Kaggles stroke dataset

Ciaran Welsh

August 14, 2019

1 Introduction

I have trained a feed forward neural network to classify people into stroke victims or not-stroke victims based on their age, their average glucose level, hypertension, heart disease, gender, bmi, marital status and work type.

1.1 Exploration

After some initial exploration I decided to that the best way to visualise this data is by using scatter matrix plots with kernel density estimators along the diagonal. An example is shown in fig. 1 which the rest can be found in root/*data*/*Plots*/*scatter_matrix*.

2 Preprocessing

The biggest problem with this dataset is that the number of stroke samples was much smaller than the number of non-stroke samples (98% vs 2%). To deal with this problem, I chose to under sample the non-stroke data such that the numbers of non-stroke sample equals the number of stroke samples in both the training and validation data. While a good place to start, this turned out to be a naive approach that disregarded the profound impact that age has on the changes of a stroke incident (fig. 1). Training using this sampling approach resulted in a large variation in the model's predictive performance, as evaluated by the performance on validation data. To solve the problem I ensured non-stroke samples had the same age distribution as the stroke samples.

Other data preprocessing steps that I used include:

1. Discard young data. There are very few people under 30 in this dataset that have had a stroke. Therefore these were removed from the analysis. Obviously this becomes less important with the age stratified under sampling.

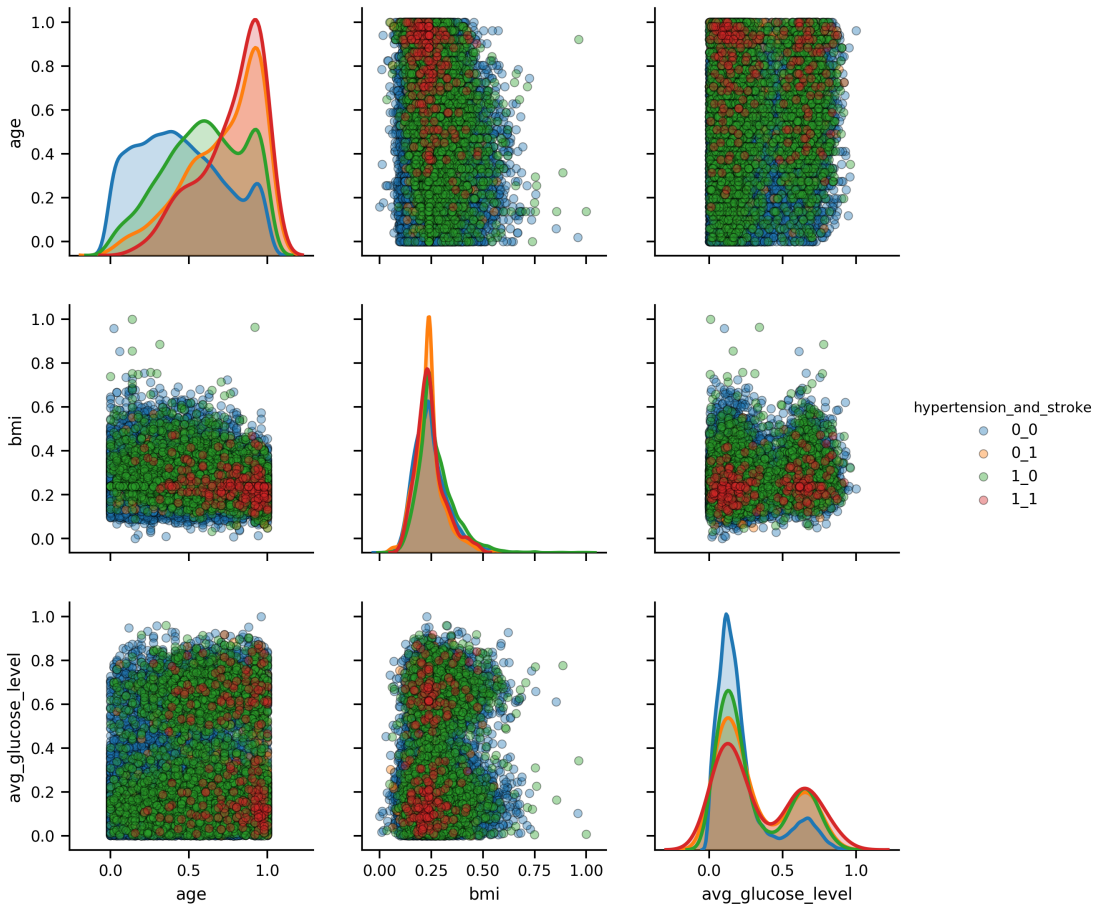

Figure 1: Scatter matrix showing relationships between bmi, average glucose levels. This plot is coloured by hypertension and stroke

2. Drop the ‘other‘ gender. There are very few ‘other’ values and none who have had strokes.

3. Impute the bmi column using the median, since only around 3% of data were missing this is reasonable.

4. Scale continuous data between 0 and 1 so they exist on a similar scale for fitting

5. One hot encode categorical and boolean variables

6. Remove the residence category: exploratory data analysis seems to suggest its not predictor of stroke incidence.

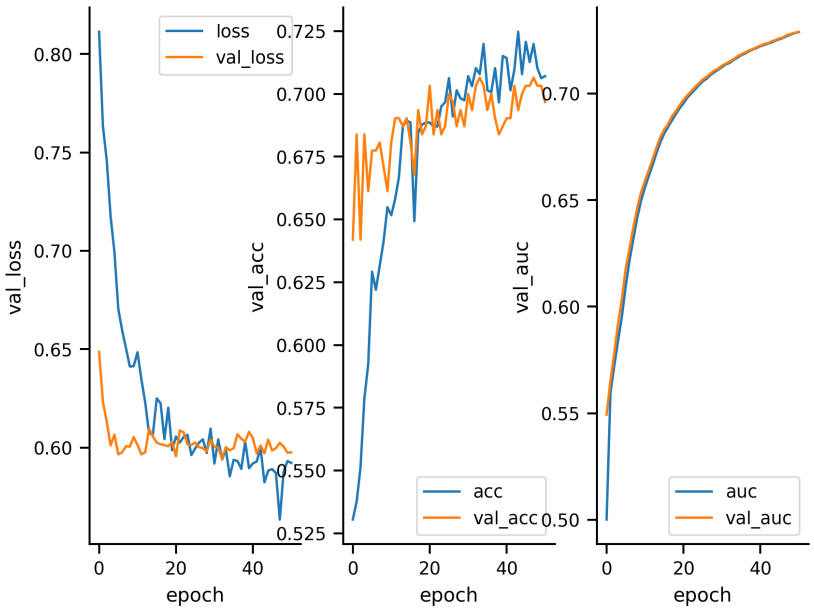

Figure 2: Training history

7. Remove smoking status category. It has too many missing values to deal with now - maybe of use for model improvement.

8. Remove any remaining samples with nan values.

Note that model performance may still be improved by modifying the preprocessing strategy. There is a package called Imbalanced-learn that would be of interest regarding sampling strategy.

2.1 Model architecture

A simple feedforward network was implemented using the tensorflow.keras interface. Dropout layers were used after each dense layer for regularization and the relu activation funciton was used in dense layers. The output layer has a single neural with a sigmoid activation function and the model was trained by minimizing the binary crossentropy objective function using the ADAM optimizer. An early stopping callback was used to prevent overfitting by stopping training when the validation accuracy begins to decline. A plot of training history can be found in fig. 2.

	loss	acc	val_loss	val_acc
Average	0.61	0.70	0.59	0.71
Std	0.013	0.014	0.020	0.029

Table 1: Average and standard deviations for 100 bootstrap models

2.2 Model architecture

Since an under sampling strategy was used to deal with the imbalanced data problem, the model only trains with a fraction of the data. To assess the models predictive robustness, a bootstrapping strategy was used (table 1). Model train and validation scores were monitored while repeating the model fitting process with a different sample from the not-stroke population with the same stroke population. Its possible this decision may have a negative impact on model generality due to over fitting the stroke population. As before, sampling was stratified by age group in addition to ensuring equal proportions of stroke and not-stroke victim.

2.3 Model Performance

Overall, the final model (which I do not have time to improve) has some issues. Even with using age stratified sampling the models performance fluctuates a lot, albeit far less than without age stratified sampling. This probably indicates that much of the samples are being categorised randomly, and therefore prone to rapid class switching between epochs. While the bootstrapping procedure indicates that the model robustly correctly classifies around 70% accuracy on the validation data, this result is probably confounded by the patience parameter of the early stopping procedure. Overall its clear that this model isn't training well and changes are required to make it perform better.

Solutions for a better classifier include modifying better feature selection. It would be valuable to train a model multiple times while dropping each variable in turn to measure the variables impact on stroke prediction. The contineous variables were scaled to be between 0 and 1, an alternative scaling method may yield better results. From preliminary simulations, I've ascertained that modifications to the model architecture does not significantly change the model performance and therefore changes to the input data may be a better strategy for taking this model forward.