Predicting Brain Stroke Using Machine Learning Algorithms

Brett Bertoni, William Nicholson, Luke Snyder

Our Topic:

Using a machine learning algorithm to predict whether an individual is at risk for a stroke, based on factors such as age, BMI, and occupation

Reason for our topic:

- Strokes are a life threatening condition caused by blood clots in the brain
- The likelihood of these blood clots can increase based on an individual’s overall health and lifestyle
- Accurately predicting whether individual may have a stroke could help save lives
- Are there any individual factors that are better at predicting strokes than others?

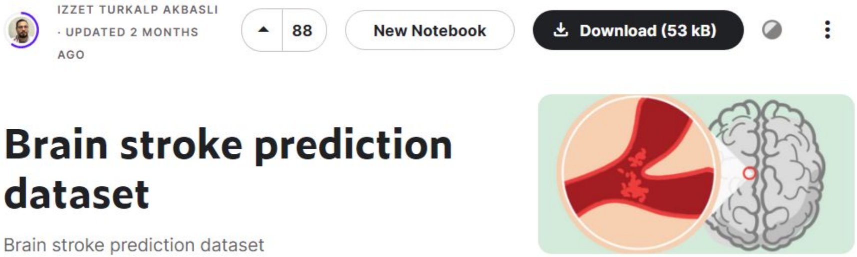
Data source:

A Kaggle database of over five thousand people, which has already been slightly preprocessed.

Factors included:

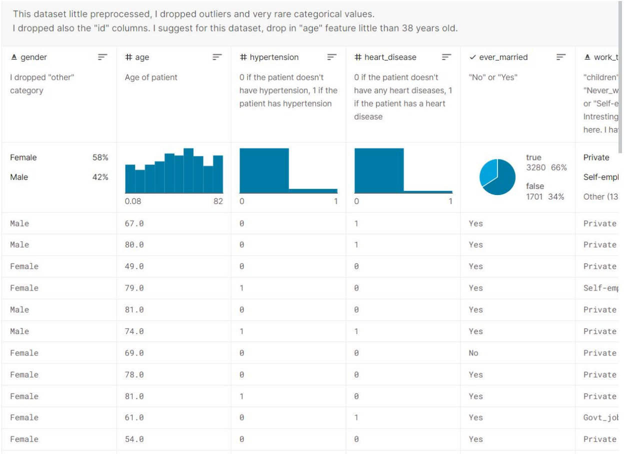
- Age
- Gender
- BMI (body mass index)
- Heart disease
- Hypertension
- Average glucose level
- Marriage status
- Residence type (rural or urban)
- Work type (government job, private, self-employed, never worked, or child)
- Smoking status

Target variable: stroke (1) or no stroke (0)



Brain stroke prediction dataset

Brain stroke prediction dataset



This dataset little preprocessed, I dropped outliers and very rare categorical values.
I dropped also the "id" columns. I suggest for this dataset, drop in "age" feature little than 38 years old.

gender	age	hypertension	heart_disease	ever_married	work_t
I dropped "other" category	Age of patient	0 if the patient doesn't have hypertension, 1 if the patient has hypertension	0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease	"No" or "Yes"	"children" "Never_w or "Self-e Intresting here. I ha
Female 58% Male 42%	0.08 82	0 1	0 1	true 3280 66% false 1701 34%	Private Self-empl Other (13
Male	67.0	0	1	Yes	Private
Male	80.0	0	1	Yes	Private
Female	49.0	0	0	Yes	Private
Female	79.0	1	0	Yes	Self-emp
Male	81.0	0	0	Yes	Private
Male	74.0	1	1	Yes	Private
Female	69.0	0	0	No	Private
Female	78.0	0	0	Yes	Private
Female	81.0	1	0	Yes	Private
Female	61.0	0	1	Yes	Govt_jol
Female	54.0	0	0	Yes	Private

Questions we hoped to answer:

- Can we create a machine learning model that can accurately predict a possible stroke?
- Which category of variable is the best predictor of a stroke (cardiovascular, employment, housing, smoking)?
- Can we predict a stroke based on an individual’s BMI alone?

Initial data cleaning/exploration

- healthcare-dataset-stroke-data.csv was read into a Jupyter Notebook file as a Pandas DataFrame
- Columns where the BMI value was “NaN” were dropped from the DataFrame
- Columns where the data values were strings were encoded into numerical form, both manually and through the Pandas get_dummies() method
- The names for the dummy columns were simplified
- To find the youngest stroke patient in the dataset, we filtered the DataFrame for ages below certain thresholds; there was only one stroke patient below the age of 20
- Since only one child had a stroke, we filtered the DataFrame for only adults (age > 17)
- The DataFrame of only adults was exported to adult.csv

Machine Learning Analysis (Segment 1)

Models tested:

- Random Forest Classifier
- Balanced Random Forest Classifier
- Logistic Regression
- Neural Network
  - ReLU input and hidden layers, 60+ nodes each
  - Sigmoid output layer
  - 100 epochs

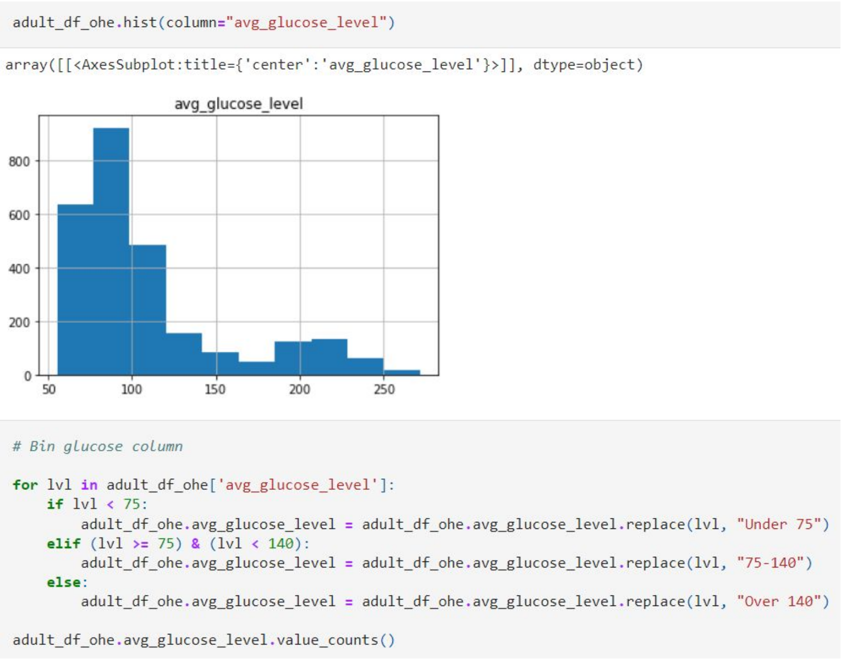
Initial Results

- The neural network achieved the highest accuracy (over 90%), but there was a high prevalence of false negatives
- False negatives are undesirable when detecting possible strokes as this is a life threatening condition
- Our dataset was modified using OneHotEncoder and ran through the neural network again, producing higher accuracy and lower loss

Machine Learning Analysis (Segment 2)

Data Binning

- Due to the prevalence of false negatives, we revisited the data preprocessing stage
- The “age”, “avg_glucose_level”, and “bmi” columns were binned into new categories based on density and value counts
    - “age” into “Under 40”, “40-59”, and “60 or Older”
    - “avg_glucose_level” into “Under 75”, “75-140”, and “Over 140”
    - “bmi” into “Obese” and “Not Obese”
- OneHotEncoder was used once again on the resulting DataFrame



```
adult_df_ohe.hist(column="avg_glucose_level")
```

```
array([[<AxesSubplot:title={'center':'avg_glucose_level'}>]], dtype=object)
```

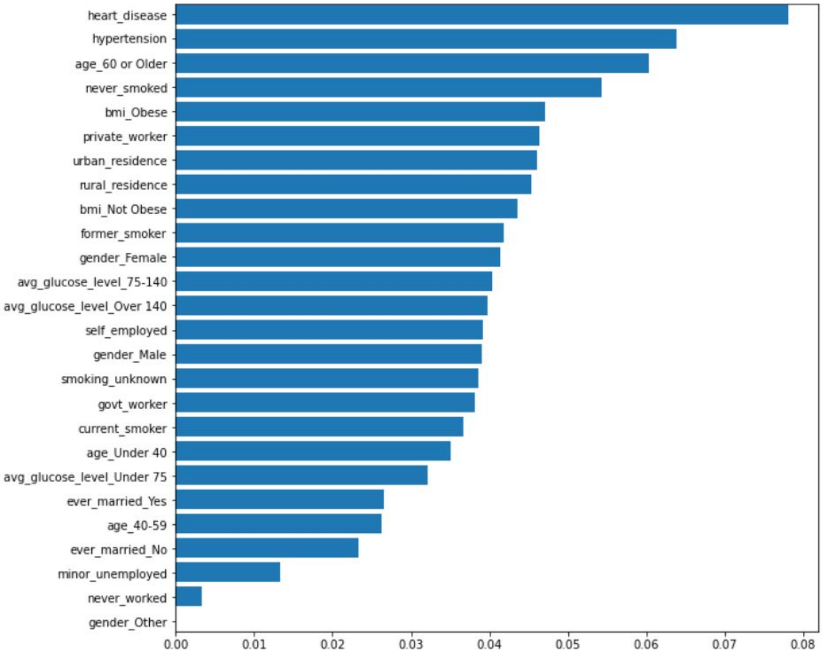
```
# Bin glucose column

for lvl in adult_df_ohe['avg_glucose_level']:
    if lvl < 75:
        adult_df_ohe.avg_glucose_level = adult_df_ohe.avg_glucose_level.replace(lvl, "Under 75")
    elif (lvl >= 75) & (lvl < 140):
        adult_df_ohe.avg_glucose_level = adult_df_ohe.avg_glucose_level.replace(lvl, "75-140")
    else:
        adult_df_ohe.avg_glucose_level = adult_df_ohe.avg_glucose_level.replace(lvl, "Over 140")

adult_df_ohe.avg_glucose_level.value_counts()
```

Binning Results

- The neural network’s nodes were decreased to 54 for both the input and hidden layers
- Performance improved slightly with each of the machine learning models and logistic regression was able to correctly predict a few stroke cases, as opposed to none
- According to the Random Forest Classifier, heart disease, hypertension, and being 60 or older were the most important features, but “never smoked” ranked highly, which interested us
- There were still many false negatives in each model, so we decided to make further improvements



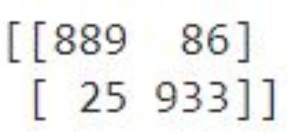
StandardScaler/SMOTE Oversampling

- Binning and OneHotEncoding were abandoned for this attempt and we started from scratch with the original adult-only DataFrame
  - Research suggests OneHotEncoding values is not ideal for tree based models as it can skew branches toward 0's (i.e. no stroke)
- StandardScaler was used on the "age", "avg_glucose_level", and "bmi" columns
- We found that the y values had many more 0's (no stroke) than 1's (stroke), so we used the SMOTE oversampling method to combat this
- The X and y values were resampled and balanced, and the dataset was split using TrainTestSplit

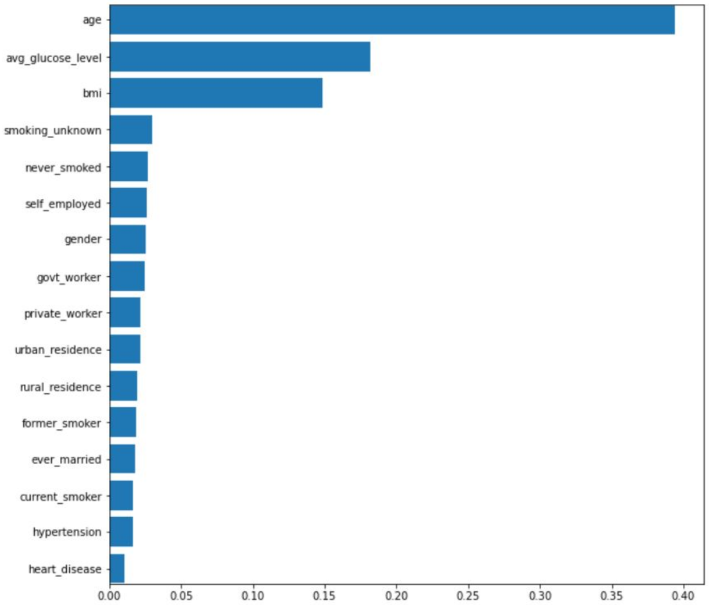
StandardScaler/SMOTE Oversampling Results

- RandomForestClassifier was used on the scaled data, and this time it showed much improved results
  - Training score: 1.0
  - Testing score: 0.943
  - Precision: 0.916
  - Sensitivity: 0.974
  - F1: 0.944
- Logistic regression and our neural network also showed improved results, but RandomForestClassifier performed best
- The feature importances looked much more reasonable, with age factoring very heavily and glucose level and bmi coming 2nd and 3rd; hypertension and heart disease dropped to the bottom
- The "never worked" column held no significance, so it was removed before graphing feature importances a second time

Confusion matrix:



```
[[889  86]
 [ 25 933]]
```



The verdict:

We will be using **RandomForestClassifier** as our final model