Final Report - Prediction with Stroke Data Set

DSC 478

Alexandra Mischou, Kevin Thompson, Molly Adam

Executive Summary

The data set we selected for our final project was health and lifestyle data of patients and if they suffered a stroke or not. Five thousand patients were included in this dataset with information about these individuals such as age, gender, marital status, residential type, BMI, heart disease, and if they had a stroke. We wanted to create a model to predict if the patient would experience a stroke based on this data. Modeling patient data is very important to improve our healthcare system and detect diseases in their early stages. Early detection may lead to more cures or longer survival rates of patients[1].

First, we cleaned and analyzed the data visually to determine distributions and if further transformations were required. During this visualization step we were able to see some trends with the health data and stroke condition. This was a promising result that we would be able to create a model with high accuracy to predict a stroke.

Principal component analysis was performed on the dataset to simplify the complexity of the data and we found the data was already straightforward and simple. This result was still used in the KNN modeling to see if it changed the accuracy of the model.

Both k-means and a Bernoulli Mixture Model were used for clustering. Using both the elbow and silhouette method, it was determined that four clusters were the optimum for k-means. These clusters did have some overlap and spread, however, they pointed to clear positive correlations between age, average glucose level, and chances of a stroke, with two clusters capturing over 90% of our instances where a patient had a stroke. A Bernoulli mixture model was attempted after creating binary dummy variables. The optimum number of components here was fifty, with most of our data falling into six clusters, with overfitting present.

K Nearest Neighbors (KNN) modeling had a very high accuracy score in predicting whether or not a patient had a stroke. Using K = 5, our model had ~94% accuracy. This model was also used on the dataset after it had been transformed using PCA. This slightly increased the accuracy, but not in a very meaningful way. While at first, we were happy to see the accuracy of the model, upon further analysis it seems that the accuracy may be more so a result of the data we are working with, and not necessarily a great predictor on a different dataset because the model was often not predicting almost any stroke patients. Our dataset has a very small percentage of people who had strokes, so KNN is having trouble classifying patients with a stroke. This model had a very high accuracy score, and this is likely due to the dataset not being generated from real patients.

As the study of data science in healthcare becomes larger, this could be very impactful to the health of patients around the world. Using models of patient data, there are many ways to optimize process efficiencies, customize medicine solutions, and improve patient outcomes[2].

Dataset Overview

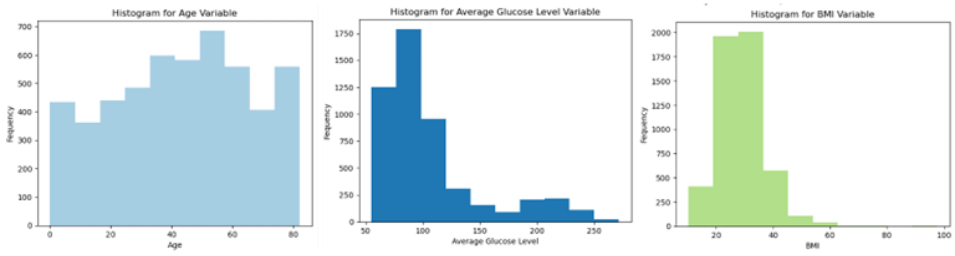
[Stroke Prediction Dataset (kaggle.com)]

The stroke dataset from Kaggle contains eleven clinical features for predicting stroke events. The clinical features are as follows: Age, Gender, Hypertension (yes or no), Heart Disease (yes or no), Marriage Status, Work Type, Residence Type, Average Glucose Level, BMI, and Smoking Status. There are 5110 instances of this data, with the features being continuous, binary, or categorical. There was very little cleaning needed as the dataset was mostly complete, however, it is worth noting that we replaced 201 missing instances of BMI with the mean.

Data Visualizations

There were three numeric variables: age, average glucose level and body mass index (bmi). In the visualization step we took these variables and plotted their histograms to see their distribution. All three histograms for these variables can be seen in Figure 1. We can see from the age histogram on the left that it has a uniform distribution with fairly equal amounts in every age group. This variable has a wide range of ages from babies aged 0 to people aged 82. From the average glucose levels distribution, we see a right skew with higher frequency at lower values. This is expected because lower levels are healthier and patients with glucose values greater than 125mg/dL are considered hyperglycemic and likely have diabetes[3]. The final numeric variable, BMI, is also right skewed with a long tail out to 97.6. Although BMI is starting to not be used as frequently because it is not a direct measure of body fat, it is still very common to gather in scientific studies because it is simple and easy to collect the data. Similar to glucose level distribution, this skew is expected due to lower bmi values corresponding to healthier individuals and BMI values greater than 25 is considered overweight and over 30 is obese[4].

Figure 1. Histograms for the numeric variables: Age, Average Glucose Level, and BMI value (listed from left to right).



In this dataset there were many nominal variables such as gender, married status, work type, residence type, and smoking status. There were also some health variables that were considered nominal because the response was either a 0 if they did not have it and 1 if the patient did have it. These were if the patient had hypertension, heart disease, or a stroke. The stroke bar chart is shown in Figure 2. In this dataset there were very few patients that had a stroke, and a large majority of patients did not have a stroke. Both heart disease and hypertension bar charts were very similar to the stroke data with a majority not having the disease. All variable bar charts can be seen in the code file “StrokeData_Visualizations” for further details. All variables were normalized before modeling to ensure the data was consistent and in a similar format.

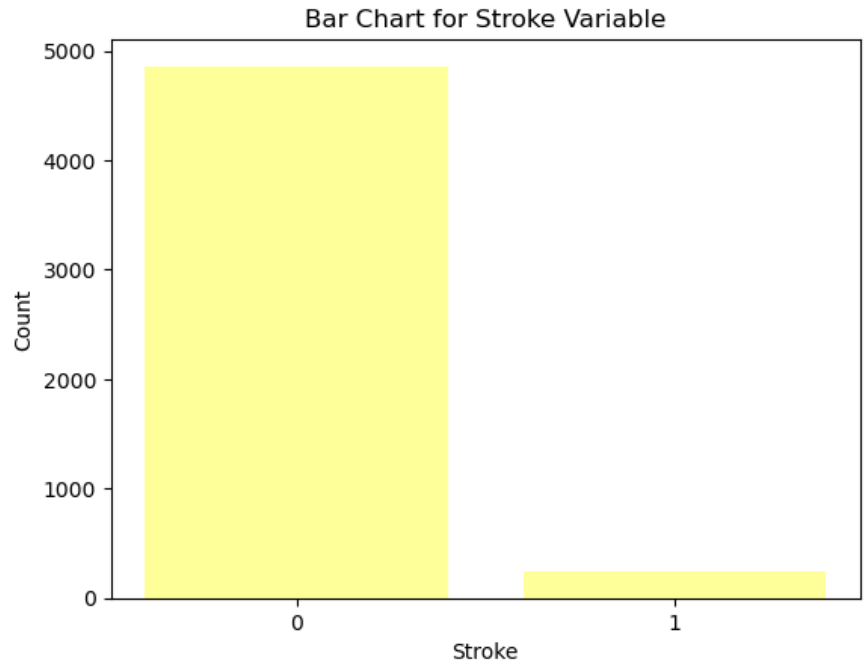
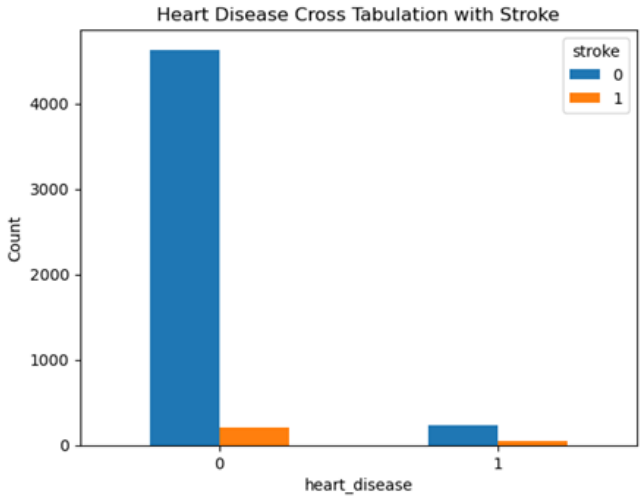


Figure 2. Bar chart for the stroke variable.

Through visualization we started to investigate which variables had an effect on if the patient had a stroke or not. Using cross tabulation, we analyzed heart disease versus stroke and the bar plot and table is shown in Figure 3. We can see a larger percentage of the patients that had a stroke also had heart disease versus the patients that did not have heart disease.

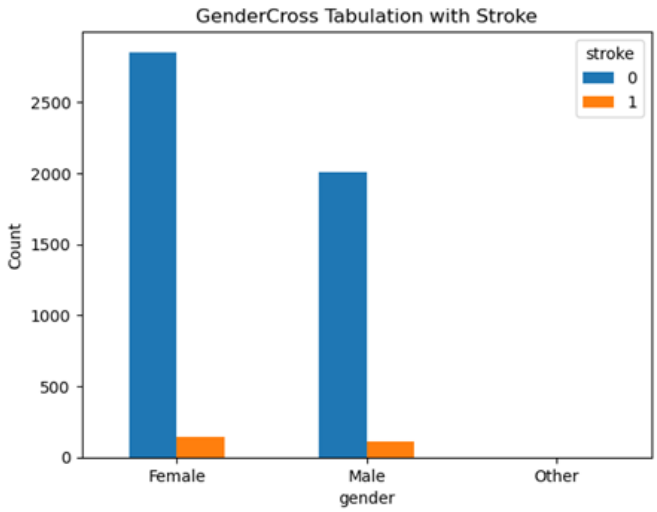
Figure 3. Heart disease cross tabulated with stroke bar plot and values.



stroke	0	1
heart_disease		
0	95.82%	4.18%
1	82.97%	17.03%

Repeating the cross tabulation with gender, there were approximately equal males and females that had a stroke. Males had a slightly higher frequency than females, but this could be within the error or variance of the dataset. Figure 4 displays these results.

Figure 4. Gender cross tabulated with stroke bar plot and values.



stroke	0	1
gender		
Female	95.29%	4.71%
Male	94.89%	5.11%
Other	100.00%	0.00%

Continuing our analysis if certain variables have a larger effect on if the patient had a stroke, we generated scatter plots where the color indicates stroke value. The purple dots corresponding to patients that did not have a stroke and yellow dots meaning yes, the patient did experience a stroke. The two numerical variables in this plot are age and average glucose levels shown in Figure 5. We can visually see a trend of increased age and glucose levels corresponding to more patients that have experienced a stroke. This data is also dense in lower glucose values and even with transparent dots, we could be not seeing some of the yellow stroke patients. Since we are able to see trends of variables having a correlation to stroke this is promising that a model can be created with high accuracy to predict if a patient will have a stroke.

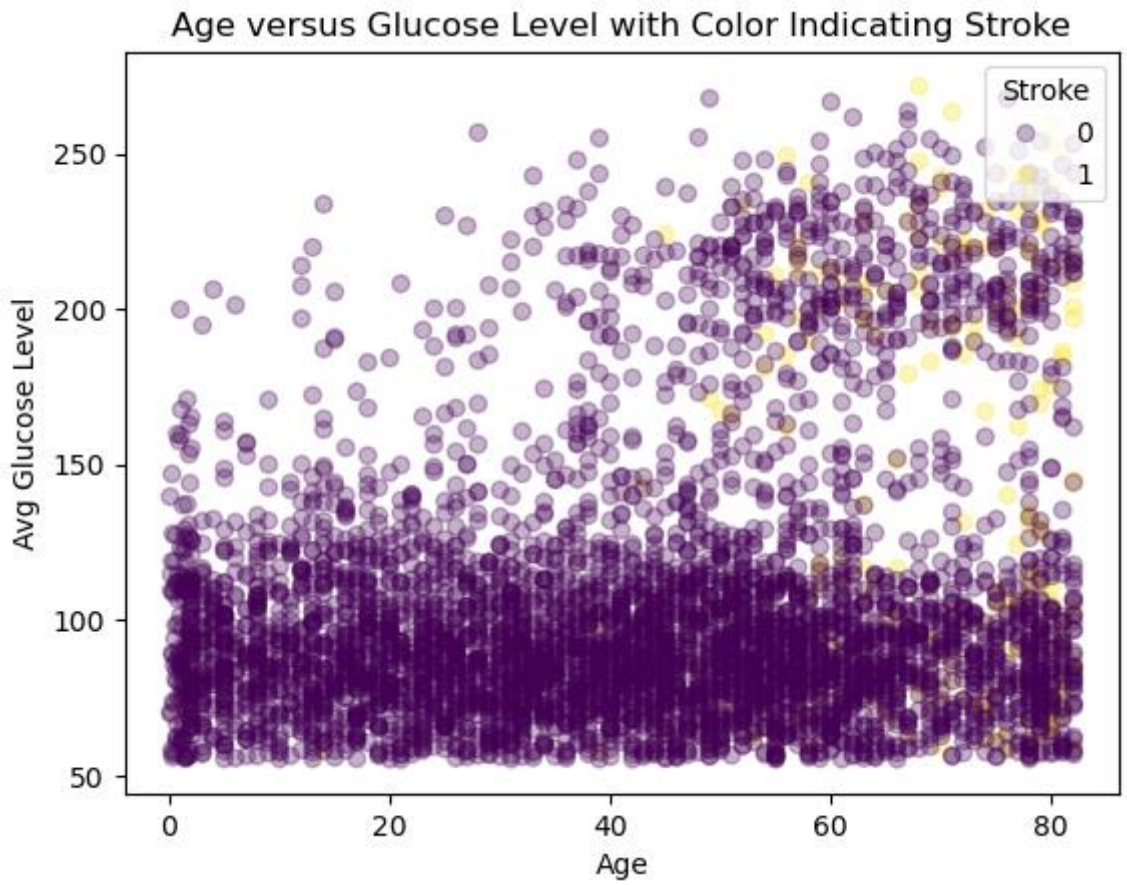


Figure 5. Scatter plot of age versus average glucose variable with color indicating if the patient experienced a stroke. The patient had a stroke is represented by the value 1 and yellow color. Purple and the value 0 in the legend means the patient did not have a stroke.

Principal Component Analysis

Principal component analysis (PCA) was performed on this dataset to simplify the complexity of the data. Since PCA was designed for continuous variables, all categorical and nominal variables were transformed into dummy variables. This increased our number of variables from 9 to 21 not including the patient ID or stroke. The data was also normalized which is critical to this analysis because PCA assumes the data is normally distributed and is sensitive to the variance of the variables. If the data is not normalized, variables with large variance will dominate other variables and your analysis will not be accurate.

After conducting PCA using the Decomposition module from Sklearn on all variables, the percent of variance captured versus the number of principal components was plotted and shown in Figure 6. We can see the line reaches approximately 95% of the variance around 10 principal components and starts to level off.

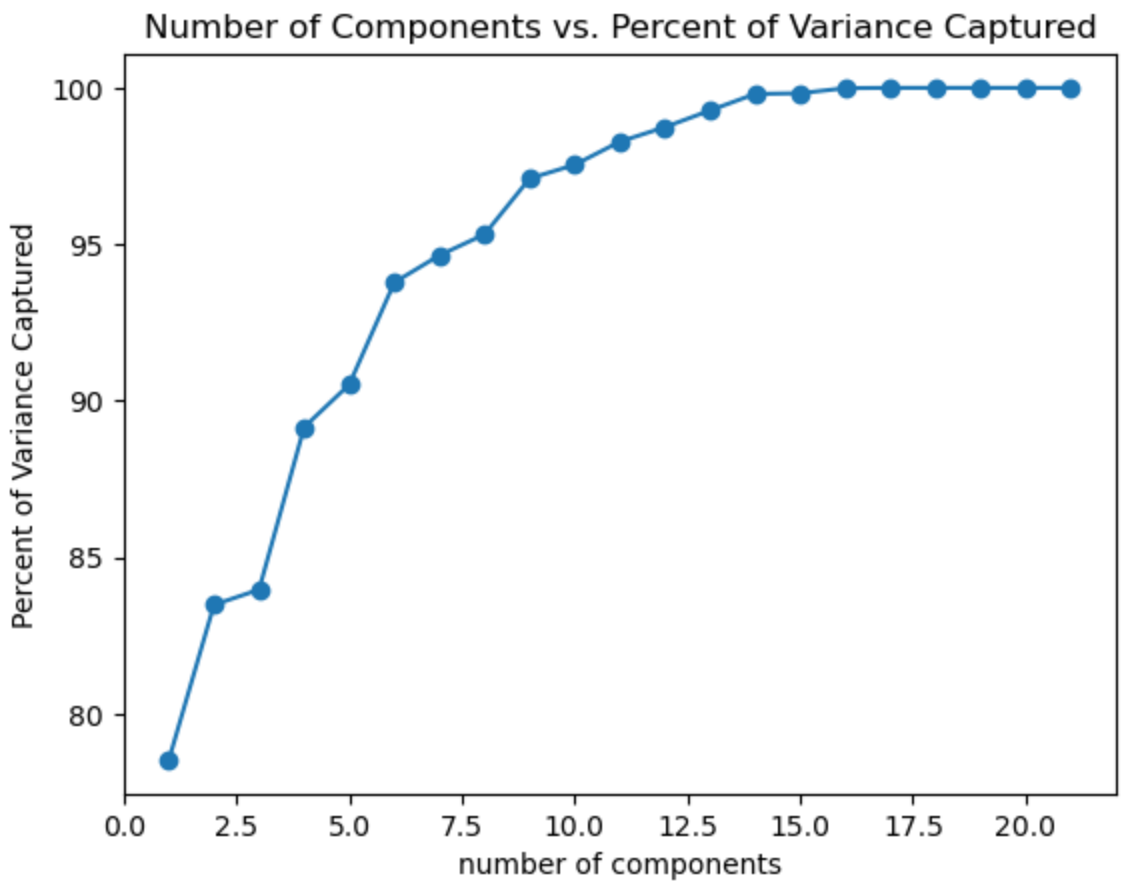


Figure 6. Percent of variance captured versus the number of principal components.

Displaying the actual values for percent of variance captured for each principal component (Figure 7) we can confirm that ten principal components capture greater than 95% of the data's variance.

```
Variance captured by Principal Components 1: 21.47
Variance captured by Principal Components 2: 37.98
Variance captured by Principal Components 3: 54.01
Variance captured by Principal Components 4: 64.85
Variance captured by Principal Components 5: 74.32
Variance captured by Principal Components 6: 80.54
Variance captured by Principal Components 7: 85.89
Variance captured by Principal Components 8: 90.57
Variance captured by Principal Components 9: 93.47
Variance captured by Principal Components 10: 95.92
Variance captured by Principal Components 11: 97.64
Variance captured by Principal Components 12: 98.9
Variance captured by Principal Components 13: 99.62
Variance captured by Principal Components 14: 99.82
Variance captured by Principal Components 15: 99.99
Variance captured by Principal Components 16: 100.0
Variance captured by Principal Components 17: 100.0
Variance captured by Principal Components 18: 100.0
Variance captured by Principal Components 19: 100.0
Variance captured by Principal Components 20: 100.0
Variance captured by Principal Components 21: 100.0
```

Figure 7. The variance captured printed out for each principal component.

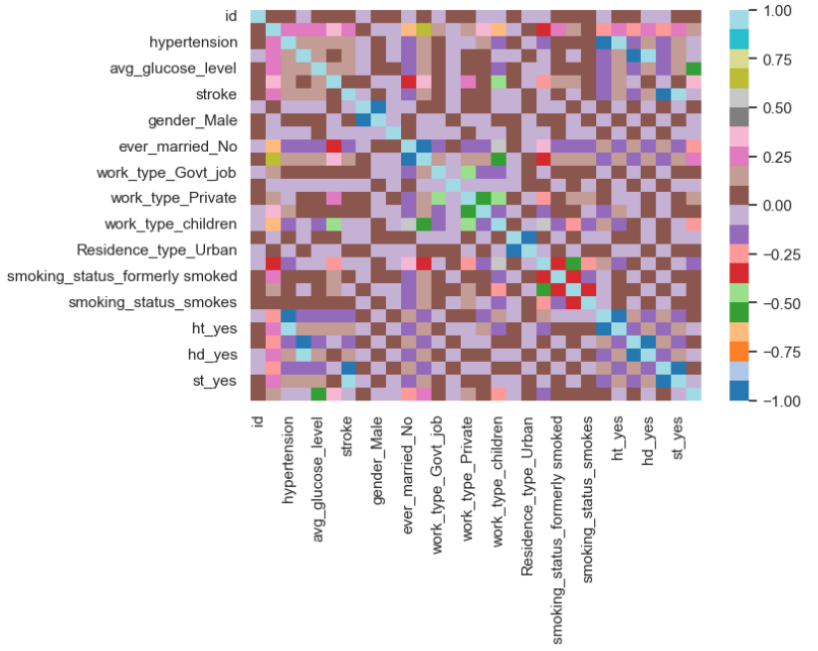
This result of 10 principal components appears to be showing that the dummy variables are being cut and not adding additional information to the dataset. We started with 9 variables and increased up to 21 using dummy variables. These 10 principal components likely are the original variables plus an additional dummy variable. Our dataset was not high dimensional from the start with only 9 variables and ~5,000 rows of data, so PCA is not as effective.

The PCA result was used for later analysis in the K-Nearest Neighbors modeling.

Clustering

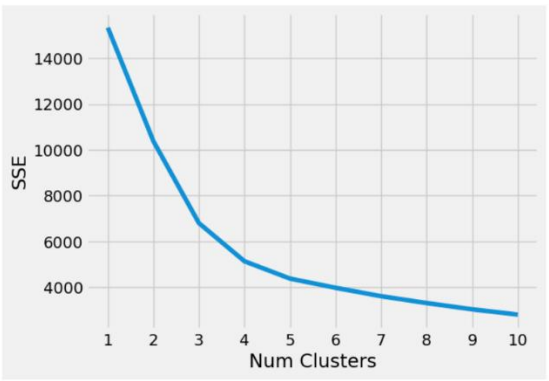
Both k-means and a Bernoulli Mixture Model were applied for clustering to capture the continuous predictive features and the categorical. Before k-means, a heatmap based on the correlation plot between all the dummy variables was used to see if there were any strong correlations between our features. There was little indication of correlation, with most correlation values falling within the range of (-0.5, .5).

```
In [88]:
M = dataset.corr()
ax = sns.heatmap(M, cmap='tab20')
```

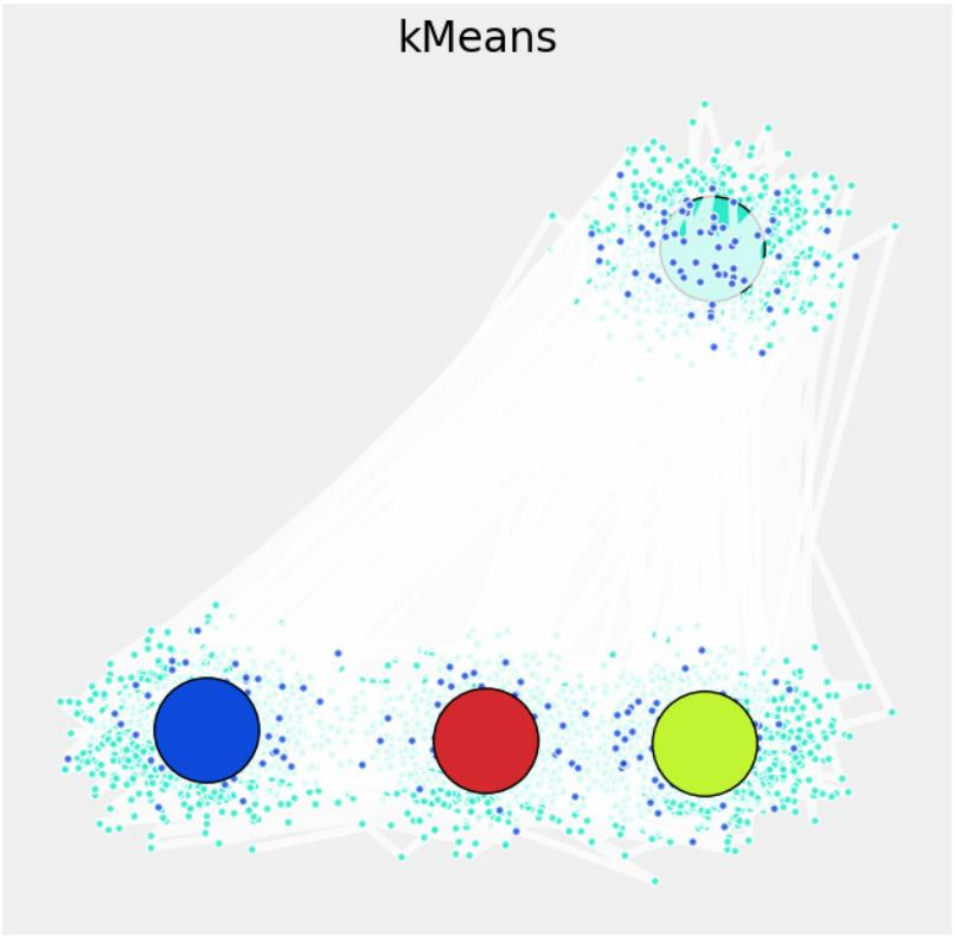


We can see here that there are few correlations with stroke and possibly a correlation between work type and hypertension. Because we will only be using continuous numeric variables when we apply k-means clustering, we do not have to worry about these for now. We applied the skLearn standard scaler function to prepare the data for clustering, which scales the features so that the mean of our data is zero and the standard deviation is one.

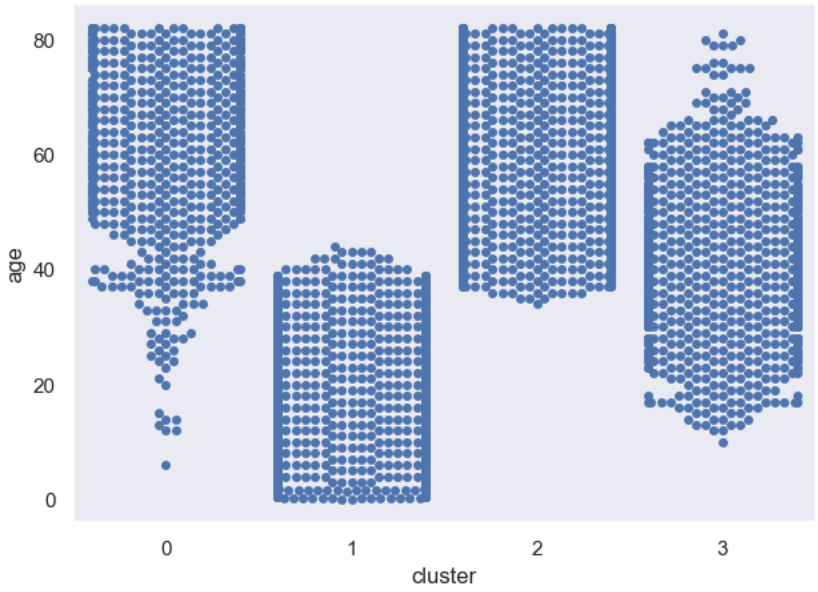
We tested to see which value for k led to the smallest sum of squared errors (SSE), and then displayed this in a plot so we could apply the elbow method to determine the optimal number of clusters. We also created a silhouette plot to see what value for k led to the most consistency within our clusters of data. This peaked at about .38, and matched the value for k we found from using the elbow method, which was four main clusters. This demonstrated to me that there were multiple features that could indicate the possibility of a stroke, and that there may be a hidden factor connecting some of our predictive features. An example would be if BMI becomes a stronger indicator of stroke risk as age increases.

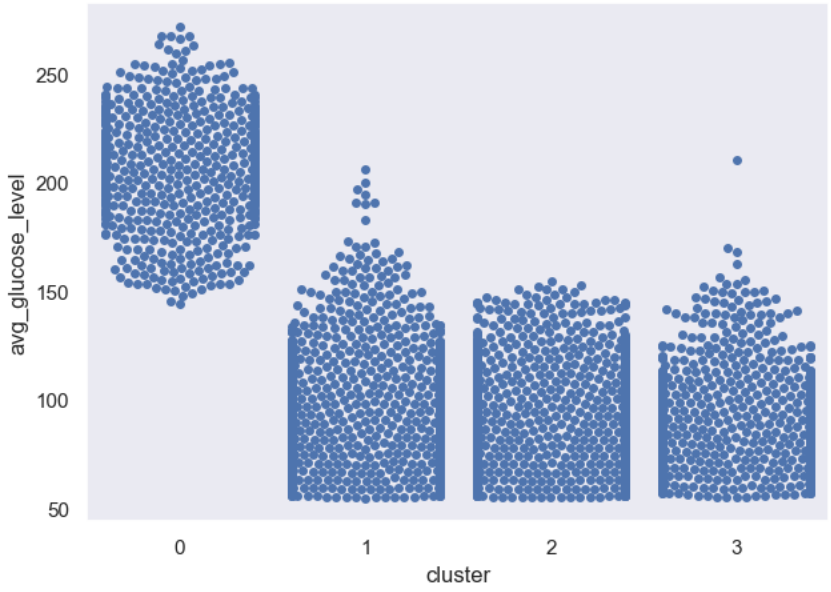


After running the model with random initialization, four clusters, and ten iterations, we could visualize the clusters as follows:



We can clearly see that there are four pockets of data, with some significant spread as well as some overlap between several of the clusters. We created swarmplots using skLearn, which combine a standard scatterplot with the ability of a boxplot to clearly show distribution. We could see that clusters 0 and 3 had captured 230 stroke instances out of 249, with those clusters having a positive relationship with age as their median age was about sixty, while the other two only had median ages of 40 and 20. The average glucose level for cluster 0 was about 200 compared to a median of 100 for the remaining clusters.

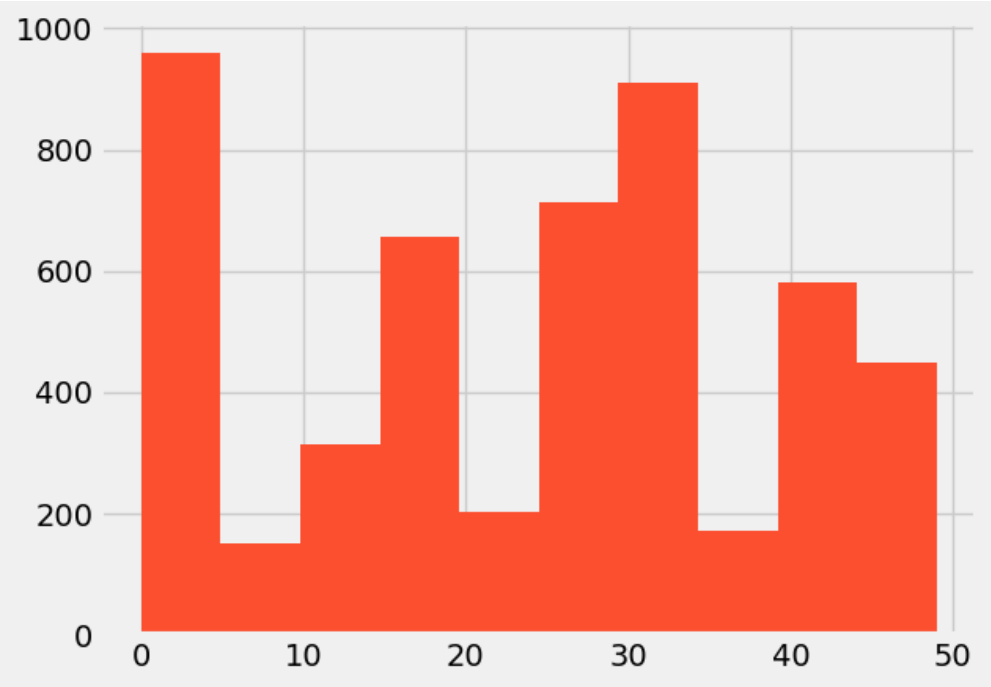





```
In [95]: print("What clusters were most likely to contain stroke_yes? \n" , dataset.grou
```

```
What clusters were most likely to contain stroke_yes?
                      id                                                    \
                   count          mean           std      min       25%
cluster st_yes
0       0          583.0  37084.048027  21046.094013    239.0  17792.00
        1           89.0  38756.977528  21708.140977   1210.0  20387.00
1       0         1633.0  36094.510104  20900.545395     67.0  17492.00
        1            4.0  47767.250000  16399.886572  31720.0  37864.00
2       0         1751.0  36259.587664  21287.534759     84.0  17372.50
        1          141.0  35427.262411  22601.689803    210.0  14248.00
3       0          894.0  37261.278523  21244.492025     99.0  18888.75
        1           15.0  40397.866667  18725.241783  12095.0  27918.00
```

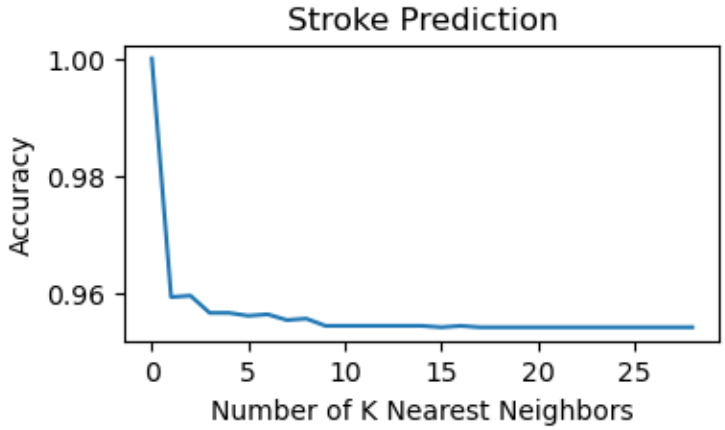
Overall, this was successful in that it helped us see some correlations that were not easily detected, however, we wanted to complete the Bernoulli Model as well to account for the categorical indicators included in the dataset. The idea is that if we converted our categorical indicators to dummy binary variables, we could run the Bernoulli clustering model, which creates a mixture of binary vectors to represent a conditional distribution of 'x' or our features. It uses the probabilities as 'weights' to determine cluster. We split the dummy variable dataset into train_test sections and ran a function to determine the optimum number of components to include in the model. The model score was 13 which indicates serious overfitting. This is likely why the data was spread among eight clusters, none of which gave a clear indication of which variables were having the strongest impact. We could also see that stroke risk was spread among clusters, and not concentrated as with the k-means approach. This indicates that some predictors only become stroke risk factors when paired with specific values of other predictors. For example, job type may only be a risk factor with certain genders or age ranges. These combinations can likely be captured in our principal components analysis. Below is an image containing the clusters and the total number of instances within each.



KNN Modeling

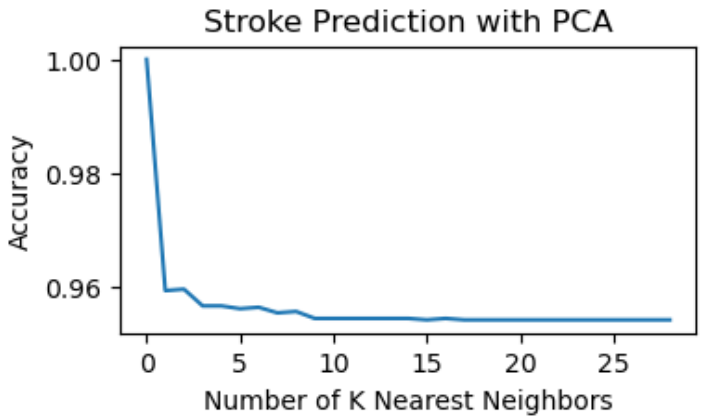
After clustering, we also wanted to look at a classifier model. KNN modeling was used to predict whether or not a patient would have a stroke. This model was used on both the PCA transformed data as well as the original set. The data was min-max normalized, and dummy variables were created to preprocess the data for KNN.

First, we compared the accuracy of our model with different values of K:



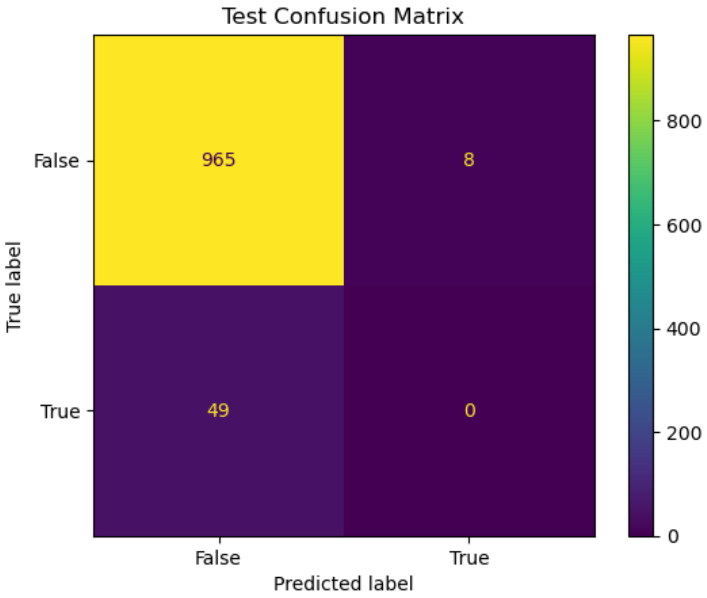
From this test we decided to continue with K=5. While values between 1 and 8 were very similar in accuracy, we wanted to not have such a small K like 1 or 2, but also not have a large K when it didn't improve the accuracy. The data was then split into testing and training sets. The accuracy held from the training to the testing set; the training set had an accuracy of 95.6% and the test prediction had an accuracy of just under 94%. This slight difference was also a good sign that the data was not overfit to our training set.

Next, KNN was used on the PCA transformed data. Testing for the best K value, the accuracies were the same as on the non-PCA transformed data:

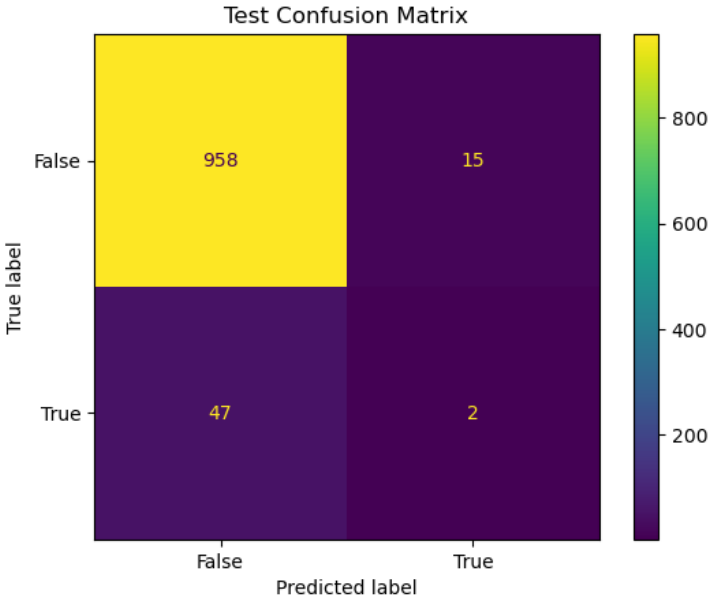


So the same K = 5 value was used. The data was once again split into testing and training but using a different seed. With PCA our test set had an accuracy of 94% as well, although it was about 0.5% higher.

From this the model seems very accurate, however looking at the confusion matrix a more interesting story developed:



As we can see from the confusion matrix, the majority of the patients were classified as not having a stroke. This is where the high accuracy number comes from because in the dataset, the vast majority of patients didn't have a stroke. But when we look at the number of True Positives, not a single one was labeled correctly. We tried multiple different seeds for splitting the data, re ran the model multiple times, and while this 0 True Positives is extreme, the general trend was true for all attempts at running the model. The model was very bad at accurately predicting the stroke patients. Using a lower value for K, such as K=3, the model was able to predict a few more True Positives in the test set:



Since such a large amount of the data is non-stroke patients, KNN is assigning almost every patient to non-stroke. And the accuracy, I would argue, is not a good indicator of the model since the confusion matrix is showing a lack of accuracy in actually predicting the strokes.

Going forward to get a better classification model, one consideration would be more bias. For example, it would be interesting to do regression analysis and see which variables are more correlated to having a stroke. Perhaps these values could be weighted more heavily in a future KNN model. The other idea is that we need more data. The dataset is not very large with only about 5,000 objects. But particularly the subset of stroke victims is very low at only a few hundred.

Conclusion

KMeans clustering was by far our best predictor of whether or not there was a stroke. Kmeans clustering also provided us with interesting underlying patterns. KNN was a very poor predictor for true positives, and we would not want to use it as a predictor for this reason on other datasets. Our data was very skewed towards non-stroke and because the dataset was fairly small, we have some concerns of overfitting in our models. Further analysis would be interesting on a much larger dataset looking at the presence of stroke.

Resources

1. Early detection of disease and scheduling of screening examinations - PubMed (nih.gov)
2. Modern Applications of Data Science in Health Care (sandiego.edu)
3. Hyperglycemia (High Blood Sugar): Symptoms & Treatment (clevelandclinic.org)
4. bmiforpactitioners.pdf (cdc.gov)