Using ML Models to Predict a Stroke

Tiara Allard, Jackson Glass, Connor Rose, Kayla Sprincis

Summary

This project explores how various machine learning models can be used to best predict whether or not a patient will suffer a stroke. We aim to classify if a patient will have a stroke or not, utilizing a dataset which contains many numeric and categorical features related to the patients' physical and mental wellbeing. In particular, we aim to determine which model performs best to predict stroke, as a model with greater predictive power will allow for greater insight into patient risk, and better patient outcomes. We utilize various combinations of models, datasets, and success metrics to determine how the problem of predicting stroke should be understood.

In our analysis, we compare the effectiveness of three different kinds of models, specifically Regression, Decision Trees, and k-Nearest Neighbor. For each model, we explored how variable selection influenced model success, using variables which were cleaned and analyzed for correlations in our initial exploratory data analysis. To have a quantitative measure of model effectiveness, we measured the Root Mean Squared Error (RMSE) and R Squared (R^2) of the model on testing data after going through the training process.

From this exploration, we determined that this is a problem best solved using logistic regression. Logistic regression specializes in predicting binary outcomes, such as whether or not a patient has a stroke. The decision tree and kNN models had lower performances on the testing dataset than logistic regression did. Despite this, our overall conclusion is that the problem is much better approached not as a binary classification problem, but that it would be more effective to predict probabilities of stroke for each patient. Logistic regression also does this effectively, as it predicts row by row probabilities of the target variable, and would work much better for overall patient outcomes than a model which predicts only binary outcomes.

Data

The stroke dataset was largely already clean, but to prepare it for analysis, we performed some minor data cleaning, as well as analysis and transformation of variables to have effective data for our models.

The initial dataset included 13 columns, including stroke, which we narrowed down to 11. We dropped one column, which had no name and no data, as well as the column 'id', as this number was just for identifying patients, and if not removed, the numerical values could influence the model. We also renamed the residence_type variable to have uniform capitalization across the dataset. Finally, the dataset had a few NaNs, all of which occurred in the BMI column. We decided to impute these values with the mean. Since BMI does not strongly correlate with stroke, and is approximately normally distributed, we decided to impute, as the other data in these rows could be valuable, even if BMI is not a strong predictor.

After performing data cleaning, the dataset included 10 predictor variables along with stroke. Numeric features remaining include

- age - The age of the patient in the survey, ranges across 80 years. Distribution is mostly uniform
- hypertension - Whether or not the patient has hypertension, 1 meaning hypertension.
- heart_disease - Whether or not the patient has heart disease 1 meaning heart disease.
- avg_glucose_level - The average glucose level of the patient, the distribution has a large peak at around 80, and a small peak at around 220.
- bmi - The body mass index of the patient, normally distributed around 30.

Along with our numeric variables, 5 categorical variables also remained in the dataset, including

- gender - The gender of the patient, a slight majority of the dataset being female.
- ever_married - Whether the patient has been married at any point, a slight majority having been married.
- work_type - The type of job of the patient, including federal employment, self employment, private employment, caretaker for children, or never worked.
- residence_type - Whether or not the patient lives in an urban or rural environment, evenly split in the dataset.
- smoking_status - Whether the patient smokes or not, includes formerly smoked, currently smokes, unknown, or never smoked.

To analyze which of these variables would be the most effective predictors of stroke, we analyzed how the variables correlated with each other and the stroke variable. To start off with this, we first created a correlation matrix of all the numeric variables. From this, we could see that age had the strongest correlation with stroke, bmi had very little correlation, and the 3 other variables had small positive correlations. This tells us that age will likely be a useful predictor, bmi will not be, and the other three variables will be slightly useful.

Next, we created dummy variables for each categorical, to allow them to be used numerically for training the different kinds of models. From these encoded dummy variables, we then analyzed each of their correlations with the stroke variable. None of the individual dummy variables had particularly strong correlations with stroke, but there were some trends that emerged across categorical variables. Ever_married, work_type, and smoking_status had the strongest correlations with stroke, and would likely be the strongest predictors among categorical variables.

To further process the data to allow for a wide possibility of features to be explored, a function was created to normalize the data. Normalizing the data can be important, especially if the magnitude of the data are wildly different across categories. Since some variables reach the hundred, whereas others are only encoded to

be 0 or 1, feature scaling can be impactful in training.

Finally, we also created polynomial features for the data up to degree 2. Polynomial features can be helpful, especially in exploiting nonlinear relationships in the data. Since there were no exceedingly strong correlations with the data, these nonlinear features could be useful in finding otherwise hidden correlations between data.

Overall, we narrowed the dataset down to a few key variables which we believe will be the strongest predictors of stroke in patients. The variables which we believe will be the best to train models on are age, hypertension, heart_disease, glucose, ever_married, work_type, and smoking_status. Though the other variables could be useful, they do not correlate as strongly with stroke, and will likely be worse predictors. Overall, these 7 variables, along with feature scaling and polynomials provide a solid base of data on which to train and compare the different models.

Results

The team created several models to find the optimal option to predict if a person would have a stroke. Each one addresses the features of interest to predict the target, and finds the R^2 and RMSE value as a measurement of how well the model fits. The goal of these models was for them to result in R^2 values greater than about 0.0872. This value was the result of a basic Linear Regression model that used all features and was created to provide a baseline for the problem being solved in this project. The models used were Linear Regression, Logistic Regression, Decision Trees, and k-Nearest Neighbors. Each model used the presplit training and testing data sets.

Linear Models were used since they are a powerful prediction method that is simple to use. To create models a function using the tools from sklearn was created. The strategy used was creating and testing a large amount of models to find the best fitting one. Tests involving numeric variables were done with both the scaled and unscaled features in case one version of the data resulted in better predictions. The variables used for this model were age, hypertension, heart disease, glucose, ever married, work type, and smoking status. For the work type variable only the dummy variables for self employed and full time child care were used since they had the highest correlation. The other dummy variables created from work type were excluded to create a more precise model. Additionally formerly and currently smoking were combined to create a general “has smoked” variable. When creating models the different variable groups, such as numerical and categorical, were looked at individually and together. A total of 12 different combinations of the variables were used to fit a model. The models were fitted with the training data and then predictions were made with the testing data. The model with the best performance was created using all scaled numerics values and the polynomial family of the numerics with a degree of two. It had a R^2 value of 0.0871 and a RMSE of 0.2060.

Logistic Regression was used since the target variable, stroke, is a binary 1 or 0. This model predicts probability of an

outcome rather than just yes or no. To create the models a function using the tools from sklearn was created. The R^2 was separately calculated since the Logistic Regression package does not provide a method to find it. All of the strategies to produce results, except for the tool used to fit models, were the same as Linear Regression. The model with the best performance was created using all unscaled numerics. It had a R^2 value of 0.0909 and a RMSE of 0.2056. It is also important to note that with this strategy a table can be created that gives a probability of stroke for each row of data, which provides a deeper insight into what increases likelihood of stroke than a binary classifier does.

Decision Trees were used as they are a very powerful model in classification. To create the models a function using the tools from sklearn was created. The R^2 and RMSE values were also calculated using sklearn functions. The cleaned data was used, which included scaled and normalized data, which has no effect on the performance of the decision tree. In addition, all categorical variables were one-hot encoded. Multiple groups of variables were run on this model, to see which best classified test data to 'stroke' or 'no stroke.' The three groups were (1) a group of all variables, including a second degree polynomial expansion of the linear data, (2) a group of all the variables, but not the polynomial expansion of the linear data, and (3) a group of only the categorical data. The models each were run at max depths of 1 to 100, with the best performing depth being picked for the model. All of the models performed poorly, with the best model being the only categorical data. This model had an R^2 value of -0.0304, and an RMSE of 0.2189. The negative R^2 value indicates that the Decision Tree models are incapable of effectively predicting stroke to any extent; This is explained by stroke being an impossible metric to fully predict. The logistic regression revealed that it is incredibly rare for someone to have a greater than 50% chance of stroke, and thus no simple classification model, such as the Decision Tree, is able to accurately predict whether or not someone will have a stroke; It has no reason to believe anyone will. This is demonstrated in the visualization of the decision tree, which is almost exclusively orange boxes, indicating it will predict no stroke. There is only one blue box, which has very specific parameters (only 15 of over 4000 training data fit these parameters), that the model would predict a stroke. The model will functionally only predict 'no stroke' for everyone. This is why a Decision Tree classifier is an ineffective model to try to predict strokes in patients.

K Nearest Neighbors is the final algorithm that we evaluated in search of a model that correctly fit the data. This strategy predicts the target of certain features based on the distance from training features. The 'k' represents how many nearest data points are being considered. To create the models a function was created that makes the model and fits the data. There were three strategies regarding which features were used. The first utilized all available features, the second used variables with high correlation to stroke (age, hypertension, heart_disease, glucose, and ever_married), and the third considered only

the numerics. It was especially important to consider the normalized version of the data since variables with majorly different averages could skew the data. For this reason the polynomial variables were also normalized. Upon creation the models were fitted with the training data and the testing data was used to make predictions. The model with the best performance was created with all the cleaned data, with a k value of 12. It had a R^2 value of -0.0514 and a RMSE of 0.2899. Despite the poor performance we concluded that this was the best version of a k-Nearest Neighbor model for this data

Conclusion

All of the models produced resulted in what would generally be considered poor fits due to the nature of the data. The data is limited in size and strokes do not occur that often. However, there is still value in this data. When the probability is considered rather than only a 0/1 prediction the results are much more applicable to real medical situations. Within the scope of this project though, it is difficult to make predictions with the limited data set. Below is a summary table of each model's best performance.

Table I
Summary of Results

Model	R^2	RMSE
Linear Regression	0.0871	0.2060
Logistic Regression	0.0909	0.2056
K Nearest Neighbor	-0.0514	0.2899
Decision Trees	-0.0304	0.2189

The shortfalls in using Linear Regression was the type variable the target was. Since stroke was a binary variable and there was a limited number of cases where a stroke did happen, it could not be accurately predicted. The data is non-linear since the target variable only has two options and it does not allow for precise predictions. Additionally, certain conditions will not always mean a stroke will or will not happen, which creates a significant amount of outliers and noise.

This shortfall once again continues in the Decision Tree model. The nature of stroke as a binary variable doesn't inherently make a decision tree classifier a bad model, but the nature of the data did. As the logistic regression shows us, there are substantial differences in the likelihood that an individual has a stroke, but this nuance is lost on a decision tree. Some individuals have an incredibly low (near 1%) chance of having a stroke, while others have a chance closer to 15%. The decision tree is only able to classify whether the individual is more likely to have a stroke or not (>50%), which

almost nobody ever is. This makes the decision tree an ineffective model, as it will almost never predict someone will have a stroke, when many people who have a 15% chance of stroke will end up having a stroke. For these reasons, we can explain why the R^2 value for the decision tree model was negative, and assert that it is a poor model for the data we are analyzing.

The use of k-Nearest Neighbor was ineffective largely due to the characteristics of the data. There were a large number of features compared to the number of observations available, which can create too much noise. Using all of the features created models that are at high risk of being overfit since it is making predictions on data that is too complex. When the features used are limited to fix this, there is not enough data to make predictions and the models are then underfit. As previously discussed many of the variables do not correlate strongly with stroke, so the models would either have had too many features or were working with too limited information.

Logistic Regression was more effective than Linear Regression since this model type was created to deal with binary target variables. Logistic Regression is able to find the connection between two data options. As seen in Table I it was slightly more accurate. However, the Logistic Regression model's value is in its ability to predict probability. For every row of data an individual probability can be computed. This value would be more valuable as a 10% chance of stroke based on previous cases is significantly more risk than if it was less than 1%. Further research on this topic would benefit from looking to fit a model that finds the probability of a stroke.

Appendix

1. Used to explain the reasoning behind why the kNN model fails
    a. https://neptune.ai/blog/knn-algorithm-explanation-opportunities-limitations#:~:text=Furthermore%2C%20Euclidean%20distance%20is%20very,all%20we've%20mentioned%20above.