Stroke Prediction

Kelly Chang, Mats Haneberg, Ryan Hilton, Annika King, Cayman Williams

December 3, 2021

1 Introduction

In the United States, a stroke occurs (on average) every 40 seconds and someone dies from a stroke every 4 minutes [1]. Since strokes are a frequent occurrence, it's important to identify those at risk of having a stroke to take preventative measures.

Since strokes are so prevalent, a lot of modeling has already been done to predict a stroke. For example, a group in Korea used linear regression on a stroke dataset that included features such as hypertension, cardiac disease, and diabetes, with a success rate of 65% [2]. Another group in Taiwan used Cox regression on a long-term stroke study and had an AUC of 78% [3]. However, there does not appear to be a study comparing different types of modeling on stroke prediction. There is a study that analyzed a collection of studies that used models to predict stroke. However, this study focused on the if the models are valid clinical models, not how well the models performed [4]. Moreover, it would not make sense to compare different studies for how well models performed due to non-uniform datasets. Thus, one of the purposes of this study is to use the same dataset on multiple modeling techniques to determine which model(s) and corresponding parameters perform the best. As a note, accuracy is always important in testing a model's performance, but since we are dealing with a dataset where false negatives are detrimental, we will also use recall as a performance measure.

Factors that increase the risk of a stroke are well known in the medical community. Some of these factors can be controlled such as obesity and smoking, while other factors such as gender and age cannot be controlled [5]. Someone could have multiple factors in the controlled category that contribute to higher risk of a stroke, and it can be overwhelming to focus on multiple factors to decrease risk of a stroke. Thus, another aspect is to first determine which features correlate the most with having a stroke. Among those features, we will see which controllable features are correlated the most with having the stroke. This will hopefully help determine an order of factors for an at-risk patient to work on one at a time.

Our data set contains 10 features, namely gender, age, hypertension, heart disease, ever married, work type, residence type (urban or rural), average glucose level, BMI, smoking status. There are 5,110 data points, either labeled stroke or no stroke (in binary form). The dataset is found at [6], from a data scientist at Kaggle. The source of the dataset is confidential; however, the contributor is a master on Kaggle with a degree in data science and has published numerous other medical related data sets. The credentials of the contributor make it less likely that the dataset is made up.

2 Data Cleaning

We note that our code can be found at [7].

We began by going through each categorical variable and checking if any features had an insufficient number of observations to be significant. which we decided needed to be at least five observations. The only category that did not meet this requirement was Other in the gender variable, which only had one observation. We next checked the continuous variables for outliers, and found that one person had one BMI observation over ninety, which is near the highest ever recorded, so we decided to drop this observation.

We also noticed that approximately thirty percent of the column smoking_status was labeled as Unknown. There were a couple ways to interpret this. Firstly, an "Unknown" value could be identical to a NaN value where the data is just simply missing. However, it could also be that an "Unknown" label tells us something about the patient, perhaps that the patient chose not to disclose the answer, and that it should be treated as its own category instead. To address the case that it was simply missing data, we applied two imputation techniques to classify these individuals into the other categories of former smoker, never smoked, and smokes. The first approach we took was to use Logistic Regression with smoking_status as the labels (and excluding stroke from the features), to estimate the smoking status of the "Unknown"'s. This method had an accuracy rating of around fifty-three percent. The second approach used a KNN classifier to do the same thing, and this imputer had an accuracy rating of around fifty percent. Since the accuracy rate for either method wasn't high enough to completely justify the imputation, we also looked at what happens if the data simply treats "Unknown" as its own category when it one hot encodes the smoking status. There weren't large differences in the model metrics (like accuracy, recall, f1 scores, etc.) when using the imputed values versus treating "Unknown" as a category. For this reason, we decided that not imputing was a safer way to go because it involves less synthetic manipulation of the data and doesn't make the model worse in any measurable way. For these reasons, the analysis section uses this this non-imputed dataset.

Moreover, we noticed there are 201 data points that have NaN in the BMI feature. Therefore, we decided to investigate several different approaches to fill in the NaN. The first method we tried was to change BMI from a numerical variable to a categorical variable with the labels underweight, healthy, overweight, and obese. After we did this, we used Logistic Regression to predict the NaN values (similar to smoking status). The imputation was roughly forty-six percent accurate. We also tried to use a KNN imputer to predict the BMI values. This approach had an accuracy of fourty-nine percent. As before, none of these accuracy scores were compelling enough to assure us that the imputation was justifiable. Because of this, We also looked into just dropping the observations with missing BMI values, since they consisted of less than four percent of the overall dataset. However, this approach introduced severe bias issues we found after further analysis, which is what we discuss next.

Overall, roughly five percent of the observations in the dataset had a stroke. Of the 201 observations missing BMI values, forty of them represented a person who had suffered a stroke, which was roughly twenty percent. This shows that the missing values are not representative of the dataset as a whole, and imputing their values or just dropping them likely biases the dataset in unintended ways. This pointed towards there being a possible significant reason behind why these BMI values are missing. Thus, to account for this possibility, we created a BMI binary indicator column that marked whether or not a BMI value was included in the original dataset and then dropped the original BMI column from the analysis.

We implemented a generalized data cleaning function that applied all that we discussed above and that can easily be applied to a similar dataset. The function takes in various parameters in order to accomplish this. The user inputs a boolean to mark whether or not they want their dataset to be one-hot encoded, and if so, they pass in a list of the columns to one-hot encode. The function also takes in a boolean to decide whether or not to change the BMI column to binary or just drop the column entirely. In addition, the user can input a minimum number of observations a feature in a categorical column is required to have in order for the feature to not be dropped from the dataset. Similarly, the user inputs a decimal percentage cutoff for when the amount of NaN in a column as a relation to the size of the dataset is small enough to drop those observations. All of these parameters have defaults to make the function work smoothly for the user. Therefore, this function would be easy to reuse on another similar dataset in the future.

We will now explore some of the data visually.

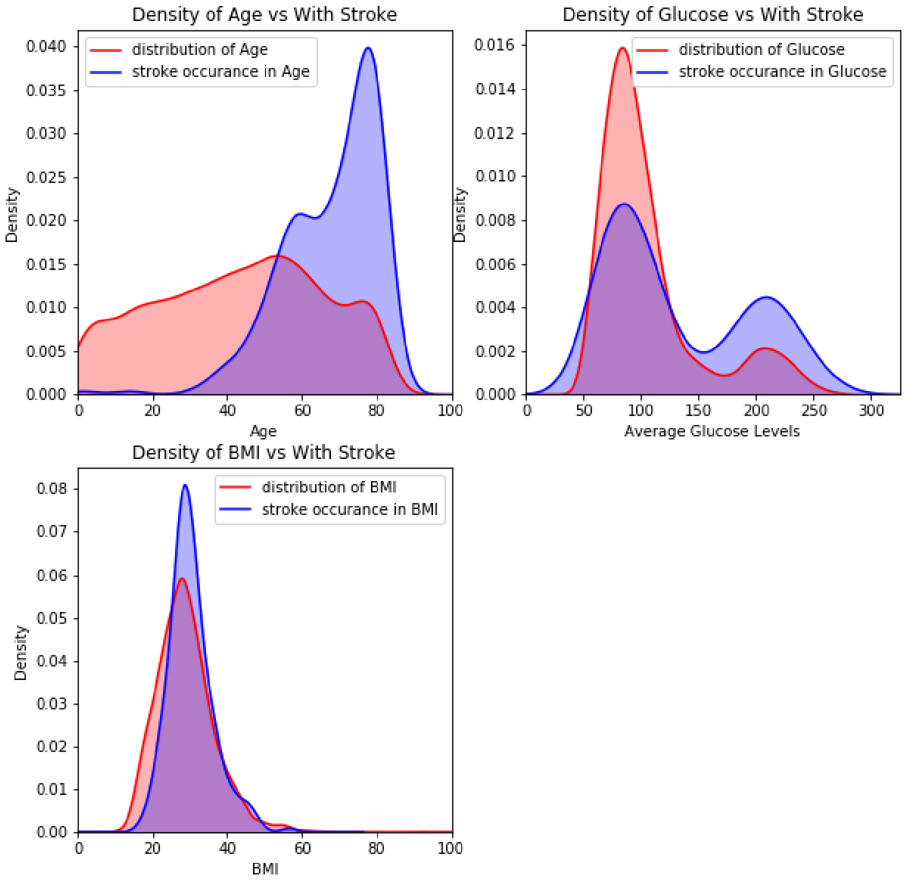


Figure 1: Comparing the estimated distribution of continuous variables in the data (age, BMI, average glucose level) with the estimated distribution of stroke occurrences with regards to the given variable. In red is the estimated density distribution using the entire dataset, and in blue is the estimated density distribution of the dataset restricted to the points labeled as stroke.

The figure above is comparing the estimated densities of the continuous features (age, BMI, and average glucose levels). Plotted in red are the respective estimated density distributions derived from the entire dataset. Plotted in blue are the respective estimated distributions on the data with stroke labels. At first glance, it may appear that BMI and average glucose levels are a good indicator of having a stroke since the red and the blue plots look similar. However, the similarity between these plots reflects that stroke occurrences mirror the underlying distribution of the dataset, suggesting that these variables do not greatly affect stroke occurrence. Basically, because there are more people with average BMI or glucose levels represented in the dataset, the stroke occurrences should also be higher in those regions. This is not enough to justify throwing these features out though. For example, we see that there are relatively more stroke occurrences at the higher glucose levels than can be accounted for by the underlying distribution. It follows that perhaps higher glucose levels relates to higher risk of stroke. Doing a similar analysis on the BMI graph suggests that perhaps the BMI is not very indicative of strokes. Since the blue graph is slightly skewed right, perhaps stroke incidence is very slightly higher for heavier people. In contrast to the glucose level and BMI distributions, the distribution of ages among the entire dataset is fairly equal, but the distribution of ages among those who have had a stroke is weighted heavily toward older subjects. This disparity between the distributions demonstrates that age is likely an extremely useful feature in predicting strokes. We note that we reworked the BMI feature to be binary (see above), but this visualization is still important since glucose levels and BMI are thought to be indicators of a stroke.

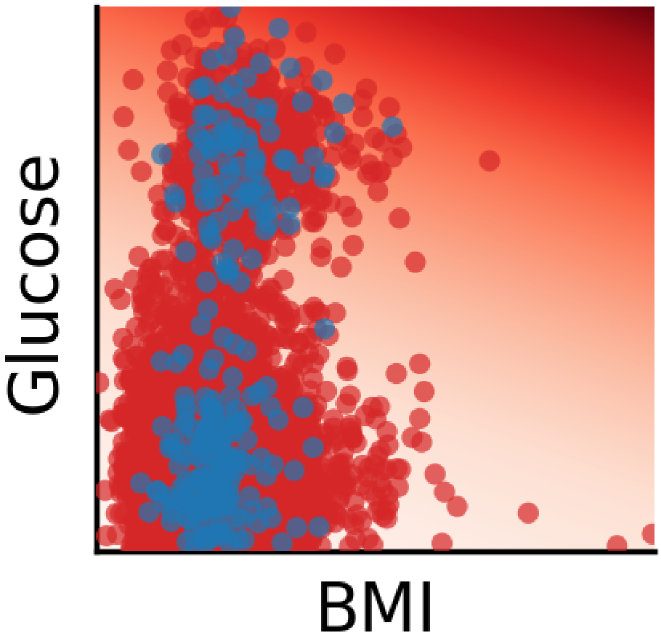


Figure 2: Here we plot BMI vs average glucose levels. Points in blue are classified as stroke and points in red are classified as no stroke. The color in the background corresponds to the probabilities that a point in that region is classified as stroke or no stroke. It's important to note that if there were regions classified as stroke, the background would be blue. This means that the darker red corresponds to high certainty of the data point being labeled as not stroke and lighter corresponds to still being classified as not stroke, but with a probability closer to 0.5.

To further investigate the relationship between BMI, average glucose levels, and stroke, we ran a Logistic Regression model on those 2 features. Figure 2 plots BMI vs average glucose levels. Points in blue represent strokes and points in red represent no stroke. The color in the background corresponds to the probabilities that a point in that region is classified as stroke or no stroke. It's important to note that if there were regions classified as stroke, the background would be blue. This means that the darker red corresponds to higher certainty of the data point being labeled as not stroke and lighter corresponds to still being classified as not stroke, but with a probability closer to 0.5. Since stroke occurrences are so rare in the dataset, this model (which basically just represents the null hypothesis that no one has strokes) would still produce about 95% accuracy but would be extremely useless in any kind of application. Therefore, classification using just BMI and average glucose levels results in an ineffective model. From this we conclude that the density plots above look similar because the BMIs and average glucose levels of those that have strokes resembles the underlying density of people in the study.

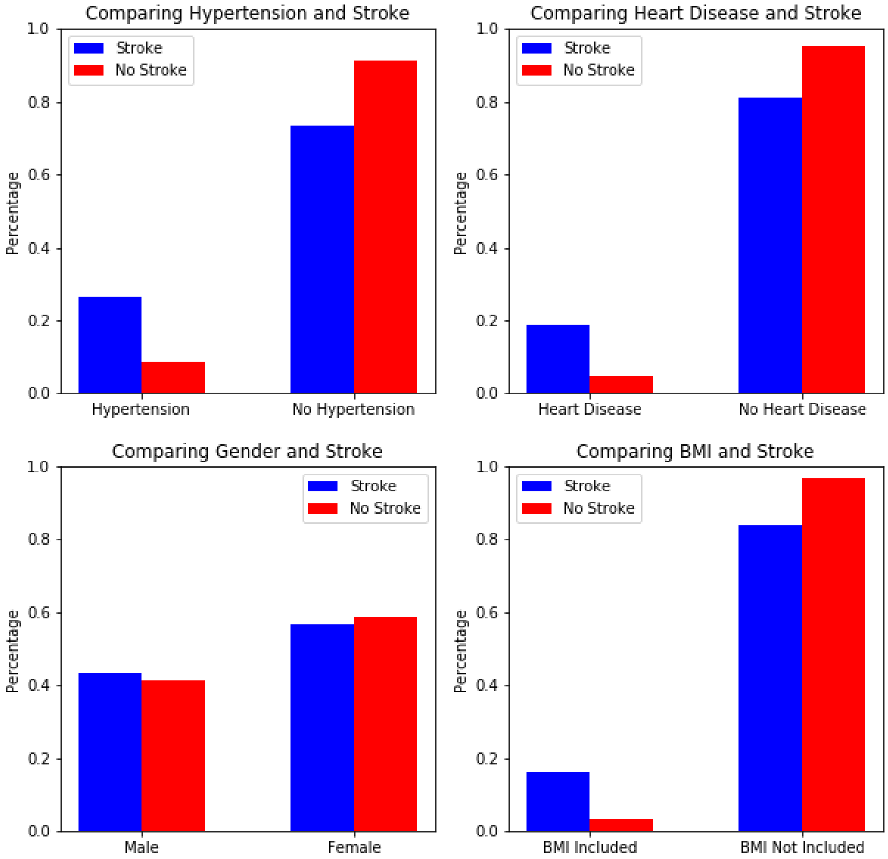


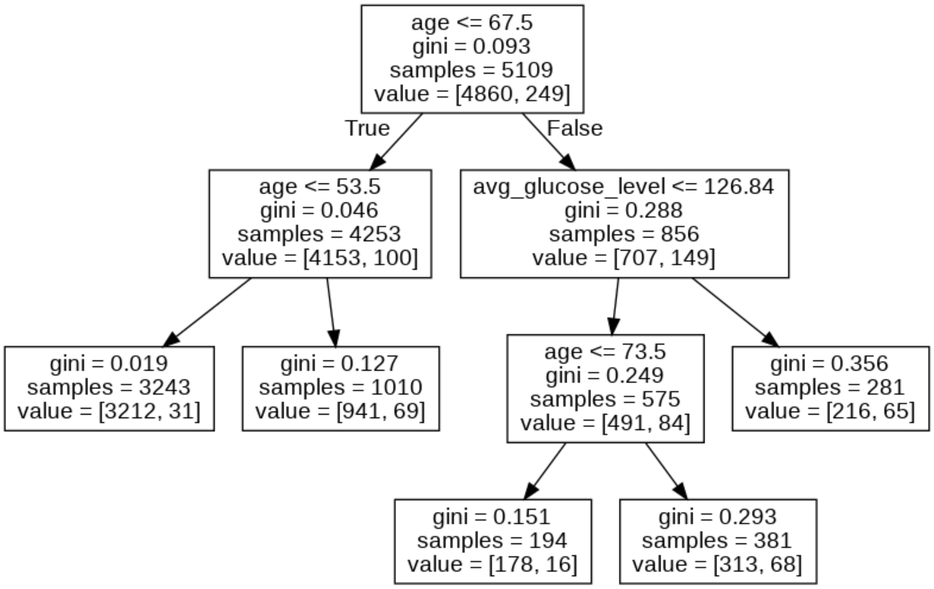
Figure 3: Comparing the percentages of some binary features (hypertension, heart disease, gender, engineered BMI) labeled stroke and no stroke.

The figure above is comparing some of the binary features (gender, hypertension, heart disease, and BMI presence) and how they are distributed among stroke and no stroke. Individually, the features are not fully indicative of having a stroke. This is part of the motivation for why we are modeling with multiple features. At the same time, we can identify some trends that indicate important features. For instance, both hypertension and heart disease features demonstrate a significantly increased likelihood of stroke. As discussed above, we also see that the exclusion of a valid BMI represents an inexplicable increase in stroke occurrence that is not representative of the rest of the data. We would be very interested in identifying what happened in the data collection phase that created this phenomenon.

3 Feature Consideration

As a preliminary test, we ran our data through an Ordinary Least Squares (OLS) model, giving us the coefficients and p-values for all the coefficients. We also cycled through many combinations of features to minimize our Akaike Information Criterion (AIC). In the end, important features included: not being married, never working (the dataset included children), and being a smoker. But as we investigated this analysis closer, the R^2 was extremely low and the p-values of most factors were very high. In summary, our efforts to use OLS were not conclusive because the data is high-dimensional and very inseparable (as seen in figure 2).

As an alternative to OLS, we investigated the columns with the most information gain by fitting our data to a Decision Tree classifier (see figure below). Thankfully, this method was more successful. By running the data through a tree with generic settings, we ended up having the following columns with the most information gain: age, glucose level, heart disease, and being self-employed. After running a grid search to find the best parameters for the Decision Tree, it created a surprisingly simple tree that split only on age and average glucose level. This is further justification for engineering the BMI column to mark the presence of a BMI value in the original dataset since it did not appear as an important feature. However, the largest caveat with finding our optimal factors this way is not having actual correlation coefficients or confidence metrics to validate our conclusions.

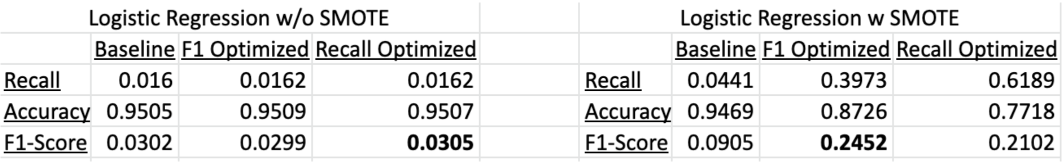


The data visualizations shown in the previous section also provide compelling evidence that the important features identified by the Decision Tree classifier (age, glucose level, and heart disease) provide important information regarding the risk of stroke. In addition, when we ran our Logistic Regression classifier (see next section), we iterated through every possible combination of features in an attempt to increase recall by dropping confounding features. Dropping gender, marriage status, work type, and residence type always resulted in a stronger model, further suggesting that these features are weak indicators of stroke, as we anticipated. In addition, dropping columns like hypertension, age, and heart disease greatly reduced the efficacy of the model, indicating that these features are useful. The use of all these methods help us isolate the features that are most important which was one of our primary objectives for this project.

4 Algorithms and Analysis

Below we analyze common classification algorithms using accuracy, recall, and f1 scores.

4.1 Results Using Logistic Regression



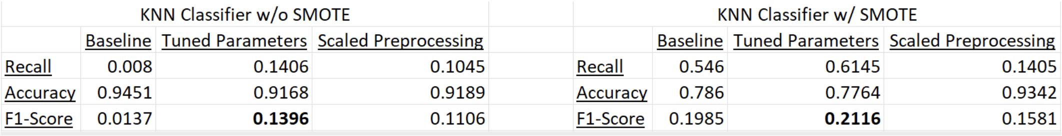
Logistic Regression w/o SMOTE

	Baseline	F1 Optimized	Recall Optimized
Recall	0.016	0.0162	0.0162
Accuracy	0.9505	0.9509	0.9507
F1-Score	0.0302	0.0299	**0.0305**

Logistic Regression w SMOTE

	Baseline	F1 Optimized	Recall Optimized
Recall	0.0441	0.3973	0.6189
Accuracy	0.9469	0.8726	0.7718
F1-Score	0.0905	**0.2452**	0.2102

4.2 Results Using K-Nearest Neighbor Classifier



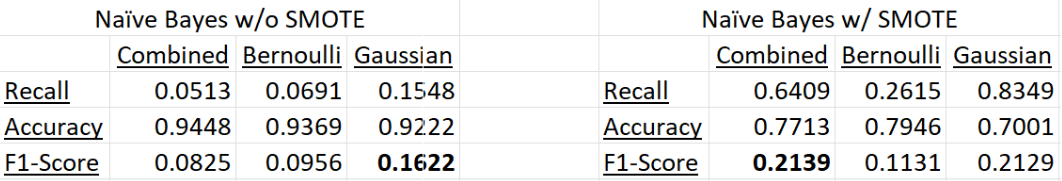
KNN Classifier w/o SMOTE

	Baseline	Tuned Parameters	Scaled Preprocessing
Recall	0.008	0.1406	0.1045
Accuracy	0.9451	0.9168	0.9189
F1-Score	0.0137	**0.1396**	0.1106

KNN Classifier w/ SMOTE

	Baseline	Tuned Parameters	Scaled Preprocessing
Recall	0.546	0.6145	0.1405
Accuracy	0.786	0.7764	0.9342
F1-Score	0.1985	**0.2116**	0.1581

4.3 Results Using Naive Bayes Classifier



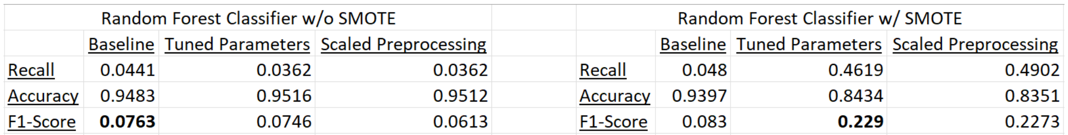
Naïve Bayes w/o SMOTE

	Combined	Bernoulli	Gaussian
Recall	0.0513	0.0691	0.1548
Accuracy	0.9448	0.9369	0.9222
F1-Score	0.0825	0.0956	**0.1622**

Naïve Bayes w/ SMOTE

	Combined	Bernoulli	Gaussian
Recall	0.6409	0.2615	0.8349
Accuracy	0.7713	0.7946	0.7001
F1-Score	**0.2139**	0.1131	0.2129

4.4 Results Using Random Forests Classifier



Random Forest Classifier w/o SMOTE			
	Baseline	Tuned Parameters	Scaled Preprocessing
Recall	0.0441	0.0362	0.0362
Accuracy	0.9483	0.9516	0.9512
F1-Score	**0.0763**	0.0746	0.0613

Random Forest Classifier w/ SMOTE			
	Baseline	Tuned Parameters	Scaled Preprocessing
Recall	0.048	0.4619	0.4902
Accuracy	0.9397	0.8434	0.8351
F1-Score	0.083	**0.229**	0.2273

4.5 Analysis of Learning Algorithms

We tried a variety of classifiers in an attempt to create a model that could accurately (and more importantly, with high recall) predict stroke occurrences using the health data provided. One important characteristic of our dataset is that it is heavily imbalanced. The null hypothesis (predicting no stroke occurrences across the board) results in a model with about 95% accuracy. Thus, it is extremely important that we focus on the recall of the model and the f1 scores, which represent a sort of balance between the precision and the recall scores. Experimentally, raising a model's recall by changing model parameters or using SMOTE to oversample the training data results in a lower accuracy so there is a practical trade-off between the two scores. However, the null hypothesis model is quite obviously useless so it's a trade-off we need to make in order to get any kind of applicable model.

Treating the natural imbalance in the data is of utmost importance. The Syntethic Minority Oversampling Technique (SMOTE) is used to take the minority class (in this case, stroke occurrences) and create fake examples that are close to the other stroke occurrences in the feature space. In all of our attempted models, oversampling with SMOTE greatly improves both recall and the f1 score from the corresponding baseline model that was trained only on the provided data. Regardless of the classifier type, the models were effectively useless without oversampling. Thus, the most important step in model creation for our dataset is using SMOTE in our pipeline. We also played around with other preprocessing steps such as using sklearn's StandardScaler, but this either reduces the efficacy of the model or results in no change at all.

Only two of the provided features are continuous/numerical, so Gaussian Discriminant Analysis was not an appropriate tool to use due to the fact that it assumes that the features are normally distributed. Instead, we turned to Naive Bayes as a potential model because it can handle both Gaussian and Binary distributed features. Due to the fact that Naive Bayes assumes independence between the features, we can split the dataset into Gaussian and Bernoulli features and get final probabilities at the end by multiplying the probabilities of the two separate models together. For each of the sets, we can train a Naive Bayes model of the corresponding type on just the appropriate features. Then, rather than using the predict method to get labels, we grab the probabilities for each class. Multiplying these newly calculated probabilities together from both the Gaussian and Bernoulli models, we get a "combined" probability that accounts for all of the features in the data. Then, assigning class labels is as easy as choosing the argmax of the resultant probabilities for each input. In addition to being theoretically sound (by the assumed independence of features), we experimentally see that the combined model improves on the f1 scores of both subset models. Using this combined model (w/ SMOTE), we end up with a recall of 0.641, an accuracy of 0.771, and an f1 of around 0.214. Unfortunately, there aren't a whole lot of model parameters that a GridSearch can iterate over for the Naive Bayes model, the only possibility being "var_smoothing", which is just used for calculation stability on GaussianNB and "alpha" a smoothing parameter for BernoulliNB. Thus, the best we can do with Naive Bayes is an f1 score of 0.214. Although this combined method results in a slightly better f1 score, the Gaussian Naive Bayes model using only the continuous variables (age and glucose levels) has an astounding .835 recall which far surpasses any other model we investigated. The accuracy and precision suffer a bit (hence why the f1 score is lower) but this may be the best model we created because false positives are much less damaging than false negatives.

Another option we explored as a way to predict stroke occurrences was a K-Nearest Neighbors classifier. Without SMOTE or any fine tuning of parameters, the model is completely useless because of its low recall value of .14 (as shown above). Oversampling immediately yields much better results with a recall score of about 0.51, an accuracy of 0.79, and an f1 score of around 0.19 which is actually already decent compared to our scores on the Naive Bayes. Unfortunately, the tuning of model parameters using GridSearch doesn't improve the SMOTE model significantly. It provides a slight jump up to a recall of .59, a slight accuracy decrease to .78, and a slight jump up to a f1 score of about .21. Thus, even with all the improvements we could make with our KNN classifier, it makes no improvements over the Naive Bayes model.

Like the other models we have run, it seems at first glance that the SKLearn Random Forest classifier performs well by looking at the OoB (Out of Bag) score, which would on average be about 95%. However, our recall was abysmal because the model predicts no strokes across the board. This makes sense because of the imbalanced dataset and the way trees averages across the members of each leaf. To counteract this, we implemented SMOTE on our dataset to get more positive stroke classifications. This increased our OoB score to 97% and doubled our recall, but it still produced much worse results than the other models we have tried. Finally, we used a GridSearch to find the optimal parameters for our random forest and we finally got results comparable to our other classifiers, as shown in Figure 6 above. This occurs because we can greatly reduce overfitting by making the max_depth parameter relatively low like 3 or 4. With its f1 score of .21, we conclude that the Random Forest classifier matches the performance we were able to get from the other models. However, its recall scores were much lower than either the K-Nearest Neighbors classifier and the Naive Bayes classifier so we would probably use a different model because of how costly false negatives could be.

Finally, we also used a Logistic Regression classifier to try to predict stroke occurrence. The baseline scores were awful, regardless of whether or not SMOTE was applied which was very discouraging initially. Trying to tune regularization parameters through a GridSearch was also ineffective. As a last resort, we looked at dropping features that may be hurting the classifier's ability to perform. To do this, we iterated through every potential combinations of features in the dataset and compared the resulting f1/recall scores. By choosing which scoring system to use, we were able to optimize the model with special regard for the attribute we thought was most important. Luckily, this finally resulted in a model that could compete with our other models. The f1 optimized score was a whopping .245, which is a degree higher than we were able to achieve in the other models. However, the recall suffered and dropped to a .397. So even though the f1 score was the highest, it was mostly because of precision increases and not a better recall. The columns that were dropped to optimize the f1 score were gender, marriage status, work type, and residence type. The columns dropped to maximize recall were gender, smoking status, work type, and residence type. Knowing what was dropped is very useful in analyzing the importance of certain features (section 3). Even when optimized for recall, our Logistic Regression performed about as well as our other models.

It is worth noting that while these f1 scores for our models may not seem very high, they exceed the scores of the other models that were being discussed/presented on Kaggle.

To conclude, with the the exception of the Random Forest classifier and the f1 optimized Logistic Regression, the tuned models get f1 scores of around .21 and recalls of about .62. These models are mostly interchangeable because their scores are so similar. However, the Gaussian Naive Bayes is of particular interest because of its extremely high recall and would be our recommended model in most situations, even though its accuracy suffers a bit.

5 Ethical Implications

The ethics of using modeling and machine learning techniques in medical diagnostics have been debated since the idea of using computers to diagnose humans has been around. One major issue is informed consent [8]. If a doctor uses a model to diagnose a patient, would the doctor have to give a lesson on Linear Regression so the patient understands where the diagnosis came from? This is of course impractical, but neglecting to inform the patient leads to uninformed consent. This is closely related to the transparency of the model [8]. Most likely, if a model is accepted as a good predictor of stroke, the creators of the model would want to profit off of it, meaning the algorithm would be kept in a black-box. This would make it difficult to determine if the algorithm is biased in any way.

One major fallacy in terms of models is believing that the features cause the outcome when really the model only tells us that they are correlated with the outcome. Our analysis does try to determine which features are more correlated with stroke, but was not very successful. If there is a successful model that determines which features are more correlated, the doctor and patient could interpret this as those features leading to stroke. With this in mind, the features that were found to be more correlated with stroke in this project should not determine a plan for a patient. That should ultimately be up to the doctor, who can choose to take correlation into consideration.

As mentioned in the introduction, the source of our dataset is confidential, meaning that we don't know how it was collected or created. For future purposes, a different dataset should be collected and used for similar analysis.

5.1 References

[1] "Stroke Facts." Centers for Disease Control and Prevention, Centers for Disease Control and Prevention,25 May 2021, https://www.cdc.gov/stroke/facts.htm#:~:text=Someone%20in%20the%20United %20States,minutes%2C%20someone%20dies%20of%20stroke.&text=Every%20year%2C%20more %20than%20795%2C000,are%20first%20or%20new%20strokes.

[2] Min S, N, Park S, J, Kim D, J, Subramaniyam M, Lee K, -S: Development of an Algorithm for Stroke Prediction: A National Health Insurance Database Study in Korea. Eur Neurol 2018;79:214-220. doi: 10.1159/000488366

[3] Chien, Kuo-Liong, et al. "Constructing the Prediction Model for the Risk of Stroke in a Chinese Population." Stroke, American Heart Association, 29 July 2010, www.ahajournals.org/doi/full/10.1161/STROKEAHA.110.586222.

[4] Wang, Wenjuan, et al. "A Systematic Review of Machine Learning Models for Predicting Outcomes of Stroke with Structured Data." PloS One, Public Library of Science, 12 June 2020, www.ncbi.nlm.nih.gov/pmc/articles/PMC7292406/.

[5] "Risk Factors for Stroke." Johns Hopkins Medicine, https://www.hopkinsmedicine.org/health/conditions-and-diseases/stroke/risk-factors-for-stroke.

[6] https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

[7] https://github.com/CaymanWilliams/StrokeProject.git

[8] Gerke, Sara, et al. "Ethical and Legal Challenges of Artificial Intelligence-Driven Healthcare." Edited by Adam Bohr and Kaveh Memarzadeh, Artificial Intelligence in Healthcare, U.S. National Library of Medicine, 26 June 2020, www.ncbi.nlm.nih.gov/pmc/articles/PMC7332220/.