Stroke prediction

Hristina Adamović SV32/2020

1. Motivation

Stroke is a second leading cause of death and disability worldwide [1]. Early prediction can facilitate timely medical interventions, potentially saving lives and reducing long-term disabilities. This project aims to leverage machine learning techniques to identify individuals at higher risk of experiencing a stroke, based on their health and demographic data.

2. Research questions

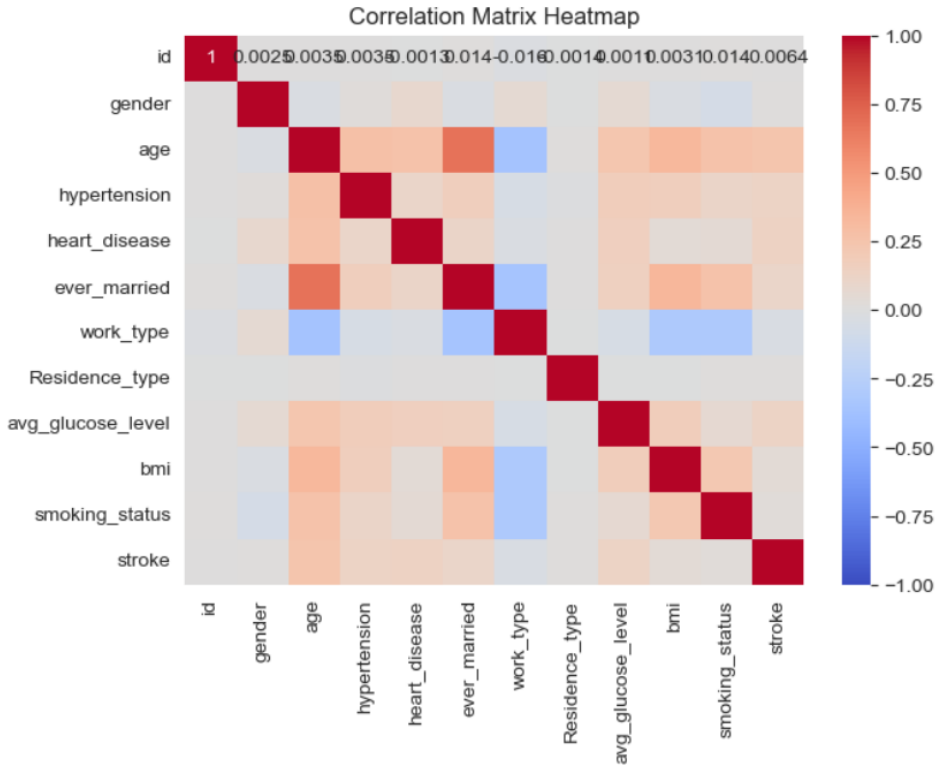
The specific problem addressed is the prediction of stroke risk, determining whether a patient is likely to have a stroke based on their health and demographic data. The dataset is sourced from Kaggle and contains 12 columns and over 5000 rows [2]. Dataset includes various features such as gender, age, heart disease, work type, bmi and other.

3. Related work

Previous work on stroke prediction has utilized various machine learning algorithms to identify individuals at risk. For example, studies have applied logistic regression, decision trees and ensemble methods such Gradient Boosting Machines and CatBoostClassifier. These studies typically focus on the importance of feature selection and data preprocessing to improve the accuracy of predictions. The Kaggle site, where the dataset was sourced, features many projects utilizing the same dataset [2].

4. Methodology

1. **Loading and Inspecting Data**: The dataset was loaded and initial exploratory data analysis was conducted to understand the distribution of features and identify any anomalies [3].
2. **Handling Missing Values**: The dataset contained missing values only in the ‘*bmi’* column. Rows with missing values were dropped to ensure data quality since they represented a small number of instances. Droping these rows showed better results than replacing missing values with the mean value of the column.
3. **Removing Outliers**: The range of values in the *'bmi'* column exceeded normal bounds, so values where ‘*bmi’* was less than 13 or greater than 55 were removed. In the *'gender'* column, a very small number of occurrences had 'other' as a value, so those rows were removed. Similarly, in the *'work_type'* column, there were also a small number of instances with *'never_worked'* as a value, which were removed as well.
4. **Encoding Categorical Variables**: Categorical variables were converted to numerical format using *one-hot encoding* [4], which performed better compared to label encoding. One-hot encoding was chosen because it did not pose a dimensionality issue given the manageable number of values in the categorical columns.
5. **Dimensionality reduction**:
   Dimensionality reduction was performed using *Principal Component Analysis (PCA)* [5], but it did not yield significant effects.



Observing the correlation matrix, the highest absolute correlation between variables was 0.679125. Since this correlation does not indicate a particularly strong dependency, no columns were removed. Only the column 'id' was removed as it holds no necessary meaning for training purposes.

6. **Oversampling**:
   Over 75% of the rows for the target variable have had a value of 0. It was necessary to perform oversampling to ensure accurate predictions for the value of 1. The *ADASYN* [6] technique was used to address class imbalance by generating synthetic samples for the minority class (stroke occurrences).
7. **Model Training**
   Three different ensemble models were used: BaggingClassifier, VotingClassifier and AdaBoostClassifier [7]. They were used with diverse base models during training.
8. **Data spliting**
   The dataset was split into training and test sets using an 80-20 split, stratified to maintain the target class distribution.

5. Discussion

Evaluation metrics used to test the model are **accuracy**, **precision**, **recall**, and **F1-score** [8]. These metrics were chosen to provide a comprehensive assessment of model performance, particularly for the minority class.
Grid search was used to optimize **hyperparameters** for the ensemble models (Bagging, Voting, AdaBoost).

The table below shows the evaluation results for each tested model:

Model	Accuracy	Precision	Recall	F1-score
VotingClassifier using RandomForestClassifier	0.96	0.96	0.96	0.96
VotingClassifier using DecisionTreeClassifier	0.90	0.91	0.90	0.90
VotingClassifier using KneighborsClassifier	0.90	0.91	0.90	0.90
BaggingClassifer using RandomForestClassifier	0.90	0.90	0.90	0.90
BaggingClassifier using DecisionTreeClassifier	0.93	0.94	0.93	0.93
BaggingClassifier using KneighborsClassifier	0.90	0.91	0.9	0.90
AdaBoostClassifer using RandomForestClassifier	0.96	0.96	0.96	0.96
AdaBoostClassifier using DecisionTreeClassifier	0.95	0.95	0.95	0.95

From the previous data, we can observe that the VotingClassifier using RandomForestClassifier and AdaBoostClassifier using DecisionTreeClassifier achieved the best results. It is important to note that the evaluation metrics were also individually assessed on the target variables, which is not shown in the table. Based on this, the AdaBoostClassifier using DecisionTreeClassifier was selected as the final solution, achieving the best performance.

6. References

[1] World Health Organisation. Available: https://www.who.int/data/gho/data/themes/mortality-and-global-health-estimates/ghe-leading-causes-of-death.

[2] Dataset. Available: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset/data.

[3] Data analysis. Available: https://github.com/hristinaina/stroke-prediction/blob/main/data_analysis.ipynb.

[4] One hot encoding. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html.

[5] PCA algorithm. Available: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html.

[6] Oversampling: ADASYN algorithm. Available: https://imbalanced-learn.org/dev/references/generated/imblearn.over_sampling.ADASYN.html.

[7] Ensemble models. Available: https://scikit-learn.org/stable/modules/ensemble.html.

[8] Evaluation metrics. Available: https://scikit-learn.org/stable/modules/model_evaluation.html.