Heart Stroke prediction

Motivation

The modern way of life is such that less and less attention is paid to health.Heart attack is one of the more common problems that occur.In order to predict a heart attack, we will introduce machine learning to help.

Research questions

We would like to predict a stroke using machine learning. A dataset is described by id, gender,age,hypertension,heart_disease,ever_married,work_type,Residence_type, avg_glucose_level, BMI, smoking_status, and stroke.
More information about attributes:

- id is a unique identifier for each row in the data
- gender can have values "Male", "Female" or "Other"
- age refers to the age of the patient
- hypertension can have values of 0 if the patient doesn't have hypertension, 1 if the patient has hypertension
- heart_disease can have values of 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease
- ever_married can have values "No" or "Yes"
- work_type have have values "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
- Residence_type can have values "Rural" or "Urban"
- avg_glucose_level refers to the average glucose level in the blood
- bmi refers to body mass index and can have a numerical value or "N/A" which means that information is not available
- smoking_status can have values "formerly smoked", "never smoked", "smokes" or "Unknown" which means that information is unavailable for this patient
- stroke can have values of 1 if the patient had a stroke or 0 if not

Related work

We found some solutions to this problem with the same dataset. They plotted data on different graphics and used heat maps for better data analysis.They used replacing, linear regression and k-nearest-neighbors for empty values in data and Random Forest and SVM for prediction.Some solutions we found on this github repository.

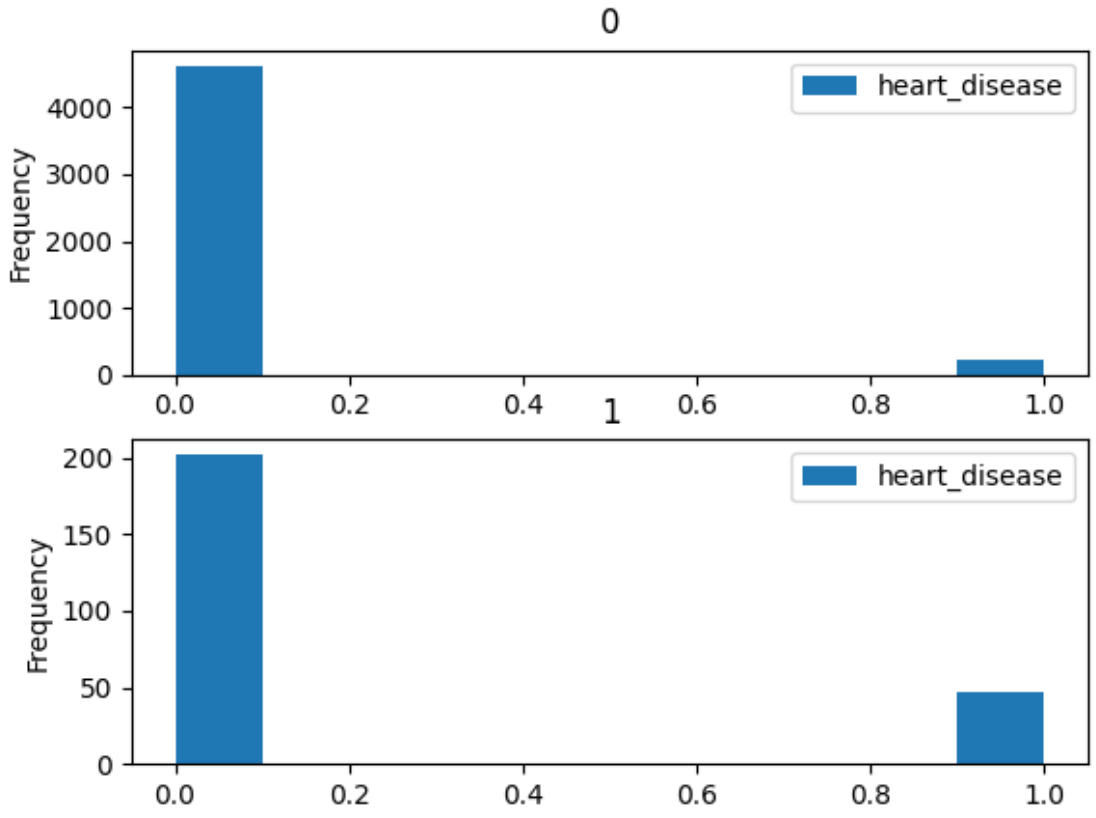
Methodology

We address the problem with preprocessing data and using SVM and Random Forest as methods.At first, we searched what method should be useful and found that SVM and Random Forest are very useful in the healthcare industry (resource). Then we analyzed the dataset and did preprocess data.Categorical features were replaced with numerics. Special were processed features smoking_status with value "Unknown" and bmi with value "N/A". "Unknown" is replaced with the most frequent value("never smoked") and "N/A" is replaced with the mean value of that column. We removed the column "id".

SVM and Random Forest were used and we were trying to get a better result.
Performance metrics we are using are accuracy, precision, recall, and f1 score with macro-averaged.Random Forest gave better results. We were trying to get better results by removing some columns and setting different values for parameters(in Random Forest we were changing values of n_estimators and random_state and with SVM we were using linear kernel and rbf kernel).We split data into training, validation, and test set(70-20-10).

Discussion

During the data analysis we found that there are more people with stroke who didn't have heart disease than those who had heart disease.



We tried to remove 'heart_disease' and the performance was better than performance which includes that column.Results were better using random forest than svm. At first the best results were achieved by removing columns *'Residence_type'* and *'heart_disease'* using **RandomForestClassifier** and parameters *n_estimators*=1 and *random_state*=11208.

Accuracy	0.9304347826086956
Recall	0.5579545454545455
Precision	0.5641025641025641
F-measure	0.5608067788518917

Results

	Stroke	Not Stroke
Stroke	3	15
Not Stroke	17	425

Confusion matrix

While we were analyzing data that was badly predicted we found some common values of columns. For bad prediction of stroke, there were 12 *hypertension* column values 0 and 15/15 has column *ever_married* value 1. Also for bad prediction of non-stroke was 12/17 hypertension value 0 and 16/17 value 1 for column *ever_married*. Our next step was removing these columns*(hypertension* and *ever_married*) and then we got better results.

Accuracy	0.9304347826086956
Recall	0.5818181818181818
Precision	0.5818181818181818
F-measure	0.5818181818181818

Results

Also we did hyperparameters optimization again and got the best result with *n_estimators*=5 and *random_state*=5436.

Accuracy	0.967391304347826
Recall	0.6488636363636363
Precision	0.913118889940082
F-measure	0.7138235660072165

Results

	Stroke	Not Stroke
Stroke	6	1
Not Stroke	14	439

Confusion matrix

So our final solution was found by removing columns 'ever_married', 'Residence_type', 'hypertension' and 'heart_disease' and with **RandomForestClassifier**, hyperparameters *n_estimators*=5 and *random_state*=5436.

References

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
https://towardsdatascience.com/how-to-split-data-into-three-sets-train-validation-and-test-and-why-e50d22d3e54c