Galactic explorations with deep learning and clustering

Michael Young, Fangfei Lan

problem/motivation

There’s a veritable deluge (think: petabytes) of astronomical data being generated from surveys like the Sloan Digital Sky Survey (SDSS). In fact, there is far more data out there than could be analyzed by expert astronomers within a reasonable time frame. In order to meet this daunting data challenge, researchers are pursuing many different strategies such as crowd-sourcing classification tasks, data mining and machine learning techniques aimed at classifying and discovering insights from survey data. This project is focused on the intersection of these methods: *we seek to explore (through dimensionality reduction and clustering) the features learned in a convolutional neural network (CNN) trained on crowd-source-labeled galaxy images.*

data explanation

The dataset we used came from a Galaxy Zoo kaggle competition launched in 2013. The dataset includes more than 130,000 images of galaxies. 61,578 of these images were labeled by the Galaxy Zoo participants. The ‘labels’ were the percentages of responses to 37 questions asked of each volunteer during the galaxy classification. In order to develop a succinct way for us to ‘label’ galaxies, we followed the questionnaire decision tree and labeled galaxies based on their highest percentage responses. This created a total of 8 different galaxy labels.

A random subset of 30,000 of these labeled images were used to train and validate our CNN. After our CNN was trained, we fed in a random subset of 20,000 images and extracted the last fully connected layer to produce a “CNN code” or feature vector corresponding to each image (https://cs231n.github.io/transfer-learning/). It was on this set of 20,000, 512-element feature vectors that we performed our dimensionality reduction and clustering.

key idea

As mentioned above, the key idea that this project is built on is to explore (through dimensionality reduction and clustering) the features learned in a convolutional neural network (CNN) trained on crowd-source-labeled galaxy images. We think this is an interesting goal to pursue for a few different reasons:

1) Machine learning interpretability: machine learning techniques (especially “deep learning” techniques) are increasingly the go-to methods for tackling a variety of problems across a startlingly wide range of disciplines. Deep learning is being used so frequently because, well, it works really well. The only catch is it’s sometimes difficult to know *why*. There’s a lot of research being done to help peer inside these deep learning models and get some sort of intuition about how they’re working and what they’re learning exactly. Our work here is a modest contribution to this effort.
2) Understanding dimensionality reduction and clustering:: Many “real world” datasets often involve objects with a large amount of features or dimensions - far too many to easily visualize and gain insight from. There are a wide variety of dimensionality reductions techniques - here we explore three: t-SNE, linear discriminant analysis (LDA), and principal component analysis (PCA). Each of these techniques come with trade-offs, especially when we pair them with clustering, and our work seeks to investigate these trade-offs.

A similar pipeline to ours was done in (https://arxiv.org/pdf/1812.02183.pdf), but their exploration did not go beyond a simple t-SNE plot paired with visual confirmation. No formal clustering methods were explored.

what we did

The first task was to train a CNN on our galaxy data ***(Done by Michael)***.

We used transfer learning with Pytorch to train our CNN. A variety of models and parameters were tested, but we found that resnet34 (pre trained on ImageNet) produced the best results. As mentioned above, we trained our net on 30,000 images. We did an 80/20 test/validation split, and used a batch size of 64 and ran the model for 8 epochs. With more computing power we could have undoubtedly trained a more sophisticated network, but decided what we achieved was good enough given that the focus of this project isn’t the deep learning part.

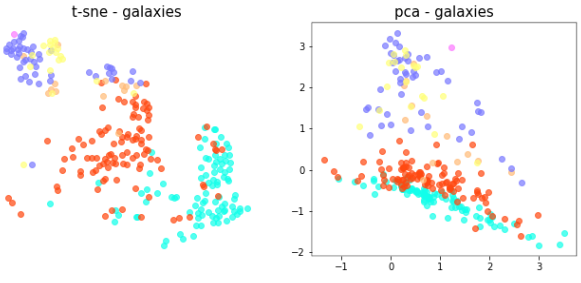


Fig 1: 500 galaxy feature vectors from Resnet34 randomly selected and plotted using t-SNE and PCA.

To determine what model worked ‘best’, we inspected the loss curves from our training and validation. Additionally, we visually inspected the t-SNE and PCA plots of features extracted from the model and chose the one that showed the most clear separation of different galaxy types. Figure 1 shows the plots we used to select resnet34 as our network. See Appendix A for details and charts showing the results of different models.

The second task was to extract the features from the network ***(Done by Michael)***.

Once we had a network which we felt had learned features that could distinguish between galaxies to a meaningful degree, we extracted the second to last layer of our network. For resnet, this layer's output is a 512-element vector of high level features learned by the network. As mentioned above, we did this by feeding 20,000 randomly selected images into the network and saved the output of this second to last layer for each of the images.

The third task was to perform dimensionality reduction and clustering on the features ***(Done by Michael)***.

The overview of our dimensionality reduction + clustering strategy was to pair 3 different types of dimensionality reduction: t-SNE, PCA, and LDA, with 2 different clustering methods: k-means, and DBSCAN.

t-SNE

t-SNE is a non-linear dimensionality reduction technique that seeks to preserve the local structure of the data. The main parameters to keep in mind when using t-SNE are perplexity and learning rate. Through trial and error we found that a perplexity of 200 and a learning rate of 50 worked well. The result is plotted to the right. t-SNE did a fairly good job at separating out the features into their distinct categories. The 'round' galaxies (pinks and purples) are all near each other toward the top. The disk-edge galaxies (yellow, green and light orange) seem to gather around the left of the plot. The spiral disk galaxies gather largely near the bottom, while the non-spiral disks are distributed near the bottom but also near the top, suggesting that the network may have had a difficult time learning the features of these.

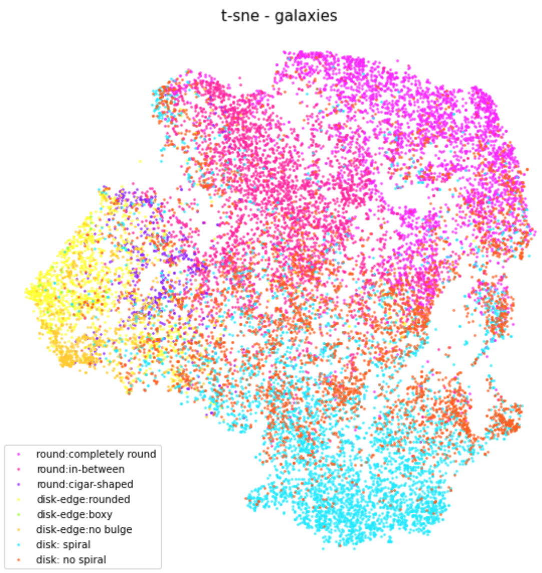


Fig 2: t-SNE on 20,000 feature vectors. Perplexity: 200, learning rate: 50.

We ran an elbow plot (Fig 3) to determine what a good choice for k clusters would be when using k-Means on t-SNE.

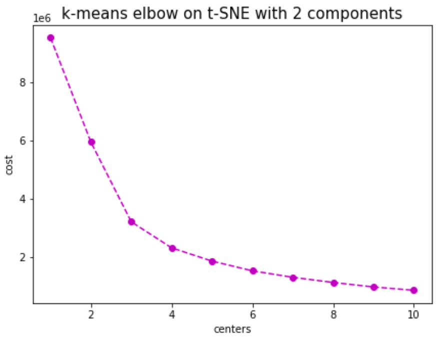


Fig 3: Elbow plot for K-means clustering on t-SNE. We chose 4, 5, and 6 as the most interesting k's to expore.

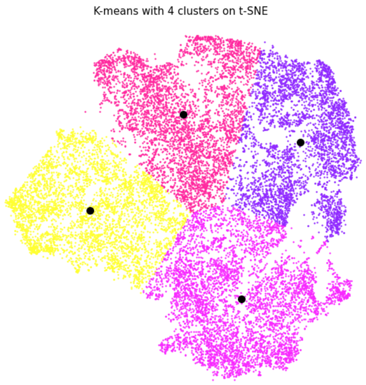


Fig 4: K-means clustering (k=4) on t-SNE.

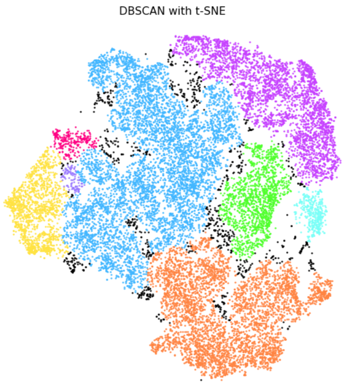


Fig 5: DBSCAN run on t-SNE. Eps: 2.0717, Min_samples: 100. Resulted in 10 clusters (9 true clusters and 1 representing all the "noisy" samples).

As seen in Figure 4, simply using k-means on t-SNE only works as well as t-SNE does at separating data.

Next, we ran DBSCAN on the same t-SNE data. A potential advantage of using DBSCAN over k-means for clustering is that it doesn't require a predetermined number of clusters to run. The main parameters of interest when using DBSCAN are the 'eps' parameter, which is the maximum distance between two points for them to be considered 'neighbors.' The other key parameter is min_samples, which is the number of points in a neighborhood for a point to be considered a core point. We experimented with many different values to obtain a clustering that looked.... well, nice. By nice we mean there aren't a gazillion clusters and there isn't just one - we seeked a happy medium. The results are shown in Fig 5.

As seen, this method potentially finds more meaningful clusters than k-means when run on raw t-SNE data, however the success or failure of this still depends heavily on t-SNE's ability to separate different neighbors from one another.

PCA

PCA has been a standard technique for dimensionality reduction for many years. It finds the principal directions in the data through singular value decomposition of the data matrix (or alternatively through eigen decomposition of the covariance matrix). An advantage of PCA over t-SNE is that we can reduce our data to more than 2-3 dimensions, which could potentially generate a more sophisticated clustering.

As shown in Fig 6, PCA doesn't do too well at separating galaxy morphologies compared with t-SNE (Fig 2). Through trial and error (and to compare with LDA later), we decided to cluster on our PCA using 7 components. As shown in Fig 7, this resulted in a fairly sophisticated clustering of the data, at least when compared with clustering over the 2 dimensions of t-SNE (we chose to present PCA clustering data on top of t-SNE because the t-SNE plot had less data overlap than the PCA plot). We once again created a k-means elbow plot.

We had less success with using DBSCAN with the 7 components of PCA. It was extremely difficult to find a value for epsilon that adequately covered the data and created smooth clusterings. This is likely a result of there being too many dimensions.

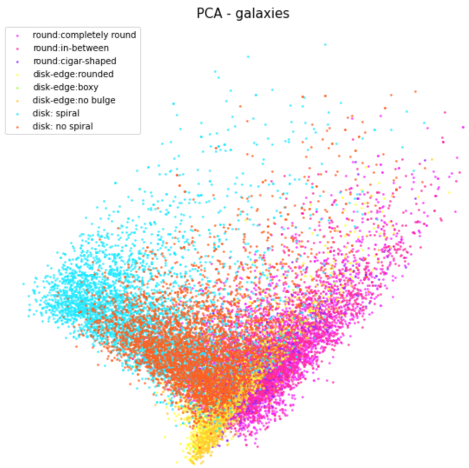


Fig 6: First two principal components of galaxy data. Can clearly see separation of galaxy morphologies here, although there is significant overlap in the center of the plot.

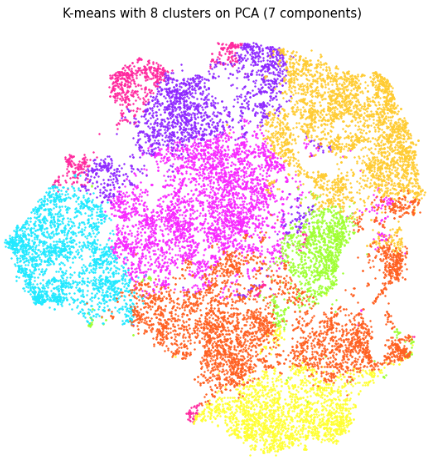


Fig 7: K-means clustering (k=8) on 7 principal components shown on a PCA plot of the galaxies. Notice the fairly sophisticated partitioning of the various galaxy morphologies.

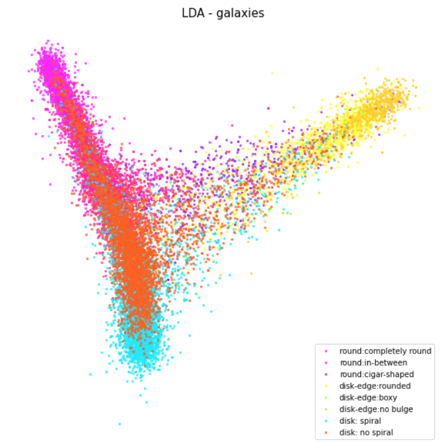


Fig 9: LDA projection of galaxy data. Notice the interesting, almost orthogonal relationship between the disk-edge galaxies and all other galaxies.

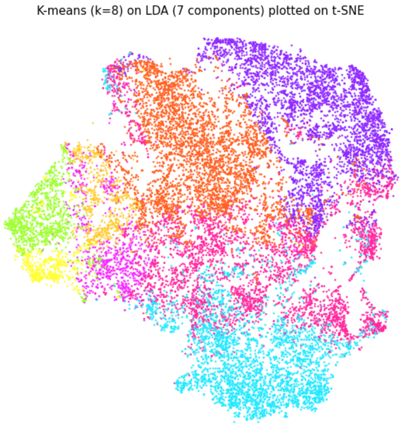


Fig 10: K-means (k=8) clustering on 7 components of LDA. Visually comparing this plot with Fig 7 and 2, we see that LDA performs the most sophisticated clustering with respect to galaxy morphology.

LDA

LDA is different from t-SNE and PCA in that it requires some a-priori knowledge of the data. It seeks to maximize the separation between known categories in the data. It is often used as a classifier in addition to a dimensionality reduction technique. As seen in Fig 9, it does a very good job at reducing the data. The disk-edge galaxies are neatly gathered to the upper right of the plot, with the round galaxies mostly gathered in the upper left. It's interesting to note that many of the round-cigar shaped galaxies span the area in between the round and disk-edge galaxies, which may make sense in that the round cigar-shaped galaxies may share many things in common visually with the disk-edge galaxies. Also, similar to t-SNE, LDA had difficulty separating the no-spiral disk galaxies from the rest of the data, which again, may make sense in that a disk galaxy with no spiral may look much like a round galaxy.

Like PCA, LDA allows one access to more dimensions than just 2-3, so we decided to cluster using the 7 dimensions that LDA provided. The rationale here was to preserve as much of the information in the data as possible for our clustering tasks. We once again ran an elbow plot for k-means to determine optimal k's and ran DBSCAN. We plotted the clustering results on top of t-SNE in addition to LDA so as to provide a point of comparison with t-SNE and PCA.

Comparing the LDA clustering results with the positions of galaxies in the original t-SNE plot shows the LDA performs by far the most sophisticated clustering. This is to be expected however, as we reduced the dimensions based on information known about the data beforehand, which may not always be known in an exploratory setting. Also it’s worth noting that DBSCAN once again was very difficult to optimize here, again likely due to the higher dimension of 7.

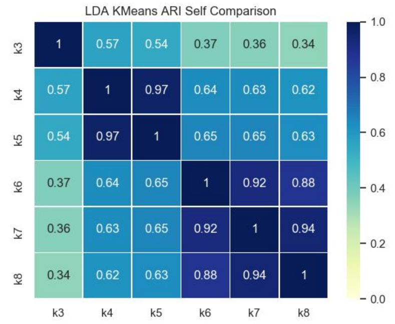
For more figures describing our results from this section, see Appendix B.

The fourth and final task was to explore the relationships between our clustering methods ***(Done by Fei)***.

There are many ways to compare clusterings and most of these methods can be classified into two categories - one is based on counting pairs of overlaps and the other is based on mutual information. We chose one from each category, Adjusted Rand Index from the counting pairs and Adjusted Mutual Information from the mutual information category. Neither methods require the same number of clusters in the clusterings.

The Adjusted Rand Index (ARI) method is an extension of the General Rand Index method which counts the number of pairs of elements that are classified into corresponding clusters. Since the expected value of the General Rand Index varies, the ARI takes the normalized difference between the Rand Index and its expected value. This results in an expected value of 0 for random clusterings and a maximum of 1 for perfect matches.

The Adjusted Mutual Information (AMI) method is based on the notion of entropy. It is also normalized based on the expected value of Mutual Information. AMI also has an expected value of 0 and 1 indicating a perfect match. Both methods can sometimes take negative values but it is difficult to interpret these values based on their magnitudes, since they are “worse” than randomly assigned clusterings. One advantage that AMI has over ARI is that ARI only takes into account the overlapping information between clusterings, but AMI considers the patterns in the entire dataset, within overlaps and outside. Below are results with the astronomy data.



	k3	k4	k5	k6	k7	k8
k3	1	0.57	0.54	0.37	0.36	0.34
k4	0.57	1	0.97	0.64	0.63	0.62
k5	0.54	0.97	1	0.65	0.65	0.63
k6	0.37	0.64	0.65	1	0.92	0.88
k7	0.36	0.63	0.65	0.92	1	0.94
k8	0.34	0.62	0.63	0.88	0.94	1

Fig 11: LDA and K-means with comparing method ARI. The matrix shows that using LDA, the clusterings with 4 and 5 clusters are very similar, so are clusterings with 6, 7 and 8 clusters.

First, we compared the k-means clusterings with different k within each dimensionality reduction method. We expect the resulting matrices to be symmetrical with 1 on the diagonal. Fig 11 is a plot of the LDA with K-Means clustering with ARI comparison method.

We plotted the K-means self comparison matrix for each dimensionality reduction technique (see Fig 12 and 13 in Appendix C) and there is a similar trend - clusterings with similar numbers of clusters are more similar than others. For example, clusterings with 7 and 8 clusters are similar in all three cases. In addition, on average LDA results in much more similar clusterings than PCA and t-SNE. Comparing Fig 14 in the Appendix with Figure 11, LDA and K-means with ARI and AMI methods, we see almost identical color patterns. This demonstrates that two comparison methods reached the same conclusion.

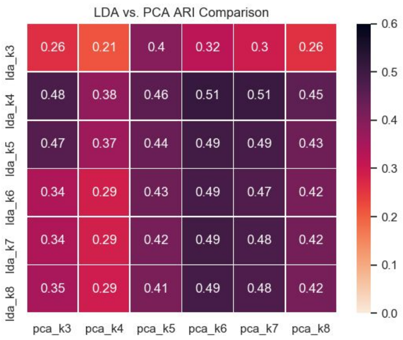


Fig 15: LDA vs. PCA compared with ARI.

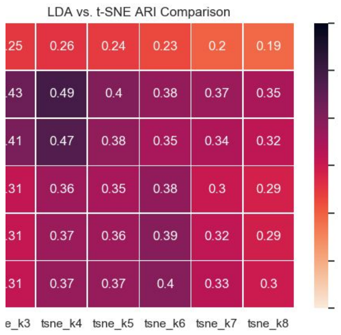


Fig 17: LDA vs. t-SNE compared with ARI.

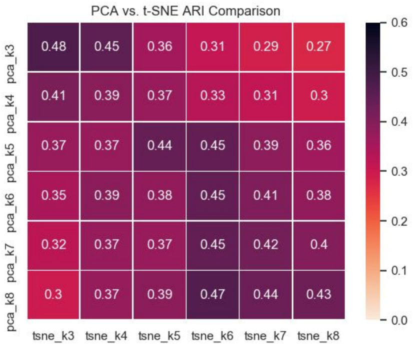


Fig 19: PCA vs. t-SNE compared with ARI.

We then performed comparison analysis across different dimensionality reduction techniques. Fig 15 shows that LDA and PCA results are most similar when LDA is clustered into 4 clusters and PCA into 6 and 7 clusters, but PCA with 6 and 7 clusters seem to be similar to most LDA clusters regardless of the number of clusters.

Figures 17 below and 18 in the Appendix show the comparison between LDA and t-SNE. It is interesting to see that in the ARI plot, LDA with 4 and 5 clusters are most similar to t-SNE with 4 clusters. The AMI plot shows some additional results, with LDA with 6, 7, 8 clusters being similar to t-SNE 6, 7 and 8 clusters. The comparisons between PCA and t-SNE are shown in Fig 19 above and 20 in the Appendix. The ARI and AMI matrix both indicate that PCA with 6, 7, 8 clusters are most similar to t-SNE with 6, 7, and 8 clusters. It is also worth noting that PCA with 3 clusters and t-SNE with 3 clusters scored high in the matrix. Looking at all 3 sets of plots, we see that the LDA and PCA had the most similar clustering results, and then it was PCA and t-SNE. With all 3 pairs, the clusterings are similar with larger numbers of clusters. In addition, AMI comparison method generally results in higher similarity scores but the trend is similar to the ARI comparison method.

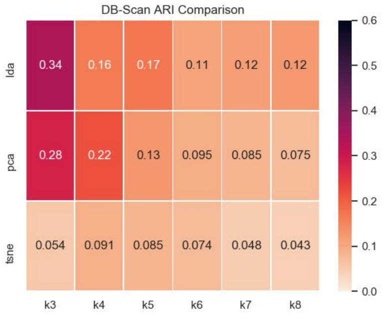


Fig 21: DBSCAN comparison with ARI.

Now let us look at the DBSCAN comparison results. Since DBSCAN did not generate meaningful results for PCA and LDA, we are only doing comparison between the DBSCAN result from t-SNE and the k-means clusterings from all three dimensionality reduction algorithms.

In both Fig 21 and 22 we see that DBSCAN scored low on all three dimensionality reduction algorithms with all numbers of clusters. The 10 clusters generated from DBSCAN are very far from the clusters generated by k-Means for all three algorithms.

what we learned

What we learned about our CNN and the features it learned: As described in detail above, by examining the features learned by our CNN through various dimensionality reduction and clustering techniques, we were able to confirm that the features learned were themselves useful for drawing morphological distinctions between galaxies. We saw that the features did a reasonable job at recognizing galaxies that held a similar overall shape (round vs. disk-edge vs. disk). We were also able to draw some potential conclusions about which galaxy morphologies our features had difficulty with. Our findings suggest that this strategy of dimensionality reduction paired with feature extraction may provide a viable addition to the quiver of machine learning interpretability tools currently in use.

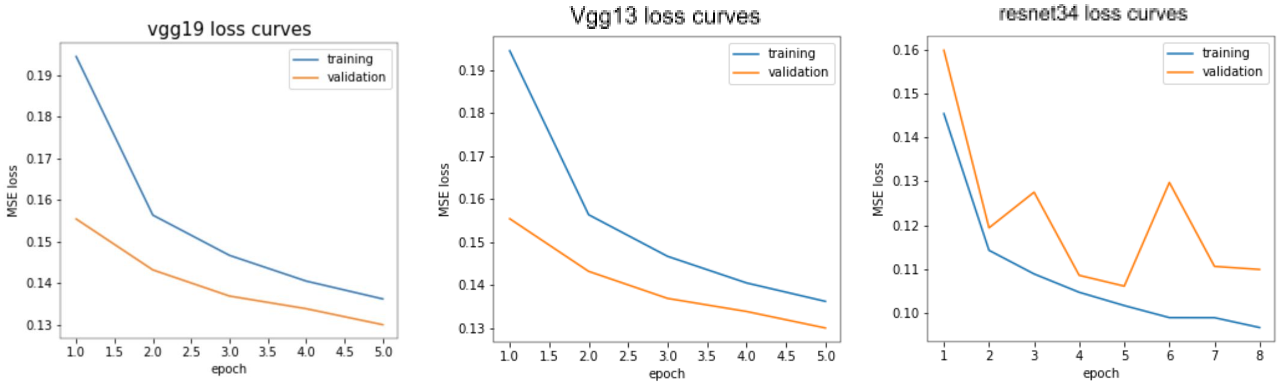
What we learned about Dimensionality Reduction:

In this particular dataset, we saw that LDA and PCA generated similar K-Means clustering results. Since we reduced to 7 dimensions for both algorithms, this means that LDA and PCA potentially generate similar dimensionality reduction results for similar dimension parameters.

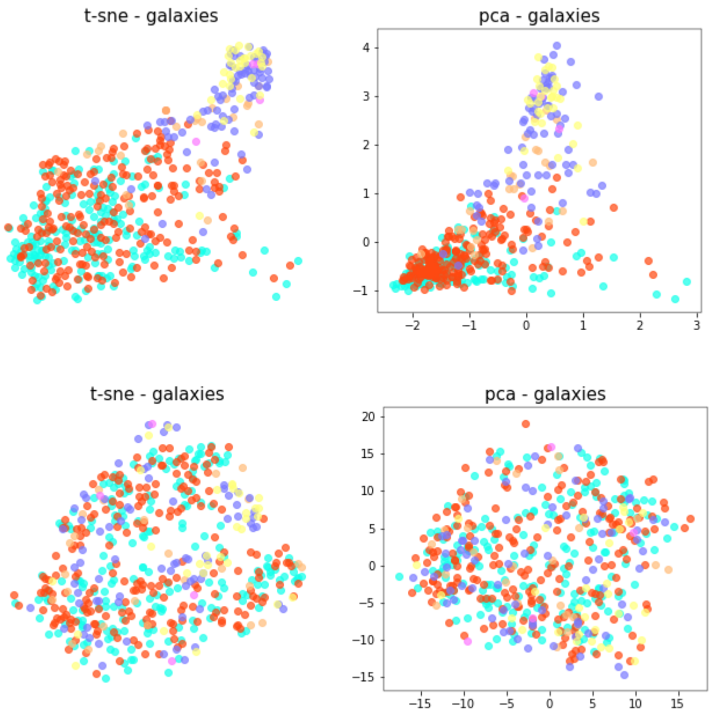
What we learned about Clustering:

We have learned, not so surprisingly, that DBSCAN does not work well with high dimensional data. We suspect that the reason could be that the density projected onto 2D could be drastically different density in higher dimension. Two points that seem close to each other in 2D could actually be far away. We also learned that comparing multiple clusterings is not a trivial task. There are many ways to approach this problem and they all have slightly different objectives.

Appendix A: CNN data

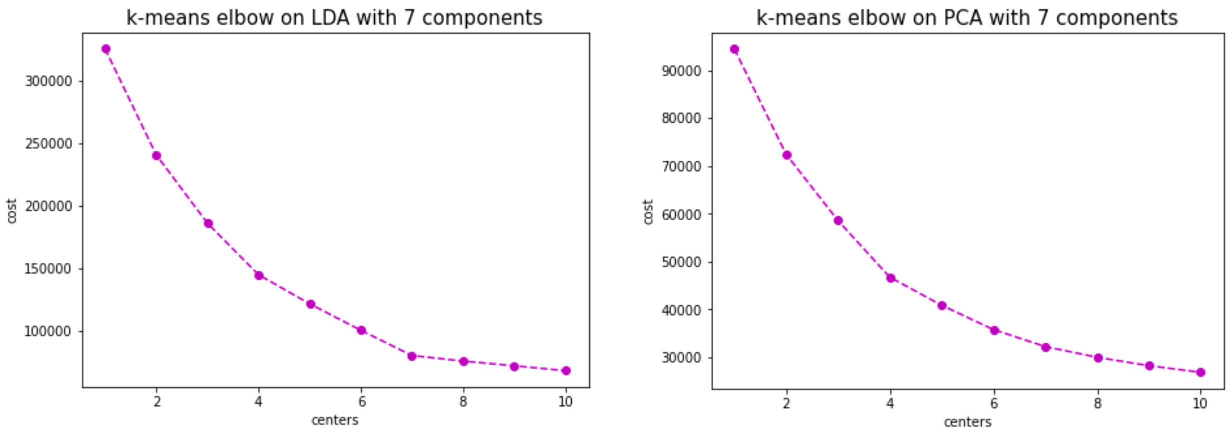


Loss curves for 3 different networks tested. While the validation performance of resnet34 was noisy, the training error achieved the lowest MSE loss.

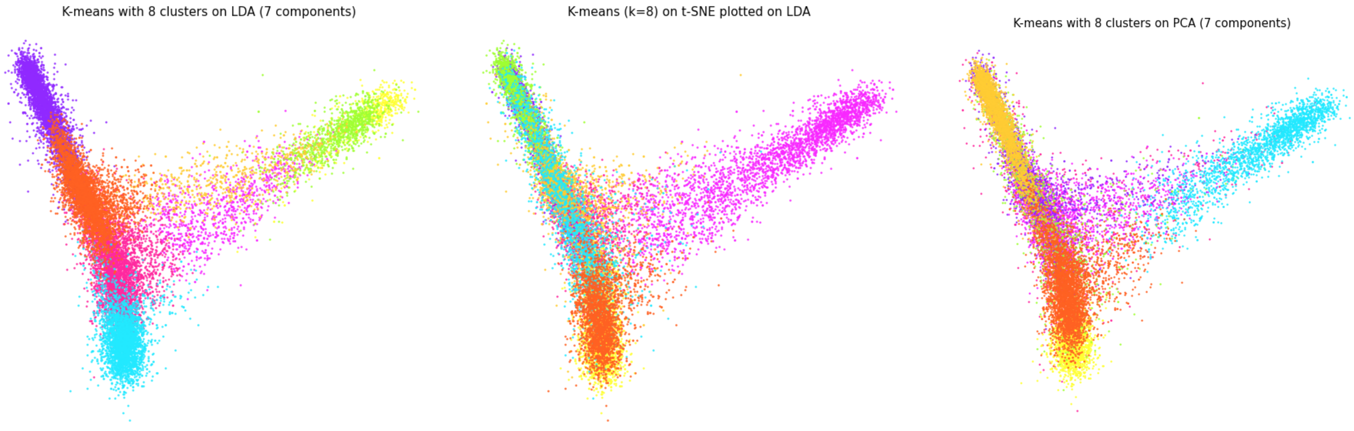


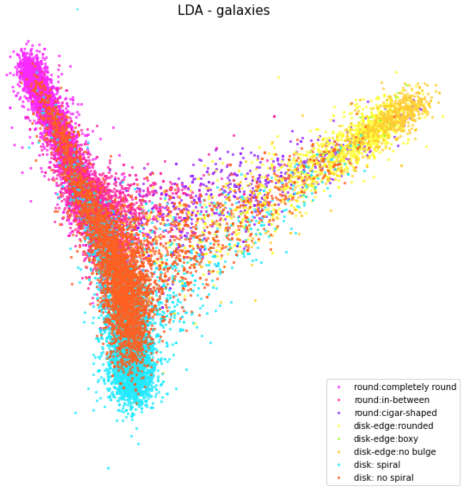
Vgg13 (top) and Vgg19 (bottom) tsne and PCA plots used to assess to what degree each network was learning "interesting" features.

Appendix B: Dimensionality reduction and Clustering

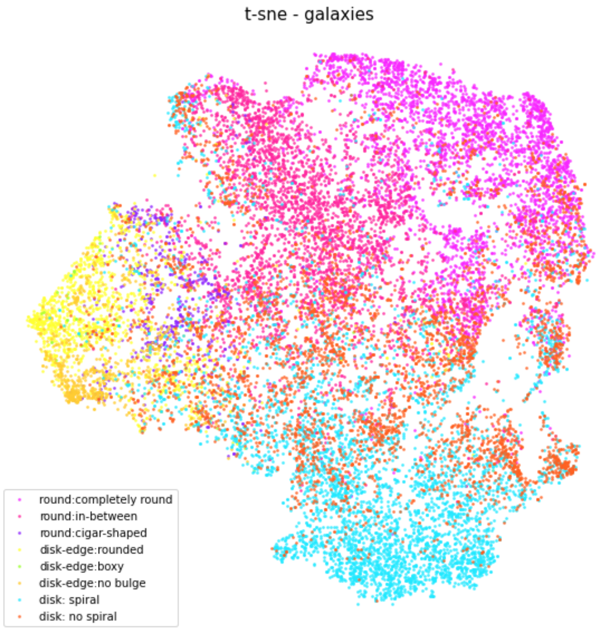


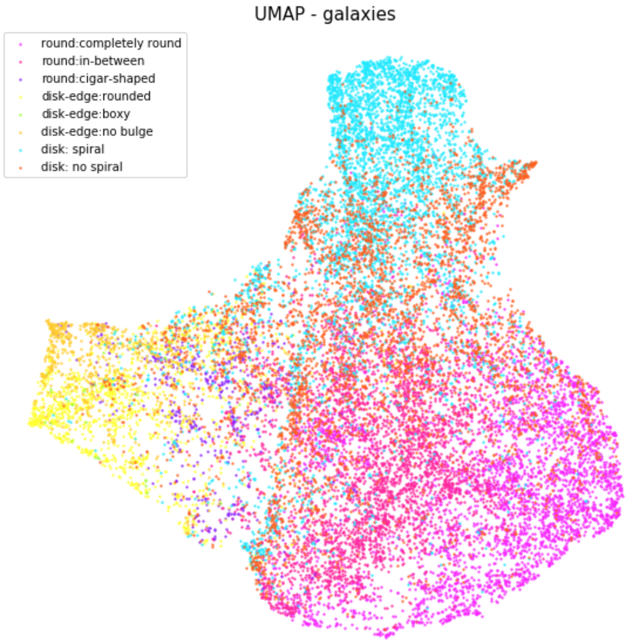
Elbow plots for LDA and PCA.





K-means clustering (k=8) for each dimensionality reduction techniques plotted on 2 dimensions of LDA. The LDA w/ labeled galaxies is presented in the bottom left as a point of visual comparison for how similarly the clustering algorithms clustered the galaxies with respect to the LDA's distribution of the galaxies.





UMAP is often touted as being 'better' than t-SNE in that there's more of a theoretical foundation to it. We considered including UMAP in our analysis, but left it out for time's sake. Seen here, it results in a galaxy distribution very similar to t-SNE. One advantage of UMAP is that we could use more than just 2-3 components. Clustering on more dimensions of UMAP may yield interesting results in a future consideration of this data.

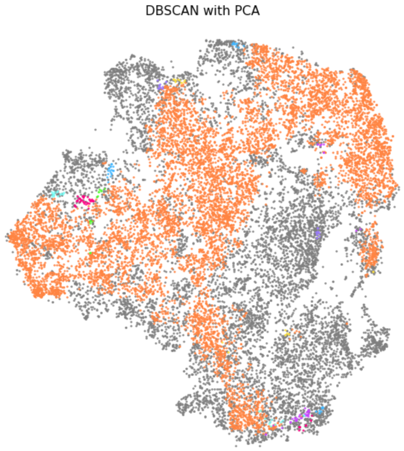


Fig 8: DBSCAN run on 7 principal components. Eps: 0.35, Min_samples = 12. This resulted in 18 clusters + 1 cluster composed of "noisy" data.

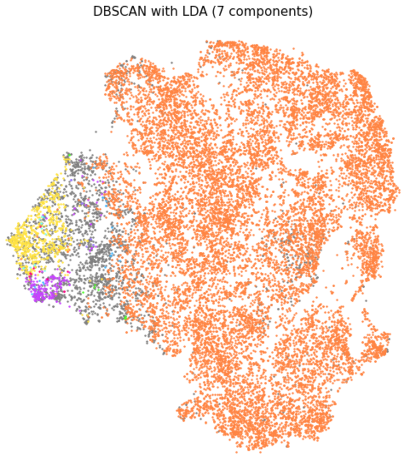


Fig 11: DBSCAN on 7 components of LDA. Eps: 1, min_samples: 10. Resulted in 12 clusters, and 1 cluster composed of "noisy" data.

Appendix C: Clustering Comparisons

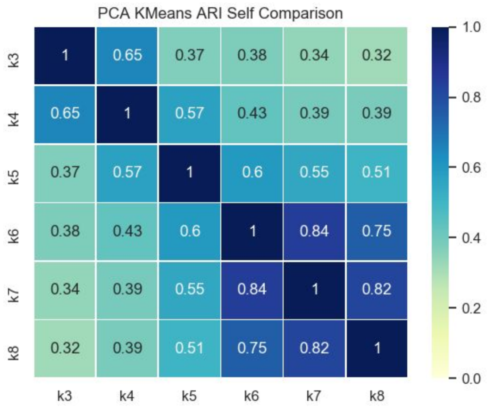


Fig 12: PCA and K-means with comparing method ARI. The matrix shows that using PCA, the clusterings are fairly similar with 6, 7 and 8 clusters.

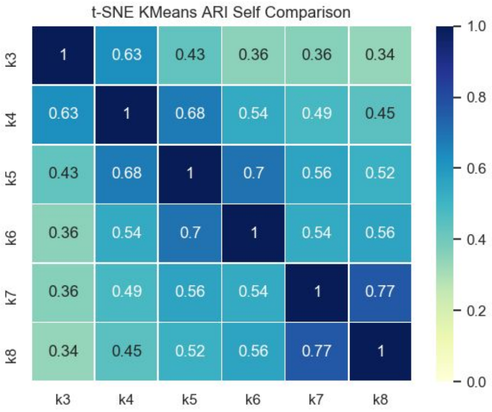


Fig 13: t-SNE and K-means with comparing method ARI. The matrix shows that using t-SNE, the clusterings tend to have a higher score with similar numbers of clusters.

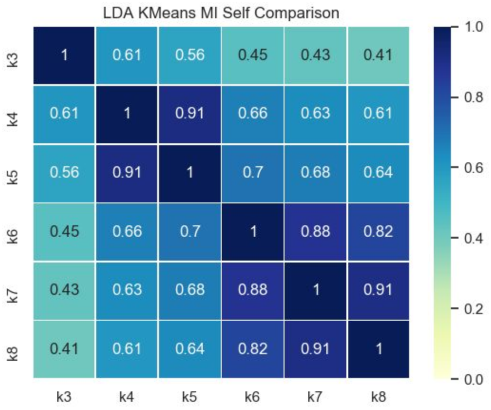


Fig 14: LDA and K-means with comparing method AMI. The matrix shows similar result to LDA and K-means with ARI (Fig 11).

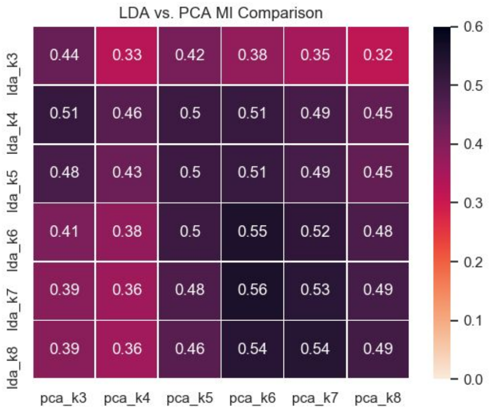


Fig 16: LDA vs. PCA compared with AMI.

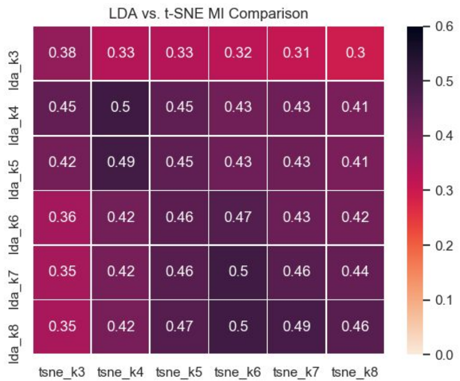


Fig 18: LDA vs. t-SNE compared with AMI.

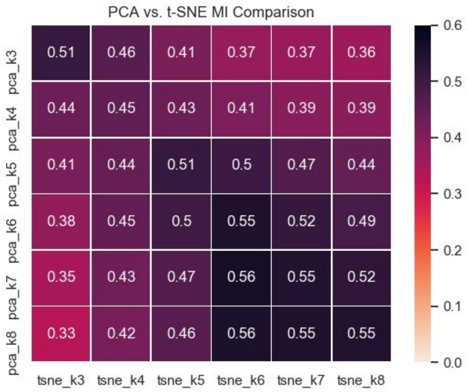


Fig 20: PCA vs. t-SNE compared with AMI.

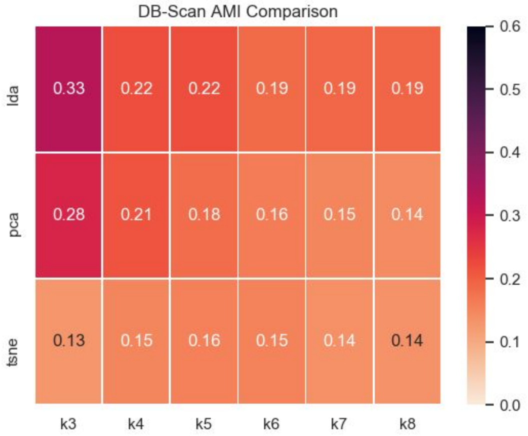


Fig 22: DBSCAN comparison with AMI.