Unsupervised Learning and Dimensionality Reduction

Branden Huggins
CS-7641

A discussion of your datasets, and why they're interesting

The first problem is predicting whether or not a LendingClub loan will be fully paid based on the information known at the selection of LendingClub loans at the time of selection. The data contained is 4 years of LendingClub data ranging from 2007 to 2011. The data mostly describes the lender’s financial status, the loan specifications and the loan status. The data will be classified on whether or not the loan status is “Charged Off” or “Fully Paid”. From the raw LendingClub data, I picked the data to get from the file based on the data available when picking LendingClub investments individually. For events that never occurred in the data set I assigned the highest possible integer value, for values such as inquiries in the last 6 months.

When loaning to strangers though the LendingClub website, the loaner would like all of the loans to be fully paid with interest. If a loan is labeled “Charged Off” then the loan will not likely be paid off. For lenders, one objective of loaning money is to lend money at the highest interest rate possible without the loan being charged off. The data used as features are the same data listed when browsing for loans on the lending club website. The information learned from this data is important because a lender can use the data to attempt to maximize their loan interest rate while minimizing the chances the loan will be charged off. A potential lender could take each of the currently listed lending notes and use the machine learning models to determine whether or not a loan is likely to be charged off. The highest interest rates loans can be fed into the model until the loaner finds the desired amount of high interest rate loans to lend to debtors who will likely repay the loan.

The second problem is divorce. Divorce seems like a major problem. It’s prevalence has created a business industry built on divorce. So being able to identify people with characteristics that make them susceptible to divorce is useful to maintaining financial health. This is because marriages and divorces both are expensive.

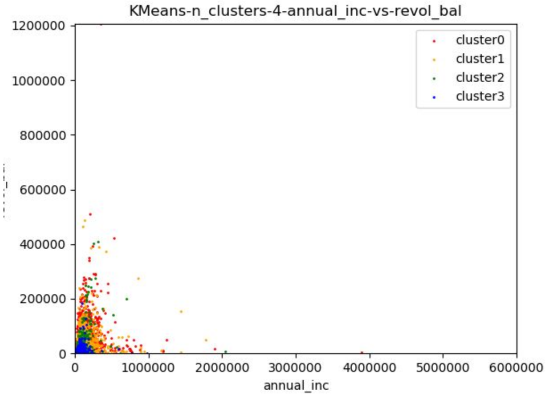
Explanations of your methods: for example, how did you choose *k*?

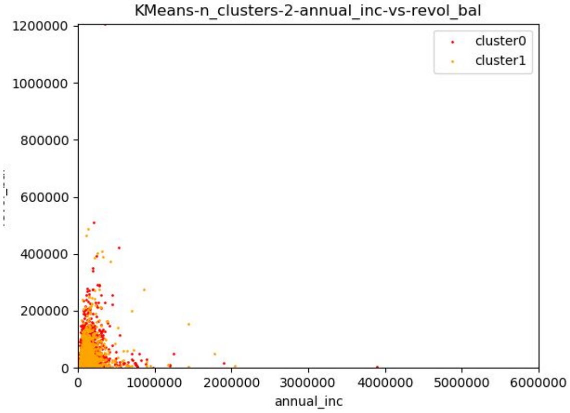
The best value of k for the KMeans algorithm will be based on the model inertia algorithm. The k with the highest inertia will be chosen as the best. The best values for the hyperparameter of the ICA algorithms will be chosen using the sum of the mixing numbers. The best values for the hyperparameter of the PCA algorithms will be chosen using the sum of the explained variance. At a sum of explained variance of 1, the tie breaker will go to the lower number. All the hyperparameters that didn’t have a specific output attribute to grade the performance are relegated to using shortest time as the parameter for choosing the best hyperparameter value. The hyperparameters for all of the algorithms will be stored until enough runs for the hyperparameters to converge at the best values for that algorithm. The following hyperparameters will be adjusted for the algorithms:

K-means: number of clusters
EM: number of clusters and reg covariance
PCA: number of components, whiten, and svd solver
ICA: number of components, algorithm, and max iterations
RCA: number of components
Feature Agglomeration: number of clusters

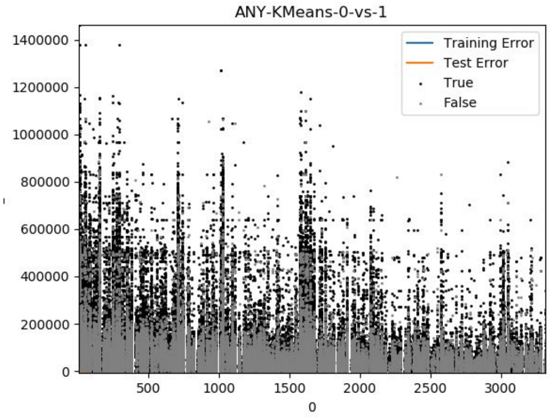
The reg covar was adjusted originally to see if it would allow the Gaussian Mixture model to run on the LendingClub data. Adjusting the reg covar did not allow the Gaussian Mixture model to run on the 24 feature sets of the LendingClub data. Reducing the feature set with reduction techniques allowed for the EM algorithm to function properly.

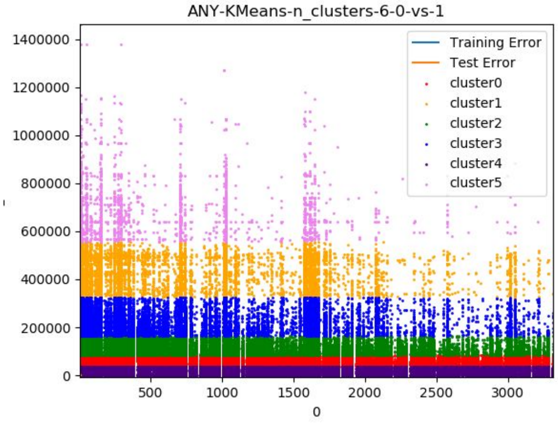
A description of the kind of clusters that you got.
For the Lending Club data most of the clusters were based on annual income. At 6 clusters of K means clustering, the clusters start to not produce as many recognizable patterns. The biggest factor of the clusters is annual income. This may mean that annual income is a predictor of the other factors recorded. A better separation is shown below with two clusters



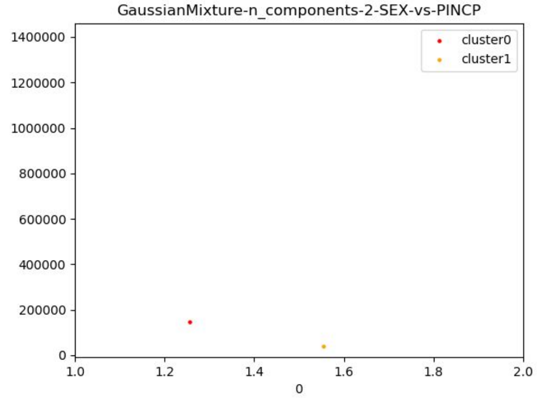


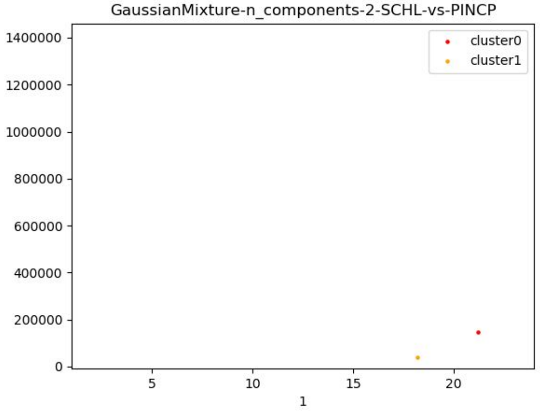
The graph above suggests that making a very low annual income tends to cap a person's credit utilization. This can be seen because the revolving balance stays lower for low income individuals and spikes higher for middle income individuals. But, someone who just makes a low or lower middle annual income may it most likely to utilize all the credit utilization they can get. People with high annual income don't feel the need to have as much credit utilization as people with low annual income and tend to keep a lower revolving balance. The EM algorithm didn't work on the initial LendingClub data. The Lending Club data needed to be reduced to a smaller feature set before the EM algorithm would work.





The two graphs above show how the annual income is clustered in the LendingClub data. As more clusters are added to the KMeans algorithm, the cluster continue to draw straight lines in annual income. The graph to the left shows whether or not the load was fully paid in black and shows a trend of higher income individuals paying back the loaned amount.





I think the two Gaussian dot cluster graphs above for the divorce data suggests that higher income and more highly educated people are more likely to stay married. The graph to the left suggests that higher earning men are the most likely to stay married. Both sexes are likely to get divorced at a low income, with low income females being the most likely to get divorced. This is shown by having cluster1 be in the center of 1 and 2 on the x-axis concluding that cluster 2 includes both men and women. Were as cluster1 mostly includes men, but this could also be showing that men make more money than women.

Analyses of your results. Why did you get the clusters you did? Do they make "sense"?

All the clusters usually were based on income for both models. The models were of the real world data. Income level probably has a high impact on decision making. Both data sets reflect decision making aspects of life. Especially, the LendingClub data which looks at long term decision making through credit score, annual income, general utilization of credit. The divorce data looks at decision making regarding occupation, education level, and personal income.

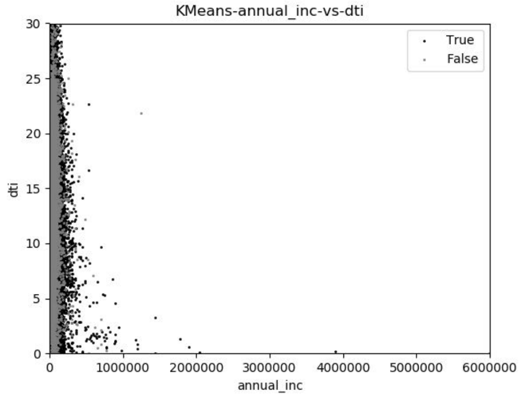
The LendingClub data also predictably reduced feature sets quickly because all of the feature set information is figured into the FICO score which is one of the features. The reason the components of ICA are a low value of two despite having 24 features for the LendingClub data is because the FICO score feature of the data is already accounts for all of the other feature sets.

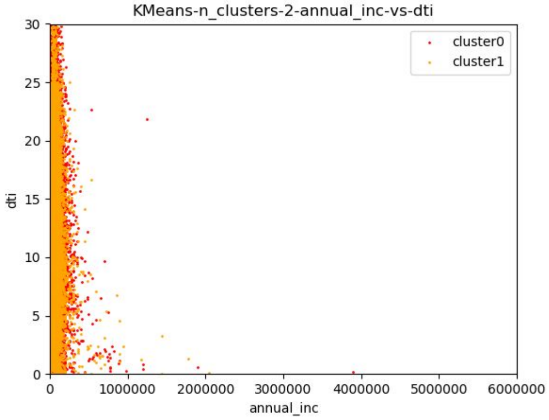
The KMeans algorithm always used a high amount of clusters because I set the best value to be the one that maximizes the sum of the explained variances. I did put a check that would say if less components explained all the variance that I would use the lower one instead. The lower amount of components never explained all of the variance.

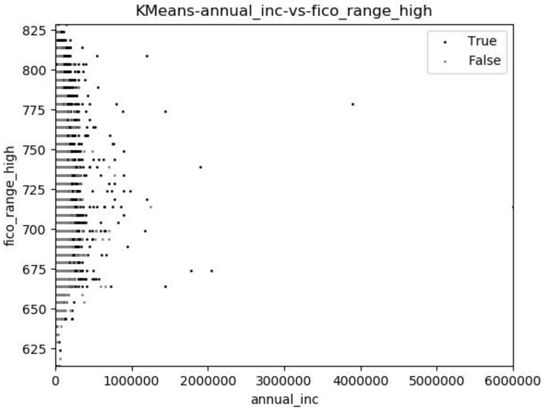
The clusters based on income do make sense. The other clusters not based on income or education level can be explained by the unquantifiable labeling of occupation in the divorce data. Visually the amount of features used in the LendingClub data set made some of the observations less explainable.

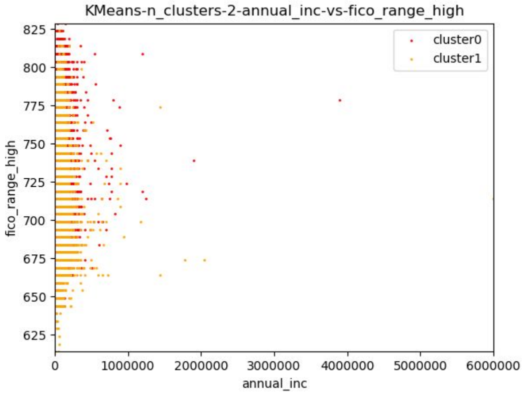
If you used data that already had labels did the clusters line up with the labels?

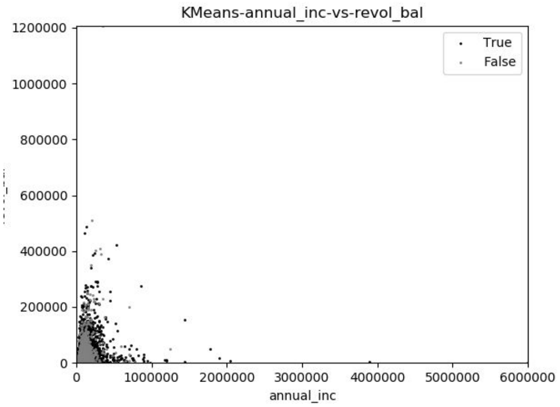
The lending club data lined up very well for the KMeans clustering algorithm as shown below. There are four examples where clustering the KMeans algorithm with two clusters almost exactly matches the classification for the training set data. Even the noisy data farther away from the group seem to match the actual and the unsupervised classification.

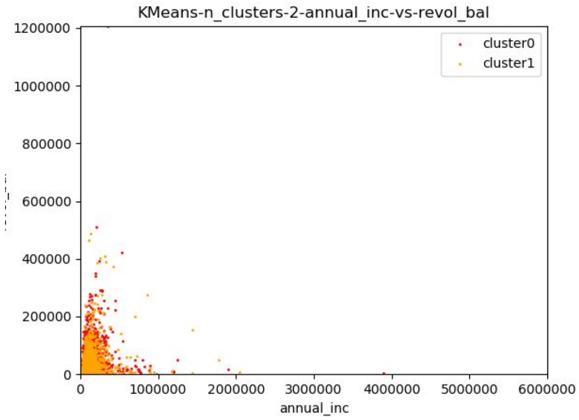

KMeans-annual_inc-vs-dti
True
False
dti
annual_inc
0
5
10
15
20
25
30
0
1000000
2000000
3000000
4000000
5000000
6000000

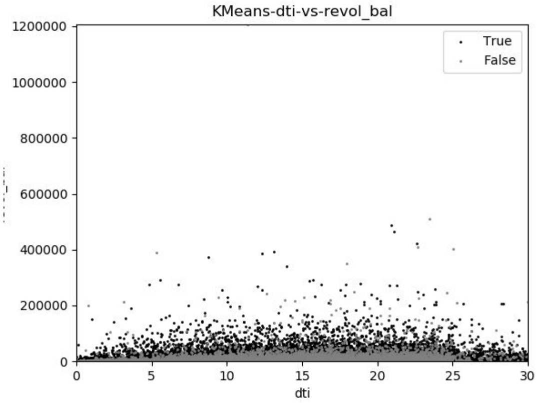

KMeans-n_clusters-2-annual_inc-vs-dti
cluster0
cluster1
dti
annual_inc
0
5
10
15
20
25
30
0
1000000
2000000
3000000
4000000
5000000
6000000

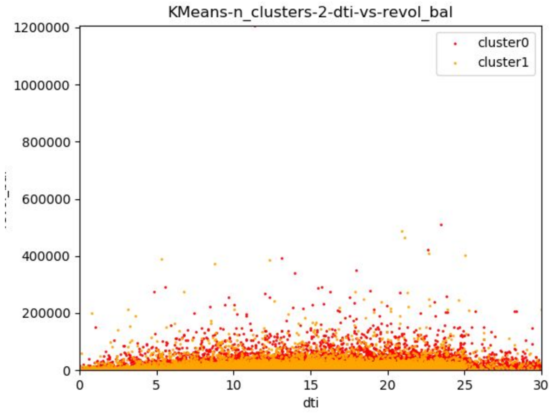

KMeans-annual_inc-vs-fico_range_high
True
False
fico_range_high
annual_inc
625
650
675
700
725
750
775
800
825
0
1000000
2000000
3000000
4000000
5000000
6000000


KMeans-n_clusters-2-annual_inc-vs-fico_range_high
cluster0
cluster1
fico_range_high
annual_inc
625
650
675
700
725
750
775
800
825
0
1000000
2000000
3000000
4000000
5000000
6000000

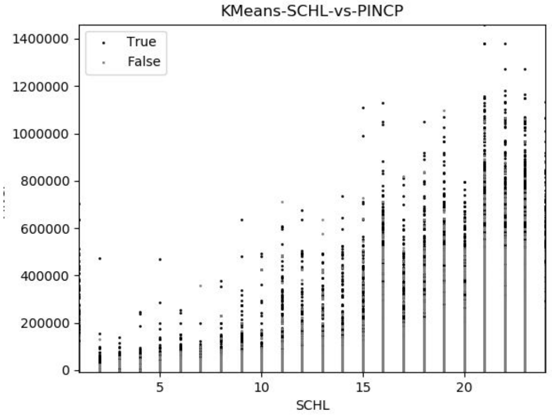


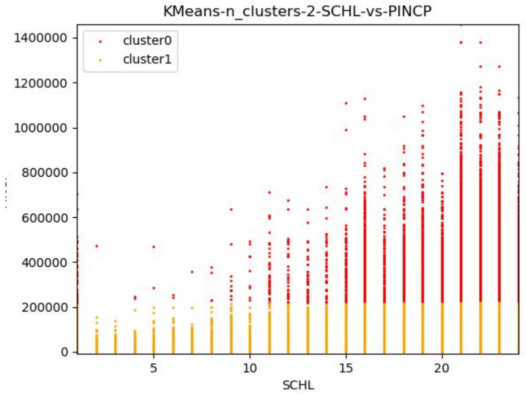






The divorce data also included unsupervised classification that were very similar to the actual classification. In the example below the people staying married not only make more than their counterparts for the amount of education they have, but the more education you have the more the person tends to make if they stay married. There is an upward trend on the dividing line between married and divorced as the education level rises. The lower area could be due to poor utilization of educational attainment or people taking on high levels of education in low earnings areas and could more over lead to mismatches in expectations during marriage.





The graph on the right above does not follow the trend. The K-clustering algorithm simply splits the groups into two based on personal annual income.

Do they otherwise line up naturally? Why or why not?

The clusters line up naturally with income and well knowing borrowing behaviors based on income. Low income borrowers feel they have to borrow. Middle income people borrow to try to keep up with others. Of the low and middle income people the ones that pay back loans typically have an already high established credit score which shows up as higher revolving balance. Then high income borrowers who most likely borrowing as an investment.

As for the divorce data clustering on annual income come as well, I don’t find it rational. Maybe, the algorithm is simply splitting on decision making ability which may be reflected in annual income and occupation used in this feature set. Also, because this is KMeans clustering the clustering algorithm could just be splitting on the most densely populated areas meaning that the cluster1 in the SCHL vs PINCP cluster represents more people in the data set. Cluster0 may be less accurately represented because they represent less of the population in the training data. KMeans algorithm will incorrectly classify because other high income people are the closest in the distance measurement used.

Compare and contrast the different algorithms.

In part 1, I wasn’t able to perform Expectation Maximization on the complexity of the LendingClub data. I was able to perform Expectation Maximization after simplifying the LendingClub data into more simple clusters. For the LendingClub, data the K-Means clustering separated the data almost identically at 2 clusters. Above 2 clusters for the K-Means algorithm the most significant factor in separation was annual income.

Overall, Expectation Maximization is different than the other algorithms because it doesn’t transform the existing values in the model. It simply generates items from a random distribution of the model. The data looks like Gaussian circles and ovals when generated. As the amount of features are reduced they circular shape of the Gaussian becomes more prominent. Expectation Maximization is good for finding the mean or average point of a cluster. Having a single point describe a set of points can be used to name or identify the point with something we observe in the real world with similar features.

PCA seeks to maximize the variance by setting a line in the dimension that gives the most variance. During the divorce data processing with a low feature count it was easy to see the PCA transformations. PCA focuses on the most varying features which tend to be the most important features

ICA wants to create features that store independent information that isn’t reliant on other features. In the divorce data graphs it’s easy to see the merged features. In the LendingClub data the data is not very independent due to being included in the FICO score. The best ICA algorithm tuning parameters set the components to only two components which was the lowest allowed components.

RCA attempts to reduce features by choosing a random projections on the sample. This method keeps random portions of the original feature set. It’s also quick to use.

The FeatureAgglomeration was similar to ICA in that it merges features. The feature merge of FeatureAgglomeration is much less aggressive. ICA is seeking to make the feature set as independent as possible where Feature Agglomeration is not seeking to do this aggressively.

What sort of changes might you make to each of those algorithms to improve performance?

```
lendingclub_observations-KMeans-{'n_clusters': 6}
lendingclub_observations-PCA-{'svd_solver': 'auto', 'n_components': 7, 'whiten': False}
lendingclub_observations-FastICA-{'max_iter': 20, 'n_components': 2, 'algorithm': 'parallel'}
lendingclub_observations-GaussianRandomProjection-{'n_components': 2}
lendingclub_observations-FeatureAgglomeration-{'n_clusters': 2}
lendingclub_observations-KMeans-PCA-{'n_clusters': 2}
lendingclub_observations-GaussianMixture-PCA-{'n_clusters': 1, 'reg_covar': 1e-07}
lendingclub_observations-KMeans-ICA-{'n_clusters': 2}
lendingclub_observations-GaussianMixture-ICA-{'n_clusters': 6, 'reg_covar': 1e-07}
```

```
lendingclub_observations-KMeans-RCA-{'n_clusters': 6}
lendingclub_observations-GaussianMixture-RCA-{'n_clusters': 1, 'reg_covar': 1e-07}
lendingclub_observations-KMeans-ANY-{'n_clusters': 6}
lendingclub_observations-GaussianMixture-ANY-{'n_clusters': 6, 'reg_covar': 1e-07}
divorce_observations-PCA-{'n_components': 4, 'whiten': False, 'svd_solver': 'auto'}
divorce_observations-GaussianRandomProjection-{'n_components': 2}
divorce_observations-KMeans-PCA-{'n_clusters': 7}
divorce_observations-KMeans-ICA-{'n_clusters': 6}
divorce_observations-KMeans-RCA-{'n_clusters': 6}
divorce_observations-KMeans-ANY-{'n_clusters': 6}
```

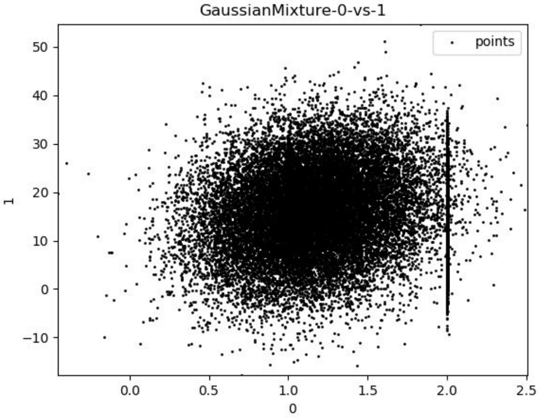
How much performance was due to the problems you chose?

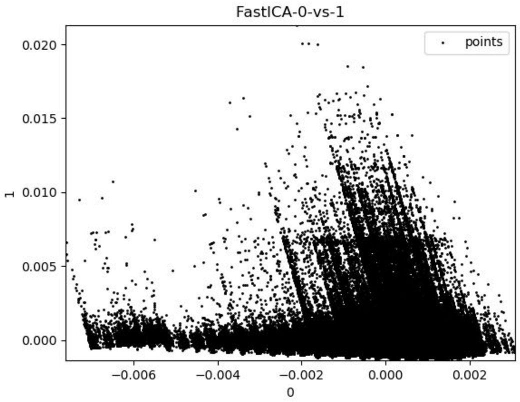
The LendingClub data was able to be clustered well because it was based on real world data. If the data generated was random then there would be no projections to represent or tight clusters to form from the data. Most of the patterns revolve around income for both of the data sets. The LendingClub data is able to be significantly reduced into less features because the FICO score component represents much of the feature sets. Divorce data was not significantly reduced into less features because it was based on four features somewhat independent. Sex and occupation and Occupation and personal income may have overlap in the feature set, but not to the degree in which they reduce the feature set below 3.

Can you describe how the data look in the new spaces you created with the various algorithms?

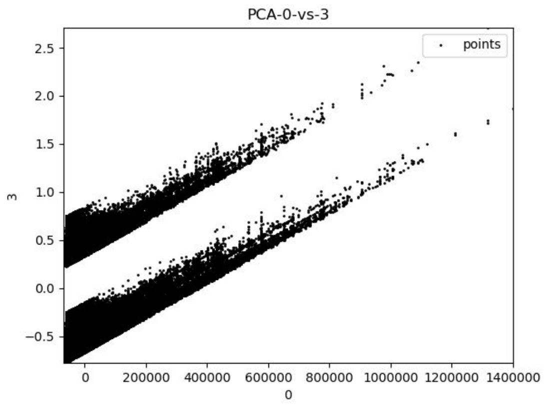
The LendingClub spaces are mostly uninterpretable because it starts with 24 features and is reduced to at most 6 features. The merge of so many features at once makes the graphs unexplainable.

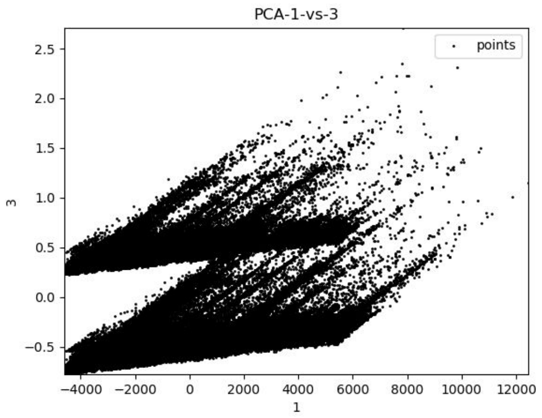
The divorce data had only four features so most reduction in feature sets were understandable as they were reduced to two or three feature sets from four.



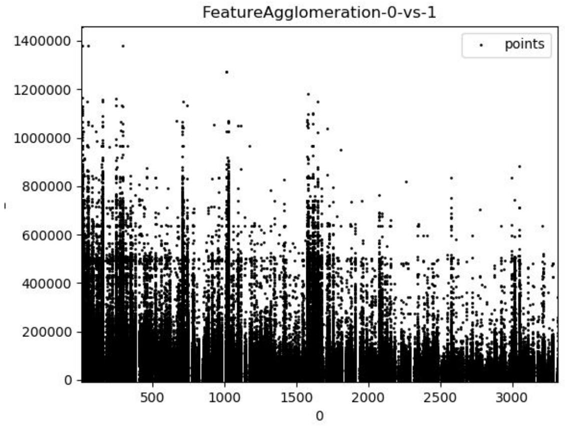


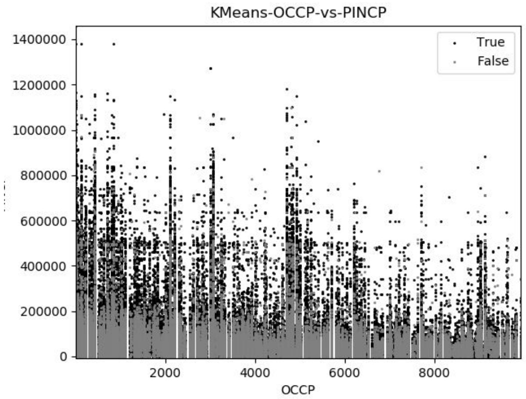
The Gaussian is centered around a central focal point with points concentrated in the center and points less concentrated farther away from the center.





Above you can see the PCA splits its features into male and female sections representing the two large clusters.





The points created for Feature Agglomeration in the graph above look just like the regular features to the right of the graph . This is because Feature Agglomeration merges features together and will keep information about the previous features.

For PCA, what is the distribution of eigenvalues?

Lending club PCA eigenvalues - n_components 7

[2.01039120e+37 6.45491555e+36 5.88993195e+35 1.98062442e+35
 9.37674867e+34 5.79161874e+34 6.10989709e+33]

divorce-PCA

[5.73242859e+09 6.56670197e+06 9.74692834e+00 2.30034460e-01]

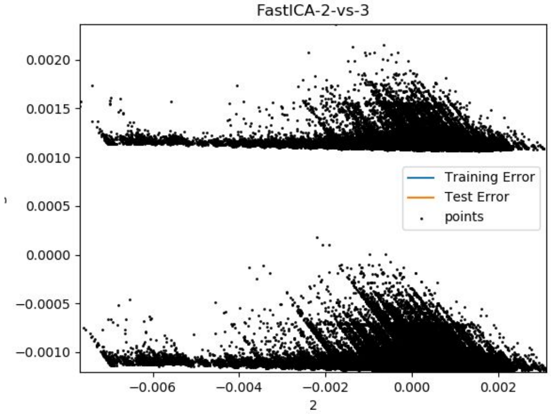
For ICA, how kurtotic are the distributions?

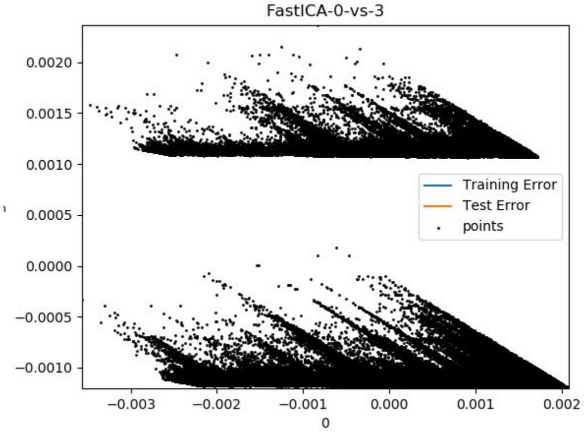
LendingClub - FastICA[6.49853784 -1.6535581]

Divorce - FastICA[28.65785361 8.85119823 -0.59380588]

Do the projection axes for ICA seem to capture anything "meaningful"?

The Lending Club ICA values were merged all the way from 24 to two components. This made the ICA graph difficult to understand.





Above we can see that ICA attempts to merge a feature for male and female creating two different sections for the appended feature. Both merged features look mostly similar which is interesting.

Assuming you only generate *k* projections how well is the data reconstructed by the randomized projections?

The LendingClub data is not well represented by randomized projections. The projections do not look the same as the other graphs.

The divorce data randomized projections for an interesting diagonal line shape across the x and y axis.

Assuming you only generate *k* projections how well is the data reconstructed by the PCA?

Only one or two projections of PCA provide most of the explained variance for both the data sets. As more projections are added the sum of the explained variances approaches 1.0 or complete explanation of the data.

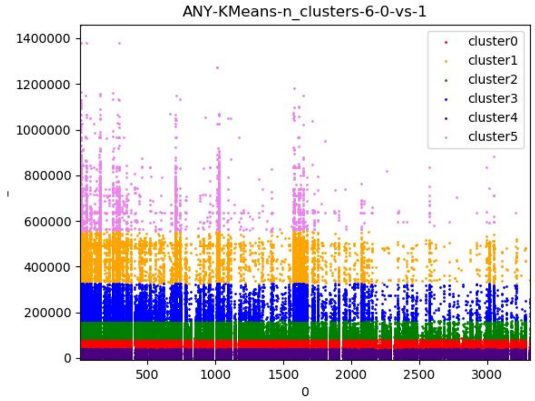
How much variation did you get when you re-ran your RP several times?

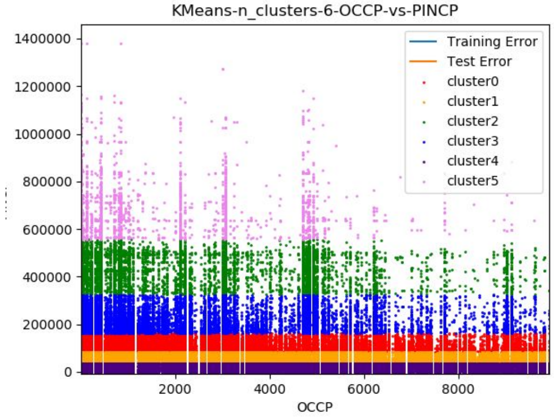
I received a lot of visual variances for the LendingClub data because the data included so many features. I received little visual variance for the divorce data because it had only a few distinct features.

When you reproduced your clustering experiments on the datasets projected onto the new spaces created by ICA, PCA, and RP, did you get the same clusters as before? Different clusters? Why? Why not?

For the LendingClub data, using the KMeans cluster algorithm after using any dimensionality reduction led to sparse graphs that looks completely different from the KMeans clustering graphs. The dimensionality reduction reduced the data from 24 features to 6 features most of the time. This created very different looking dimensions for the data that were unrecognizable.

Unlike the LendingClub data, the divorce data has only 4 features and has very similar clusters after using the reduction algorithms.





The EM clusters all look like Gaussian circles with different shapes. The reduction didn’t help differentiate between the multiple circular shape. Nor do the circular shapes resemble each other on the edges on the Gaussian.

When you re-ran your neural network algorithms were there any differences in performance? Speed? Anything at all?

Using dimensionality reduction on the LendingClub data allowed for actually decent error rates from all reduction techniques. Using the clustering algorithms didn’t significantly improve the performance of the neural network.

Divorce Data - Supervised Learning
Error

MLPClassifier	0.14826135121858297	0.14680568435299135

Time

MLPClassifier	17.0886628	0.01854870000000375

LendingClub Data - Supervised Learning
Error

MLPClassifier	2.3213331999999998	0.0219599999999998

Time:

MLPClassifier	2.3213331999999998	0.0219599999999998

Citations

Hayes, Genevieve. “Machine Learning, Randomized Optimization and SEarch¶.” *Mlrose*, 2019, mlrose.readthedocs.io/en/stable/.

"Personal Loans Borrow up to $40,000 and Get a Low, Fixed Rate." *Peer to Peer Lending & Alternative Investing*, LendingClub, 2019, www.lendingclub.com/info/download-data.action.