Predicting Stroke Outcomes from patient health data

Lukas Kania
Data 1030: Andras Zsom
December 7, 2021
Github Repo: https://github.com/lkania1/data1030project

Introduction:

In this project, I investigated data collected from patients to determine whether a patient had a stroke or not. This dataset provides insights to what the World Health Organization calls the second highest cause of death in the world (Kaggle). This classification problem can be used to help anticipate a patient's health outcomes and inform health providers on certain indicators that can lead to a stroke.

This project included multiple facets of a patient's health that all impact the stroke positivity or negativity. These features included metrics such as age and gender as well as more in-depth health related metrics such as presence of hypertension and heart disease. Other factors beyond bodily health are included with metrics such as residential type, marital status, and work type. All these metrics provided insights to a patient's overall health and were used to determine their risk for a stroke. The risk is a binary scale of 1 and 0, with 1 indicating a having/had a stroke and 0 indicating not having/had a stroke. The dataset provides these metrics for 5110 patients.

This Kaggle dataset has been used in multiple studies, including the paper in the IEEE called "Predicting Stroke from Electronic Health Records." They used this dataset to do just what the title describes. They performed a principal component analysis of the features to see what datapoints contributed strongly towards a stroke. Another paper, "Using Machine Learning to Improve the Prediction of Functional Outcome in Ischemic Stroke Patients" does not explicitly use the Kaggle set but provides insights to machine learning and the prediction of strokes.

Feature Descriptions:
ID: Identifier of a patient that provides no insight other than an identifier of a patient. Will not be used in analysis.
Gender: Identify the gender of a patient with values of "Male", "Female", "Other".
Age: Identify the age of a patient.
Hypertension: Describe whether a patient has hypertension or not. 1 = hypertension, 0 = no hypertension.
Heart_Disease: Describes whether a patient has hypertension or not. 1 = heart disease, 0 = no heart disease.

Ever_Married: Describes whether a patient has ever been married. “No” = never married, “Yes” = has been married.
Work_type: Describes the type of work a patient does for a living. Values: “Children”, “Govt_jov”, “Never_worked”, “Private”, “Self-employed”.
Residence_type: Describes the type of residence that a patient resides in. Values: “Rural”, “Urban”.
Avg_glucose_level: Indicates the average glucose level of a patient.
Bmi: Indicates the body mass index of a patient.
Smoking_status: Indicates the smoking habits of a patient. Values: “formerly smoked”, “never smoked”, “smokes” and “Unknown”.

EDA:

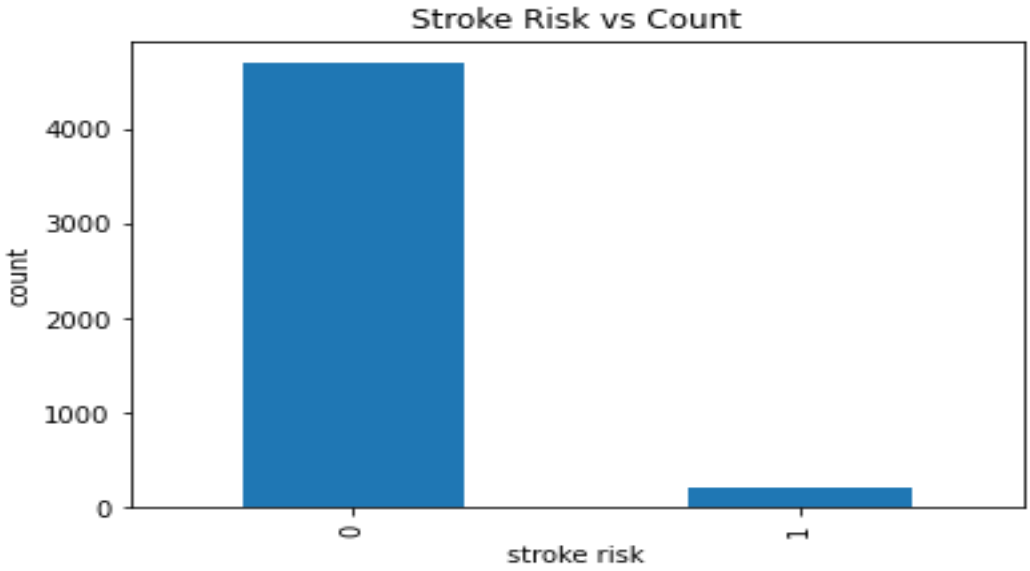
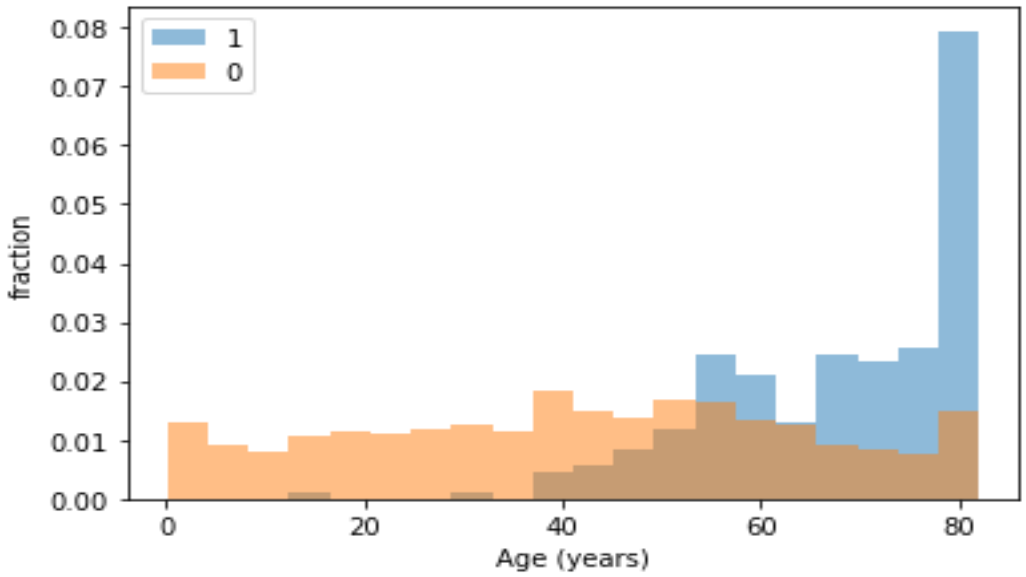
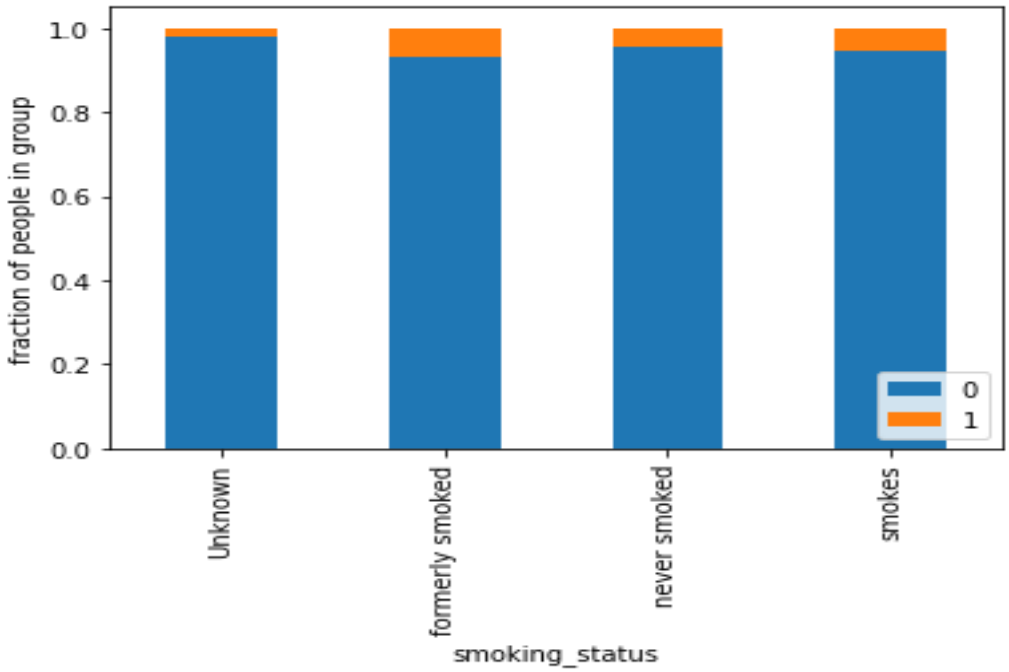


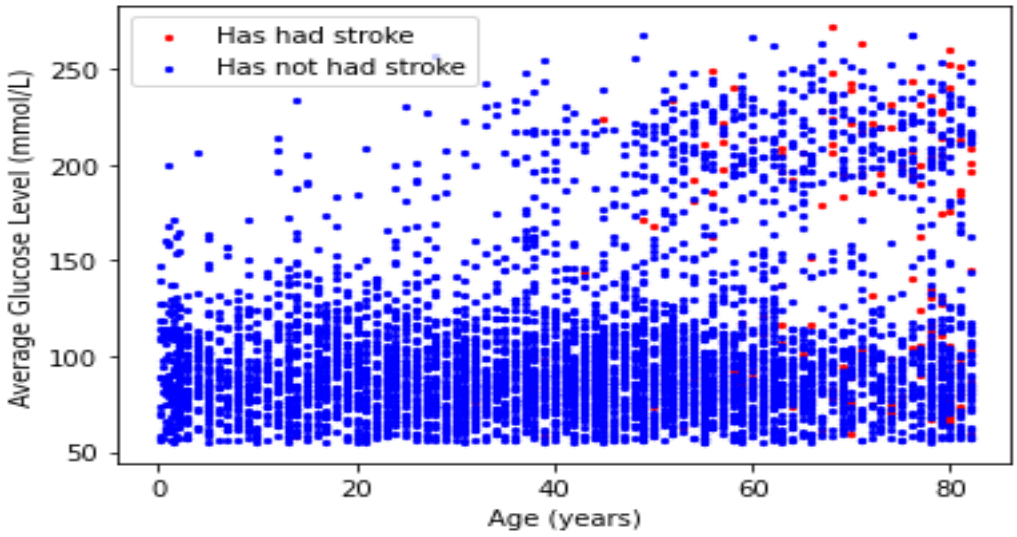
Figure 1: This bar plot compares our target variable outcomes. A patient either hasn’t had a stroke (“0”) or has had a stroke (“1”). This plot shows that our data is unbalanced and splitting the data will need to reflect this.



<u>Figure 2:</u> This histogram shows how the age of patient compares to their risk of having a stroke. We can see that there is an even spread throughout all ages if someone had not had a stroke. However, if a patient has had a stroke, they tend to be above 40 years old.



<u>Figure 3:</u> This stacked bar plot compares the risk of a stroke verses the smoking status. As described in the dataset description, "Unknown" means information about their smoking habits was not provided by the patient



<u>Figure 4:</u> This plot shows the compares the continuous variables of age and average glucose level. It also makes the distinction between patients that have had a stroke and have not. We can see that being older in age has an impact of the risk of having a stroke. Average glucose level seems to be more spread-out regarding likelihood of a stroke.

Methods:

Splitting the Data:

I decided to use a stratified k-fold split on my data since it is quite unbalanced. Only 209 patients have been labeled a stroke risk. This is only 4.26% of the datapoints. The data is IID because these are individual patients. This dataset is also not a timeseries as there is no information to tell when this data was collected. I utilized an 60/20/20 stratified k-fold split due to the size of the dataset. There are only 5,110 datapoints, 201 of which were eliminated because they did not provide a body mass index for the patient. The reason I chose to eliminate these datapoints is discussed below in the preprocessing section.

Preprocessing:

After eliminating the ID feature column, 10 features will be used to predict the target variable. Since “id” is a given identifier for a patient and provides no information for predicting stroke. I used the StandardScaler preprocessor for the age column because this continuous feature roughly follows a normal distribution. I then used OneHotEncoder for the "work_type", "smoking_status", 'gender', 'ever_married', and 'Residence_type' features because these categorical columns needed to be given dummy variables so the machine learning pipeline would function. As I mentioned above, I decided to eliminate the 201 rows that were missing values for body mass index. I did this because this is data about the healthiness of a patient, and we are classifying their outcome of having a stroke. This is very sensitive data and imputing a value could make our classification less accurate. Since these datapoints only account for 3.93% of the dataset, it should not impact the results.

ML Models:
I selected four models from SchiKit Learn to perform the analysis of this project: Logistic Regression, support vector classification (SVC), Random Forest Classifier, and XGBoost Classifier.

Logistic Regression is a simple binary classification model that I suspected would suit well with this dataset. I implemented the model with the L2 penalty because L1 would send many of the feature coefficients to zero and with limited features, this is not ideal. The tuning of this model worked with C, or the inverse of alpha, which controls the strength of the regularization. I used C values, which were the reciprocal of a uniform log space on the interval of -5 to 5 with 51 values.

I used SVC for similar reasons to Logistic Regression. However, SVC will provide a non-linear attempt at making classifications. To tune this model, I focused on gamma and C, which control the kernel coefficients and regularization, respectfully. For gamma, I utilized a log space with 11 values from -2 to 2. For C, I utilized the reciprocal of 11 log space values from -2 to 2. With multiple parameters to tune, I utilized a grid and looped through each of these combinations.

Next, I modelled predictions using a random forest classifier. To tune this model, I selected the parameters: maximum depth and maximum features. These two features control the number of branches that are formed in each tree and the number of features used in each tree, respectively. Max depth was spaced linearly with 20 values between 1 to 20. Max features was linearly spaced with three values between 0.5 and 1.

Finally, I implemented an XGBoost classifier. The power of XGBoost as compared to other models is impressive and is typically the best choice for a multitude of datasets. I tuned three parameters for XGBoost: learning rate, column sample by tree, and sub sample. Learning rate was tuned with 4 values linearly space from 0 to 4 while column sample by tree and sub sample were tuned with 5 values linearly spaced from 0.01 to 1.

Results:

Baseline: This problem was a binary classification problem, where 1 indicates a patient had a stroke and 0 indicates a patient did not have a stroke. Seeing that the data was heavily unbalanced (class 1: 209, class 0: 4700), we could not use accuracy scores but instead F1 score was utilized. I developed baseline scores for each parameter by setting all 4909 patients' outcome to class 1, which indicated they had a stroke. This provided a F1 score of 8.17% respectively. I then used this value to compare against each optimal model's mean and standard deviation of F1 score. From this analysis, I found which models performed worse than the baseline due to simplicity of the model versus the complexity of the data and which models did better than the baseline.

Logistic regression performed the worst of all the models. It seemed to be much too simple of a model for this dataset and results in an 85.90% decrease in F1 score (3.5148 standard

deviations). The Random Forest model unexpectedly performed poorly as the data was quite suitable for a decision tree algorithm. It yielded a F1 score that decreased by 62.90% (1.5432 standard deviations). SVC also performed poorly with a 0.38% decrease in F1 score (0.007206 standard deviations). XGBoost performed the best for this problem. It provided a 61.8% increase in F1 score (0.8127 standard deviations). It may seem odd that the percent changes are so large, but this makes sense given the lack of precision in the baseline for F1 score. We can look at the standard deviations, which are quite small to ensure that the uncertainties due to splitting, and random states is also small.

F1 scores	Logistic Regression	SVC	Rand Forest Classifier	XGBoost
Average F1 (%)	1.152	8.139	3.0303	12.163
Percent change (%)	-85.90	-0.38	-62.9	48.87
F1 Standard Deviation	0.01996	0.03906	0.03329	0.04916
Number of Std. Dev.	3.5148	0.007206	1.5432	0.8127

Table 1: Depicts the results of the four optimized models, with an emphasis on the number of standard deviations the baseline F1 Score is from each model's average F1 Score.

Feature Importance: Utilizing the XGBoost model that performed the best of all models analyzed, I calculated the feature importance of each feature. To do this, the perturbation was used, and this resulted in the plot below. Age, body mass index (BMI), and average glucose level were the top three features that impacted the model . This is what we expected from EDA as show in figure 2 and 4 above. The values for age, BMI, and average glucose level are 0.0833, 0.2105, and 0.2307 respectively.

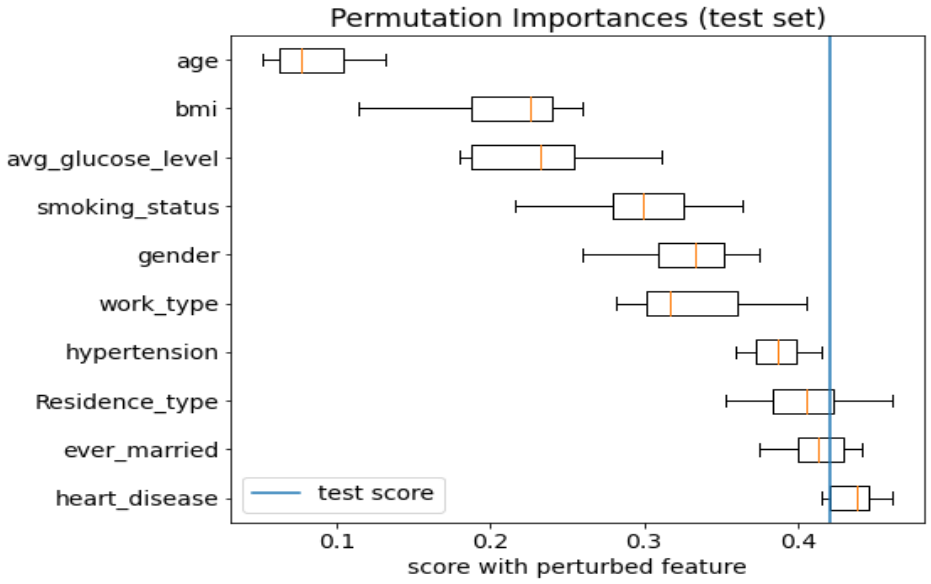
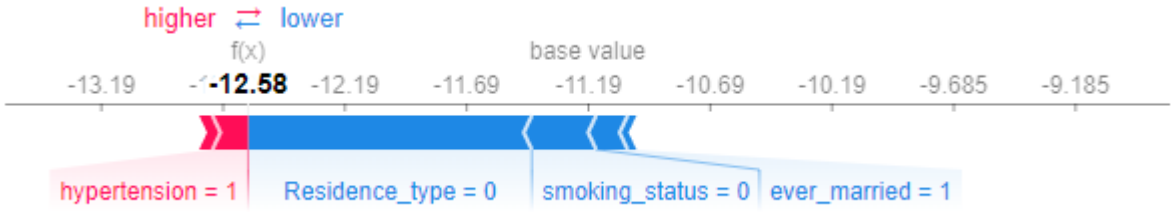


Figure 5: Feature importance plot which indicates age, BMI and average glucose level strongly influence stroke outcome.

SHAP: Utilizing SHAP we can see that for given local features, which ones impact the outcome of a patient. From figure 6, we can see that their residence type and smoking status heavily sway them into the class 0 prediction.



<u>Figure 6:</u> Demonstrates the local feature importance of one of the patients within the dataset.

Outlook:

The amount of data and its balance was a huge limitation of this project. It is difficult for a model to make prediction of class 1 since even with a 60/20/20 split there is only about 42 class 1 outcomes out of 982 total in a testing/validation set. There are multiple similar analyses done with this data that utilize SMOTE which can help with this issue. However, this means sampling the data and adding to the dataset, which can be costly in an analysis of a patient's health outcomes. We do not want to infer about a patient. We need to know for sure what their parameters are to make an accurate prediction. Including additional features would also help a model to build stronger predictions because it can for example build deeper trees in XGBoost. Such features could include, fitness of the patient, eating habits, or drinking habits to see if these have any feature importance to predicting a stroke outcome.

References:

C. S. Nwosu, S. Dev, P. Bhardwaj, B. Veeravalli and D. John, "Predicting Stroke from Electronic Health Records," 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2019, pp. 5704-5707, doi: 10.1109/EMBC.2019.8857234.

M. Monteiro et al., "Using Machine Learning to Improve the Prediction of Functional Outcome in Ischemic Stroke Patients," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 15, no. 6, pp. 1953-1959, 1 Nov.-Dec. 2018, doi: 10.1109/TCBB.2018.2811471.

Kaggle, https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

Github Repo: https://github.com/lkania1/data1030project