BANA 273: MACHINE LEARNING ANALYTICS

GROUP PROJECT REPORT

STROKE PREDICTION ANALYSIS

Group 15

GROUP MEMBERS:

Lei Ye: lye10@uci.edu

Haixin Zhao: haixiz3@uci.edu

Flavio Wang: flaviow@uci.edu

Ryan Donaghy: rdonaghy@uci.edu

Chirag Madhukar: cmadhuka@uci.edu

Executive Summary

Stroke has become a common disease recently. The CDC estimates that every 4 minutes, someone dies of stroke and that every year, more than 795,000 people in the United States have a stroke.Experts believe that the best prevention for strokes is a healthy lifestyle, but which healthy habits are the best to prevent this medical emergency? Our dataset contains various parameters which are used to predict the likelihood of getting a stroke. We make some exploratory data analysis on the given parameters. We also build six machine learning models to make predictions on if people are likely to get strokes. The six machine learning models include logistic regression, decision trees, random forests, SVMs, Gaussian Naïve Bayes and KNNs. We will analyze these six models and make some adjustments on them to optimize their accuracy performance.

https://www.canva.cn/design/DAExFF4eyxQ/VSbPN6jymky_RIG30ml7qw/view?utm_content=DAExFF4eyxQ&utm_campaign=designshare&utm_medium=link&utm_source=publishsharelink

Table of Contents

Introduction

As access to both information and healthcare have become more readily available, there has been a steady increase in the number of people interested in monitoring their health. Many illnesses that were previously regarded as incurable are now able to be treated as technology and innovation in the medical space has grown. However, there are still a number of diseases modern medicine has yet been able to solve. One of the most common, and deadly is Stroke, a common disease resulting from the sudden death of brain cells due to lack of oxygen. “Every 4 minutes, someone dies of stroke. Every year, more than 795,000 people in the United States have a stroke.” (Stroke facts 2021)
Although modern technology has been able to provide cures for a multitude of ailments and diseases, scientists have yet been able to cure this disease. Popular opinion indicates that the most effective way of decreasing the probability of having a stroke is by building a healthy lifestyle, however, which habits are the most important in reducing the probability of a person suffering a stroke? Our group plans to explore several of the most commonly accepted healthy habits to determine their efficacy, as well as weighing to what extent they might lower someone's risk of stroke.

Historically stroke was viewed as a disease suffered exclusively by older people, although in reality it can affect people of all ages. For some time it has been commonly accepted that stroke has a close association with heart disease and smoking history, although the data suggests this might not always be true. Our team built six machine learning models and compared their performance by evaluating their accuracy and ROC scores. In an effort to better understand our results, we also resampled the data and added feature selections to the models. In this report, we will introduce our findings and our machine learning models which were able to predict the likelihood of suffering a stroke with high accuracy. Finally, we will provide some suggestions to the public on which kind of lifestyle can reduce the risk of stroke. We all only live once, so it is crucial that we do everything in our power to make healthy choices.

Exploratory Data Analysis

Data Summary and Description

Our data is retrieved from Kaggle, which is an online repository where users can find and publish a variety of datasets. Our dataset contains 5110 rows and 12 columns. The target variable is “stroke” (1 represents that patient has a stroke; 0 represents that patient does not have a stroke). Below are 10 independent variables:

1. gender: Categorized as “Male”, “Female” or “Other”
2. age: age of each patient
3. hypertension: 1 represents that patient has hypertension; 0 represent that patient does not have hypertension
4. heart_disease: 1 represent that patient has heart diseases; 0 represent that patient does not have hypertension
5. ever_married: “Yes” or “No”
6. work_type: “Private”, “Self-employed”, "children", "Govt_jov" or "Never_worked"
7. Residence_type: “Urban” or “Rural”
8. avg_glucose_level: average glucose level in blood
9. bmi: body mass index of each patient
10. smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown"

Data Preparation

The dataset we retrieved was well organized and easy to use. It didn’t contain a significant number of missing values or meaningless attributes which made it significantly easier to both navigate and use. We dropped the first column “id” since it was not required for our analysis. The dataset also contained 201 missing values in “bmi” attributes. Since the dataset was not large enough to ignore these missing values, we decided to replace these missing values with the average value of “bmi”. We then changed the categorical data into dummy variables. Therefore, we will use this new modified dataset for machine learning. It contains 5110 rows and 17 columns. This is what our dataset looks like:

	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke	gender_Male	gender_Other	ever_married_Yes	work_type_Never_worked
0	67.0	0	1	228.69	36.600000	1	1	0	1	0
1	61.0	0	0	202.21	28.893237	1	0	0	1	0
2	80.0	0	1	105.92	32.500000	1	1	0	1	0
3	49.0	0	0	171.23	34.400000	1	0	0	1	0
4	79.0	1	0	174.12	24.000000	1	0	0	1	0
...	...	...	...	...	...	...	...	...	...	...
5105	80.0	1	0	83.75	28.893237	0	0	0	1	0
5106	81.0	0	0	125.20	40.000000	0	0	0	1	0
5107	35.0	0	0	82.99	30.600000	0	0	0	1	0
5108	51.0	0	0	166.29	25.600000	0	1	0	1	0
5109	44.0	0	0	85.28	26.200000	0	0	0	1	0

5110 rows × 17 columns

work_type_Private	work_type_Self-employed	work_type_children	Residence_type_Urban	smoking_status_formerly smoked	smoking_status_never smoked	smoking_status_smokes
1	0	0	1	1	0	0
0	1	0	0	0	1	0
1	0	0	0	0	1	0
1	0	0	1	0	0	1
0	1	0	0	0	1	0
...	...	...	...	...	...	...
1	0	0	1	0	1	0
0	1	0	1	0	1	0
0	1	0	0	0	1	0
1	0	0	0	1	0	0
0	0	0	1	0	0	0

Data Visualization

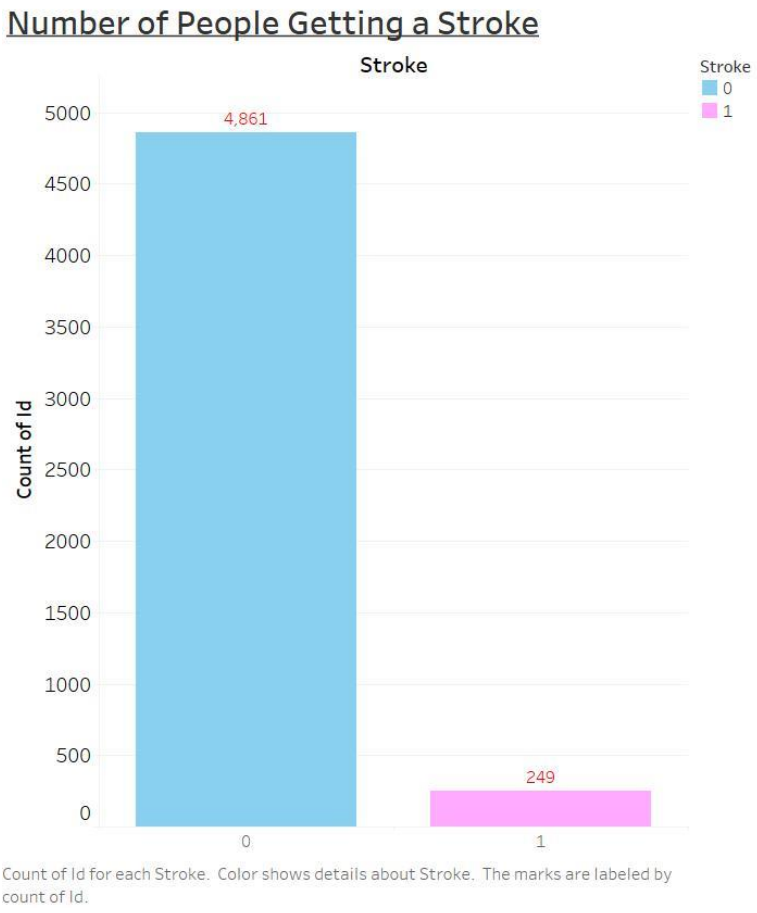
In this part, we will analyze the relationship between each attribute in the dataset and present those relationships in graph format for easier interpretation. We will first analyze the data that is composed from both stroke sufferers and those who have not suffered a stroke. The purpose of this is to understand what kinds of testing groups participated in this survey. Then, we will focus on analyzing the stroke group only. We will explore this group's characteristics in an effort to determine what the most important attributes are that can increase the chance of having a stroke. In the end of this section, we will also provide our predictions regarding the result of this report.

Topics for Analyzation

(1) Analyzing Data of Both Non-stroker and Strokers

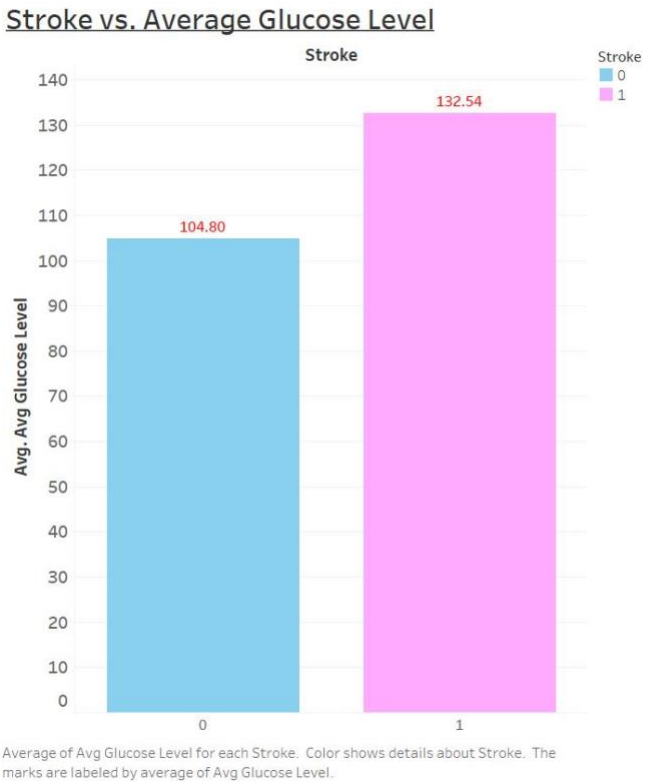
1. **Count of participants who get Stroke**

   The dataset is composed of 249 stroke sufferers and 4861 people who have not. Additionally, stroke sufferers comprise 0.051224028 or 5.1% of the entire cleaned dataset.



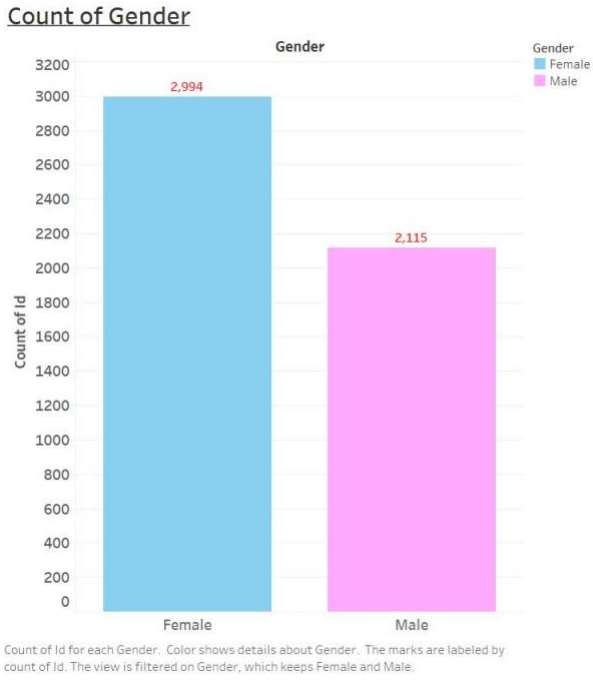
2. **Stroke vs. Average Glucose Level**

   As we can observe from the graph below, for those who suffer a stroke(stroke=1), their average glucose level is higher than those who have not had a stroke(stroke=0). This suggests that in most cases, higher average glucose levels increase the chances of having a stroke. However, according to mayoclinic.org, a blood sugar level resulting from random testing of less than 140 mg/dL is considered normal.



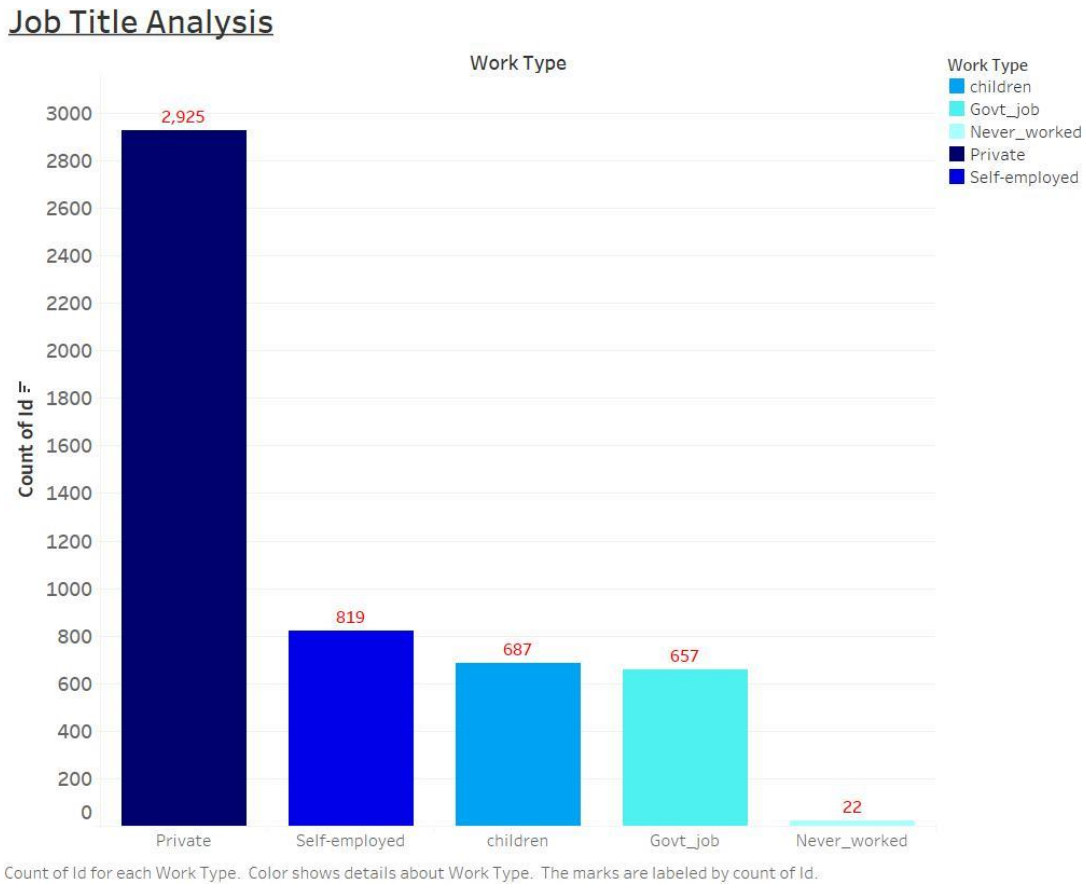
3. Count of Gender

In this dataset, we have 2994 female and 2115 participants.



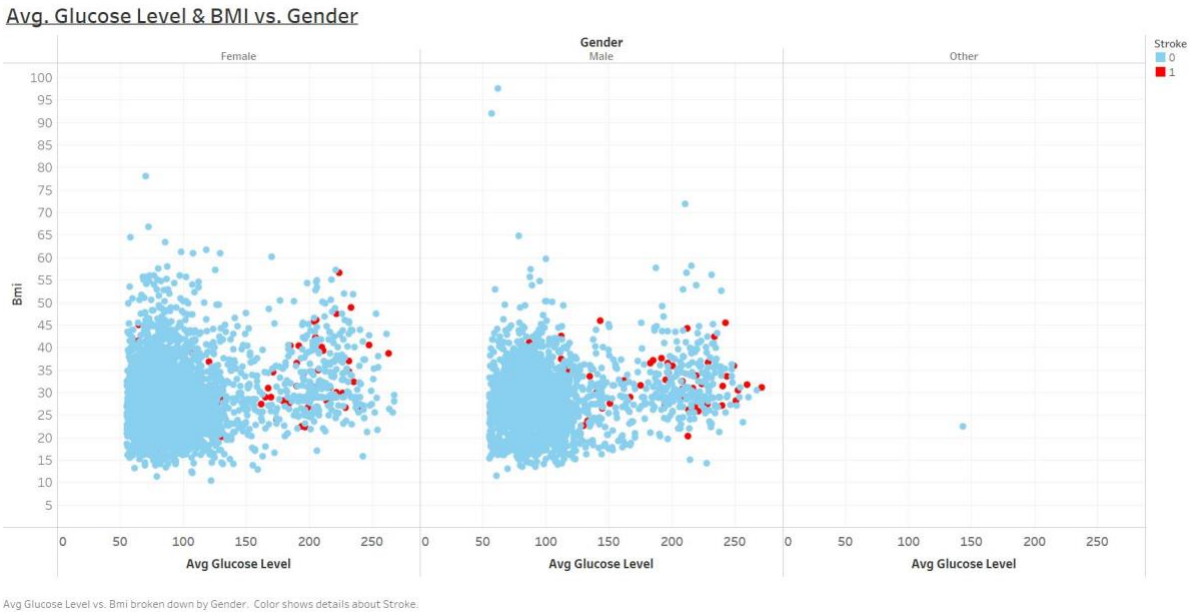
4. **Job Title Analysis**

   Of all patient records gathered in this dataset, patients whose job title is “private” appear the most frequently with a count of 2925.



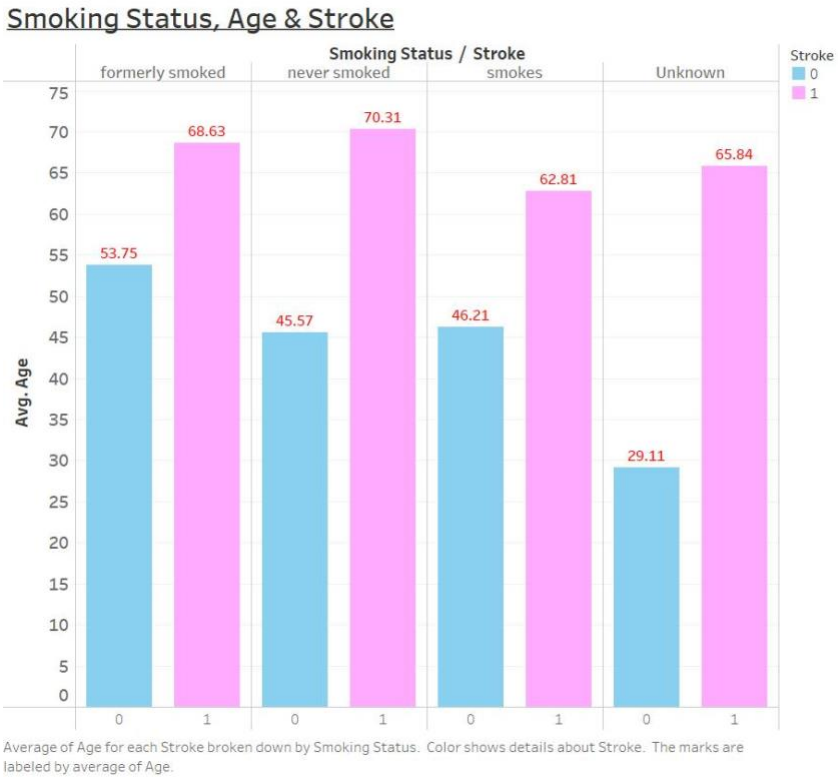
5. **Avg Glucose Level & Bmi vs. Gender**

   In general, most of the people who have had a stroke report a bmi in the range of 25-40. This suggests that people who are classified as overweight based on the bmi standard have a higher chance of suffering a stroke. We don’t see significant difference regarding gender and the likelihood of stroke - both female and male participants reveal similar outcomes.



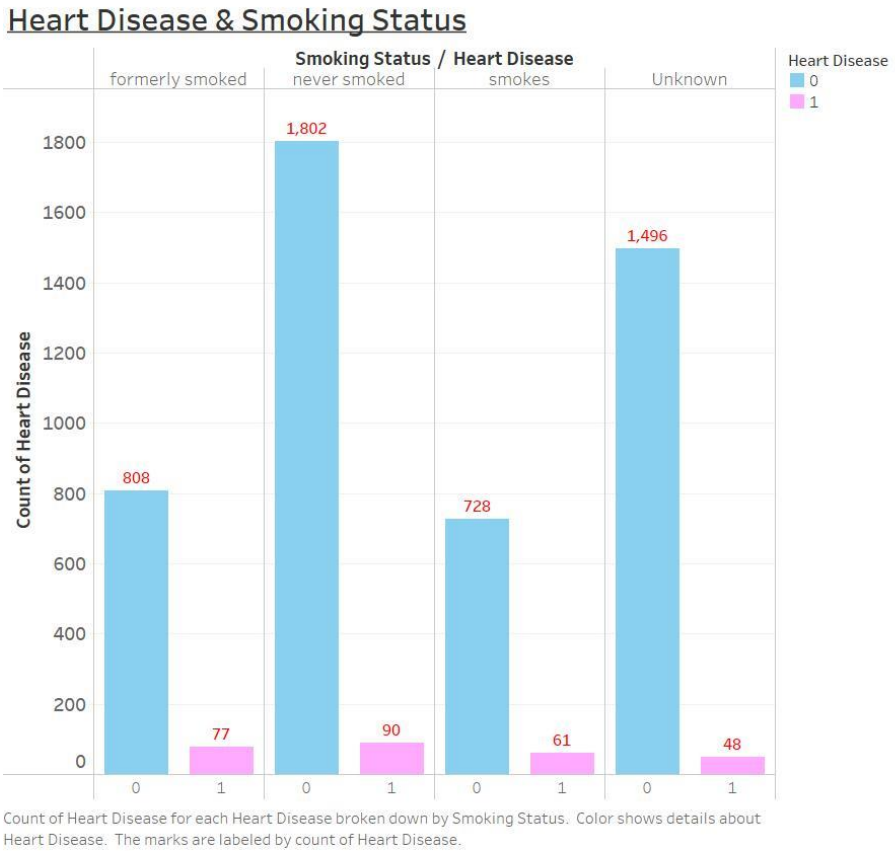
6. Smoking Status, Age, and Stroke

In this section we analyze the relationship between age, smoking status, and stroke. Here we observe that the average age for stroke sufferers is higher than those who have not had a stroke.



7. **Heart disease & smoking status**

In the graph below, one of the most interesting observations we found was between the number of people who never smoked and have heart disease (90) which is higher than the number of people who have smoked and have heart disease(61).

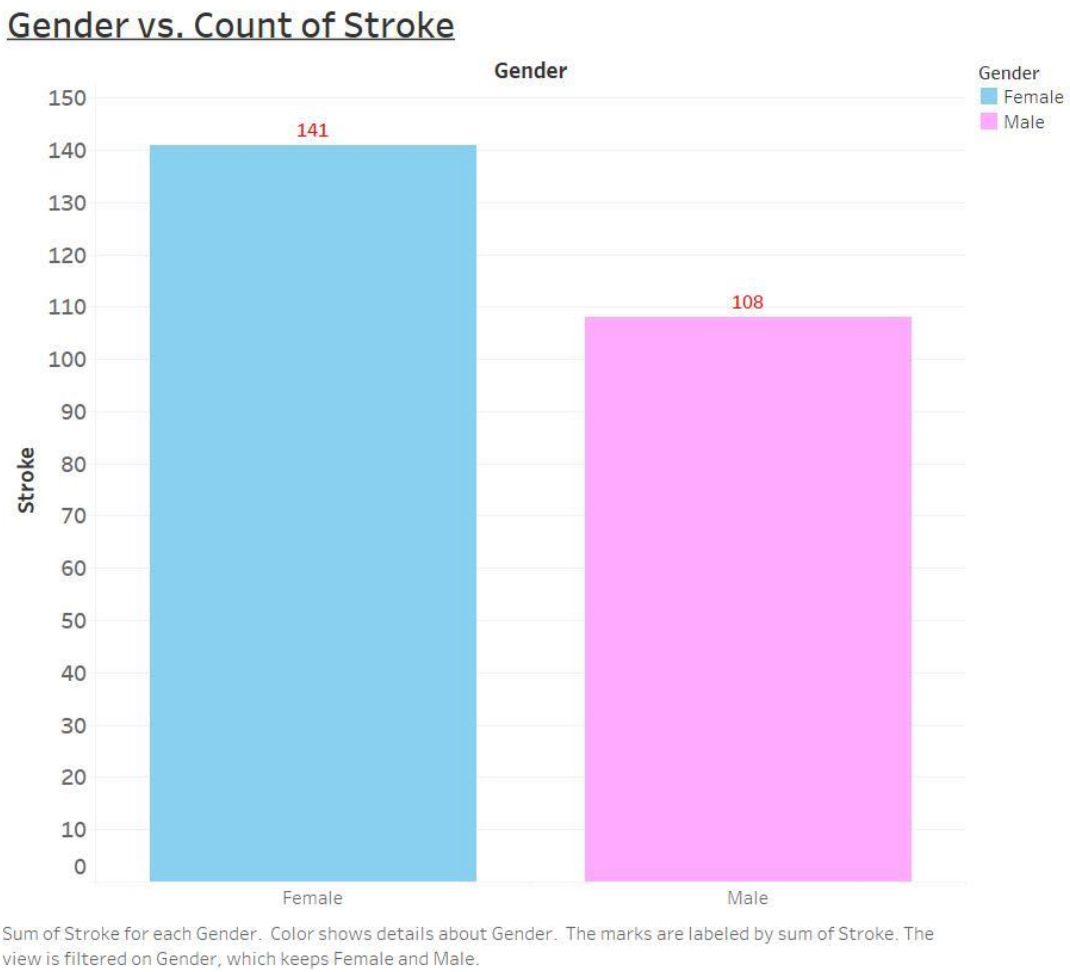


(2) Analyzing Data of Strokers (stroke = 1)

The following visualization will mainly focus on data of strokers only.

1. **Gender Counts**

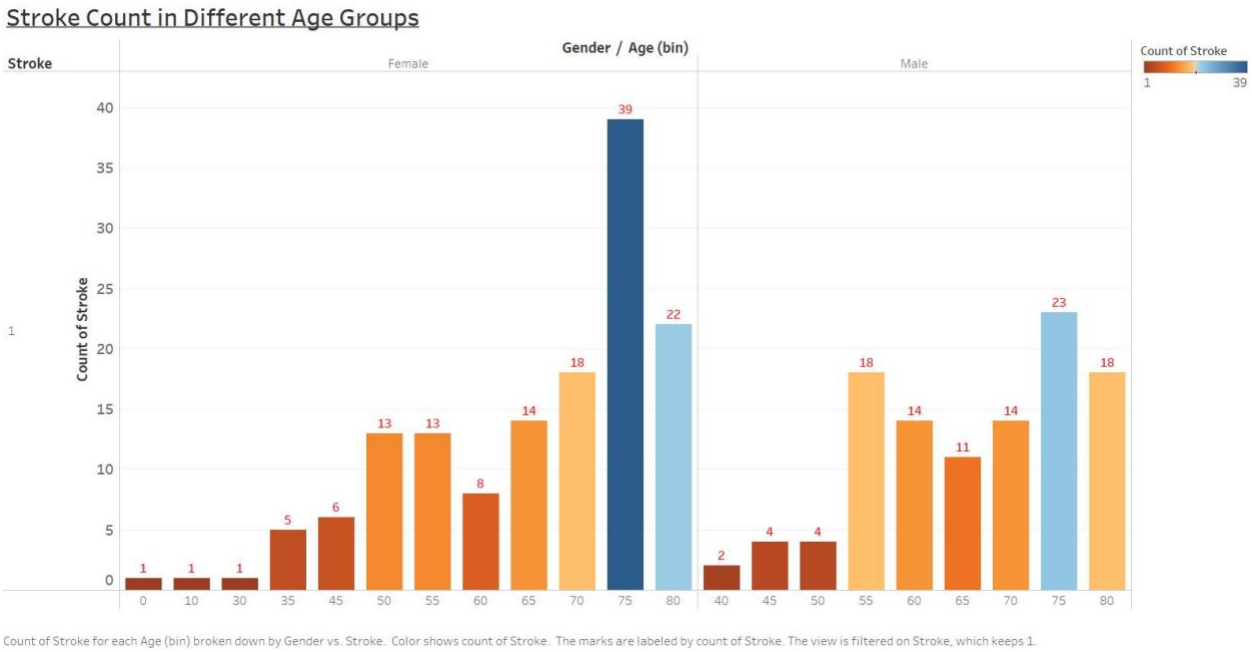
   The count of female stroke sufferers in this dataset is higher than that of males who have had a stroke. This observation could be explained by the large numbers of elderly women living in the United States. According to Statista.com, as of July 2020, “there were 162.26 million males and 167.23 million females(Statista.com)”.



2. Age Distribution

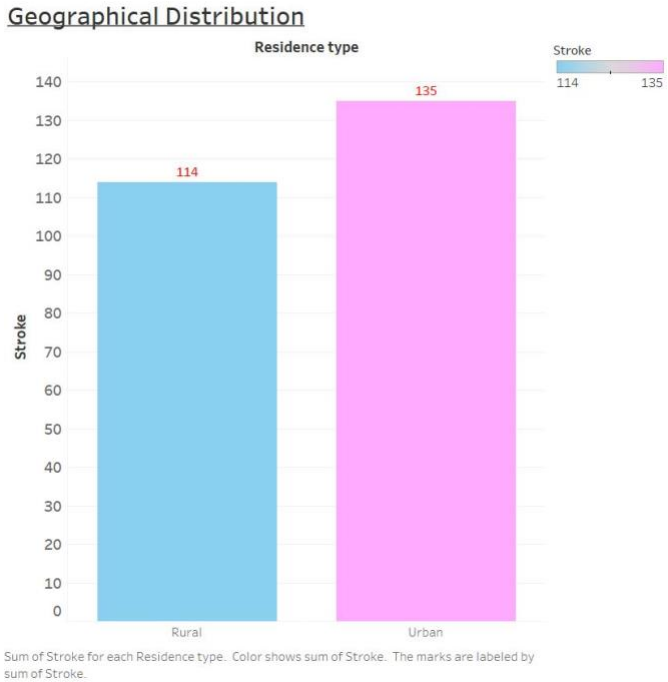
The average age of people who have a stroke is 67.72. As observed in the given histogram below, typically older people have a higher chance of having a stroke. In the age graph, people who are in the age 75 category for both the female and male groups have the highest chance of having a stroke.

In rare cases, it is possible for newly born babies to have a stroke. This is why we observe a tester with age 0 is being labeled as stroke. This condition is called pediatric stroke, and affects only 1 in every 4000 babies.



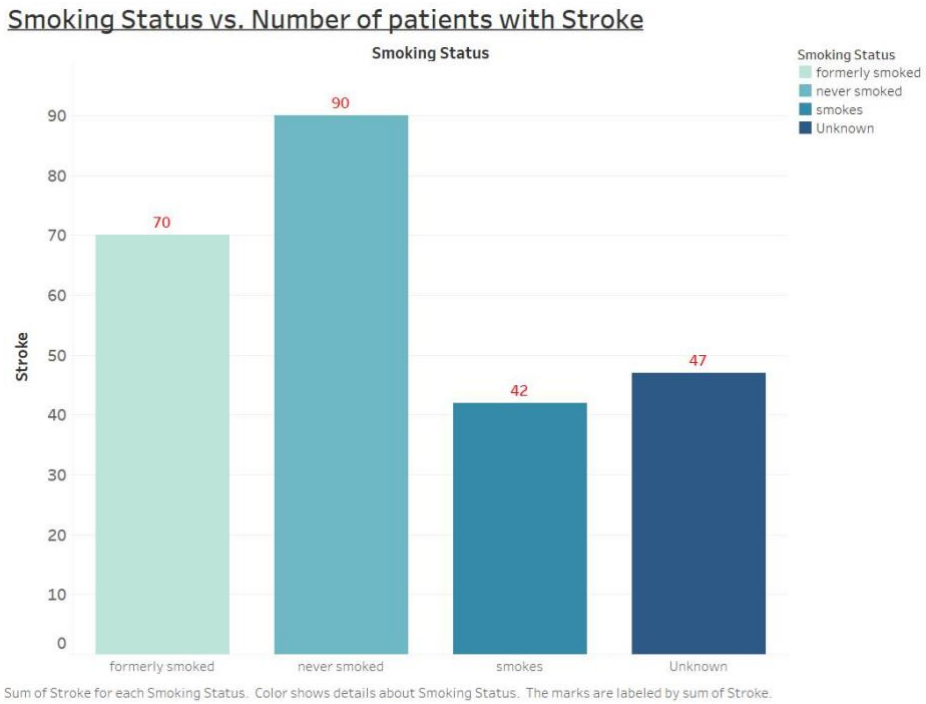
3. Geographic Distribution (Residence Type & stroke)

In this dataset, the patient's living environment is recorded and classified into two categories: Rural and Urban. The geographic distribution is as follows, 114 patients live in rural areas and 135 live in urban areas.

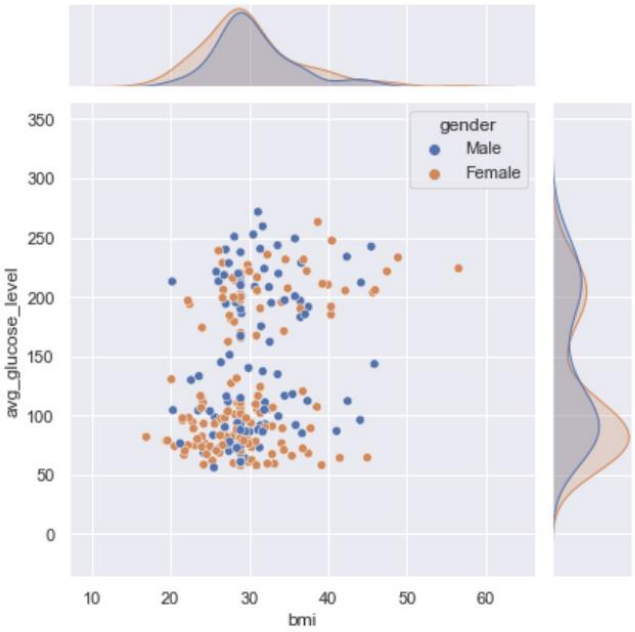


4. Count of Smoking Status

For the group of patients which is labeled as stroke, people who never smoked in the past have the highest count. While patients who are currently smokers(‘smokes’) have the lowest count.

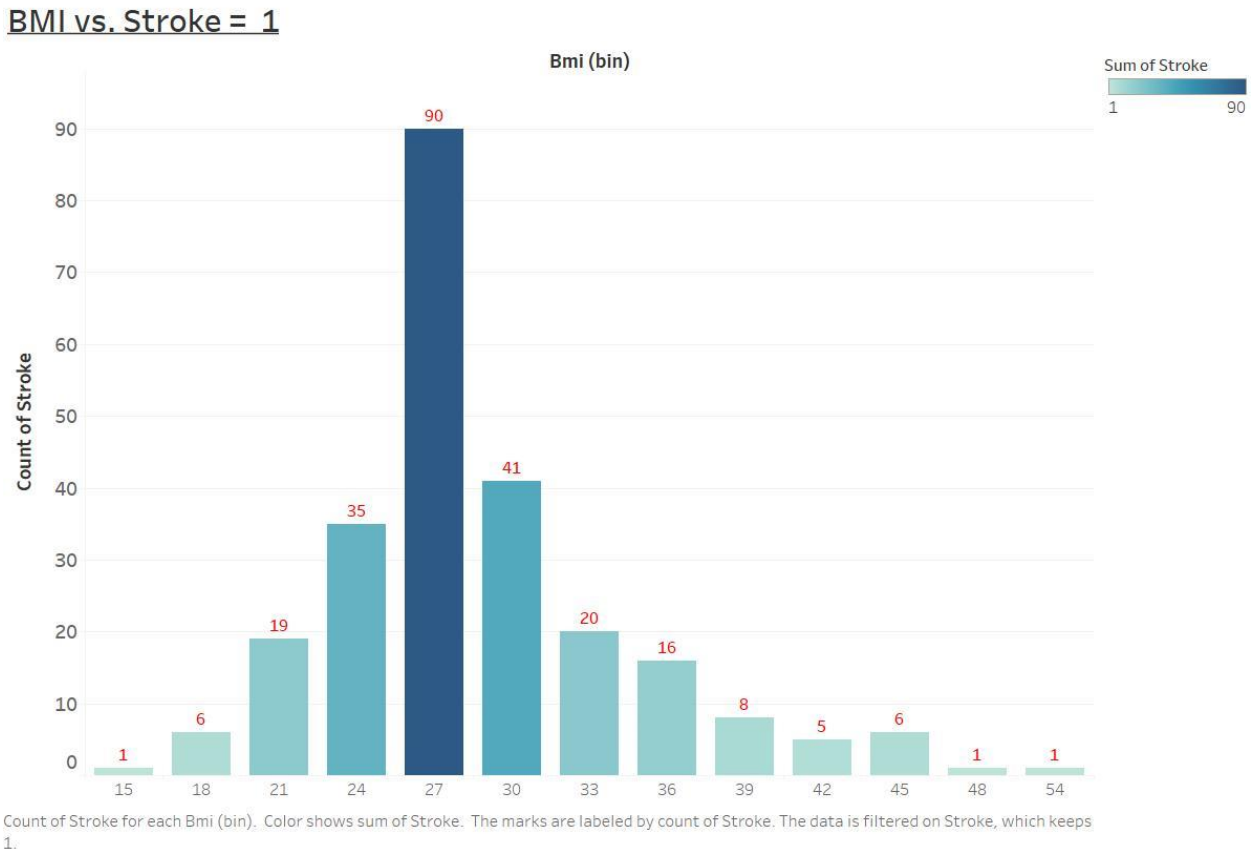


5. **Bmi, Avg glucose Level & Gender**



6. **Bmi vs. Stroke=1**

   The graph below contains the BMI ranges for those who have suffered a stroke. We observe that the most common BMI reading within that range is 27, which has 90 counts in the stroke group.



Predictions

Before our group applied any machine learning models to analyze this dataset, we already had several hypotheses regarding which factors would be the most determinant in causing a person to have a stroke.

At this stage, we believe the following factors will play an important role in increasing the chances a person will have stroke:

- **Bmi**: people with bmi above 25 have a higher probability of suffering a stroke
- **Age**: people who are in the 75 years old age bracket have a higher probability of a stroke
- **Average Glucose Level:** higher than 130
- **Smoking Status**: We expect smokers will have a higher incidence of stroke than those who do not smoke

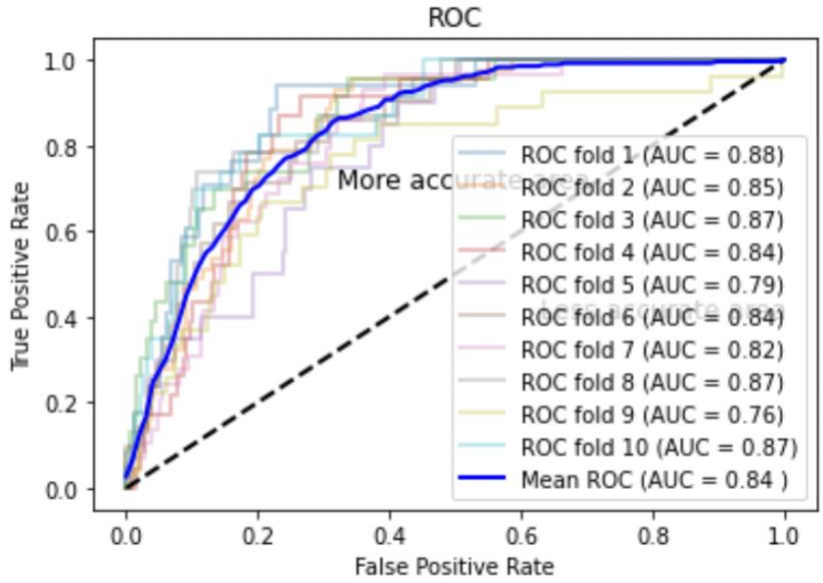
Analysis on Machine Learning Models

In this section, we are going to build six machine learning models to make predictions on a patient's likelihood of having a stroke based on the parameters contained in our dataset. The attributes are mixed with numeric data and categorical data. We will use the edited dataset which was introduced in the data preparation section to do the further analysis. The six machine learning models are logistic regression, decision trees, random forests, SVMs, Gaussian naïve bayes and KNNs. Then, we are going to add some preprocessing steps such as feature selections and resamplings to each model. We are going to use 10-fold cross validation to split the dataset because our dataset is not large enough to simply 70/30 train-test-split. Additionally, 10-fold cross validation can reduce any bias on the data selection in each model. We will use average AUC as our main evaluation measurement between the six models. We will also take their accuracies as the secondary metric. Since our dataset contains 249 stroke sufferers and 4861 non-strokers, accuracy cannot predict evenly. For most of the cases, it will predict the majority class. Therefore, an imbalanced dataset may have a high accuracy and lead to misleading measurement. In contrast, AUC is a more comprehensive measure. In this way, we can consider not only the proportion of true positives and negatives, but also how true positive rate and false positive rate trade off in the whole data set as we want to maximize TPR and minimize FPR. Due to the nature of our project, we want the highest TPR even if the false positive rate is high, since the cost of a false negative is really high and the cost of a false positive is not as much.
Based on these performance changes, we will give our analysis and insights.

Logistic Regression

Before preprocessing data

The first machine learning model we built is logistic regression. We used the sklearn package to get its accuracy with 10-fold cross validation to 0.951467710 with 0.0135 standard deviation. The mean AUC is 0.84. Below is a graph that plots all the ROC curves of the 10 folds. We can see that the logistic regression model performs well because it fits our mainly binary categorical data.



Although this model already works well, we want to improve it as much as possible. Also, our dataset is not balanced. As previously mentioned, the dataset contains 4861 individuals who have not had a stroke and 249 who have suffered a stroke. There are also too many attributes in the dataset since we created dummy variables for each unlabeled categorical variable. We decided to make feature selections and re-sample the dataset to prune and balance it.

Feature selection

We used **RFECV** (Recursive Feature Elimination and Cross-Validation Selection), an algorithm which eliminates irrelevant features based on validation scores, from the Sklearn library in Python to make feature selections on the model. By utilizing this library, we can find the best features to keep for every model, i.e. with which features each model performs the best.

For logistic regression, the model does not take out any features as the best result comes from the regression model with all the features included.

Re-sampling (oversampling)

Since the data is imbalanced, we decided to resample the dataset. We use SMOTE (Synthetic Minority Oversampling Technique) to create new examples in the minority class. SMOTE selects examples that are close in the feature space, drawing a line between the examples in the feature space and creating a new sample at a point along that line. Even though the new items don't add any additional information to the model, oversampling can give a more balanced classification to our models.

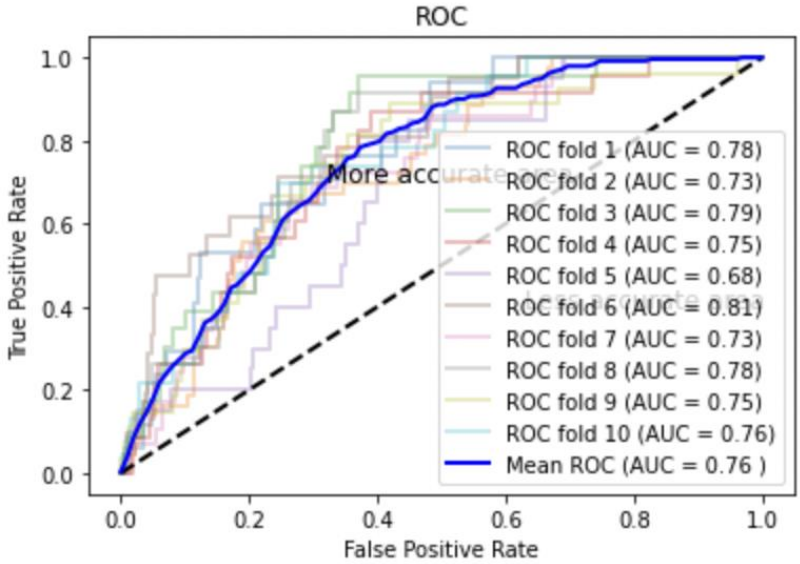
Instead of undersampling the majority class, we are attempting to keep the valuable data and make a data augmentation for the minority class in this step. Since we want to see just the results on resampling, we are going to implement the technique into the original dataset.

After oversampling the data, our new logistic regression has a lower accuracy than before, which is 0.8365949119373777. The new average AUC is also lower at 0.76. Since our dataset was imbalanced, the accuracy can be high. Our resampled data

causing a lower accuracy is understandable because the model can predict with a best possible result about 50%.

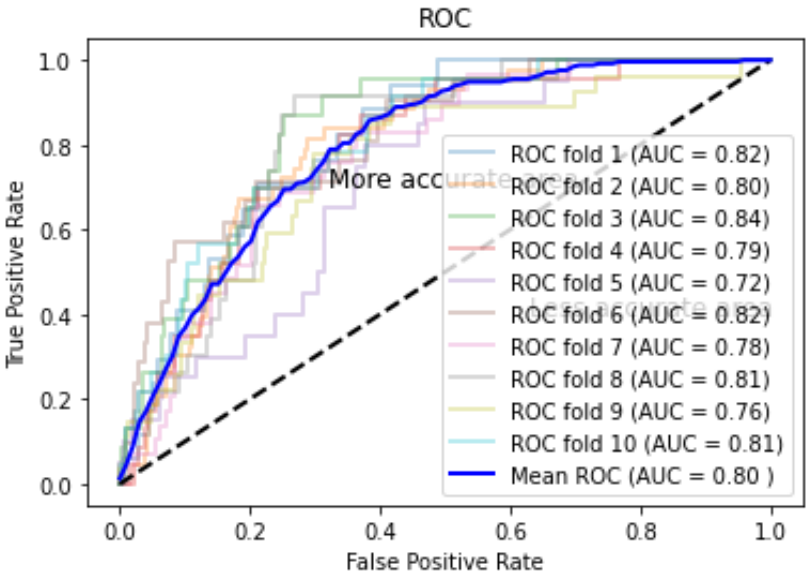
Re-sampling (undersampling and oversampling)

Instead of just oversampling the minority class 1, we chose to use undersampling to decrease the majority class 0 using RandomUnderSampler from scipy package and



oversample the minority class 1 together within each fold with SMOTE. Running this model with the mixed resampling method apparently takes longer time. However, the results based on this more reasonable dataset are better than other preprocessing methods. Its outcome has a 0.82 mean accuracy and 0.80 mean AUC.

Mean accuracy: 0.82

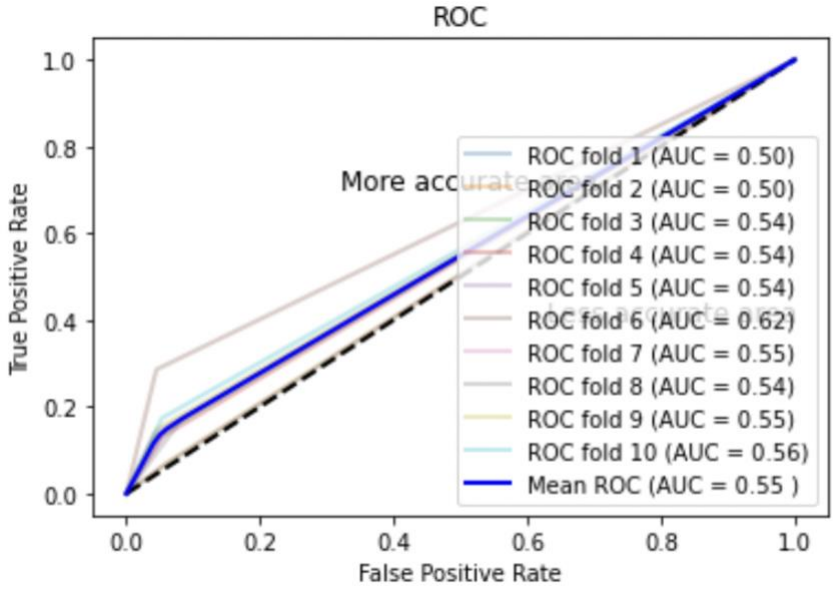

ROC
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC fold 1 (AUC = 0.82)
ROC fold 2 (AUC = 0.80)
ROC fold 3 (AUC = 0.84)
ROC fold 4 (AUC = 0.79)
ROC fold 5 (AUC = 0.72)
ROC fold 6 (AUC = 0.82)
ROC fold 7 (AUC = 0.78)
ROC fold 8 (AUC = 0.81)
ROC fold 9 (AUC = 0.76)
ROC fold 10 (AUC = 0.81)
Mean ROC (AUC = 0.80)

Decision Trees

Our second machine learning model is decision trees. A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

Before pre-processing

The performance of decision trees was not as good as we expected. It's accuracy is 0.904892368, with 0.0173 standard deviation. The average AUC score from 10 folds is 0.55. Since the dataset is imbalanced, the decision tree is apparently not very suitable to our data. However, we still want to see what the measurements are going to change after we use feature selection and resampling.



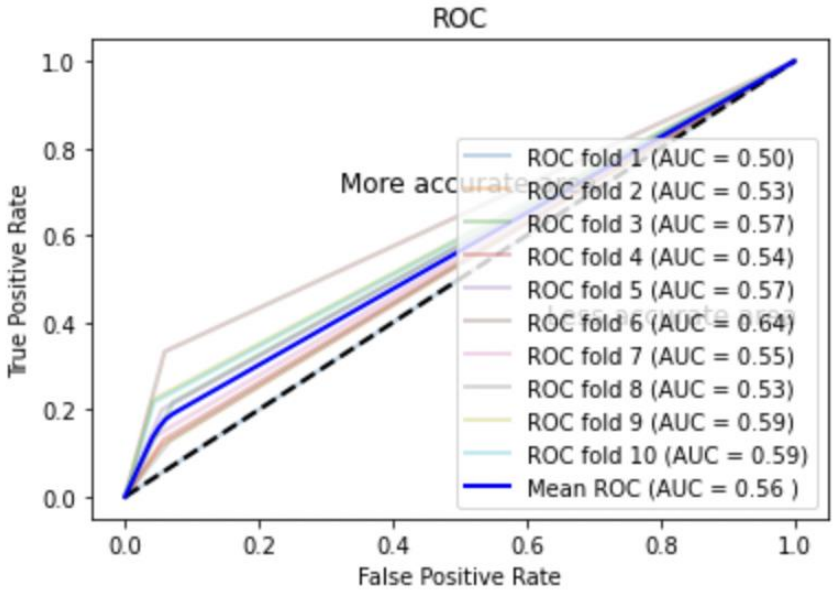
Feature Selection

By using RFECV, we found that only 9 features are selected as optimal and useful for the decision tree. Here are the model's outcomes after we preprocessing the data:

The optimal number of features is 9

The selected features are:

['age', 'avg_glucose_level', 'bmi', 'gender_Male', 'ever_married_Yes', 'work_type_Private', 'smoking_status_formerlySmoked', 'smoking_status_neverSmoked', 'smoking_status_smokes']

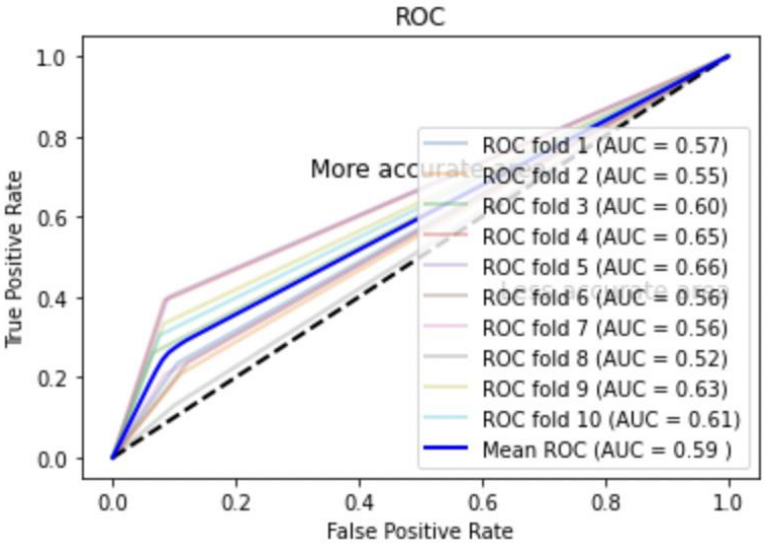


Its accuracy is 0.91 and the average AUC turns out to be 0.56, which is slightly higher than before.

Re-sampling (oversampling)

Decision trees very much rely on data balancing. So we are trying to resample the data. We use SMOTE to balance the data. Here are the new outcomes of the model, which has 0.8767 accuracy and 0.59 average AUC. The measurements are getting better, but we want to see how much they can improve if we balance the dataset.

Accuracy: 0.8767123287671232

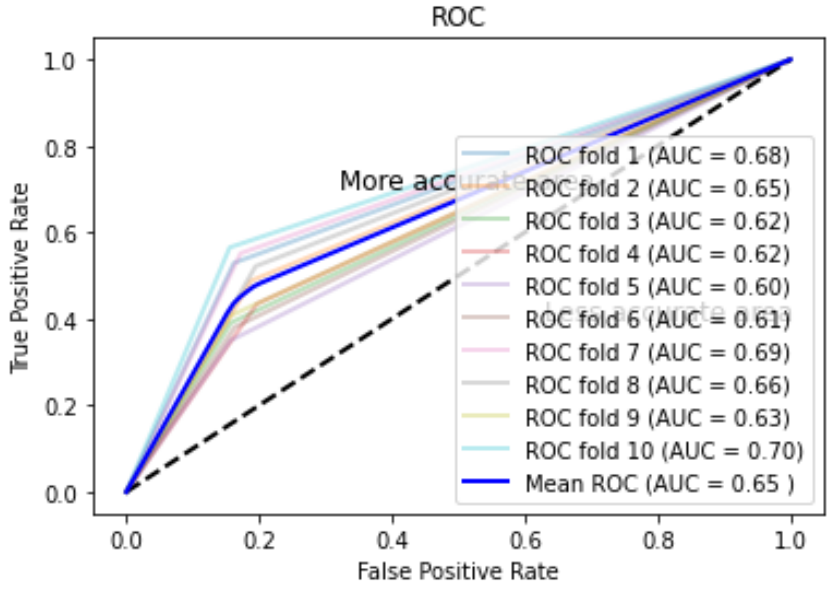


Re-sampling (undersampling and oversampling)

Our last attempt for this model is utilizing undersampling and oversampling at the same time. Since we have more than 4,000 rows of class 0 and around 200 rows of class 1, our dataset needs to be re-balanced. Instead of simply increasing the number of observations in the minority class, we try to cut out some of the majority class and extend the minority class at the same time. By doing this, we believe the model will perform better.

After running the model with resampling, we get 0.81 mean accuracy and 0.65 mean AUC. The performance is better than simply oversampling the minority class.

Mean accuracy: 0.81



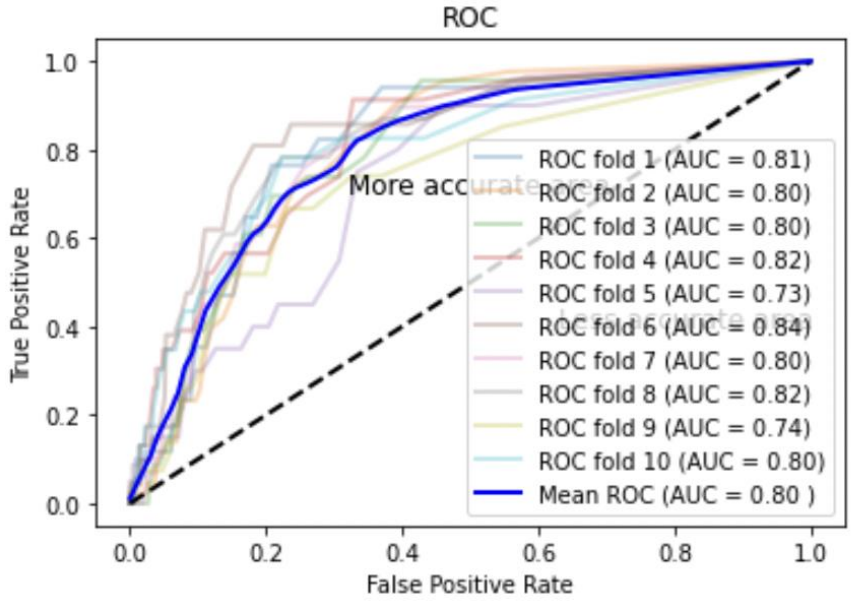
Random Forests

The third machine learning model we applied is a random forests classifier. Random forest is an ensemble learning method that operates by constructing a multitude of decision trees on various subsets of the given dataset. It takes the average to improve the predictive accuracy of that dataset. Random forest can sometimes reduce the overfitting problems that happen to decision trees.

Before pre-processing

Before making any preprocessing on the dataset, we run a random forest model. We get 0.950489237 accuracy with 0.0132 standard deviation and 0.80 as the average AUC. Its performance is much better than a single decision tree model because it is a highly accurate and robust method consisting of many decision trees.

Accuracy: 0.950489237 (0.013245222)



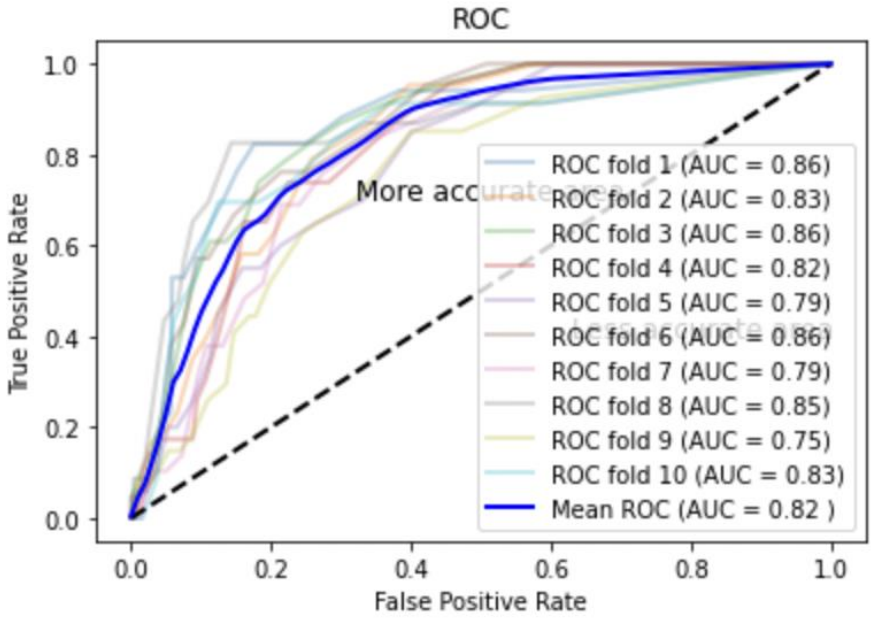
Feature selection

Feature selection is extremely important to the random forest. The model with RFECV method gets the optimal number of features which is 9. Using these new features to run the random forest again, we get 0.95 accuracy and 0.82 average AUC. Here are the optimal features and a ROC curve graph.

The optimal number of features is 9

The selected features are:

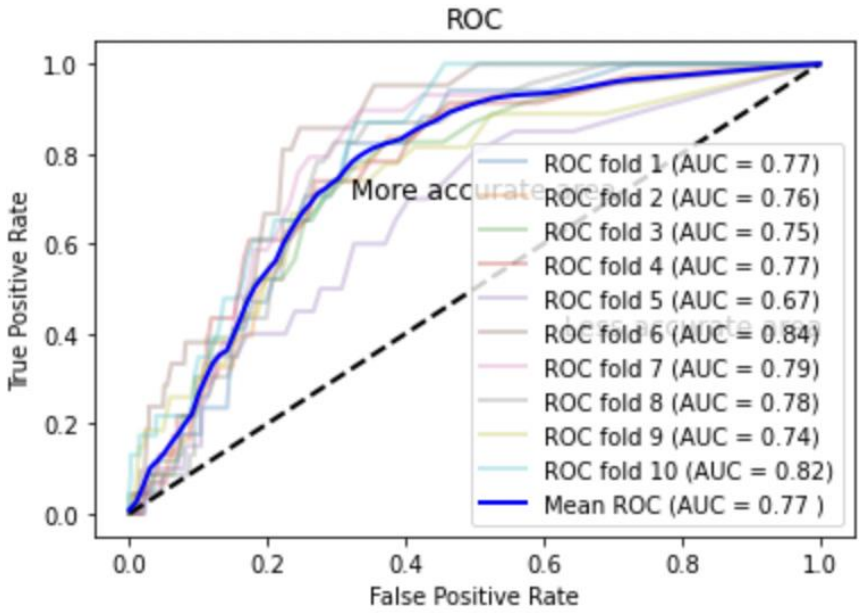
['age', 'hypertension', 'heart_disease', 'avg_glucose_level', 'bmi', 'gender_Male', 'work_type_Private', 'Residence_type_Urban', 'smoking_status_neverSmoked']



Re-sampling (oversampling)

Using a similar process as we did for the other models, we randomly oversample using SMOTE on the original dataset to make the dataset balanced. After we oversample the minority class, our model has worse performances: 0.91 accuracy and 0.77 mean AUC. Since the model does not perform as well, we will try to mix the undersampling on the majority class and oversampling the minority class at the same time in the next model.

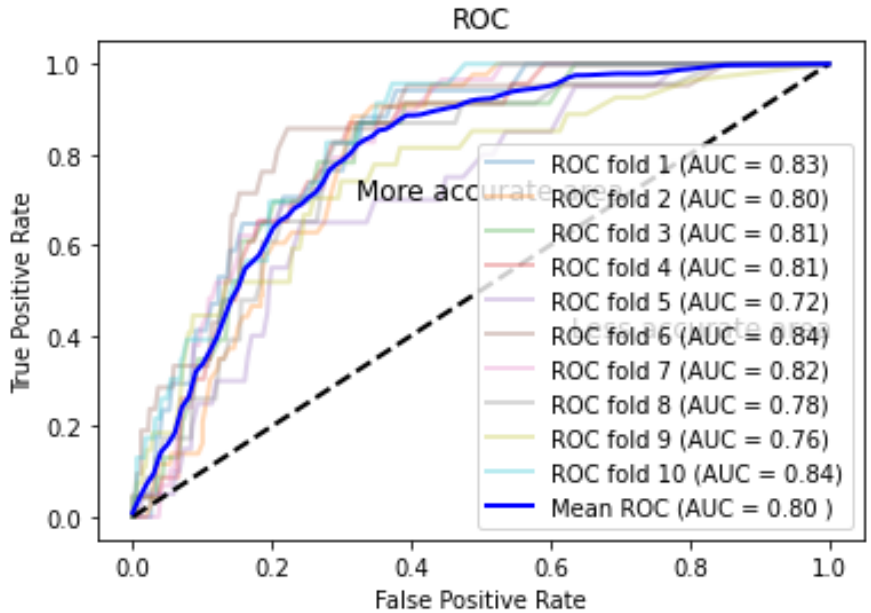
Accuracy: 0.9093933463796479



Re-sampling (undersampling and oversampling)

The next method we will try is what we introduced in the previous step. We use the SMOTE method to automatically resample the data in each fold. We get 0.85 mean accuracy and 0.80 mean AUC. This is better than a single oversampling method.

Mean accuracy: 0.85

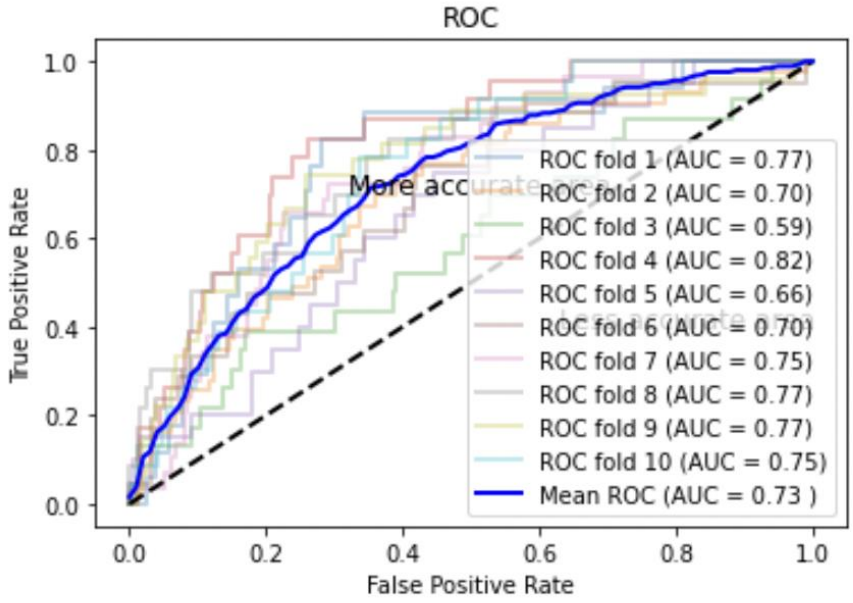


Support Vector Machines

Another model we have used is Support Vector Machines. This type of model can be used for both regression and classification analysis, although it is mostly used for the latter one.
In SVM every data is plotted as a point on a N-dimensional space, where N is the number of features.
The goal of this model is to find a hyperplane in this dimensional space that clearly classifies distinct data points, which is the one that has the maximum margin, and the maximum distance between data points of both classes and the hyperplane. In order to create the best decision boundary, the model uses support vectors, which are the data points closest to the hyperplane and affect the position and orientation of the boundary.

Before pre-processing

Accuracy: 0.951272016 (0.013343166)



SVM with the cleaned data performs really well with an accuracy of 95%. However, this is probably most likely due to the imbalance distribution of the classes. Looking at the AUC, we can see that it does not perform as well as other models in classifying stroke.

Feature Selection

By using RFECV, we found out that the best SVM model has the following 13 features:

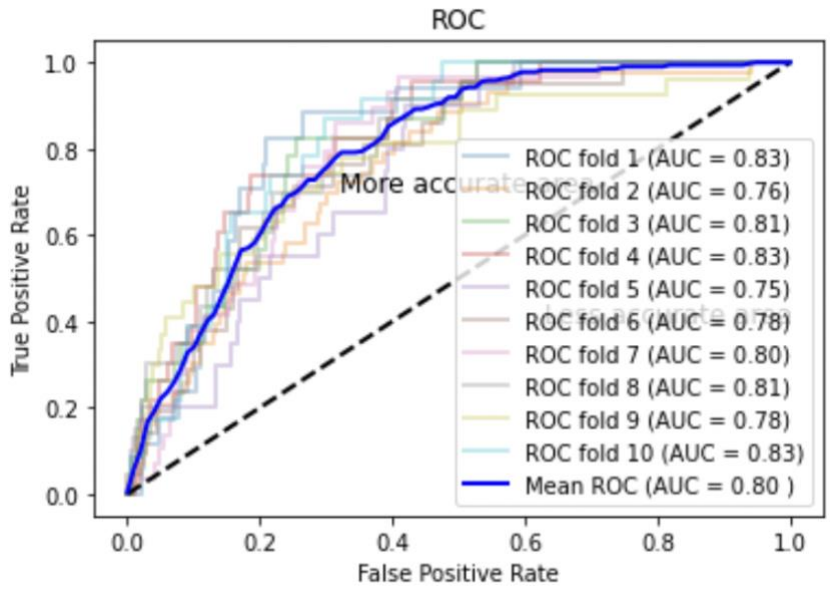
The optimal number of features is 13

The selected features are:

['age', 'hypertension', 'heart_disease', 'avg_glucose_level', 'gender_Male', 'ever_married_Yes', 'work_type_Private', 'work_type_SelfEmployed', 'work_type_children', 'Residence_type_Urban', 'smoking_status_formerlySmoked', 'smoking_status_neverSmoked', 'smoking_status_smokes']

The accuracy hasn't decreased after dropping three features and the mean AUC across all folds increased by 0.03. This model is slightly better than the benchmark in classifying people's risk of stroke. The accuracy is still high, since we haven't changed the distribution of the class so far. In order to see if our model is not biased towards class 0, we'll next see how it performs after resampling.

Accuracy: 0.951272016 (0.013343166)

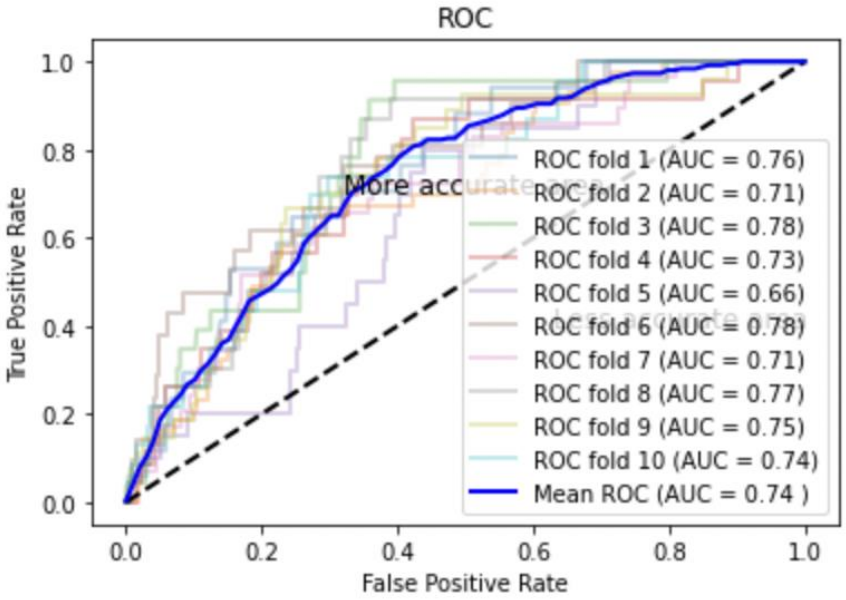


Re-sampling (oversampling)

Accuracy: 0.8377690802348334

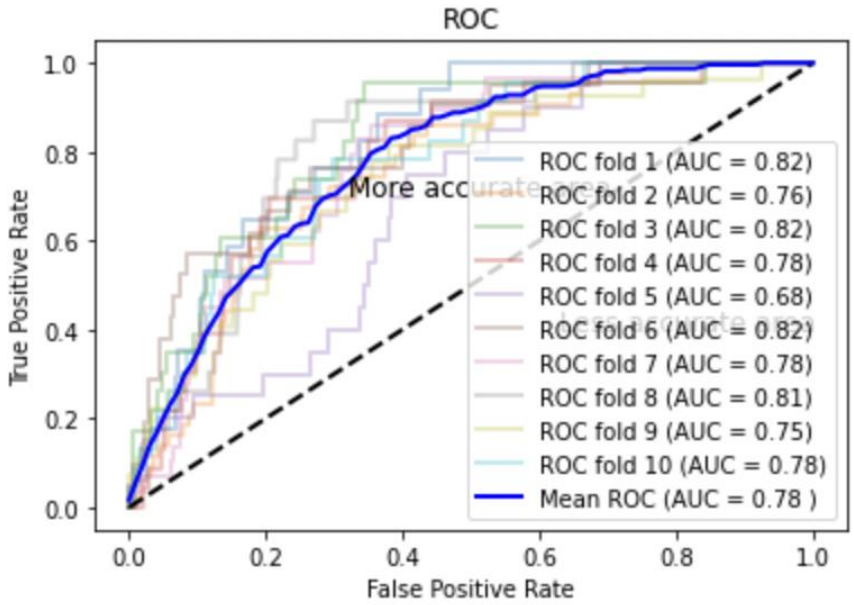
By oversampling we see that the model's AUC is 0.01 points higher than the benchmark, but 0.06 points lower than the feature selection one, meaning this model has a lower probability to rank randomly chosen positive instances higher than randomly chosen negative ones. Although we tried to even the distribution of the class,

the AUC is somewhat similar to the benchmark, but worse than the model with just the significant features.



Re-sampling (undersampling and oversampling)

Accuracy: 0.827788649706458



Since oversampling didn't improve our model, we tried to undersample together with oversampling. The results were better than the model using exclusively oversampling but still not as high as the one with only the significant features, although it's close. The AUC is just 0.02 points lower, but it's 0.05 points higher than the benchmark one.

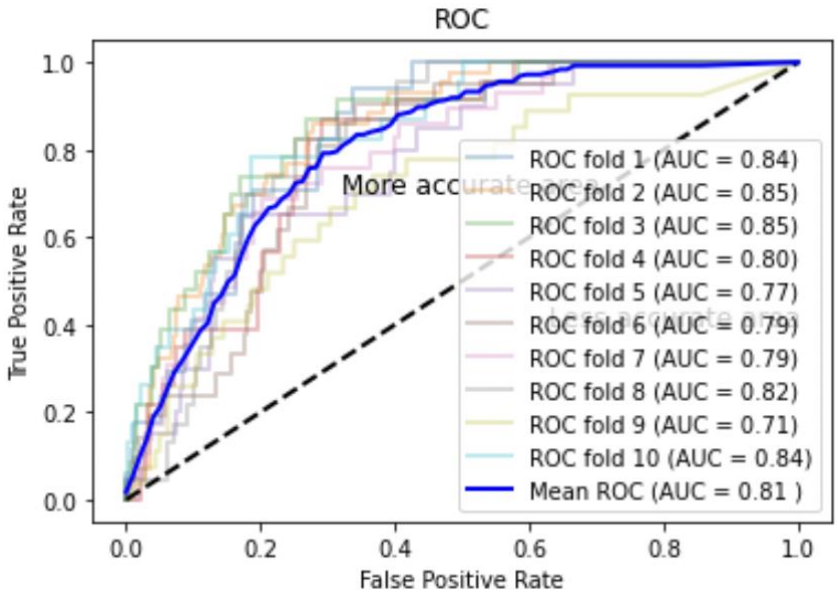
Gaussian Naïve Bayes

Naive Bayes are a group of supervised machine learning classification models and are based on Bayes Theorem (conditional probability).

Gaussian Naive Bayes is one type of Naive Bayes that follows Gaussian normal distribution and can use continuous data. The assumption in this model is that the values within each class are distributed according to a normal (Gaussian) distribution.

Before pre-processing

Accuracy: 0.417025440 (0.056286985)

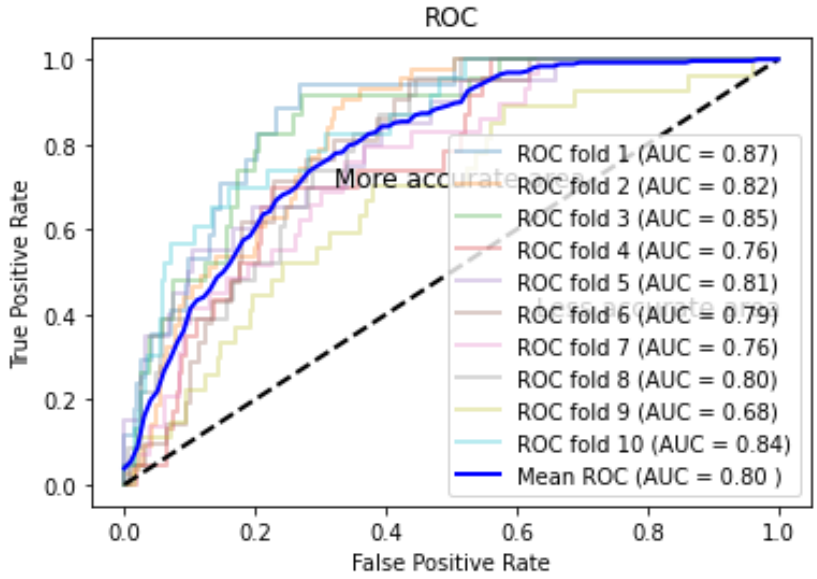


This model has the worst performance out of the 6. With an accuracy of just 0.42 despite the imbalanced distribution. However, the AUC is still relatively good at 0.81, meaning our model is predicting class 1 without too many false positives. This is because when classifying a given data point, the Naive Bayes classifier first calculates the probabilities with which it believes the data points belong to each possible class label. It produces a classification by selecting the class label associated with the largest probability. Frequently the largest probability is already associated with the correct class label. However, beyond the (important) close correlation between the largest probability and the correct class label, the probabilities are not correlated with classification confidence. The largest probability is frequently close to 1, with other probabilities close

to 0. This indicates extremely high classification confidence, much more than empirically demonstrated actual classification accuracy justifies.

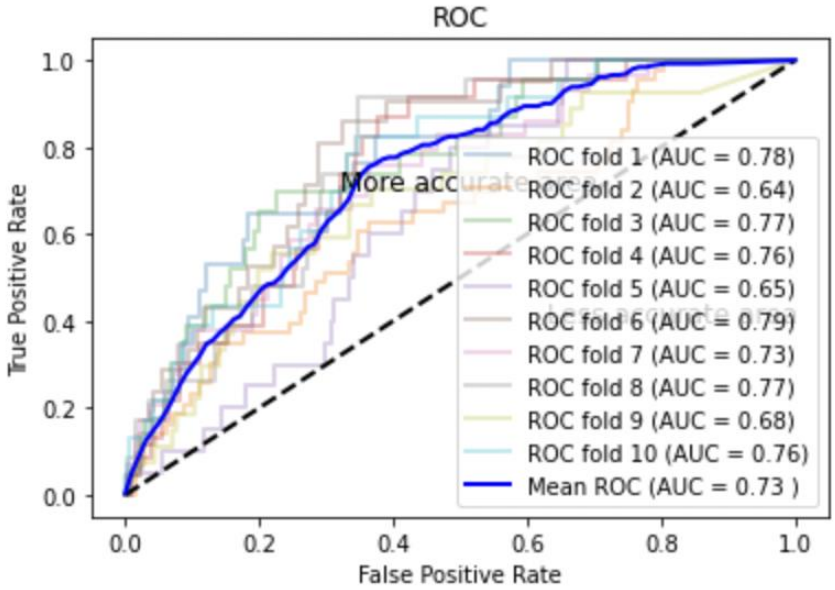
Feature Selection

Mean accuracy: 0.91



Re-sampling (oversampling)

Accuracy: 0.8125813873212792



Now we try the Gaussian Naive Bayes with oversampling with SMOTE. We can see that with an even distribution of the class, the model performs better with regards to

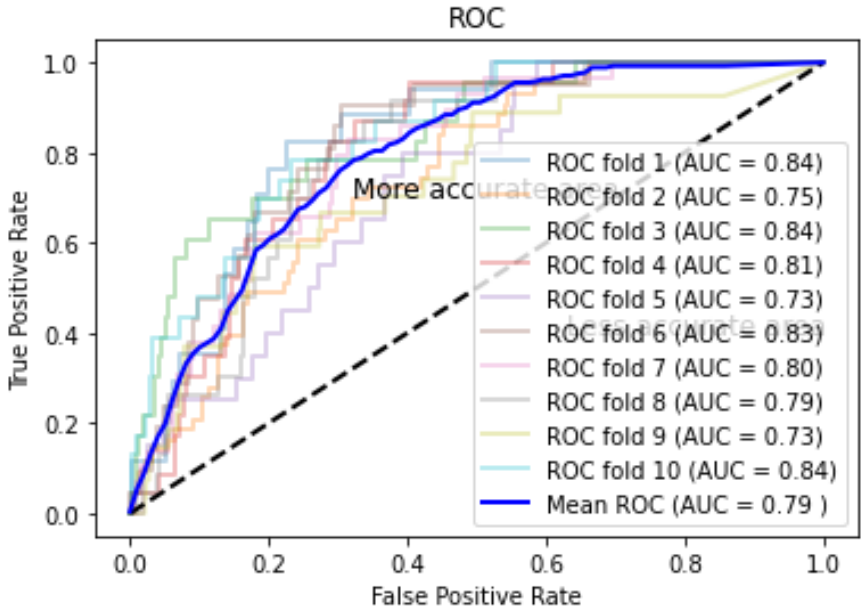
accuracy, increasing by 40%. However, AUC decreased by 0.07 as the model contains more false positives than the benchmarked model, most likely owing to the fact that by oversampling we created replicas of class 1 observations.

Re-sampling (undersampling and oversampling)

This time we tried to equalize the resampling by under and oversampling. By merging these two methods, we obtained a dataset of about 2000 observations, almost evenly distributed in class.

Using this processed data, the model reached an accuracy of 0.83 and AUC increased by 0.06 points. Not only does the model more correctly label the classes, but it also has a lower false positive rate.

Mean accuracy: 0.83



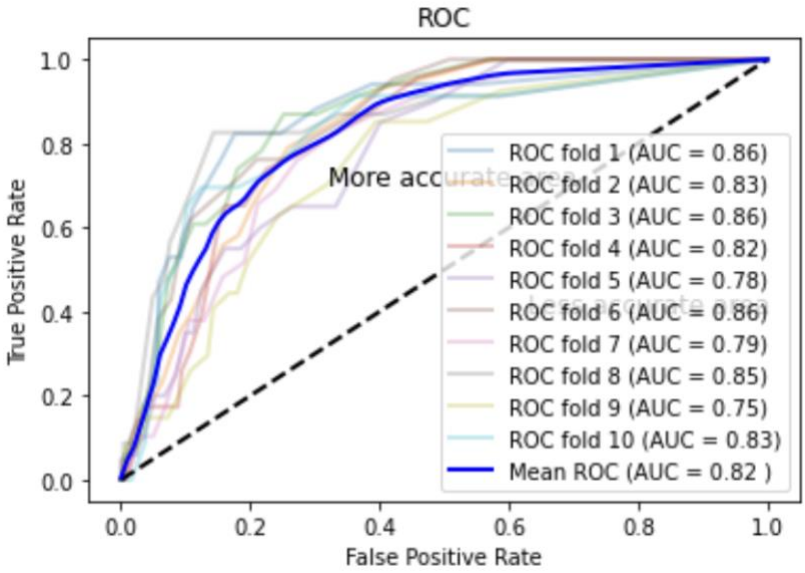
K-nearest Neighbors

Another model we have tried is the K-nearest neighbor. It's a supervised machine learning algorithm that can be used for both classification and regression problems. The KNN algorithm assumes that similar data points are close to each other. KNN is based on this idea of similarity, calculated by using distance measurements like Euclidean distance. To predict which class a data point belongs to, the algorithm determines which group is nearest to the data point.

We have set the number of neighbors to be the square root of the training sample. Since we are using a 10-fold cross validation, k will be equal to 90% of the total sample and if it's an even number, we add 1 to it.

Before pre-processing

Accuracy: 0.951272016 (0.013343166)



KNN is one of the best performing models with our base data. It has 95% accuracy with AUC 0.82. However, as for the other models, accuracy is not a good measurement for a model's performance as our data is highly imbalanced toward class 0. Indeed, after resampling, we'll see a drop in accuracy but AUC won't change as much.

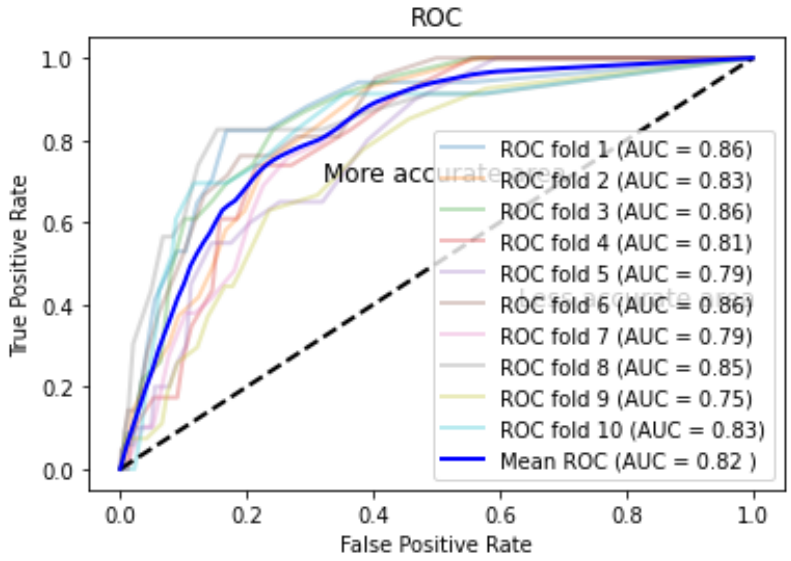
Feature Selection

In order to perform feature selection on KNN models, we had to do it manually instead of relying on RFECV. To find the most significant features, we used ANOVA for the numerical variables (age, bmi and average glucose level) and for the other categorical variables, we've found the p values through chi square distribution.

The features we have kept are: age, average glucose level, bmi, children, work type self-employed, heart disease, ever being married, hypertension and smoking status formerly smoked.

We can see from the results below that dropping the other 6 features has not changed the model's performance at all. We can conclude that the other variables are not as relevant in our analysis on this model.

Mean accuracy is: 0.95; Mean AUC: 0.82

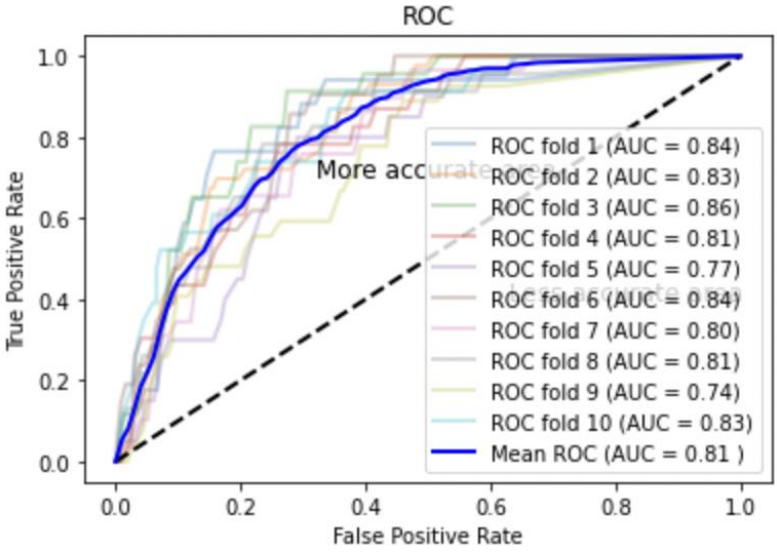


Re-sampling (SMOTE)

After oversampling the data and getting an even distribution of our class variable, the accuracy significantly decreased, which is relatively normal. The model previously had 95% accuracy because we had almost 4900 class 0 observations and 200 class 1 observations; which means if the model classifies an observation as class 0, it has a higher than 90% probability of being right.

The AUC, which is our standard measurement for a model's performance, instead, hasn't changed much, dropping by 0.01. Oversampling made our model less accurate in predicting class 1.

Mean accuracy is: 0.6937377690802348; Mean AUC is: 0.8125813873212792

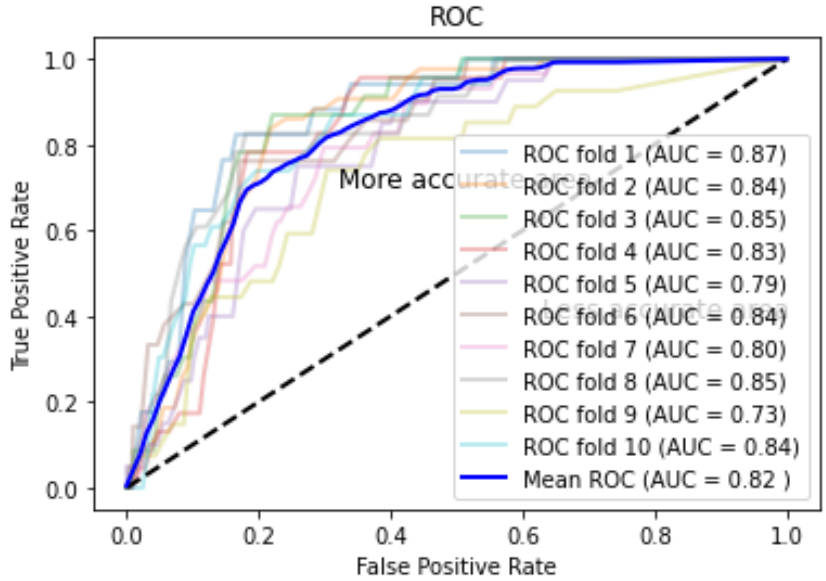


Re-sampling (undersampling and oversampling)

Since oversampling hasn't improved our model, we tried undersampling as well. However, undersampling caused our dataset to get too small, therefore we first oversampled class 1 and then undersampled class 0. The new dataset now contains about 2000 observations, almost evenly distributed.

Our model now performs slightly better than the exclusively oversampled one. Accuracy increased by 2% and AUC increased by 0.01.

Mean accuracy is: 0.71Mean AUC is: 0.82



Takeaways and Conclusion

Comparison between the performance of six models

Measure by AUC. When tie, compare accuracies	Original Model	Add Feature Selection	Oversampling	Mixed Resampling	Best Preprocessing Step
Logistic Regression	**0.84**	**/**	**0.76**	**0.82**	**0.84 (original model)**
Decision Tree	**0.55**	**0.56**	**0.59**	**0.65**	**0.65 (mixed resampling)**
Random Forest	**0.80**	**0.82**	**0.77**	**0.80**	**0.82 (feature selection)**
SVM	**0.73**	**0.80**	**0.74**	**0.78**	**0.80 (feature selection)**
Naïve Bayes	**0.81**	**0.80**	**0.73**	**0.79**	**0.81 (original model)**
KNNs	**0.82**	**0.82**	**0.81**	**0.82**	**0.82 (feature selection)**

After comparing the AUC performance across all our models, we observed that the original logical regression performs better than all the other models. We experimented by adding feature selections, oversampling and mixed resampling. However, the original model without any preprocessing outperformed them. We can safely infer that half of the population will not have strokes in the real world. With 0.95 accuracy and 0.84 AUC, logistic regression can give a relatively accurate prediction as to if a patient is likely to suffer a stroke.

After running all six models, we can see that the decision tree performs poorly compared to the others. We believe this is because the dataset is too biased.

Based on the table above, we can see that feature selection is the best preprocessing step for most of the models because we have too many features and most of them are dummy variables. Pruning out some features gives the models higher accuracy.

Feature selection throughout machine learning models

	Decision Trees	Random Forest	SVM	Naïve Bayes	KNNs	Total
age	**1**	**1**	**1**	**1**	**1**	**5**
avg_glucose_level	**1**	**1**	**1**	**1**	**1**	**5**
bmi	**1**	**1**	**0**	**1**	**1**	**4**
ever_married_Yes	**1**	**0**	**1**	**1**	**1**	**4**
smoking_status_formerlySmoked	**1**	**0**	**1**	**1**	**1**	**4**
hypertension	**0**	**1**	**1**	**1**	**1**	**4**
heart_disease	**0**	**1**	**1**	**1**	**1**	**4**
gender_male	**1**	**1**	**1**	**0**	**0**	**3**
work_type_Private	**1**	**1**	**1**	**0**	**0**	**3**
smoking_status_neverSmoked	**1**	**1**	**1**	**0**	**0**	**3**
work_type_SelfEmployed	**0**	**0**	**1**	**1**	**1**	**3**
work_type_children	**0**	**0**	**1**	**1**	**1**	**3**
smoking_status_smokes	**1**	**0**	**1**	**0**	**0**	**2**
Residence_type_Urban	**0**	**1**	**1**	**0**	**0**	**2**

We summarized the feature selections throughout six machine learning models into a table and ranked them from largest to smallest. Since logistic regression's feature selection does not have any difference to the original model, we decided not to add its results to the table.

We can see that age and average glucose level are selected as the most important features that affect the likelihood of having a stroke. Bmi, ever_married_Yes, smoking_status_formerlySmoked, hypertension and heart_disease are the features that also affect the possibility of having a stroke. By looking at the significant features, we were surprised that ever_married_Yes and work_type_Private were there. We think that these two variables are significant because of the demographic distribution of the data. In fact, there are way more married people and people working in the private sector in the dataset. We recommend collecting samples demographically evenly distributed to improve the models and see if the variables are significant or not.

Our conclusion is that the public health authorities should consider these factors above the others if they want to provide accurate predictions of whether patients are likely to have a stroke. Additionally, logistic regression machine learning models have proven the most accurate models for them to use.