Understanding Lending Club Data Through SVD and PCA

Jonathan Hilgart

Lending Club is an online marketplace that allows individuals to buy debt in a peer-to-peer marketplace. Currently, Lending Club has loan information on over 800,000 loans from 2008-2015 online at kaggle.com. In analyzing this dataset, two groups of loans were created based upon loans status; either ‘Paid Off’ for one group, or another group of ‘Default’ and ‘Charged Off’. Paid off means that the loan has been paid off when the data was pulled. Default means that a borrower has gone more than 120 days without a payment and Charged Off means that a borrower has gone more than 150 days without a repayment. I grouped Default and Charged Off together because ~90% of the principal is typically lost after this time period(1). This analysis is interesting to understand factors that inform risk when lending money. In addition, understanding how the number of features needed to predict risk changes across different profiles.

For this analysis, SVD and PCA was used to understand the underlying variance in Lending Club data, and SVD was used predict which group different people would fall into. In order to analyze this data, the following alterations were made.

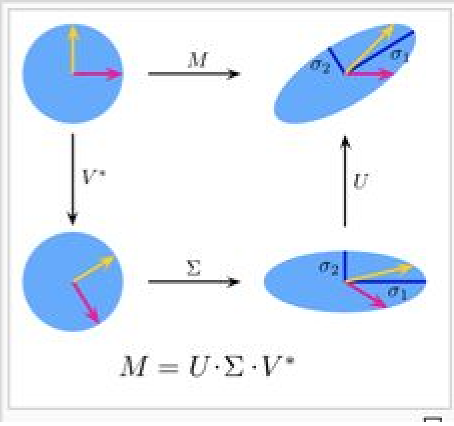
1. Loans were grouped into two categories, paid off loans and default/charged off loans.
2. All categorical features, as well as loan indicator variables (i.e. loan id) were dropped
3. Features were normalized using a standard scalers (mean = 0 variance =1). The reason for this standardization is to compare the variance of features that have vastly different measures (i.e. loan amount vs interest rate), and this allows computation via PCA.
4. SVD was computed on a sample of the original matrix due to resource/time constraints on the local machine.
5. PCA was computed on the entire matrix

SVD

$$\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$$

Formula for SVD where V* is V_T

SVD is a method to decompose a matrix, M, into three different matrices that each represent different characteristics about your original matrix. SVD produces three elements; U, D, and V_T. The U and D matrices are both orthonormal basis vectors for the matrix space. The matrix U corresponds to the eigenvectors from the A*A.T. These are the eigenvectors that correspond to your feature space, because it maps rows to ‘concepts’ in your data). For the Lending Club matrix, the elements that map to represent U are the strength of each person to the ‘concepts’ space. The V.T matrix corresponds to the eigenvectors from the A.T*A space, which is often called your sample space and maps features to ‘concepts’ . For Lending Club data, V.T maps attributes about loans to concepts. Finally, the matrix D corresponds to the eigenvalues from the matrices A*A.T and A.T*A (The eigenvalues are the same for both of these matrices). These eigenvalues give a relative strength of each mapping along the eigenvector for your feature space or your sample space. Concepts, in the sense of this paragraph, are underlying relationships in your matrix between different features and different rows (usually a combination of features dictated by rows).



(7)

One way to think about these matrices is that the V_T (V* in the picture above) matrix rotates your data sample into the concept space (related to each feature, which is a change of basis), D scales the strength of each concept along the different directions in your concept space, and then U rotates your data back to the original basis and transforms these concepts into the rows of your data set (transpose back to the elements of each loan by reversing the feature space) (7). This will recompose your original matrix A. The last feature of SVD is that it can be used for dimensionality reduction by comparing each element in your diagonal matrix / sum of the trace of your diagonal matrix. This process of dropping the smallest eigenvalues minimizes the RMSE of the matrix (To minimize the norm of the original matrix minus a low rank matrix approximation squared)(4).

$$\|M - M'\|^2$$

$$M = U * E * V.T$$

If the original matrix is, represented by the equation above after setting a criteria for amount of variance in the original matrix desired, approximate the original matrix by a lower rank approximation of E', then the equation below is true.

$$M' = U * E' * V.T$$

Now, the differences between these two matrices is represented by the differences in the singular value matrices E - E'. This is true because matrix U is column orthonormal while V.T is row orthonormal with the same eigenvectors between each (with one transposed). By the properties of column orthonormality, when squaring these matrices, the dot product will be zero between columns and will be 1 for the dot products of the same column.

$$(M - M') = U * (E - E') * V.T$$

Therefore, the RMSE terms are dictated by the difference in E - E’ singular values that zeroed. Note, a full proof is beyond the scope of this paper.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

For SVD, and PCA, it is possible to minimize the RMSE while maintaining a threshold level of variance. For this analysis, a criteria of keeping 90% of the original variation was set. Unfortunately due to the restrictions of computing SVD on my local machine, a sample of each of the original matrices was used for SVD while the entire matrix was used for PCA. I used sample size of 10,000 for this analysis via SVD.

First, the **default charged off group**, was analyzed with sample size 10,000 with 5 trials. In order to determine the number of components to keep, the formula below was used.

$$\sum \frac{E_{ij}}{\sum E_{ij}} >= Retained - Variance - Goal$$

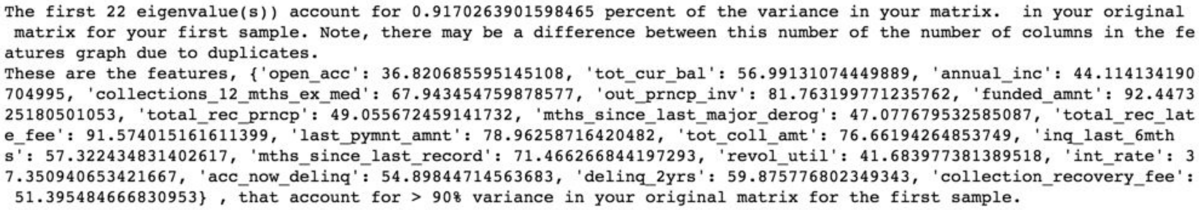
After setting a retained variance goal, then you will take the sum of the explained variance of each eigenvalue until you reach this threshold. Next, The importance of each feature in the original matrix was computed using the following formula.

$$max(abs(V.T_{ij}))$$

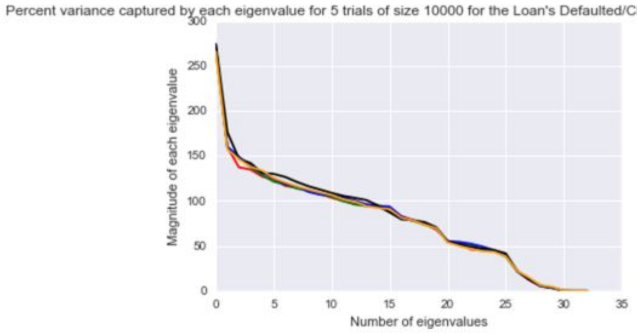
Using the i=rows and j=columns index, use this index to find the corresponding feature from the original matrix. Then, to find the magnitude of importance for each features, each of these values from the eigenvector were scaled by the corresponding eigenvalue in the E matrix.

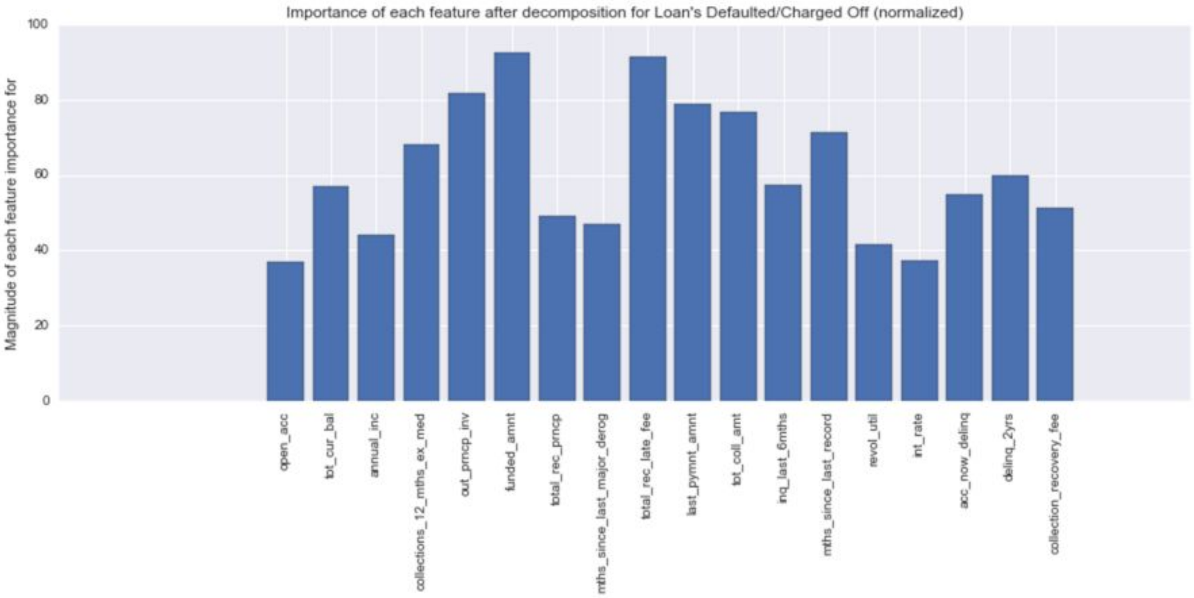
$$max(abs(V.T_{ij})) * E_{ij}$$

Finally, a bar graph was made with each feature in the original matrix against the magnitude of importance.



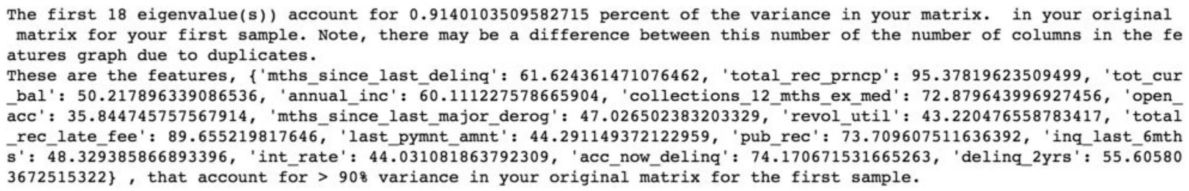
```
The first 22 eigenvalue(s)) account for 0.9170263901598465 percent of the variance in your matrix.  in your original
 matrix for your first sample. Note, there may be a difference between this number of the number of columns in the fe
atures graph due to duplicates.
These are the features, {'open_acc': 36.820685595145108, 'tot_cur_bal': 56.99131074449889, 'annual_inc': 44.114134190
704995, 'collections_12_mths_ex_med': 67.943454759878577, 'out_prncp_inv': 81.763199771235762, 'funded_amnt': 92.4473
25180501053, 'total_rec_prncp': 49.055672459141732, 'mths_since_last_major_derog': 47.077679532585087, 'total_rec_lat
e_fee': 91.574015161611399, 'last_pymnt_amnt': 78.96258716420482, 'tot_coll_amt': 76.66194264853749, 'inq_last_6mth
s': 57.322434831402617, 'mths_since_last_record': 71.466266844197293, 'revol_util': 41.683977381389518, 'int_rate': 3
7.350940653421667, 'acc_now_delinq': 54.89844714563683, 'delinq_2yrs': 59.875776802349343, 'collection_recovery_fee':
 51.395484666830953} , that account for > 90% variance in your original matrix for the first sample.
```



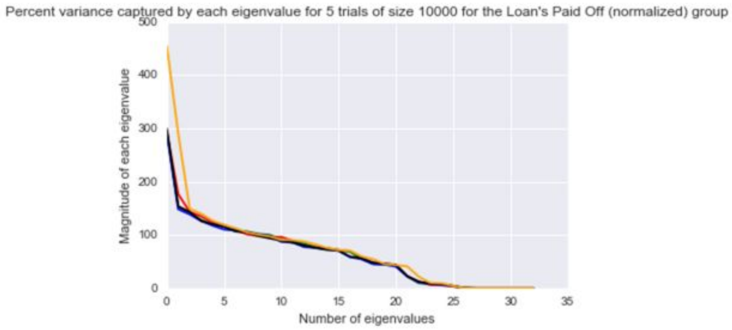


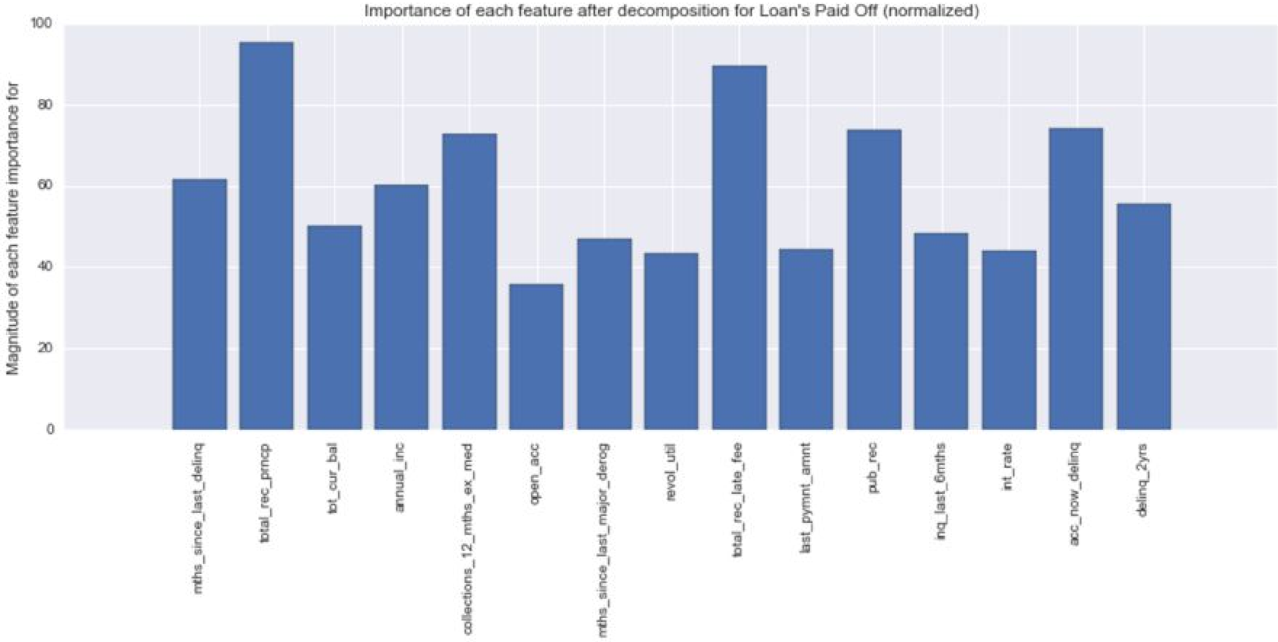
Above, notice that there is a difference in the importance of eigenvalues for different samples of the matrix. From this graph, the most important features for people who default or charge off their loans appears to be tot_rec_late_fee and funded_amnt.

A similar process was carried out for the **paid off loans.**



```
The first 18 eigenvalue(s)) account for 0.9140103509582715 percent of the variance in your matrix.  in your original
 matrix for your first sample. Note, there may be a difference between this number of the number of columns in the fe
atures graph due to duplicates.
These are the features, {'mths_since_last_delinq': 61.624361471076462, 'total_rec_prncp': 95.37819623509499, 'tot_cur
_bal': 50.217896339086536, 'annual_inc': 60.111227578665904, 'collections_12_mths_ex_med': 72.879643996927456, 'open_
acc': 35.844745757567914, 'mths_since_last_major_derog': 47.026502383203329, 'revol_util': 43.220476558783417, 'total
_rec_late_fee': 89.655219817646, 'last_pymnt_amnt': 44.291149372122959, 'pub_rec': 73.709607511636392, 'inq_last_6mth
s': 48.329385866893396, 'int_rate': 44.031081863792309, 'acc_now_delinq': 74.170671531665263, 'delinq_2yrs': 55.60580
3672515322} , that account for > 90% variance in your original matrix for the first sample.
```





The major difference between the Paid Off group and the Default/Charged off group is the number of features that account for 90% of the variance (22 for sample size 10k vs 18 for sample size 10k for Default/Charged off vs Paid off respectively). Notice, that there is a difference between the number of features that account for 90% of the variance in the elbow plot, and the number of features present in the bar graph above. This is because when calculating the index of the most important features from each eigenvector, there might be duplicate indexes. This makes sense because eigenvalues are a combination of all features in the matrix - so a particular feature may be present more than once.

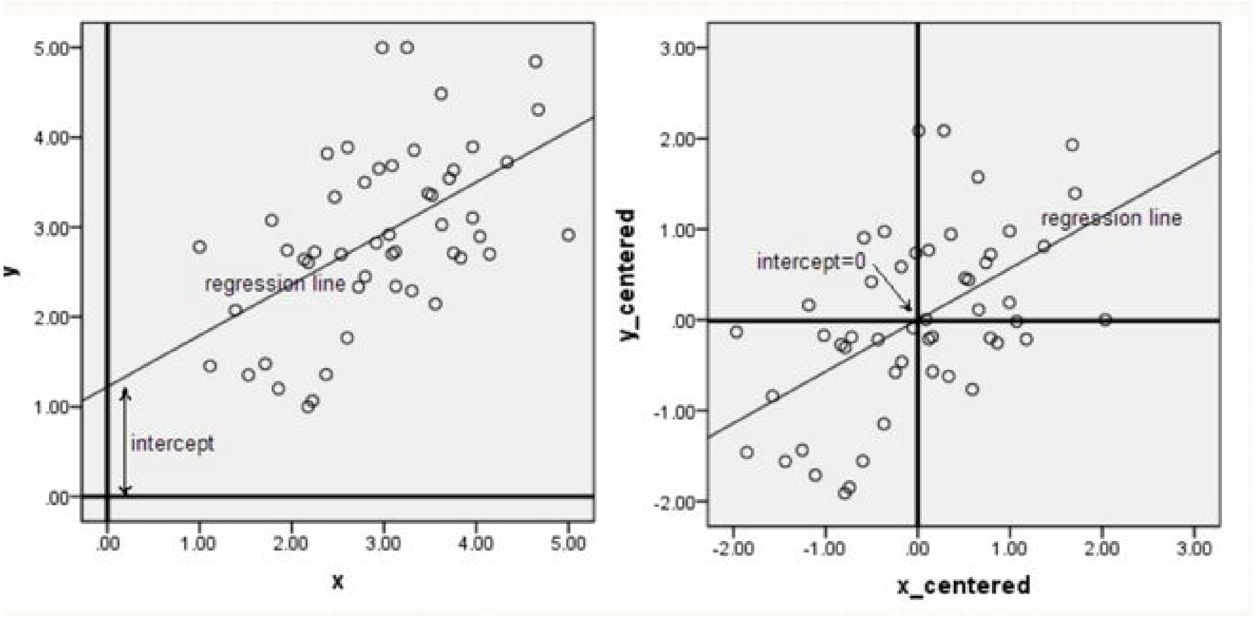
This analysis indicates that there are fewer features needed to predict if a person will repay a loan compared to if they will default on a loan. The additional features needed are ['out_prncp_inv', 'open_acc', 'dti', 'recoveries', 'tot_coll_amt']. The features that are the same across both groups are {'mths_since_last_delinq', 'tot_cur_bal', 'annual_inc', 'collections_12_mths_ex_med', 'funded_amnt', 'total_rec_late_fee', 'last_pymnt_amnt', 'pub_rec', 'inq_last_6mths', 'acc_now_delinq', 'delinq_2yrs'}.

PCA

Next, PCA was performed to compare the results of dimensionality reduction to the SVD analysis. PCA is a similar process to SVD, except that it is a requirement to center the data at zero and have a variance of 1 (compared to SVD where this is not a requirement). The reason is that to compute PCA, a covariance matrix is used and this covariance matrix is computed from values that are centered (the mean of the column is subtracted out from them). Below, X is the centered data and C is the covariance matrix.

$$C = \frac{X.T * X}{N - 1}$$

In addition, centering the data forces the intercept to go through the origin to better compare distance of each element (as a distance from the origin). Otherwise, it is harder to understand what a distance of an element means (5).

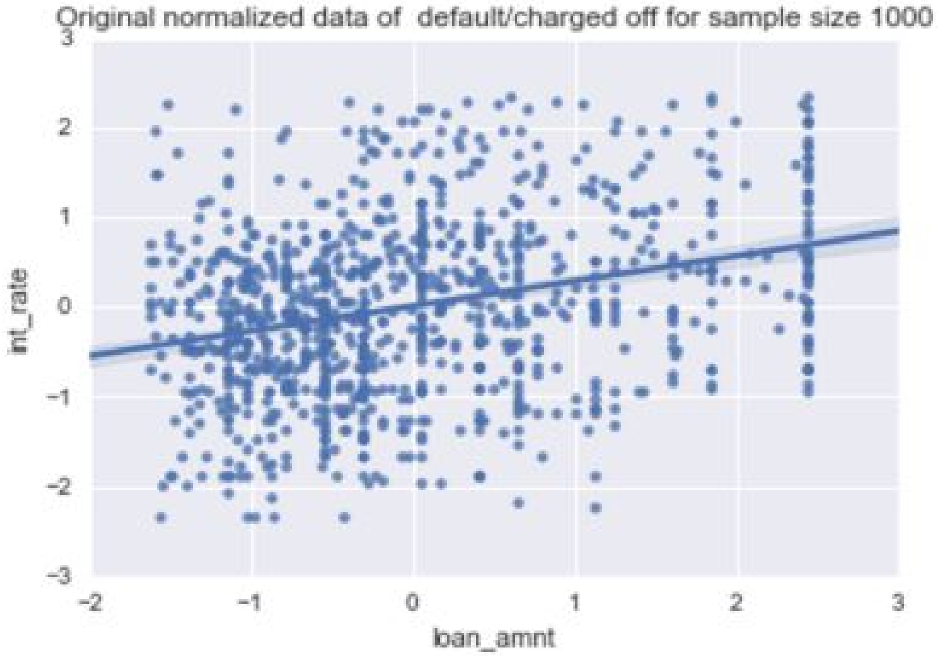


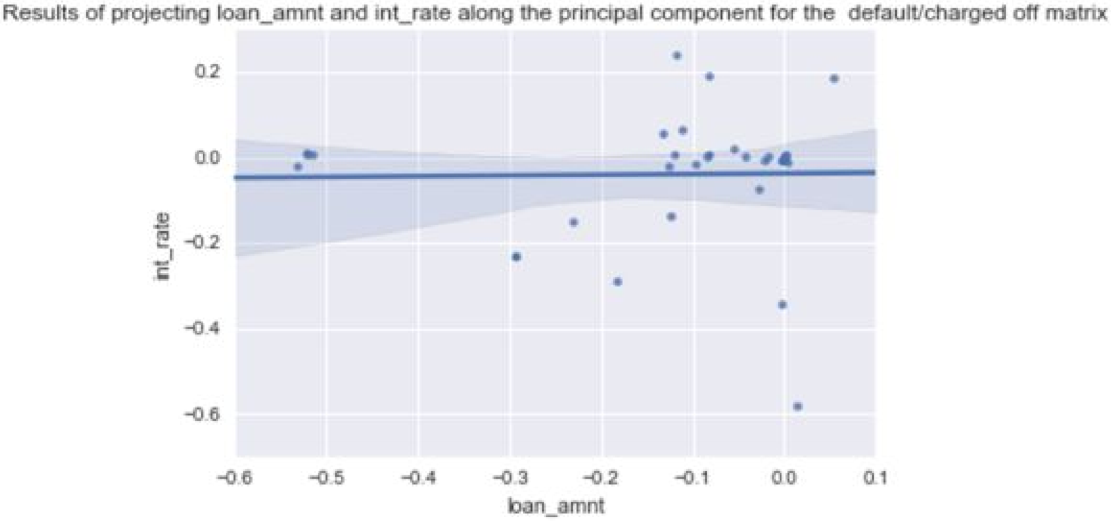
After centering your data, and computing your covariance matrix, it is possible to finish your PCA analysis by either computing SVD on the C matrix, or solving for the eigenvalues and eigenvectors directly from the C matrix.

The goal of PCA is to eliminate features that have a high covariance related to each other (only keep features where the distance between each is maximized - up to the desired variance level). One problem with PCA is that it is harder to interpret than SVD (since the covariance matrix obscures the results). However, it is useful for dimensionality reduction. The output from PCA are components (directions) and magnitudes. The first principal component corresponds to a line that passes through the mean and minimizes each point's distance from the line (8).

PCA was first computed on a sample of 1,000 rows from the **default/charged off** loans matrix. The original matrix contains all features - 33 while the PCA matrix was computed on the reduced matrix - 16. The number of components (to maintain 90% variance of the original matrix) was computed using sklearn's PCA package via a similar process outlined above for SVD (Principal axes in feature space, representing the directions of maximum variance in the data(6)).

Next, I compared the results for loan_amnt against int_rate in the original matrix sample vs the projection of these variables along the principal component.

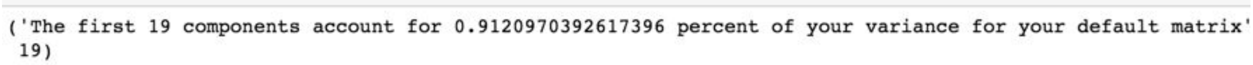




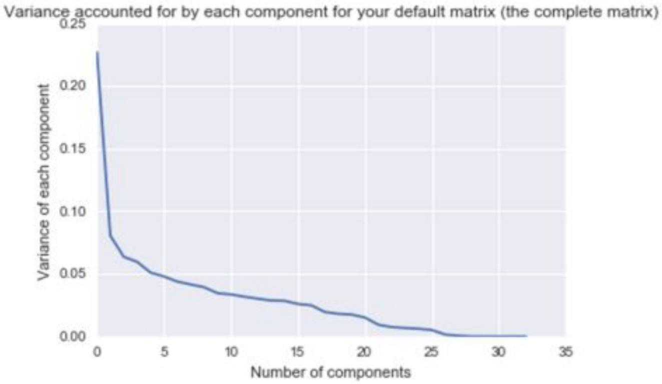
Results of projecting loan_amnt and int_rate along the principal component for the default/charged off matrix

Above, notice that PCA attempts to 1) remove highly correlated features and 2) represent the original relationships between columns with fewer data points.

Next, determine how many components account for 90% of the underlying variance in the data.



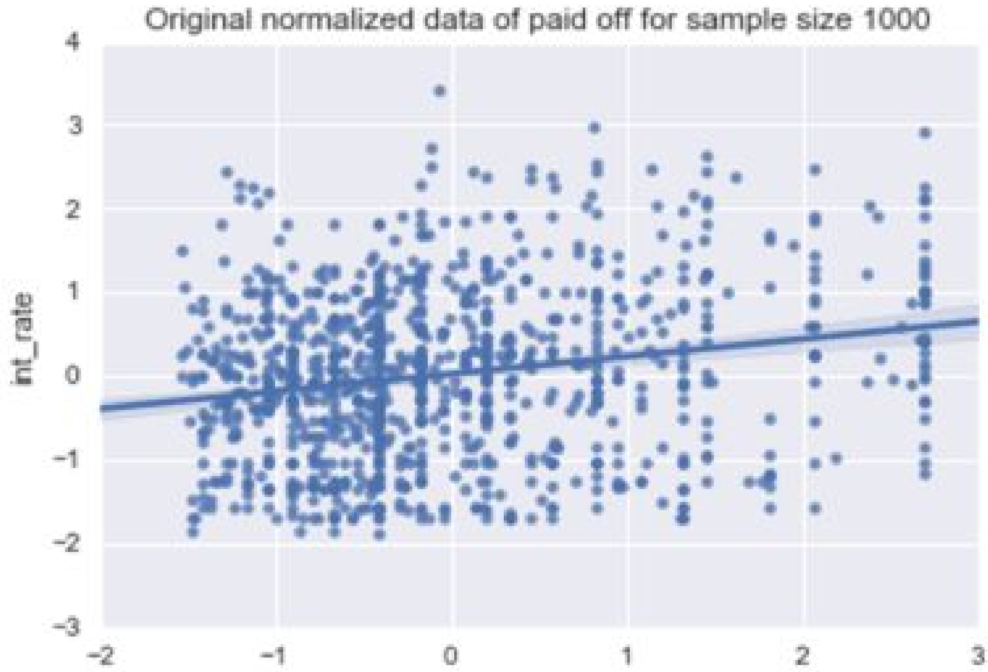
```
('The first 19 components account for 0.9120970392617396 percent of your variance for your default matrix'
 19)
```

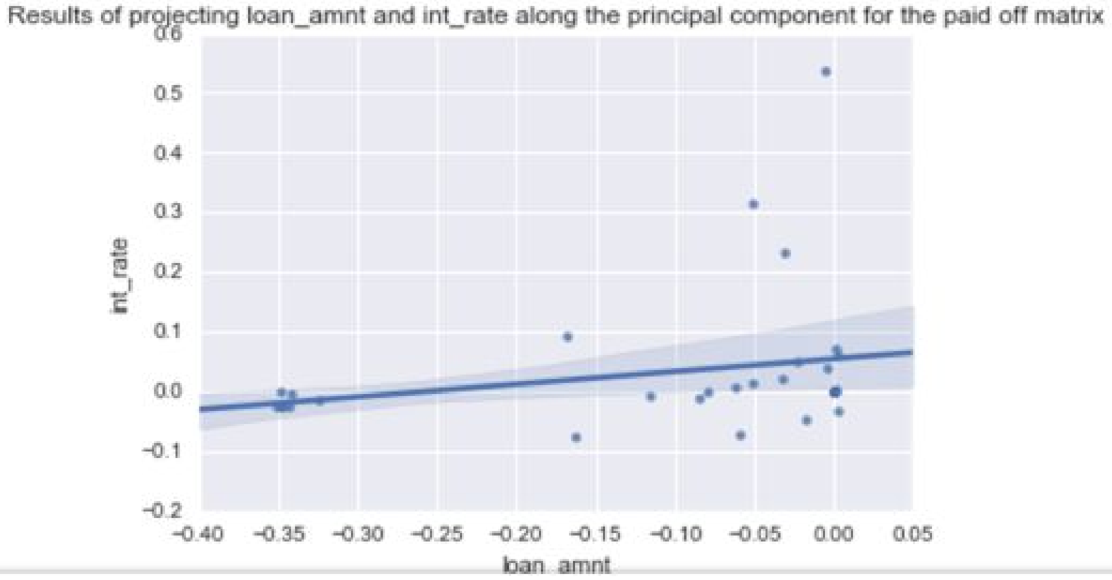


Variance accounted for by each component for your default matrix (the complete matrix)

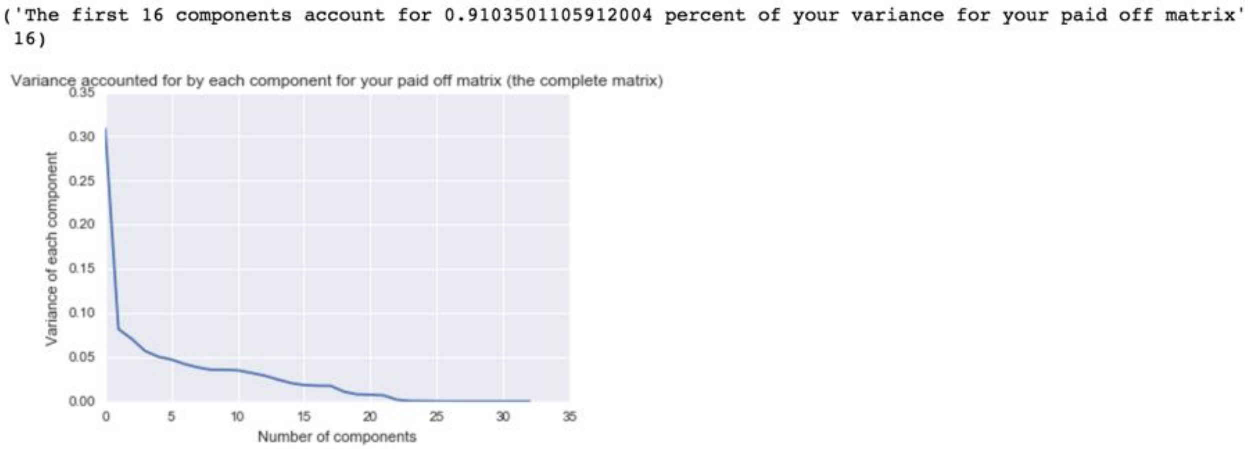
For a sample size of 1,000 for, there are 19 components that explain 90% of the variance in the data (for default/charged off).

Next, I completed this same process on the **paid off** loans.





For a sample size of 1,000 for, there are 16 components that explain 90% of the variance in the data (for paid off loans). Above, you can also see that there is significantly less covariance between the elements in the PCA matrix. Next, determine how many components retain 90% of the variance for the entire matrix.



```
('The first 16 components account for 0.9103501105912004 percent of your variance for your paid off matrix'
 16)
```

After these alterations, with a goal of retaining 90% of the underlying variance in the data, SVD could reduce the dimensionality of the paid off loans matrix by **45%** [(33 features-18 features kept)/33 features] while PCA could reduce this matrix by **51%** [(33 features - 16 features kept) / 33 features]. For the default/charged off loans, SVD could reduce the dimensionality by **33%** [(33 features-22 features kept)/33 feature], while PCA could reduce this dimensionality by **42%** [(33 features-19 features kept)/33 features].

In order to understand why there was a difference in number of features to be kept (for 90% variance) between PCA and SVD, a t-test was computed using the formula below (The sum of the median for each column of your matrix. Then, take the difference between the population and the sample.).

$$\sum_j Pop.\,medianX_{ij} - \sum_j Samp.\,medianX_{ij}$$

This was compared for a sample matrix of 10,000 vs the entire matrix (with each matrix normalized) for the two groups. The p-value calculations are at the end of this paper.

Paid off loans sample versus original

- ~0% p-value.
- This means there is a very low chance of observing the values in the sample matrix by chance from the original matrix. Therefore, reject the null that the difference =0. This translates to an expectation that there is a difference in the number of principal components between the sample and the original matrix.

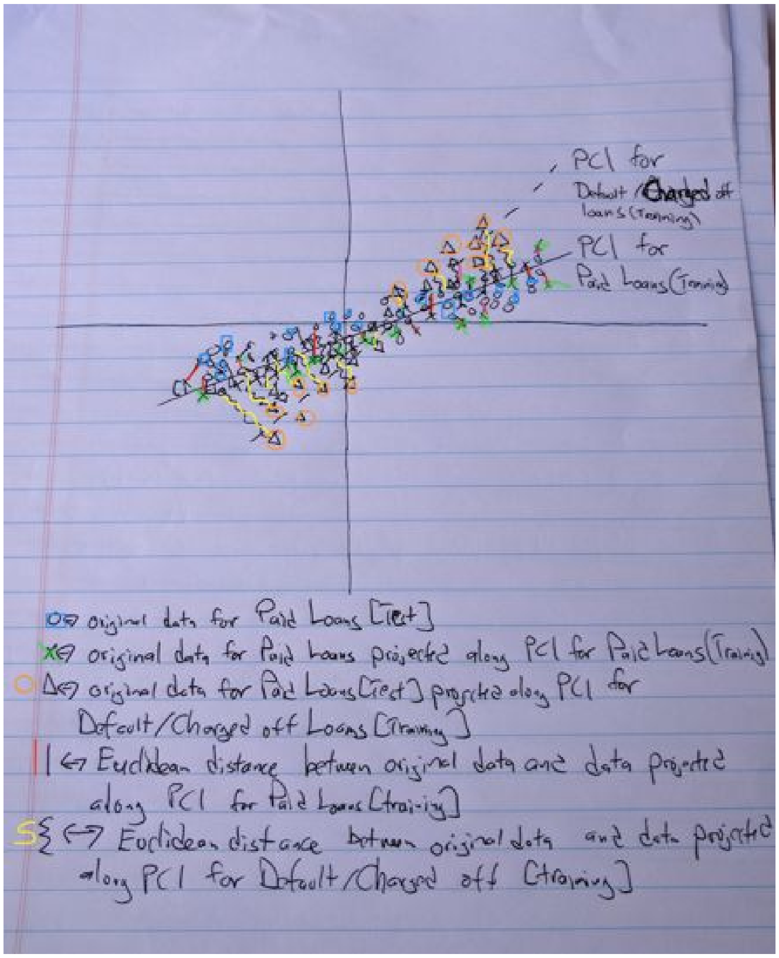
Default/Charged off loans sample vs original

- ~4.61e-80 % p-value.
- This means there is a very low chance of observing the values in the sample matrix by chance from the original matrix. This translates to an expectation that there is a difference in the number of principal components between the sample and the original matrix.

The difference in p-values, while small, most likely corresponds to the difference in the size of the original matrix (default/charged off loans (46,467) vs paid off loans (207,723)).

Prediction

After decomposing a matrix, it is possible to use this decomposition to predict values. For this data, I wanted to predict if users will be in the paid off of the default/charged off group. To do this, I created a training set of data ranging from 1% - 30% of the data from the original matrix(normalized as before), and compared 2,000 rows of test data projected along the eigenvector of the training data. In order to compare, I projected each row of the test data along the first row of the V.T vector from the training data. This will project my data along the direction that explained the principal direction of the training data. Next, I took the euclidean norm of each element from the test vector minus the result of the projection of my test data along the V.T vector of my training data. Next, compare these distances between projections onto the training eigenvector (from either the default /charged off group or the paid off group). The number that is smaller corresponds to the group that the SVD ‘predicts’ each user would fall into.


PC1 for Default/Charged off loans (Training)
PC1 for Paid Loans (Training)
□ ⇐ original data for Paid Loans [Test]
X ⇐ original data for Paid Loans projected along PC1 for Paid Loans (Training)
○ Δ ⇐ original data for Paid Loans [Test] projected along PC1 for Default/Charged off Loans [Training]
| ⇐ Euclidean distance between original data and data projected along PC1 for Paid Loans [training]
S ⇐ Euclidean distance between original data and data projected along PC1 for Default/Charged off [Training]

Above is an example of how prediction works for the test vectors and training components. Theoretically, the yellow euclidean distance should be larger than the red euclidean distance.

1) Note, for below this is an outer product between each vector and the V.T matrix. Here, only the first direction was used so i=0 for the V.T vector.

$$A = \sqrt{(\sum_{i=1}([X_{TestDefault_{ij}}] - [X_{TestDefault_{ij}} * V.T_{TrainingDefault_{ij}}])^2)}$$

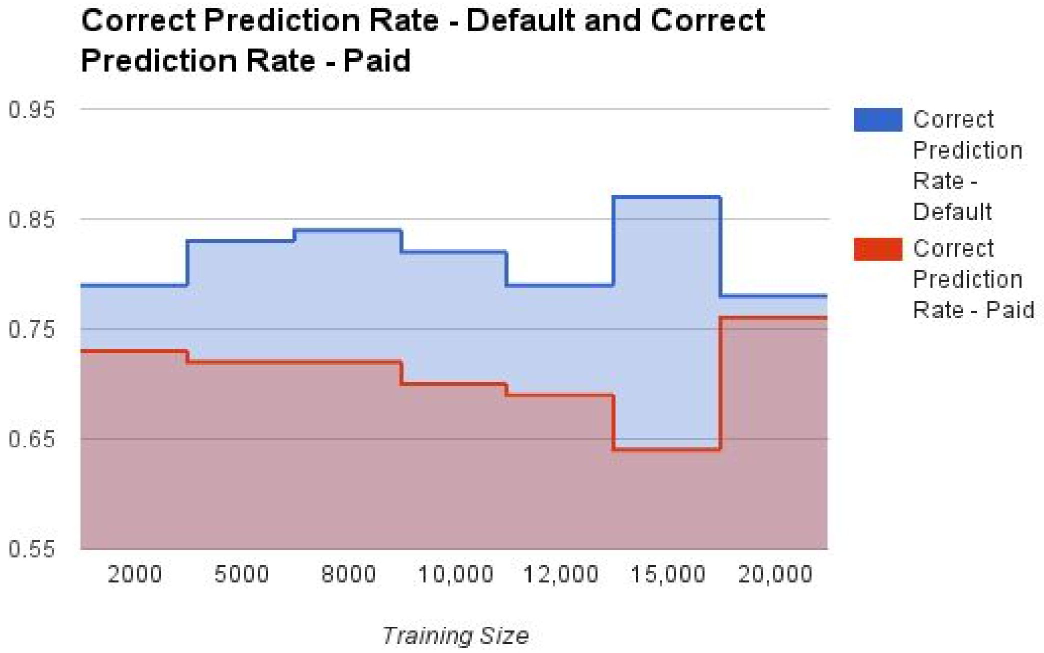
$$B = \sqrt{(\sum_{i=1}([X_{TestPaid_{ij}}] - [X_{TestPaid_{ij}} * V.T_{TrainingDefault_{ij}}])^2)}$$

$$C = \sqrt{(\sum_{i=1}([X_{TestPaid_{ij}}] - [X_{TestPaid_{ij}} * V.T_{TrainingPaid_{ij}}])^2)}$$

$$D = \sqrt{(\sum_{i=1}([X_{TestDefault_{ij}}] - [X_{TestDefault_{ij}} * V.T_{TrainingPaid_{ij}}])^2)}$$

First, compare A and B for your test group. If A is less than B, then the prediction states this person would belong to the default/charged off group. If A is greater than B, then this is the error percent for this model. Next, compare C and D for your test group. If C is greater than D, then this prediction states this person would belong in the paid group. Otherwise, this is the error term for paid prediction.

The prediction achieve accuracy in the range of 76-86% for people who will default/charge off their loan and 64-75% for people who will pay off their loan.



Throughout this paper, SVD and PCA techniques were utilized on Lending Club data. After this analysis, there was a difference in the number of features kept to explain 90% of the variance between the two groups - typically three to four features additional features for the default/charged off group across both techniques (SVD, PCA). This suggests that there are more features that influence whether a person will default/charge off compared to pay off their loan. Moving forward, a computation of SVD on the complete matrix should be done to verify that the difference in the number of features kept was due to sampling. In addition, this prediction could be improved moving forward in two ways.

1) Increase the size of the training set for both groups of loans (up to 80% of total data)
2) Increase the number of eigenvectors used to predict data along (instead of just the first direction).

For future work, it would be interesting to predict additional loan statuses (instead of paid off vs default/charged off) such as ‘Grace Period’ (0-15 days late for a payment) ‘Late 16-30 Days’ ...etc.

Footnotes

(1) https://www.lendingclub.com/info/demand-and-credit-profile.action

(2) Paid off loans sample versus original calculation of t-value

- Sum of median values for each feature for the sample of paid off loans = -5.418709236010741
- Sum of median values for each feature for the original matrix of paid off loans = -6.4759121631146455
- The standard error is calculated of the square root of the sample variance / sample size. The sample variance =1 due to the standardization process, and the sample size is 10,000. Therefore, the standard error = sqrt(1/10,000) = .01
- The t-statistic for the paid off loans is -(5.418709236010741 + 6.4759121631146455) / .01 = 105.72029271039041
- Next, if you take a t-distribution with 10,000 - 1 degrees of freedom (-1 for the sample), and calculate 1- CDF(105.72)
- P-value ~0%

(3) **Default/Charged off loans sample vs original calculation of t-value**

- Sum of median values for each feature for the sample of paid off loans = -7.455245587709658
- Sum of median values for each feature for the original matrix of paid off loans = -7.264425105004517
- The standard error is calculated of the square root of the sample variance / sample size. The sample variance =1 due to the standardization process, and the sample size is 10,000. Therefore, the standard error = sqrt(1/10,000) = .01
- The t-statistic for the paid off loans is -(-7.455245587709658 + 7.264425105004517) / .01 = -19.082048270514118
- Next, if you take a t-distribution with 10,000 - 1 degrees of freedom (-1 for the sample), and calculate 1- CDF(105.72)
- P-value ~~4.61e-80 % p-value.

(4)https://books.google.com/books?id=jBpKBQAAQBAJ&pg=PA401&lpg=PA401&dq=why+does+dropping+lowest+singular+value+minimize+rmse+svd&source=bl&ots=tbda17g1DB&sig=vLH7IzBWKjxm1lh0VT_qq3ABZw4&hl=en&sa=X&ved=0ahUKEwjC9Izkv-LPAhVqs1QKHSU4A6sQ6AEIIjAB#v=onepage&q=why%20does%20dropping%20lowest%20singular%20value%20minimize%20rmse%20svd&f=false

(5)
http://stats.stackexchange.com/questions/22329/how-does-centering-the-data-get-rid-of-the-intercept-in-regression-and-pca

(6) http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

(7) https://en.wikipedia.org/wiki/Singular_value_decomposition

(8) https://en.wikipedia.org/wiki/Principal_component_analysis