SDSC3011 Social Data Processing and Modelling
City University of Hong Kong

SDSC3011 Midterm Project

Instructor:
Qing KE

Stroke Patient Prediction from World Health Organization (WHO) stroke dataset

Matthew CHEUNG	56201843
Shan Wa YEUNG	56632434
T-touch PATTARAVARODOM	56325772

Date: 02-03-2023

Table of content

Part 1: Data Processing

1.1 Background and Motivation

The Stroke Patient Prediction dataset is from the World Health Organization (WHO), which is a specialized agency of the United Nations responsible for international public health. Our motivation is to predict stroke correctly because, according to WHO, Stroke carries a high risk of death. So people can prevent other health issues caused by stroke if they can get medical treatment early, and our mission is to increase the accuracy score.

The dataset is a 'Social Data,' containing relevant information about patients who have been to a Stroke diagnosis, provided by the data scientist at Kaggle, Fedesoriano. This part explains the data pre-processing and how the quality of data is assessed. Furthermore, the missing values problem is also handled in this part.

1.2 Data Semantics

The most critical variables in the dataset are described as follows. The remaining columns are not discussed either because their meaning is too obvious or because their discussion is delayed to the following sections.

- *gender*: a categorical attribute defining three gender types (Male, Female, Others).
- *hypertension*: a binary attribute that identifies if the patient has hypertension or not;
- *heart_disease*: a binary attribute that identifies if the patient has heart disease or not;
- *work_type*: a categorical attribute defining five working statuses (children, Govt_jov, Never_worked, Private, Self-employed);
- *Residence_type*: a binary attribute defining if the patient has a Rural or Urban residence;
- *avg_glucose_level*: a continuous numerical attribute indicating the average glucose level in blood;
- *bmi*: a continuous numerical attribute indicating the body mass index;
- *Smoking_status*: a categorical attribute defining four smoking habits (formerly smoked, never smoked, smokes, Unknown);
- *stroke*: the target variable. It states if the patient was diagnosed with a stroke or not.

1.3 Data Transformation

Data in this dataset come in different types. Therefore, it's valuable to ensure consistency by managing data types. The issue with this dataset is there are many categorical variables; 'gender,' 'hypertension,' 'heart_disease,' 'ever_married,' 'work_type,' 'Residence_type,' 'and smoking_status.' The conversion of these categorical variables to numerical vectors by 1-hot encoding is applied. From adding a dummy variable to each categorical variable, there are a total of 24 input attributes and one target attribute in the dataset.

Some of the information contained in these attributes is irrelevant to the target attribute and is dropped from the dataset, the ID attribute.

The correlation matrix in Figure 1 reveals that most of the attributes are weakly to not correlated to each other. There is a moderate association between 'married' and 'age,' 'children work_type' and 'age,' 'children work_type' and 'never_married', which are all general assumptions for the age-related correlation. Interestingly, 'children' has a moderate negative association with 'bmi' and 'private' work type.

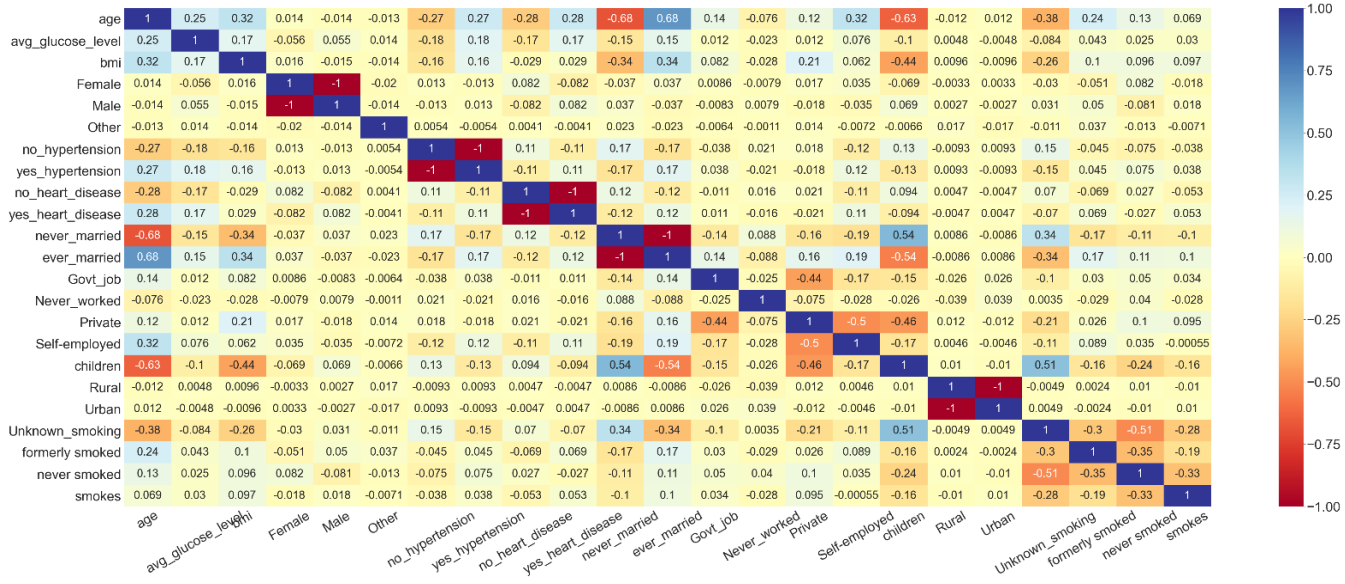

Fig. 1: Correlation Matrix

1.4 Attribute Distribution

In this part, we will analyze the distribution of some particular attributes, showing interesting statistical plots. The first thing to note is that the target variable (Stroke) is highly imbalanced. Stroke patients are only 4-5% of the dataset, while the remaining 95-96% are non-Stroke patients. Data Balancing by SMOTE which will be discussed later (Part 2.1).

In the plot shown in Figure 2, we offer the distribution of the three numerical attributes, which point out a different range of distribution. There are some outliers in the avg_glucose_level and bmi, which will be discussed later (Part 1.4). The distribution was normalized to be in the same range of (0, 1) to improve the accuracy and integrity of the data.

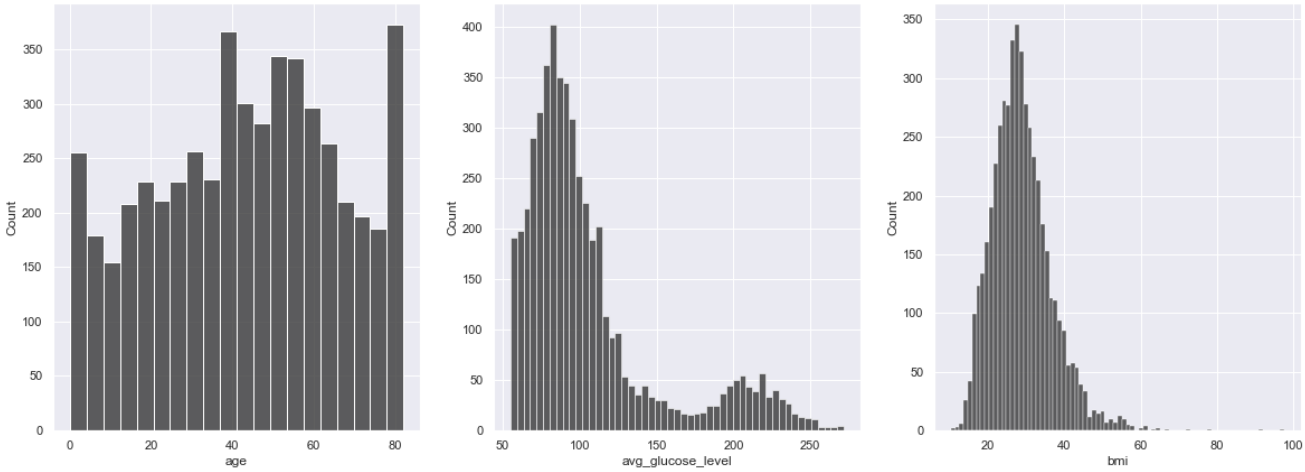

Fig. 2: Distribution (from left to right) of age, avg_glucose_level, and bmi

1.5 Data Quality

According to the Data Quality Continuum, the problem occurred from Data Transformation; the *'Other'* attribute from *'Gender'* only contains one instance; dropping this attribute, therefore, was committed.

As far as the outliers are concerned, and the distribution was explained (Part 1.3), we kept many instances with a high average glucose level and some instances with a high bmi. Still, we could not label them as outliers, as we thought that this behavior was to be expected. So we decided to keep all in the data set.

By looking at Figure 3, it is possible to recognize the missing values of the dataset (identified by the white lines). The attribute 'bmi' has some missing values, thus we decide to impute them with the median of its attribute.

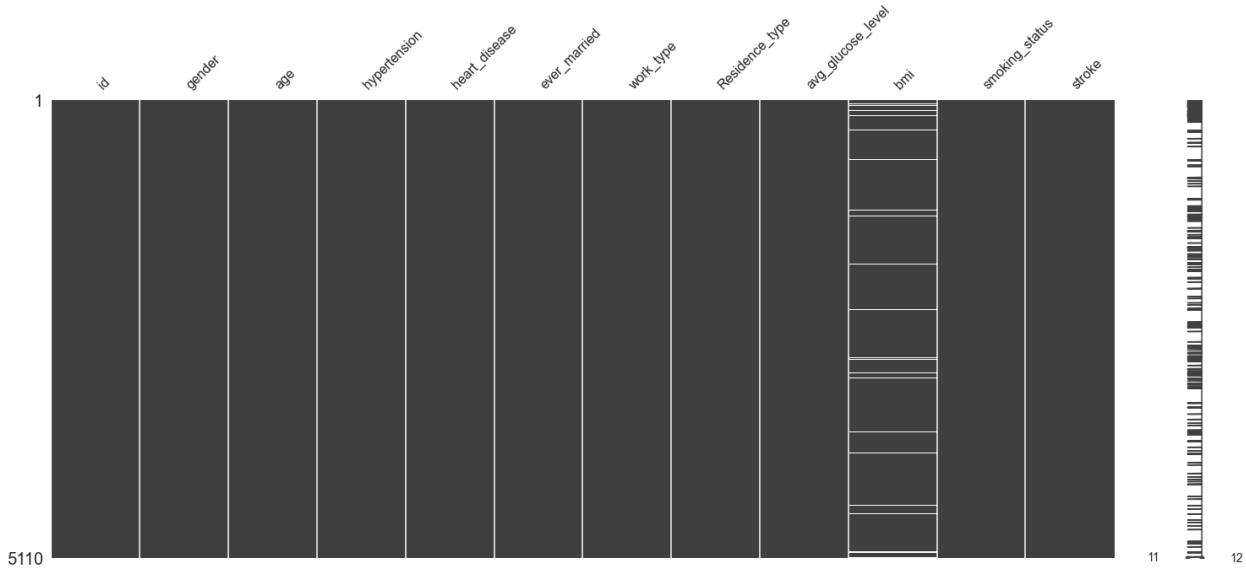

Fig. 3: Missing values

Part 2: Model Development and Classification

In the following part, three algorithms, Random Forests Classifier, Logistic Regression, and Multilayer Perceptron (MLP), are used during the classification. The main goal of this task was to predict the target attribute called 'Stroke,' which indicates whether a patient has a diagnosed stroke or not with 23 input attributes. The criteria used to evaluate each algorithm are the accuracy and F1 scores.

2.1 Methodology

The first step was to split the dataset into test and train datasets. Regarding the proportion of the train-test data, 70-30 was committed. The test set ("test.csv") will be used as ground truth to verify the performance of the final models. For the training set ("training.csv"), cross-validation was applied to each model and used as input data.

For each algorithm of classification, the input dataset is *balanced* and *undersampled.* Then, three different methods in each algorithm were compared for the best performance.

2.1.1 Data Balancing

SMOTE is applied to the dataset to improve the class imbalance.

- The *undersampling* technique resulted in 174 instances equally for each class.
- The *balancing* technique resulted in 3,403 instances equally for each stroke class.

2.1.2 Feature Selection

Due to the Data Balancing, the feature importance from the *balanced*, *undersampled*, and *imbalanced* datasets are relatively different. Therefore, the feature selection is different for each method. The feature selection was made by the *mutual_info_classif* function by setting k (number of features) equal to ten features.

Table 1. Feature Selection

Dataset	*Feature Importance*
Imbalanced	**'age', 'avg_glucose_level'**, 'bmi', 'yes_hypertension', 'no_heart_disease', 'yes_heart_disease', 'never_married', 'Govt_job', 'children', 'formerly smoked'
Balanced	**'age', 'avg_glucose_level'**, 'bmi', 'never_married', 'ever_married', 'Private', 'Self-employed', 'children', 'Unknown_smoking', 'never smoked'
Undersampled	**'Age', 'avg_glucose_level'**, 'no_hypertension', 'yes_hypertension', 'ever_married', 'Govt_job', 'Self-employed', 'children', 'never smoked', 'smokes'

According to the table 1, some common features appeared in every dataset, ‘age’, and ‘avg_glucose_level,’ showing to be vital attributes that contribute to the target attribute. Figure 5 also visualizes the level of importance of each feature in each dataset. We then only select the ten features for the model developed in each method.

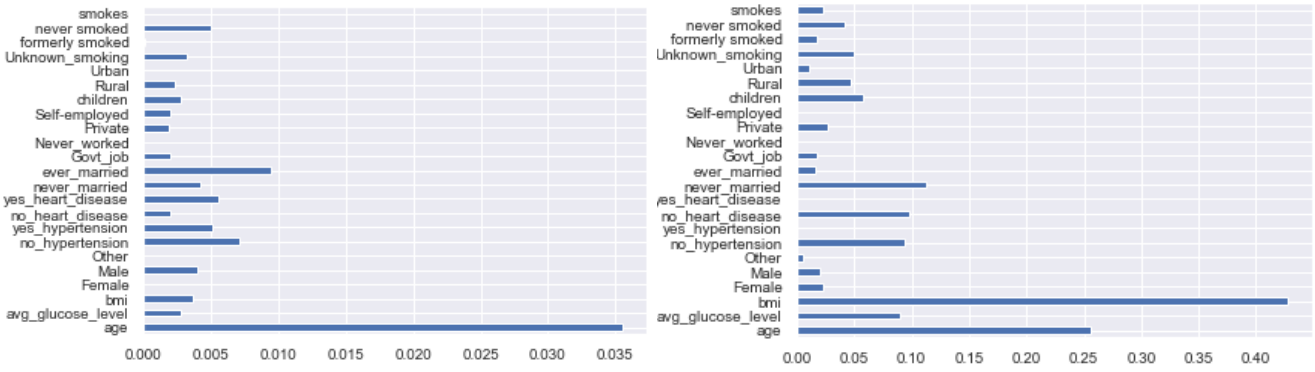

Fig.5: Features Importance (from left to right) of the imbalanced and balanced dataset

2.2 Classification Results

Once the best settings for each method and algorithm were discovered, all models were trained. Regarding the input data, we adopted the whole training set that has been processed and normalized from the first part. Afterward, we verified the performance using the given test set. Below's table reveals that the best approach for this task in terms of accuracy is MLP Classifier with imbalanced data.

Table 2. Classification Results from each method and algorithms

	Accuracy	*F1-Score*
Logistic Regression	0.7547	0.2450
Logistic Regression (Balanced)	0.7495	0.2351
Logistic Regression (Undersampling)	0.7495	0.2289
Random Forest Classifier	0.8995	0.2376
Random Forest Classifier (Balanced)	0.8513	0.1739
Random Forest Classifier (Undersampling)	0.5649	0.1735
MLP Classifier	0.9511	0
MLP Classifier (Balanced)	0.7991	0.2143
MLP Classifier (Undersampling)	0.7443	0.2129

2.3 Hyperparameters Tuning

To discover the best parameters to predict the target attribute, we tested different settings using BayesSearch Cross-Validation. We first define a range of values for each hyperparameter. The model will utilize Bayesian Optimization where a predictive model referred to as "surrogate" is used to model the search space and utilized to arrive at good parameter values combination. Table 3 shows the tuning of the MLP classifier. Interestingly, all three methods have the same hyperparameters setting.

Table 3. Hyperparameter Tuning

Algorithm	*Hyper-parameters*
MLP Classifier MLP Classifier (Balanced) MLP Classifier (Undersampling)	*activation : relu* *alpha* : 0.0001 *batch_size* : auto *beta_1* : 0.9 *beta_2* : 0.999 *early_stopping* : False *epsilon* : 1e-08 *hidden_layer_sizes* : (100,) *learning_rate* : constant *learning_rate_init* : 0.001 *max_fun* : 15000 *max_iter* : 500 *momentum* : 0.9 *n_iter_no_change* : 10 *nesterovs_momentum* : True *power_t* : 0.5 *random_state* : 420 *shuffle* : True *solver* : adam *tol* : 0.0001 *validation_fraction* : 0.1 *verbose* : False *warm_start* : False

Part 3 Discussions and Interpretation

3.1 Insights and Conclusion

For the Model Development and Classification task, we used different algorithms to understand whether a patient has a stroke.

In the *Data Processing* part, we addressed issues related to data semantics, data type, data distribution, and data quality, like data transformation, managing data types, data normalization, fixing missing values, and deleting redundant attributes. Our main goals in this part were to clean up the Model attributes (which are the deciding factors for the target attribute) and have a basic understanding of variable distributions.

In the *Model Development and Classification* part, we tried to predict the value of the target attribute, namely 'stroke.' We performed 10-fold cross-validation with various classifiers and configurations, attempting Dataset Balancing also in both *undersampling* and *balancing* techniques. The *Undersampling* performed worse than the *Balancing* technique. However, the *Imbalanced* (Non-balancing) dataset has the relatively best performance. Regarding the Feature Selection, all three dataset shows some common parameters among the three datasets. The best model that outperformed all the algorithms is Multilayer Perceptron (MLP), with the highest accuracy. After that, we found that all three MLP methods tuned the same optimized hyperparameter setting. The True Positive is very high, while the True Negative is low for the prediction performance.

3.2 Optimized Undersampled Multilayer Perceptron: Interpretation

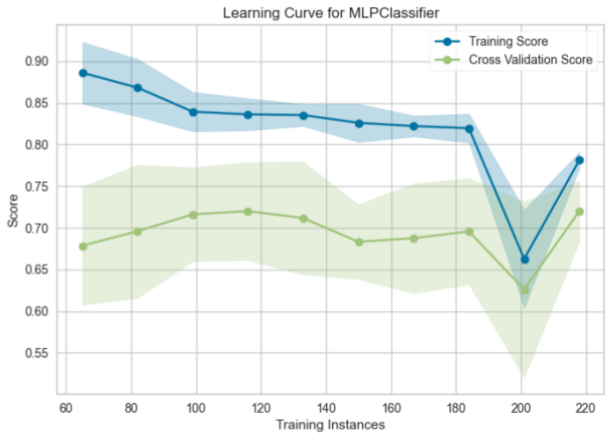

Fig. 6: The Learning Curve for Undersampled MLP

The learning curve concludes with both moving up, and the training score remains higher than the validation score across all training, as seen in Figure 6. It indicates that the model was not learning well and could not adequately capture patterns in the data. The model may have a large bias or underfitting. This can happen if the model is overly basic and unable to grasp the patterns in the data, resulting in a large training and validation loss.

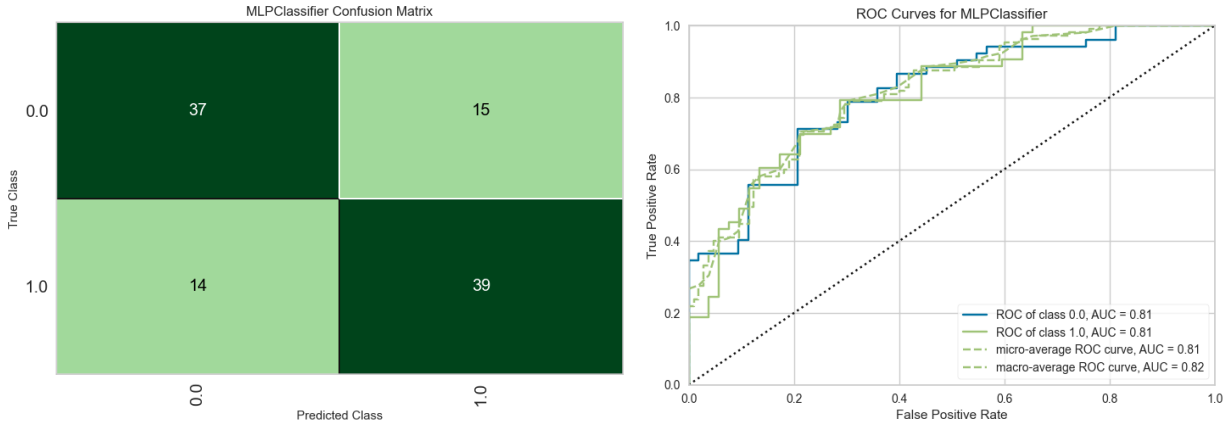

Fig. 7: (from left to right) The Confusion Matrix and ROC curve for the Undersampled MLP

3.3 Optimized Balanced Multilayer Perceptron: Interpretation

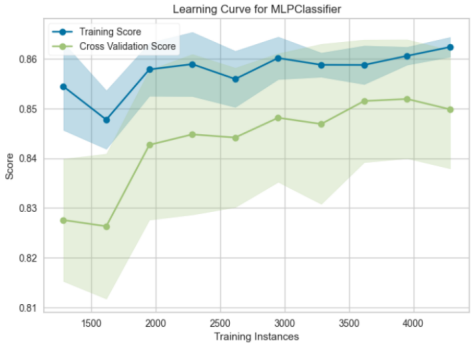

Fig. 8: *The Learning Curve for Balanced MLP*

Figure 8 shows that as the instance rises, the validation score decreases and the training score climbs near the learning curve's end. That can be because the model was overfitted. The model gets excessively complicated as a result of learning the noise in the training set, making it less able to generalize to new data (test set).

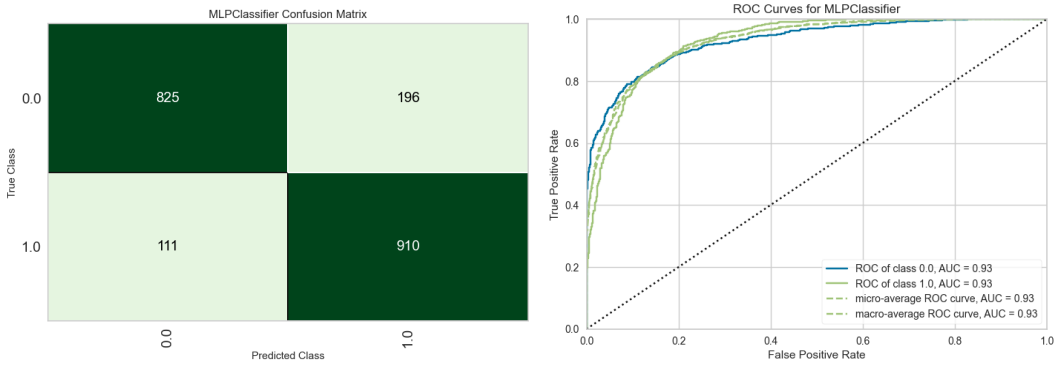

Fig. 9: (from left to the right) The Confusion Matrix and ROC curve for the Balanced MLP

3.4 Optimized Imbalanced Multilayer Perceptron: Interpretation

During the training process, the model learns to minimize the loss on the training data, which measures the difference between the predicted and actual labels. As the model becomes more complex (e.g., by adding more hidden layers or neurons), it may start to fit the noise in the training data, resulting in a low training error but a high validation error. The validation error measures the model's performance on a separate validation set that the model has not seen during training.

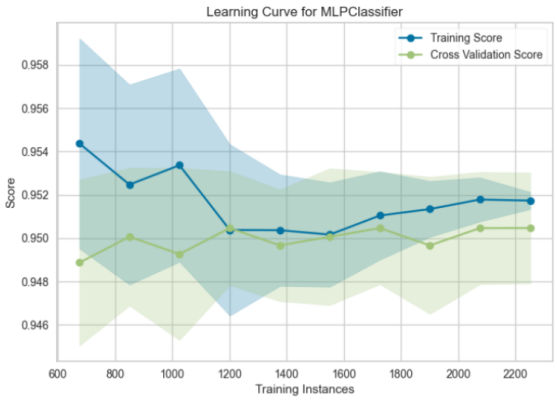

Fig. 10: *The Learning Curve for Imbalanced ML*

Suppose the training curve surpasses the cross-validation curve at the end of the training process. In that case, it suggests that the model is overfitting the training data and is not generalizing well to new data. The training error continues to decrease, but the validation error starts to increase, indicating that the model is no longer improving on the validation set.

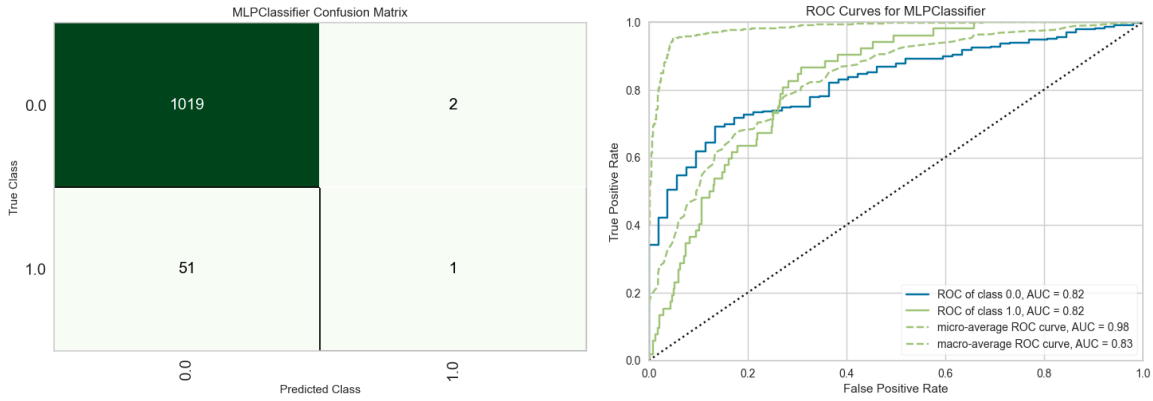

Fig. 11: (from left to the right) The Confusion Matrix and ROC curve for the Imbalanced MLP

3.5 Limitations and Reflections

*Using **Random Forest Classifier** as an example (Noted the test dataset is unbalanced, only the train dataset is balanced for case 2)*

Although the true positive rate we first did (using normalized data) was pretty good (100%), we still wanted to improve our model to increase the true negative rate.

However, the dataset was considerably imbalanced; 249 instances for 'stroke' labeling out of 5,110 instances in the original dataset, making the prediction result also highly biased, e.g., we tried different models. At the same time, most of them cannot predict true 'stroke' labeling (perform poorly in true negative rate). Resampling the data was therefore committed; we then tried under-sampling the 'non-stroke' labeling, but the results remained unimproved, e.g., the true negative rate increased from 0% to 5.3%. Moreover, the true positive also decreased from 100% to 97.15% (using a normalized and balanced dataset). The total accuracy score was also reduced, so it may not always be better for balancing the dataset.

In this case, the true negative rate is the focus of our study (correctly predicting stroke labeling), as we believe if people were diagnosed and labeled as stroke patients under a high true negative rate case, meaning there is a high probability that they tend to have a stroke. They can consider medical treatment (early), so the high accuracy in true negative rate is more important than that of the high accuracy in the true positive rate in this case.

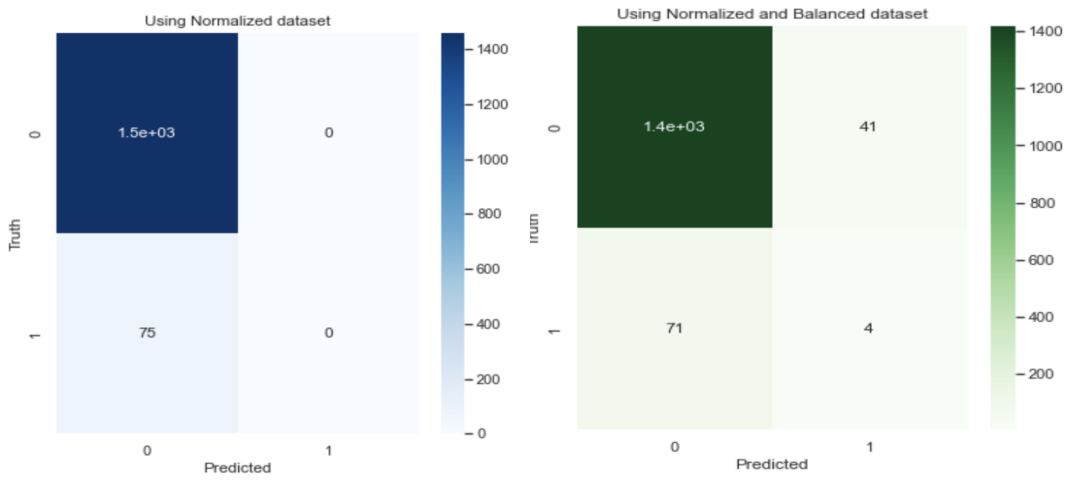

Fig. 12: (from left to the right) The Confusion Matrix by Random Forest using the imbalance dataset and using the balanced dataset

AUC, F1, and Accuracy

Table 2 shows that MLP Classifiers perform best when given unbalanced data. The high accuracy of utilizing the imbalance data is deceptive, as we discovered when we examined the confusion matrix, AUC-score, and F1-score of the three techniques in more detail. While the imbalance model "exactly" predicts the non-stroke class, it can obscure the imbalance model's subpar performance in predicting the minority class and hence have greater accuracy. In reality, because the "had stroke" class makes up just 5% of the overall data, the model has inadequate training data and is unable to forecast the class accurately. Consequently, when choosing the best-performing model, we also need to consider several metrics.

Reference

Fedesoriano. (2021). Stroke Prediction Dataset. Retrieved from https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset.

Appendix:

Link of code folder
https://drive.google.com/drive/folders/1z4lBRLmeXY8xdLTbEPBx15lYXjlbpWrb?usp=share_link