Written Report – 6.419x Module 2

Name: ajland

Problem 1

1.1. (3 points) Provide at least one visualization which clearly shows the existence of three main brain cell types as described by the scientist, and explain how it shows this. Your visualization should support the idea that cells from different groups can differ greatly.

Solution:

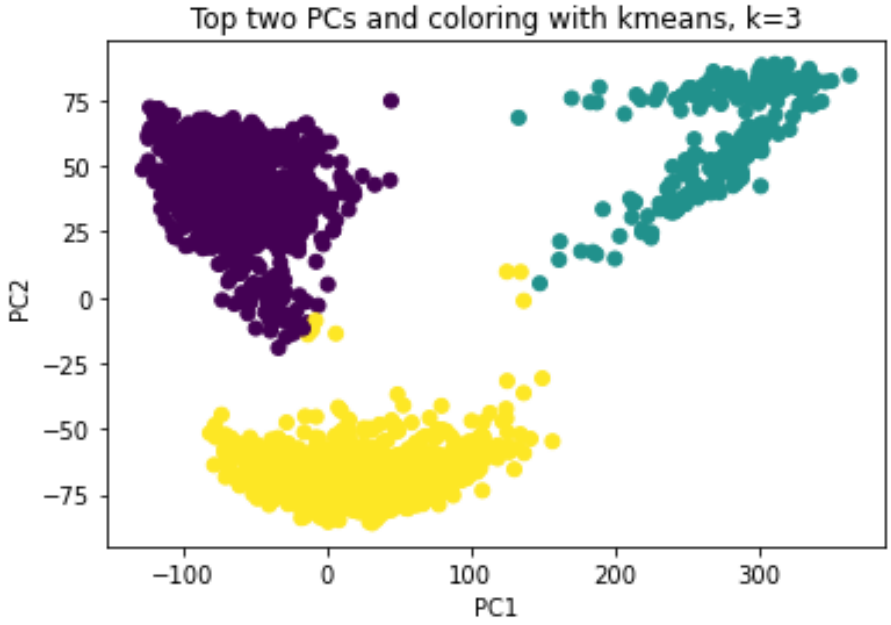

Figure 1: Three visually distinct cell clusters colored using k-means with k=3

I begin by performing k-means clustering with $k = 3$ clusters on the first 500 principle components (PCs) of the log transformed gene expression data. I then plot the top two PCs and color them with the cluster assignments from k-means, resulting in Figure 1. It is apparent in this view that there are three main clusters, which are nearly linearly separable, which coalesces with the scientist's statements about three main brain cell types. You can even begin to see the sub-clusters indicating the cell sub-types.

1.2. (4 points) Provide at least one visualization which supports the claim that within each of the three types, there are numerous possible sub-types for a cell. In your visualization, highlight which of the three main types these sub-types belong to. Again, explain how your visualization supports the claim.

Solution:

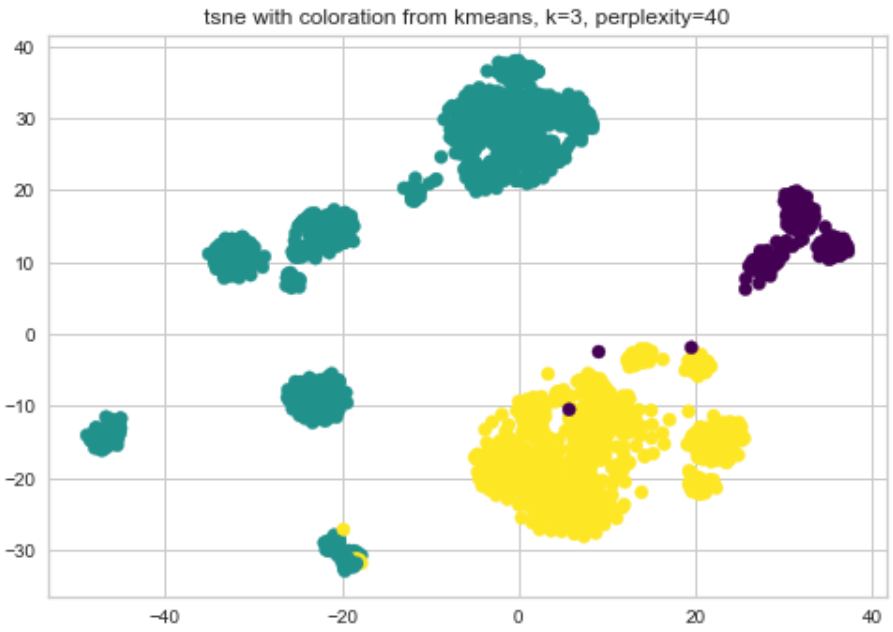

Figure 2: T-SNE non-linear projection showing several sub-types within larger cell types

Again using k-means with $k = 3$, Figure 2 shows a T-SNE projection. Here you see three mostly separable distinct cell types (illustrated by color) along with cell sub-types within each color/main cell type. For example, within the yellow clusters there appear to be 4-5 subtypes, while in the teal main cell type there are 5-8 cell sub-types.

Problem 2

2.1. (4 points) Using your clustering method(s) of choice, find a suitable clustering for the cells. Briefly explain how you chose the number of clusters by appropriate visualizations and/or numerical findings. (to cluster cells into the subcategories instead of categories)

Solution: Since the goal is to determine an appropriate k for subcategories, we can set $k = 9$ based on the elbow plot in Figure 3, which is approximately the point where the curve "elbows."

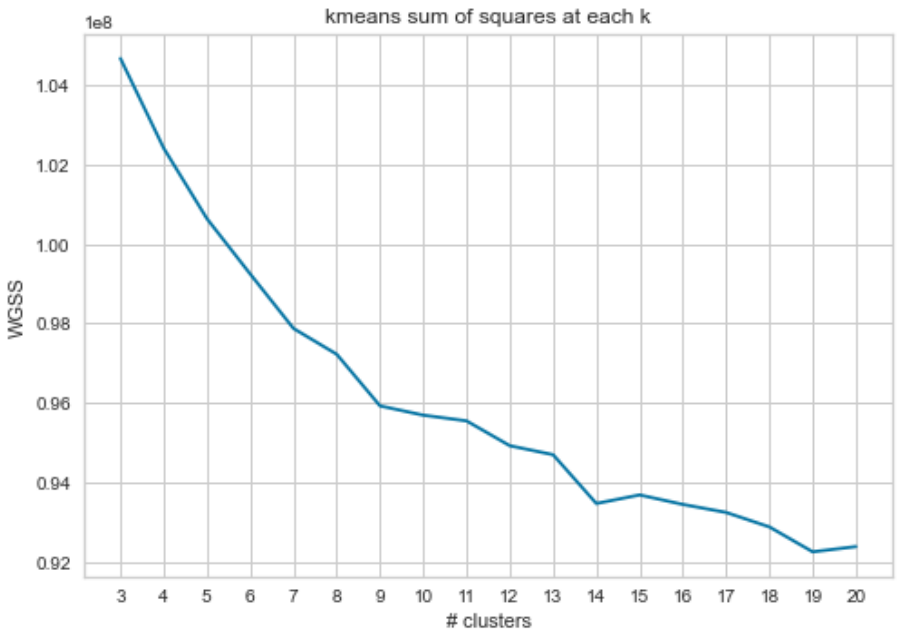

Figure 3: Elbow plot showing WGSS for each cluster; for $k = 10$, $WGSS = 86e6$

This time I used 1253 principle components since this count explains 85% of the variance. This cluster count also aligns reasonably well with the number of subcategories seen in Figure 4, which is a 3D plot of the top three PCs.

While some clusters look as though they could be one cluster, it is possible that they are rightly many in higher dimensions (note this is even be seen in going from a 2D to 3D plot in Figures 1 and 4, respectively).

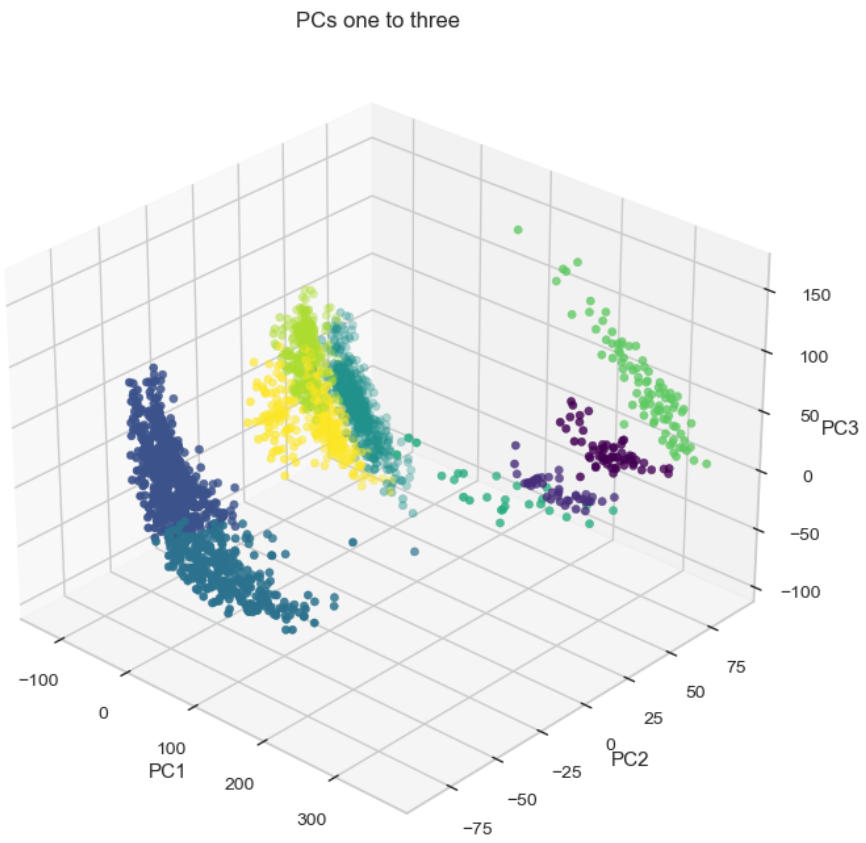

Figure 4: 3D scatter of top three PCs showing k-means coloring; k=9

2.2. (6 points) We will now treat your cluster assignments as labels for supervised learning. Fit a logistic regression model to the original data (not principal components), with your clustering as the target labels. Since the data is high-dimensional, make sure to regularize your model using your choice of ℓ_1, ℓ_2, or elastic net, and separate the data into training and validation or use cross-validation to select your model. Report your choice of regularization parameter and validation performance.

Solution: Using the results from $k = 9$, I performed logistic regression with the following code,

```
X_train, X_test, y_train, y_test = train_test_split(
X, y_85, train_size=0.8, random_state=523, shuffle=True
)

log_reg = LogisticRegressionCV(cv=5,Cs=[0.01,0.1,1,10],max_iter=5000,
penalty="l2",solver="liblinear",multi_class="ovr", n_jobs=-1)
log_reg.fit(X_train,y_train)
```

which used 5-fold cross-validation on 80% of the data with l2 regularization. The model achieved a score of 1.0 on the train data and a 0.968 on the remaining 20% of the test data.

2.3. (9 points) Select the features with the top 100 corresponding coefficient values (since this is a multi-class model, you can rank the coefficients using the maximum absolute value over classes, or the sum of absolute values). Take the evaluation training data in p2_evaluation and use a subset of the genes consisting of the features you selected. Train a logistic regression classifier on this training data, and evaluate its performance on the evaluation test data. Report your score. (Don't forget to take the log transform $\log_2(x+1)$ before training and testing.) Compare the obtained score with two baselines: random features (take a random selection of 100 genes), and high-variance features (take the 100 genes with highest variance). Finally, compare the variances of the features you selected with the highest variance features by plotting a histogram of the variances of features selected by both methods.

Solution: Taking the highest 100 features corresponding to the sum of absolute values of coefficients over the $k = 9$ clusters, I fit a new logistic regression model using the same parameters as above. Using only these features on the log-transform of X_{train}, I achieve train score of 0.999; evaluating the model on the log-transform

of X_{test}, I achieve 0.923. Comparatively, the train and test scores for the 100 randomly selected features are 0.539 and 0.360, respectively. Similarly, the train and test scores for the 100 highest variance features are 0.997 and 0.938, respectively. Finally, a histogram showing the variances of the 100 highest-variance features and 100 top-coefficient features is shown in Figure 5.

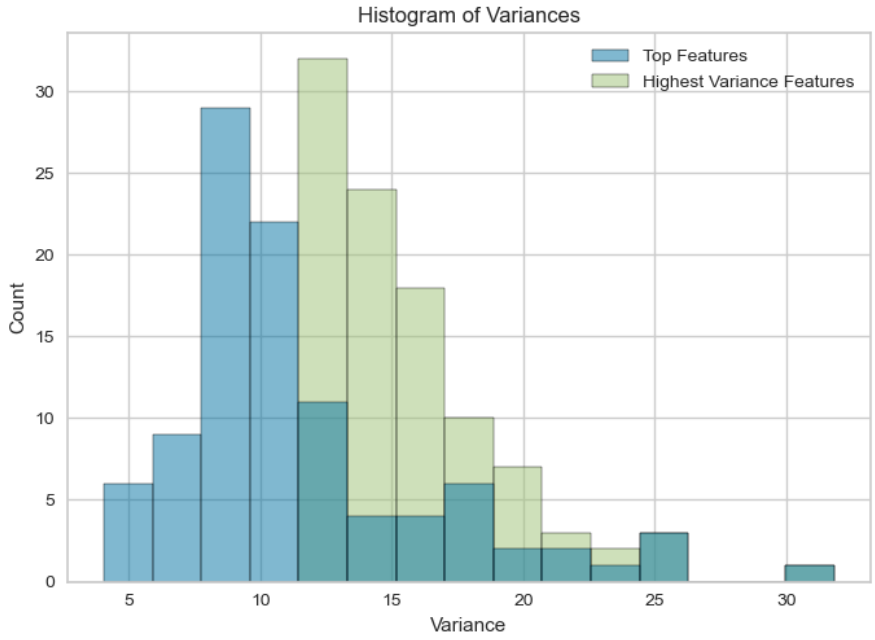

Figure 5: Two histograms where green is the showing the distribution of variances of the 100 highest-variance features and blue is showing the distribution of variances of the 100 features corresponding to the highest coefficients in the previous problem

Interestingly, these histograms are similar but are not the same. In the feature selection histogram, the labels were determined using k-means on data projected onto several principle components. That is, some of the directions of highest variance played a role in bringing data closer together (recall that the data is originally sparse). The subsequent logistic regression model then prioritized high-variance features (i.e. directions in the original basis) due to the direct influence of the k-means label on the logistic loss function. There may be several reasons why the histograms are not the same, one of which I believe relates to how many principle components I projected onto. Another may be how y_train and y_test were determined in the p2_evaluation folder, which used 36 clusters. Another still may be

the number of chosen clusters or how the important features were picked with an absolute sum of the weights over the clusters.

Problem 3

3.1. (3 points) When we created the T-SNE plot in Problem 1, we ran T-SNE on the top 50 PC's of the data. But we could have easily chosen a different number of PC's to represent the data. Run T-SNE using 10, 50, 100, 250, and 500 PC's, and plot the resulting visualization for each. What do you observe as you increase the number of PC's used?

Solution: Using the data in p2_evaluation_reduced and plotting the results, Figure 6 is produced.

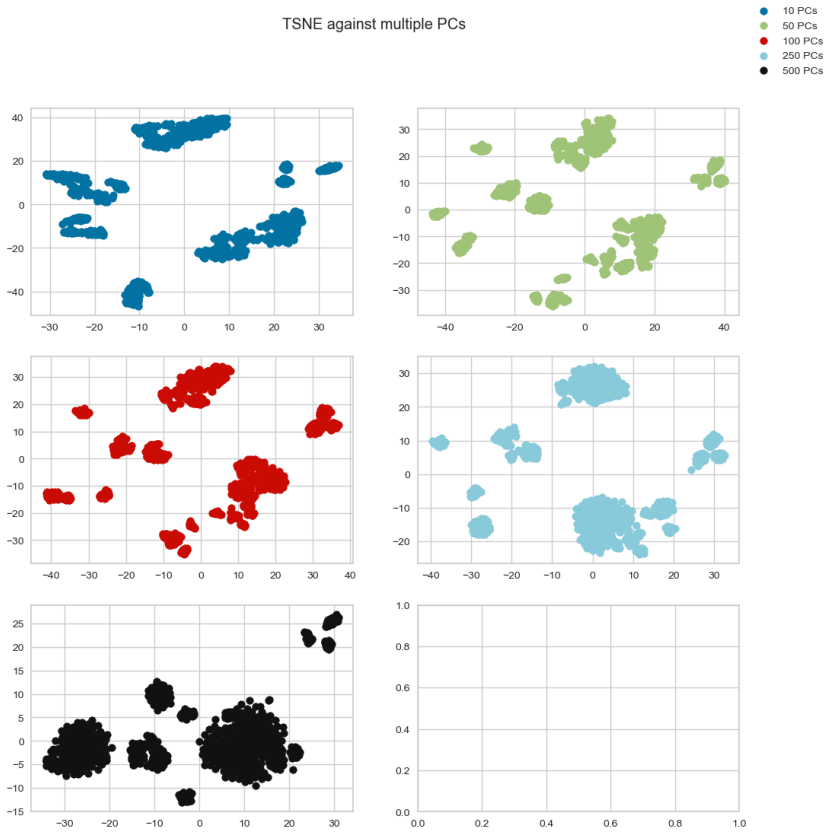

Figure 6: T-SNE non-linear projection onto several principle components. Perplexity = 40 for all plots.

It appears that as the number of PCs increase, the clusters tend to become larger. At first, however, the clusters appear to break apart, as in the transition from 10 to 50 PCs. After 50, there appears to be less distinction between smaller clusters. Some things maintain consistency, however, such as the clover pattern in the upper right of each plot, though the spread of the petals varies slightly.

3.2. (13 points) Pick three hyper-parameters below (the 3 is the total number that a report needs to analyze. It can take a) 2 from A, 1 from B, or b) 1 from A, 2 from B.) and analyze how changing the hyper-parameters affect the conclusions that can be drawn from the data. Please choose at least one hyper-parameter from each of the two categories (visualization and clustering/feature selection). At minimum, evaluate the hyper-parameters individually, but you may also evaluate how joint changes in the hyper-parameters affect the results. You may use any of the datasets we have given you in this project. For visualization hyper-parameters, you may find it productive to augment your analysis with experiments on synthetic data, though we request that you use real data in at least one demonstration.

Solution: From A I will cover "T-SNE learning rate" and from B I will cover "Effect of number of PC's chosen on clustering" and "Type of regularization (L^1, L^2, elastic net) in the logistic regression step and how the resulting features selected differ."

Part A

1. T-SNE learning rate

In examining how learning rate affects T-SNE, I will keep all other parameters constant. To this end, I will only use the top 50 PCs, set the perplexity to 40 and use a fixed random_state. All other parameters will be left to default.

To begin, Figure 7 depicts the results of several runs of T-SNE with different learning rates. All runs converged within the max iteration limit.

Surprisingly, the end result does not appear to be affected much by the range of different learning rates in the range of 10 to 1000. The higher two learning rates begin to deviate more, but even still have striking similarities. Plotting the KL-Divergence against learning rate (Figure 8), we can see which learning rate minimizes the similarity measure.

Notably, the KL-Divergence with the given parameters is minimized at a learning rate of 300, but becomes an order of magnitude larger at 5000 and 10000.

Let's examine the effect of learning rate on a toy example where the starting dimensionality is two and we have two gaussian mixtures with different means and same variances (Figure 9). After several runs, the 500 learning rate appear to

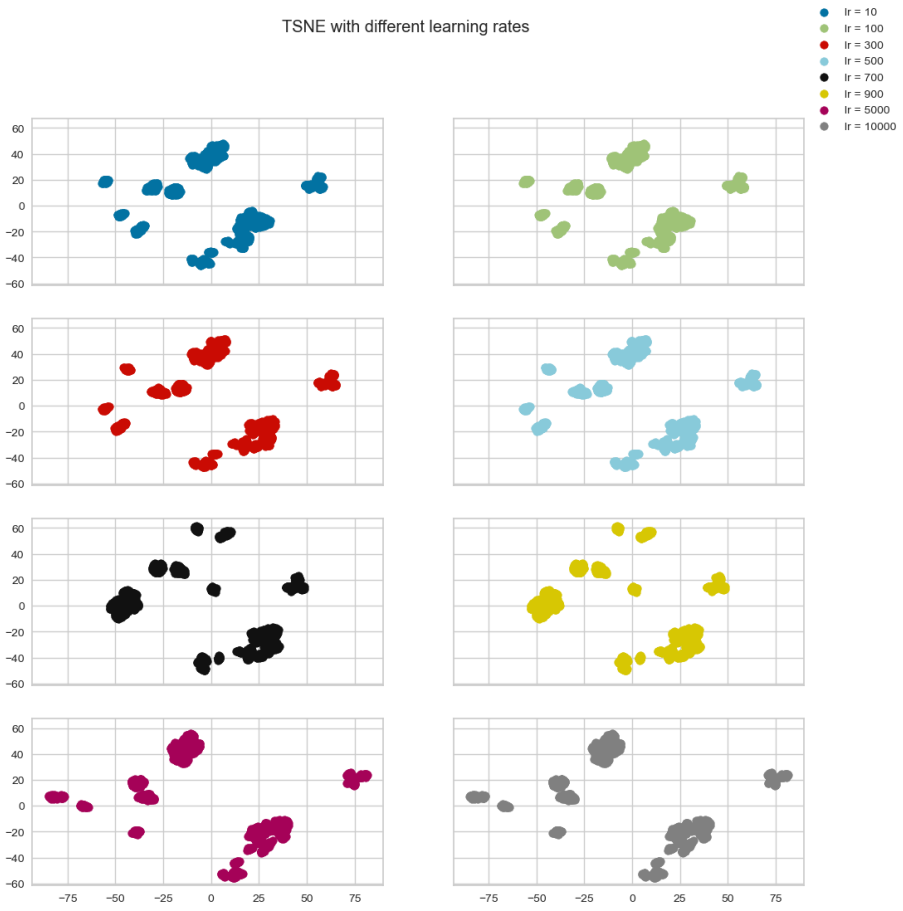

Figure 7: Result of T-SNE algorithm with varying learning rates

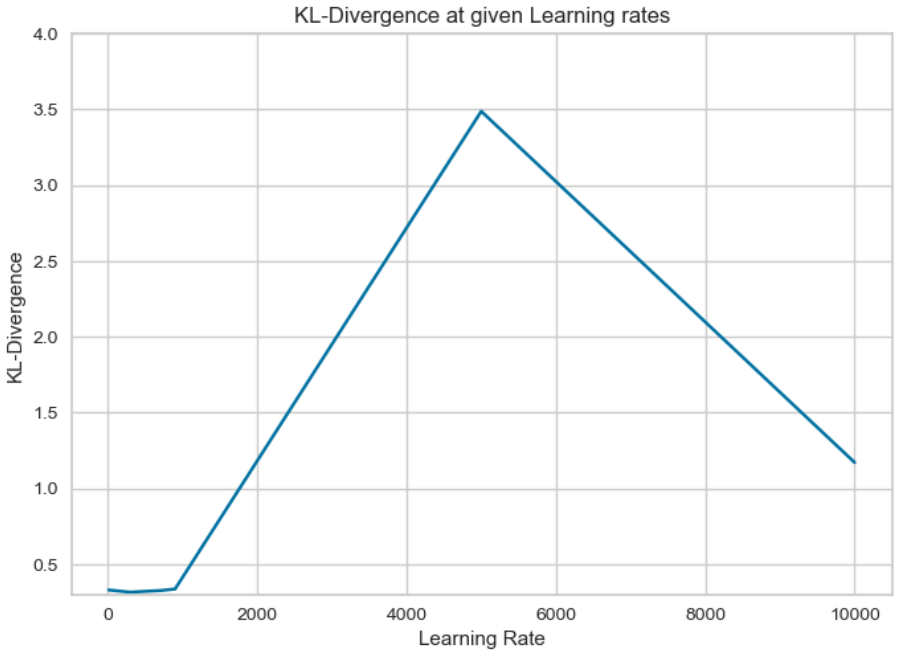

Figure 8: KL-Divergence as a function of learning rate in T-SNE algorithm

be able to consistently separate the data and have similar structure to the original. The smallest learning rate consistently separates the two clusters and simultaneously elongates them. Learning rates 750, 1000 and 5000 have largely inconsistent results after multiple repeated runs. This variation comes from the random initialization of the original Gaussian's - results do not change if the seed is fixed.

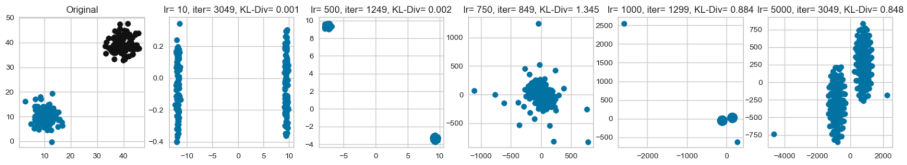

Figure 9: Left most plot is original Gaussian mixture and moving to the right is the result of increased learning rate. Perplexity = 100, which is the number of points originally in each cluster.

The only acceptable plots according to the KL-Divergence corresponds to learning rates 10 and 500, while the 500 seems to represent the original structure best. Both results would have one conclude that there are two clusters, whereas the other learning rates could have you believing there are several more clusters which in reality are outliers in the original. Depending on the initialization, the higher learning rates could lead to several conflicting conclusions since it can converge to substantially different results.

Part B

1. Effect of number of PC's chosen on clustering

For this portion of the analysis, we will use $k = 9$, which is the count I selected previously, and evaluate the effect of the number of PCs used in clustering. We first do this by plotting the WGSS as a function of number of PCs as shown in Figure 10. Note that we will always choose the top PCs; for example, if the number of PCs is 10, we will use the top 10. Furthermore, the p2_evaluation_reduced has 1077 dimensions, so the maximum number of PCs is 1077. The plot should be read from right to left.

In a similar fashion to a WGSS vs number of clusters plot, the loss function of k-means decreases as the number of PCs used decreases. This is because as we project the data into lower and lower dimensions, the intra and inter-cluster variances will correspondingly decrease (assuming the data spans all the dimensions). Notably, this graph follows the same shape as the percent of explained variance shown in Figure 11. Perhaps intuitively, then, the inter-cluster variability will de-

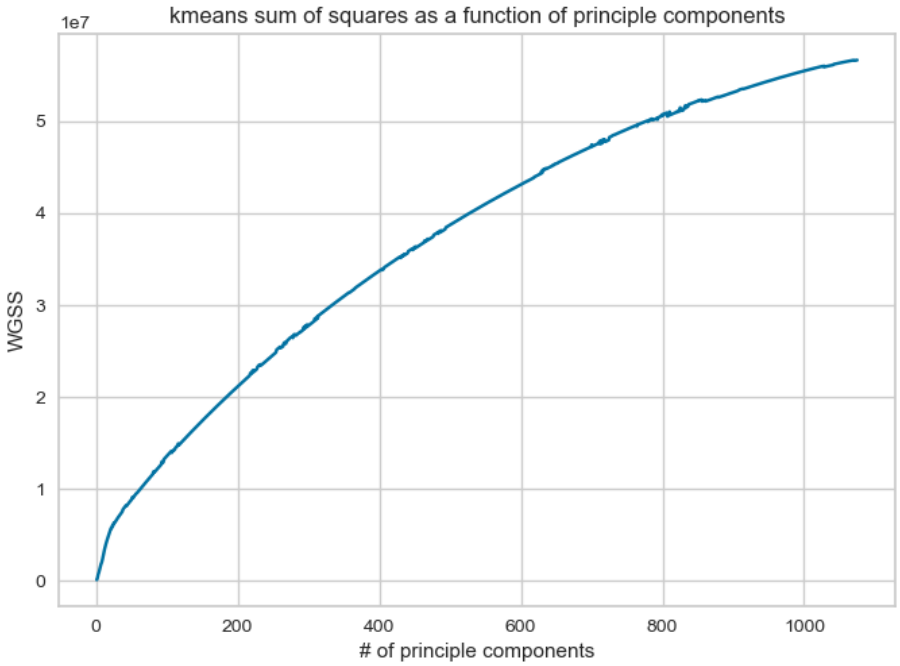

Figure 10: Effect of # of principle components on Within-Group Sum of Squares (WGSS) loss for k-means algorithms. Plot should be read from right to left.

pend on the overall variability.

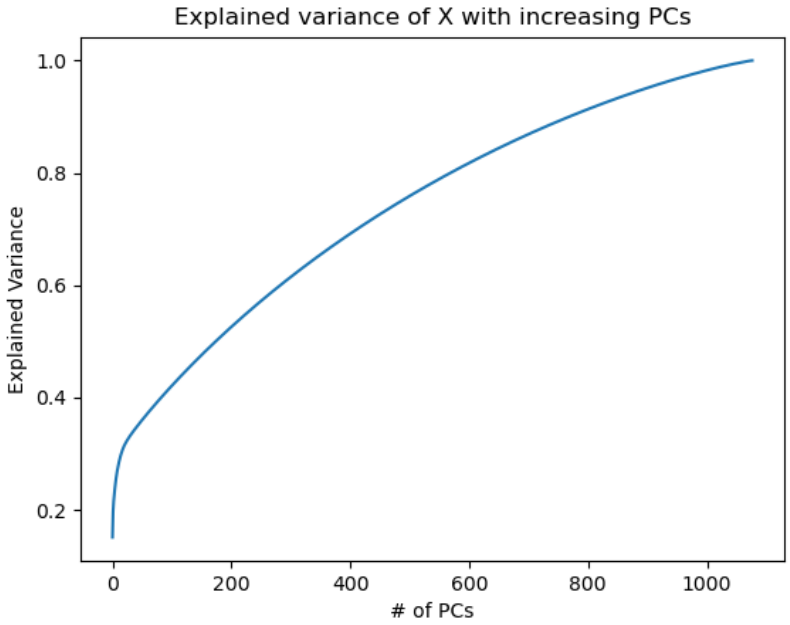

Figure 11: Percent of explained variance given the number of principle components used

Now we will run k-means on a select number of PCs and visualize the results in 2D.

Figure 12 above shows the results. Interestingly, everything above 2 PCs in the figure appears to show visually similar clusterings. Figure 13 is the same kind of plot as before but focused in on smaller PC counts.

To evaluate the impact of number of PCs on feature selection, I use the same method as in question 2.3 where I select the top 100 features corresponding to large magnitude coefficients. I do this for a range of principle components by training logistic regression models on the data in p2_unsupervised_reduced. I use $k = 9$ and train these initial logistic regression models using labels produced from a k-means solution on PCs that explain 85% of the variance. Finally, the top 100 features are selected for each of the corresponding PCs used to classify the k-means 85% clusterings. These features are then used to train several new logistic regression models and are subsequently evaluated using the data in p2_evaluation_reduced. The results are shown in 14

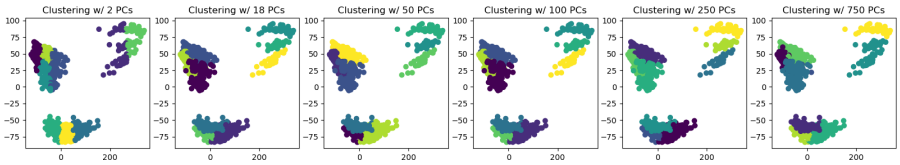

Figure 12: K-Mean clustering result with increasing number of PCs then projected onto the top two PCs

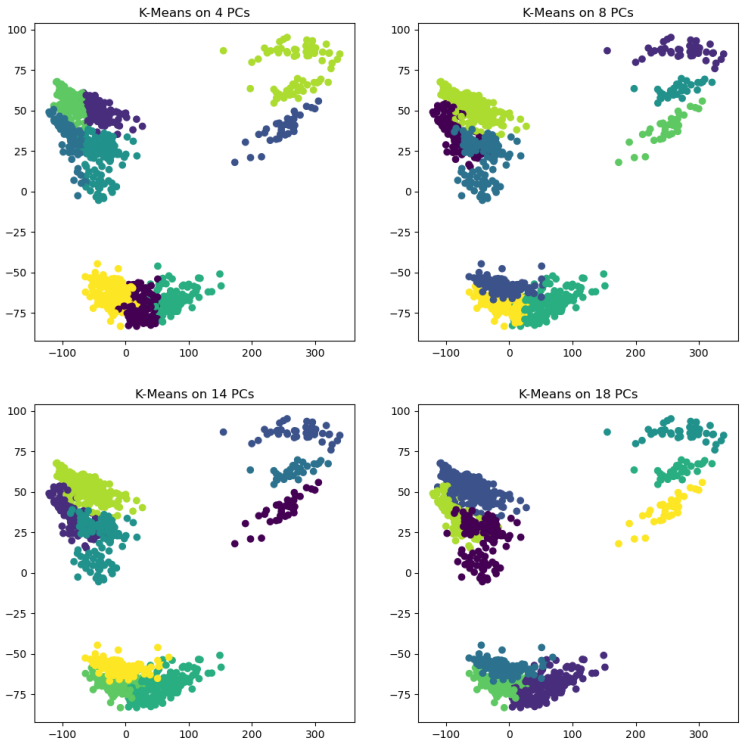

Figure 13: K-Mean clustering results with up to 18 PCs used. Results seem to suggest steady state clustering result after 18 PCs

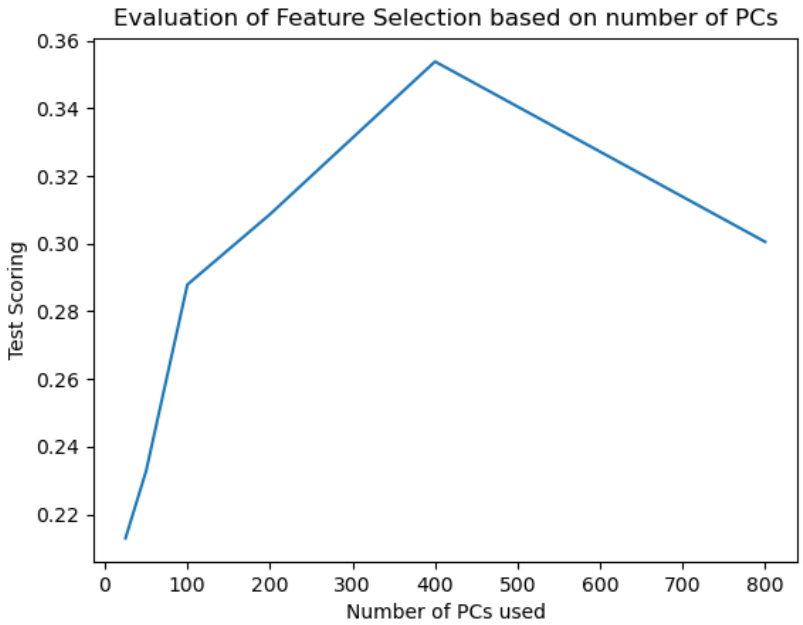

Figure 14: Evaluating top 100 features selected when using increasing numbers of principle components for classifying cluster assignments. Cluster assignments were determined using k-means on PCs explaining 85% of the variance. Higher PCs tend to produce better results

Knowing that using all 85% of explained variances PCs produces 0.35 test score from my analysis, I am inclined to think that the 800 PC score is an outlier. Thus, using more PCs for feature selection helps up to a point, then it helps no longer after this point. For this problem, this point is at most 400 PCs.

2. Type of regularization (L^1, L^2, elastic net) in the logistic regression step and how the resulting features selected differ.

For this analysis, we will again use $k = 9$ and use the top 400 principle components. We will use the 'saga' solver from scikit-learn's LogisticRegressionCV model. The penalty will be set to 'elasticnet,' and the l1_ratios will be [[0], [0.5], [1.0]], corresponding to fully l2, weighted l1 and l2, and fully l1 penalties, respectively (the center entry is the elasticnet portion, where the model will return the best mixing ratios for each of the $k = 9$ classes; in this case, it returns the only entry in the list). Figure 15 gives a visual representation of the selected features for each penalty scheme.

Interestingly, the selected features appear to be quite similar in that they have the same shape. Using these features to train another set of logistic regression models in p2_evaluation_reduced, the test scores are computed and given in Figure 16.

This second figure corroborates the suspicion that the selected features are similar since there is little difference between scores. Therefore, with this data, penalty schemes appear to play a small role in feature selection. More options could be explored, however, to ensure that this is indeed the case, such as truly allowing for varying l1_ratios instead of fixing them for each cluster to be the same.

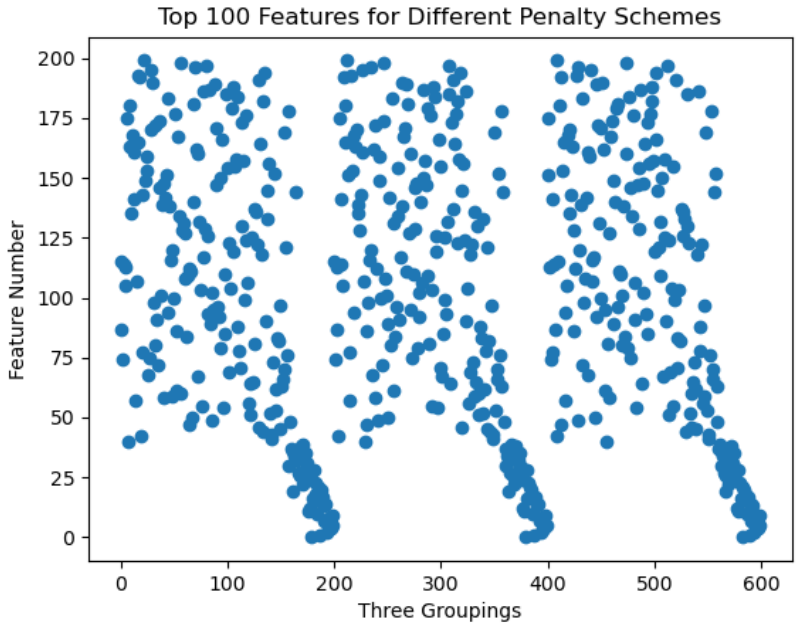

Figure 15: Visualization of features with largest coefficients for three penalty schemes. Features are represented by their index in the data. the l2 penalty corresponds to the first 200 x-ticks, elasticnet to the following 200, and l1 to the last 200.

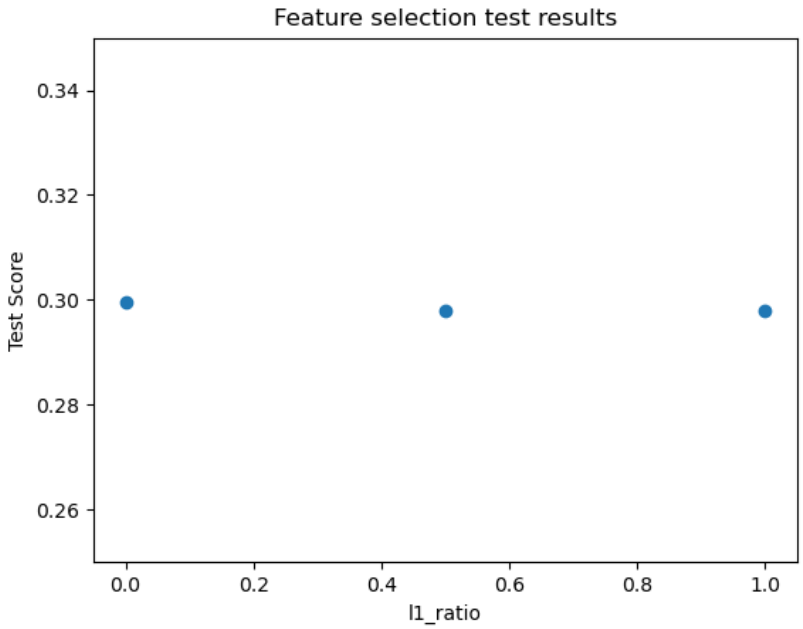

Figure 16: Test score using top 100 features for each penalty scheme.

References