National Taipei University of Technology
Computer Science and Information Engineering

Spring 2022 Machine Learning
Group Project Report

Stroke Prediction

Group 4 Members
110598010 徐紹崴
110598087 謝狄烽

I. Introduction

A stroke is a medical emergency, and prompt treatment is crucial. Early action can reduce brain damage and other complications.

Strokes happen in two ways. In the first, a blocked artery can cut off blood to an area of the brain. And this is known as an ischemic stroke. 85% of strokes are of this type. The second type of stroke happens when a blood vessel can leak or burst. So the blood spills into the brain tissue or surrounding the brain. And this is called a hemorrhagic stroke.

II. Problem Statement

There are people who have all the risk factors, but are young enough where they simply have not had a stroke yet, thus, it is important to find the risk for them to have stroke and to find those potentially stroke patient.

III. Proposed Approach

In order to reach our goals, we have proposed an approach, as Fig.1, to do our experiment.

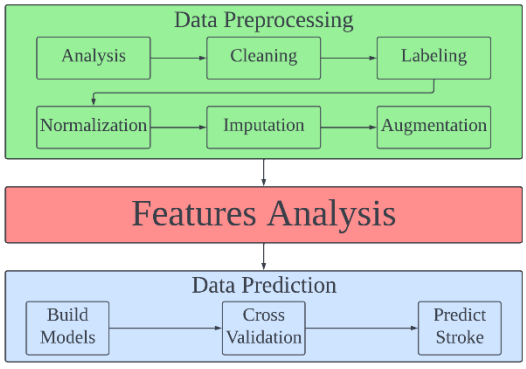

Fig.1: Process of experiment.

A. Collect and analysis dataset.

B. Cleaning Dataset

C. Relabel columns, if needed.

D. Normalize dataset

E. Missing data imputation, if needed.

F. Data augmentation, if needed.

G. Analysis features.

H. Build model for prediction.

I. Apply cross validation technique.

J. Predict stroke.

IV. Experiments

A. Collect and analysis dataset

We collect dataset from Kaggle[1] and has 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", "Unknown"
12. stroke: 1 if the patient had a stroke, 0 if not

*Note: "Unknown" in smoking_status means it is unavailable for this patient.

B. Cleaning Dataset

Id column is dropped in this step.

C. Relabel columns

There are some features that have yes-no value, and some features have categorical values, so we do label encoding and one-hot encoding to those columns.

D. Normalize dataset

Using min-max normalization to balanced

features weights for model which are sensitive to different interval of features.

E. Missing data imputation

We Applied KNN imputation[2] rather than using mean value of corresponding feature, so the imputed samples are only influenced by neighbor samples instead of all samples in dataset.

F. Data augmentation

Due to unbalanced dataset, 249 samples for stroke and 4861 samples for non-stroke, models can simply get 0.949 accuracy by predict all samples as non-stroke, to avoid such phenomenon, data augmentation is a kind of way to solve it.

Two types of augmentations are considered, Border-line SMOTE[3] and ADASYN[3], it is more ideal to make the augmented dataset similar to the original one because 95% of stroke samples are augmented, which means they are not from the real world.

We found that Border-line SMOTE change the correlation between features more than ADASYN in our study case, also, it does not have similar distributions of features from the original dataset as Fig.2, we then considered ADASYN is more appropriate to our case.

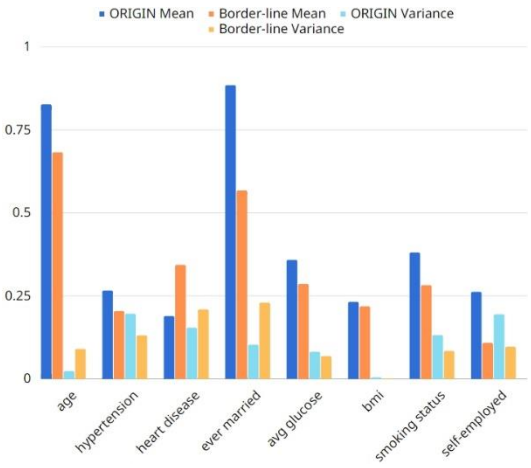

Fig.2: Features distributions after apply Border-line SMOTE.

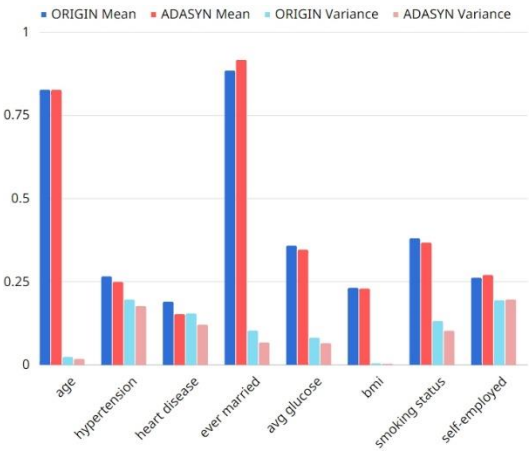

Fig.3: Features distributions after apply ADASYN.

G. Analysis features

By analyzing correlation of features, features having higher correlation with stroke are considered as more important factors, 8 features are selected:

1. age
2. hypertension
3. heart_disease
4. avg_glucose_level
5. ever_married
6. bmi
7. smoking_status
8. work_type_self_employed

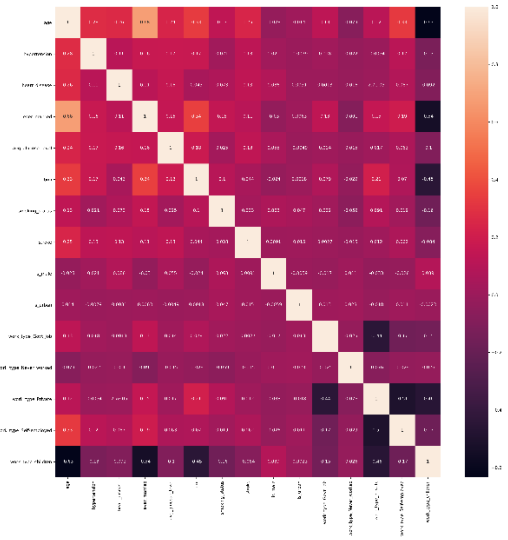

Fig.4: Heat map of original dataset.

H. Build model for prediction

For classification model, we have KNN where K=3, SVM, and AdaBoost, which has 100 estimators, and for stroke detection, we constructed four layers AutoEncoder which contains 2 encoders and 2 decoders.

I. Apply cross validation technique

StratifiedKFold[4] is a kind of the design and implementation of cross validation technique, we can give a value K and it will help us split the dataset to K folds with same or close to same number of samples, all folds will be used for validation one time, check Fig.4 for more implement detail of StratifiedKFold.

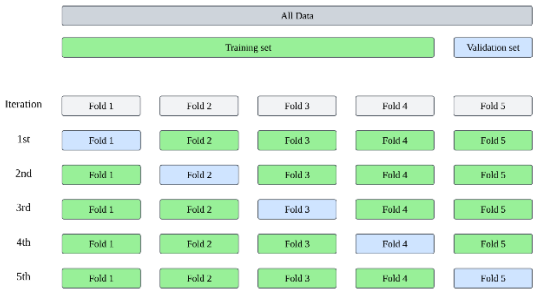

Fig.5: StratifiedKFold implementation detail when K=5 (also called 5-Fold).

In this step, we collect variances of dataset by processed different data preprocess steps, giving them identifier and corresponding steps as:

1. **CLF-D1**: Cleaning + Relabeling + Normalization + Imputation
2. **CLF-D2**: Dataset-CLF-1 + Only keep 8 features selected in step G
3. **AE-D1**: Cleaning + Relabeling + Normalization + Drop all rows with missing values
4. **AE-D2**: Dataset-AE-1 + Imputation
5. **AE-D3**: Dataset-AE-1 + Only keep 8 features selected in step G

Next, train all the classifiers in 3 ways, each way assigns 3 different datasets and apply 10-Fold cross validation.

First, assign **CLF-D1**, use 9 folds of training set to fit model and 1 fold of validation set to predict in each iteration, after 10 iterations, we will get average metrics from cross validation.

Second, assign **CLF-D1** and do augmentation on training set before fitting model, but keep validation set as original samples for prediction, in the third time, do the same steps as second time but assign **CLF-D2** instead.

After training classifiers, we continue to train AutoEncoder in 3 ways, each way assigns 3 different datasets, and apply 5-Fold cross validation.

First, assign **AE-D1**, split the chosen validation set again to 50% test set, 50% validation set, so we get 60% training set, 20% test set, 20% validation set in each iteration, then use training and test set to fit model, do predictions with validation set, after that, we will get average metrics from cross validation.

For the second and third time, do the same steps as the first time, but assign **AE-D2** and **AE-D3** respectively.

J. Predict stroke.

Models make prediction in every iteration of cross validation, a prediction will have a metrics, after model fit and predict in all iterations have ended, we get average values of those metrics w.r.t. each model, then we derived metrics such as precision, recall, f1-score to evaluate the model performance.

V. Result of experiments

KNN and AdaBoost model have 0.95 accuracy but only 0.1 or less than 0.1 f1-score when assigning dataset **CLF-D1** for cross

validation, the problem of unbalanced dataset mentioned in previous section has occurred, SVM model can avoid this problem by adjusting its hyperparameters, AutoEncoder only need to trained by dataset of one class, so it does not have such problem.

For the summary results of each model, trained by different datasets, refer to Table.1 to Table.5

Table.1: KNN (K=5) model performances.

	CLF-D1	CLF-D1 +ADASYN	CLF-D2 +ADASYN
Accuracy	0.9489	0.8205	0.8065
Precision	0.4104	0.1222	0.1221
Recall	0.0643	0.4345	0.4785
F1-score	0.1057	0.1903	0.1944
ROC	0.5292	0.6374	0.6509

Table.2: SVM (kernel = linear, class weight = balanced) model performances.

	CLF-D1	CLF-D1 +ADASYN	CLF-D2 +ADASYN
Accuracy	0.7195	0.756	0.7125
Precision	0.1301	0.1214	0.1219
Recall	0.8311	0.6468	0.7877
F1-score	0.2249	0.2042	0.211
ROC	0.7725	0.7042	0.7482

Table.3: AdaBoost (n_estimators=100) model performances.

	CLF-D1	CLF-D1 +ADASYN	CLF-D2 +ADASYN
Accuracy	0.9500	0.7579	0.7403
Precision	0.1	0.1248	0.1242
Recall	0.004	0.6618	0.7112
F1-score	0.0076	0.2098	0.2113
ROC	0.5012	0.7123	0.7265

Table.4: Auto Encoder (non-stroke, MAE) model performances.

	CLF-D1	CLF-D1 +ADASYN	CLF-D2 +ADASYN
Accuracy	0.7146	0.68	0.7616
Precision	0.3181	0.3308	0.3848
Recall	0.4966	0.555	0.5138
F1-score	0.3877	0.4144	0.4399
ROC	0.6298	0.6335	0.6653

Table.5: Auto Encoder (non-stroke, MSE) model performances.

	CLF-D1	CLF-D1 +ADASYN	CLF-D2 +ADASYN
Accuracy	0.7519	0.7017	0.7989
Precision	0.334	0.3409	0.4338
Recall	0.3655	0.4947	0.3416
F1-score	0.349	0.4036	0.3820
ROC	0.6016	0.6247	0.6211

VI. Conclusion

Classifier who trained with imputation and augmentation dataset has higher sensitivity and precision, classifier trained with imputation, augmentation and only has selected features dataset, has highest f1-score, we may consider that having domain knowledge of the disease can help the prediction if we know how representative a feature is, to a disease.

Data generated by augmentation is learned by some models, hence, they may have wrong predictions when predict the original validation dataset.

We can train AutoEncoder models without data augmentation, and still have about 0.4 f1-score, using MAE as loss function may have better ROC, AutoEncoder models have the best performances for our stroke prediction study.

VII. References

[1] “Kaggle — Stroke Prediction Da-taset,” https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset, 2021.

[2] “sklearn — KNNImputer,” https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html

[3] “資料科學(一)：處理不平衡資料幾種方法,” https://lufor129.medium.com/%E8%B3%87%E6%96%99%E7%A7%91%E5%AD%B8-%E4%B8%80-%E8%99%95%E7%90%86%E4%B8%8D%E5%B9%B3%E8%A1%A1%E8%B3%87%E6%96%99%E5%B9%BE%E7%A8%AE%E6%96%B9%E6%B3%95-39c25d06a6cb

[4] “sklearn — StratifiedKFold,” https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html