Predicting factors influencing stroke: building a predictive machine learning model

Godfrey Leung
godfrey.leung.cosmo@gmail.com

Abstract—This report summarise the predictive machine learning (ML) model we build to classify patients who have stroke, which is an imbalanced class binary classification problem, using the Kaggle stroke dataset: link_here.

1 Preprocessing

Three main preprocessing steps are considered here: 1) imputing missing values, 2) label encoding for categorical variables and 3) standardise/normalise for numeric variables.

1.1 Imputation

Brief exploratory data analysis (EDA) suggests that only [bmi, smoking_status] have missing values, ∼ 3.4% for bmi and ∼ 31% for smoking_status. We pick the following simple imputation strategy: median for bmi and a new label 'no_info' for smoking_status. In Fig.1 (left panel), we show how our imputation strategy affects the model performance.

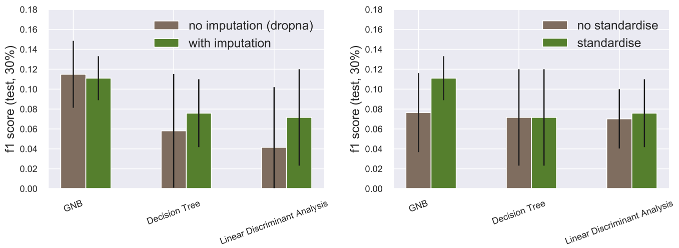


Figure 1: ML f1 score with/without imputation (**left panel**); ML f1 score with/without standardise (**right panel**).

1.2 Label encoding

Next is the label encoding preprocessing step, i.e. converting categorical variables into numeric form. We use a custom mapping for [smoking_status]; pd.get_dummies for nominal. This preprocessing step is necessary for training certain ML models.

Here we use using a custom mapping (∼LabelEncoder) for smoking_status, pd.get_dummies (OneHotEncoder) for [gender, ever_married, work_type, Residence_type]).

1.3 Standardise (scaling)

The final preprocessing step is to standardise any numeric variable to similar ranges (usually [0,1]). In Fig.1 (right panel), we show how our standardising the numeric variables affects the model performance.

2 Check for potential overfitting issues

To avoid overfitting, we perform K-fold cross validation tests, splitting the dataset into (train, test)=(60%, 30%) in 10-fold using **ShuffleSplit in sklearn**. The results are summarised in Fig.2, for the Gaussian Naive Bayes classifier.

We also perform feature selection to avoid using all the available feature variables, as too many can result in overfitting as well.

3 The ML model

Using the f1 score as metric (or KPI), we compare the performance of different ML classifiers on predicting strokes in the dataset. In the end, we

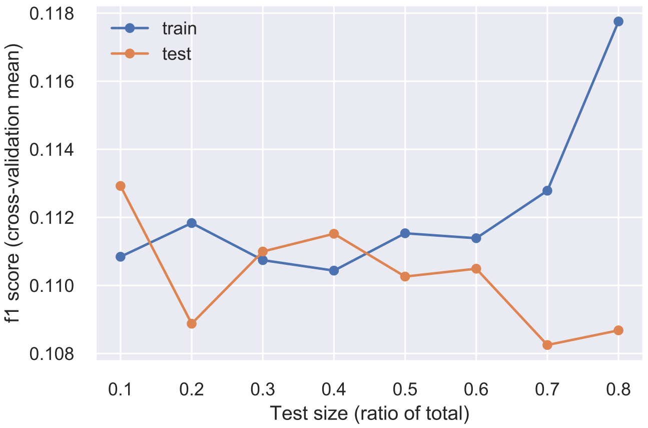


Figure 2: ML f1 score for different test sizes. Note that 90% of the whole dataset is used here.

choose the Gaussian Naive Bayes (GNB), which is simple, quick to train and gives the best f1 score.

After some hyper-parameter tuning and feature selection, we choose the following model parameters: var_smoothing=0.001, features = ['avg_glucose_level', 'bmi', 'age', 'smoking_status', 'hypertension', 'heart_disease', 'ever_married'], training on 60% of the whole dataset.

4 Model evaluation

We show the comparison of our GNB model compared with some other ML classifiers in Fig.3. We can see GNB model outperforms most other ML classifiers.

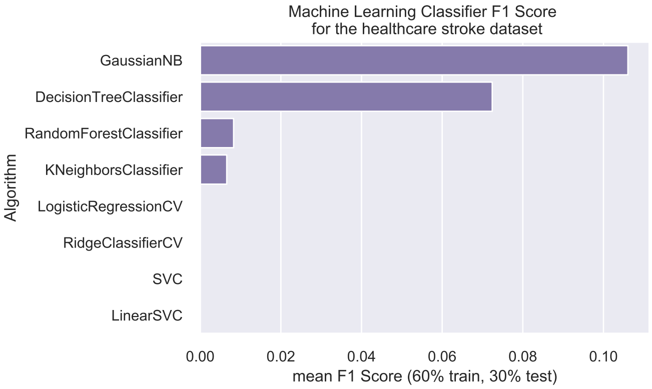


Figure 3: ML f1 score for different test sizes. Note that 90% of the whole dataset is used here.

We also show the normalised confusion matrix of our GNB model in Fig.4. Our model gives a 32% accuracy of predicting stroke and 93% for not stroke.

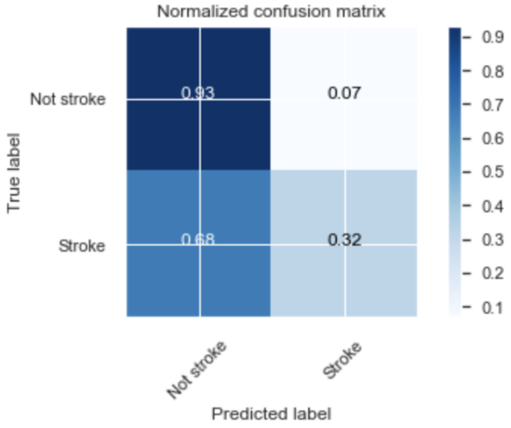


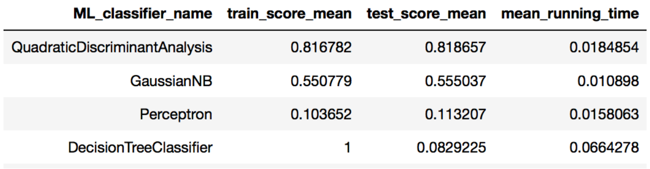
Figure 4: Normalised confusion matrix of the final GNB model.

5 Discussion and conclusion

Question: what is the appropriate metric here in this problem?

In fact, what metric to use should depend on the ultimate goal of the predictive model. For instance, if the goal is to have a predictive model for both binary classes as accurate as possible, 'f1 score' or 'roc_auc' would then be a good choice. However, if we only want to predict patients with stroke as accurate as possible, 'recall' would be more appropriate.

A brief investigation on using 'recall' as the metric suggests GNB is also one of the best models in that case as shown in Fig.5.



ML_classifier_name	train_score_mean	test_score_mean	mean_running_time
QuadraticDiscriminantAnalysis	0.816782	0.818657	0.0184854
GaussianNB	0.550779	0.555037	0.010898
Perceptron	0.103652	0.113207	0.0158063
DecisionTreeClassifier	1	0.0829225	0.0664278

Figure 5: Normalised confusion matrix of the final GNB model.

Based on using the f1 score as metric, we conclude that the simple GNB model is good enough to classify patients who have stroke.

Note: Please refer to the attached Jupyter notebook for the detailed analysis.