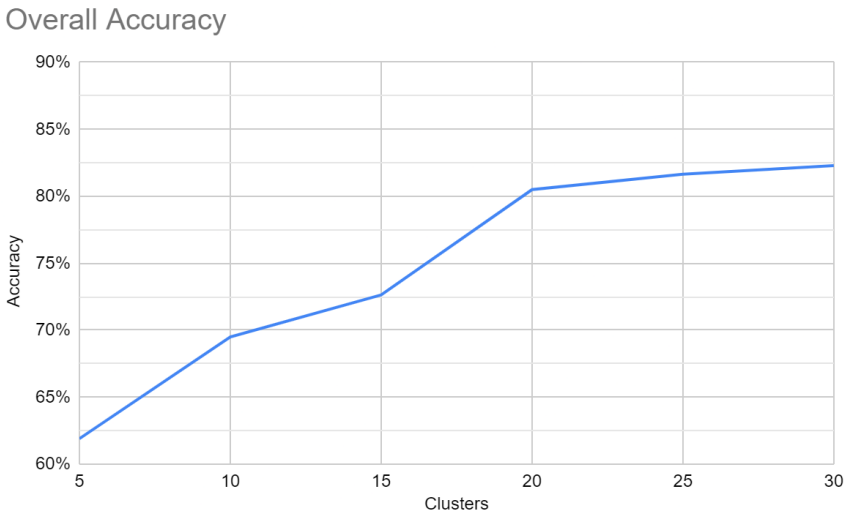
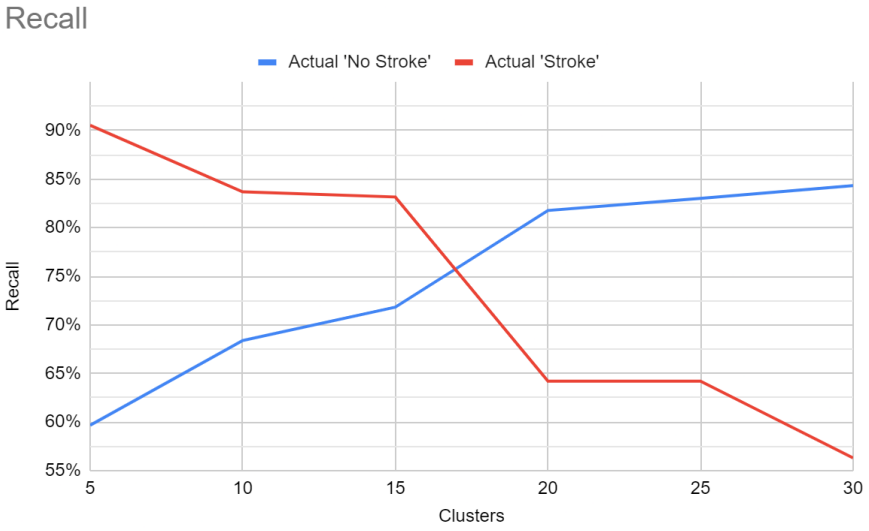
K-Means Classifier

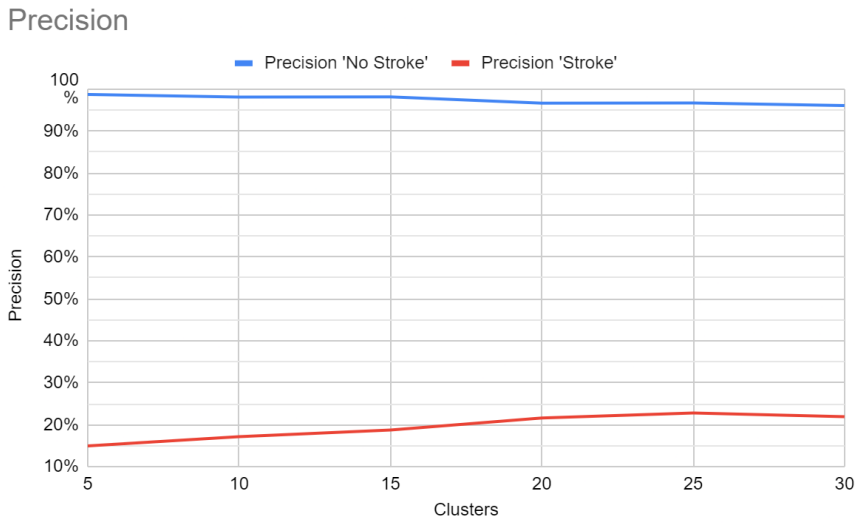
We attempted to create a k-means classifier for the stroke data set. The data set includes 17 data dimensions and a classifier. Some of the data is one-hot encoded categorical data, such as occupation, marital/children status, and smoking history. There are 2430 patients who haven't had a stroke and 190 who have.



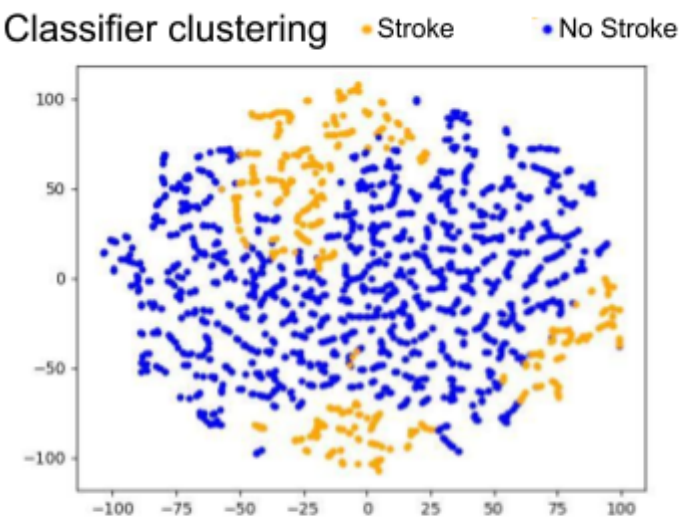
As the number of clusters was increased, the Overall Accuracy of the classifier increased to a peak of 82.29%. This accuracy is a result of more 'no stroke' patients being correctly identified. Since patients who have no had a stroke make up over 90% of the patients, the accuracy increases come from better classification of this group, but this has problems, discussed next.



This accuracy falls short of our goal of 95+% accuracy. For low numbers of clusters, the Recall for identifying patients who've had strokes is fairly high, over 90%. Unfortunately, this comes at a cost of misidentifying many 'No Stroke' patients as having had a stroke. As the number of clusters increases, this misidentification decreases and the recall on Actual 'No Stroke' patients peaks at 84.32%. The recall of Actual 'Stroke' patients drops to just over 55%.



Finally, the precision of correctly identifying 'stroke' and 'no-stroke' patients was a mixed result. The classifier was precise at correctly classifying patients who hadn't had a stroke, but absolutely terrible at precisely classifying stroke patients. The precision peaked at 22.8%.



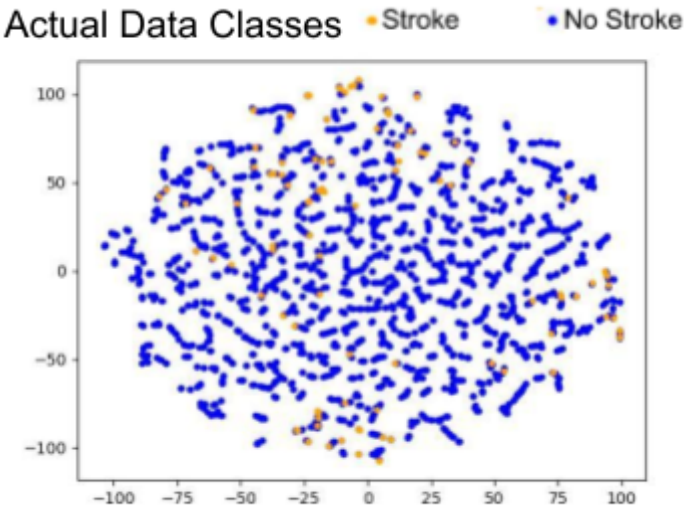


Figure 1 & 2: 15 cluster classifier, dimensions reduced by t-SNE.

I believe the problems with the classifier are due to the sparsity of the data. There aren't clear clusters of data that can be partitioned. Where the data for stroke patients is clustered, there are more 'no stroke' data points within those clusters. A higher cluster count decreased the misclassification of 'no stroke' data but at the same time increased 'stroke' misclassification. This led to a net improvement in accuracy simply because of the size of the 'no stroke' class. More relevant data dimensions might help separate the classes.

There is another possible reason for the error. The 'no stroke' data that falls into a cluster classified as stroke could be an indicator of increased stroke risk. Which would mean the classifier is working better than it appears. Perhaps switching to a Gaussian probability score rather than discrete clustering would improve performance as well.

Data:

5 Clusters	Predicted NO STROKE	Predicted STROKE	Recall
Actual NO STROKE	1450	980	59.67%
Actual STROKE	18	172	90.53%
Precision	98.77%	14.93%	
Accuracy	61.91%		

20 Clusters	Predicted NO STROKE	Predicted STROKE	Recall
Actual NO STROKE	1987	443	81.77%
Actual STROKE	68	122	64.21%
Precision	96.69%	21.59%	
Accuracy	80.50%		

10 Clusters	Predicted NO STROKE	Predicted STROKE	Recall
Actual NO STROKE	1662	768	68.40%
Actual STROKE	31	159	83.68%
Precision	98.17%	17.15%	
Accuracy	69.50%		

25 Clusters	Predicted NO STROKE	Predicted STROKE	Recall
Actual NO STROKE	2017	413	83.00%
Actual STROKE	68	122	64.21%
Precision	96.74%	22.80%	
Accuracy	81.64%		

15 Clusters	Predicted NO STROKE	Predicted STROKE	Recall
Actual NO STROKE	1748	686	71.82%
Actual STROKE	32	158	83.16%
Precision	98.20%	18.72%	
Accuracy	72.64%		

30 Clusters	Predicted NO STROKE	Predicted STROKE	Recall
Actual NO STROKE	2049	381	84.32%
Actual STROKE	83	107	56.32%
Precision	96.11%	21.93%	
Accuracy	82.29%		