Iteration 4

HR Analytics

Data Magnets - Thuan Tran & Chayse Summers

Overview:

This project aims to understand the question : “Why are our best and most experienced employees leaving prematurely?” from HR perspective.

Motivation:

Most companies will come to have the same questions eventually, “Will an employee leave the company in the future? If so, when?”

Most managers, and even those at higher levels of an organization, often don’t know exactly why their employees choose to leave the company. Each case is different, so there is no single unique reasons for why people leave their jobs. However, some companies do a survey each year to record the level of satisfaction of their employees. The following attributes are commonly encountered:

- Employee satisfaction level
- Last evaluation
- Number of projects
- Average monthly hours
- Time spent at the company
- Whether they have had a work accident
- Whether they have had a promotion in the last 5 years
- Department
- Salary

By combining all of that history with the result of whether an employee left or not, we hope to have a dataset that demonstrates a pattern to help predict if an employee is on track to leave their current position.

Through analysis of the different attributes within the datasets, we will attempt to find a pattern for which attributes contribute the most to an employee leaving their position, or which attributes will help companies to better retain their key employees.

Application Area:

This project shows promise in application of the Human Resources department for employee retention by helping understand the key attributes that may be the cause of an employee’s termination or resignation.

Technical Approach and the data set

We found an interesting site for the data science community which has provided this data set to us conveniently in CSV form. The dataset consists of 15 thousand records along with 10 attributes. The link of the dataset can be found below in the “References” section. Since this dataset is already collected, we do not need to collect additional data Most the values in the dataset are continuous numbers, except for some attributes that are categorical like salary (low,medium or high).

Inside the dataset, there will be some missing values. After further analysis, we will determine our course of action for those missing values. One course of action we can think of at the moment is to perform analysis on all other records, determine their means, and variances. Then, based off what we learn from that, we will assign new value to the record that is missing.

We also expect that there will be outliers as well, e.g.someone who is really happy at the company but decided to leave. This could cause a drastic change to our classification. However, thanks to the large amount of data, we feel these outliers shouldn’t be a problem. Just to be sure, we will perform visualization and testing for the outliers to determine the appropriate action.

After further consideration, we decided to use all features of our data set. Some of the features are qualitative, and we will determine our course of action for those features. One possible approach is to transform those features into quantitative data, but still have the relationship to the original qualitative data. For example, high salary can be 1, medium can be ⅔, and low salary can be ⅓.
In addition, we will create a new feature set by using PCA in order to eliminate redundant, or correlated features, making it easier to visualize as well. We will also calculate the covariance matrix to see if features are independent or not.

Plots

For this data set, we think that a scatter and pdf plot will be useful for us. The script used to generate the plot will be attached with the document

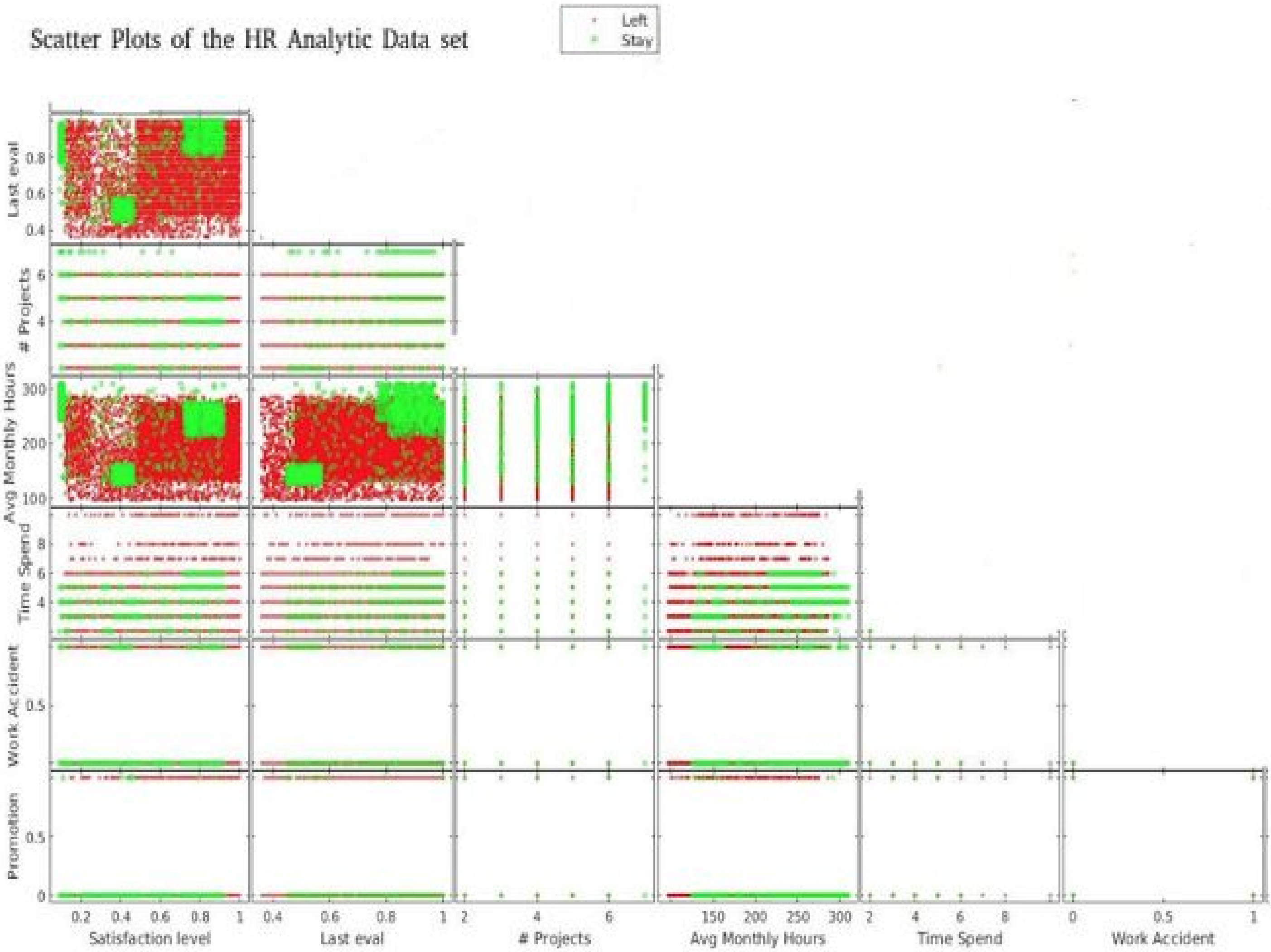
The scatter plot will help us to visualize what the distribution of each feature compare to another feature and help us to understand if is there a line that will be able to help distinguish whether an employee left or not . A preview of the scatter plots we generated will be provided below. We will also attach the image along with this document. (The script created the scatter matrix with duplicate attributes, so I used Paint to remove those duplicate plots)

The pdf plot will help us to determine if our data set follow a normal distribution. And if it follows a normal distribution, we can use the linear discriminant function in order to generate a line that will be able to classify our observations

Scatter Plot

Looking at the scatter plot in two dimensional, it is hard to generate a linear line that can separate the data. Since the data set has 15000 points and we used all of them for plotting, there are multiple cases where the data got blend in. However, after we plotted 3D graphs way below, we can see that there is a clean separation in the new space of the data

Scatter plot matrix of our data set



PDF plots

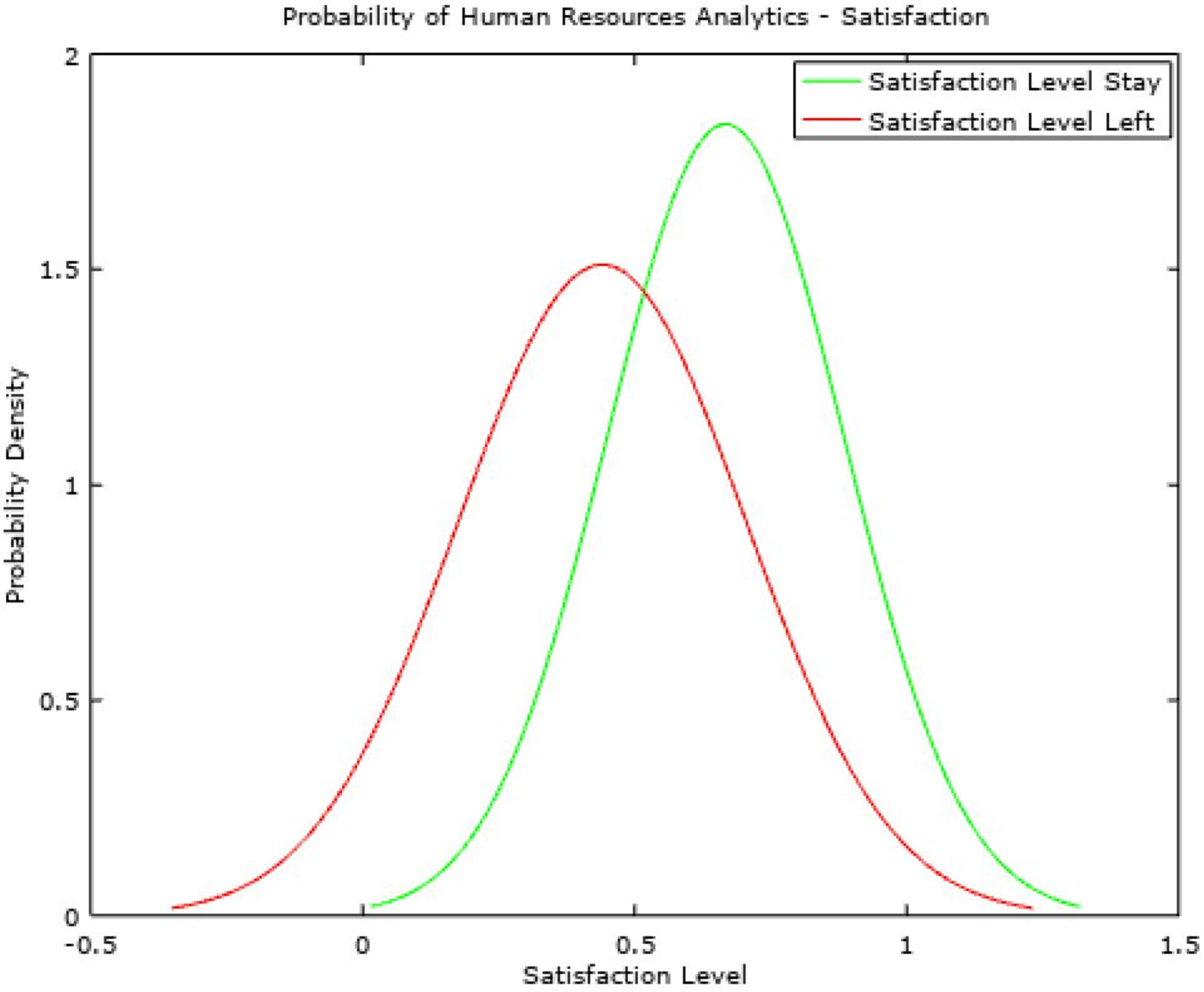


Figure 1: Employee Satisfaction Level PDF

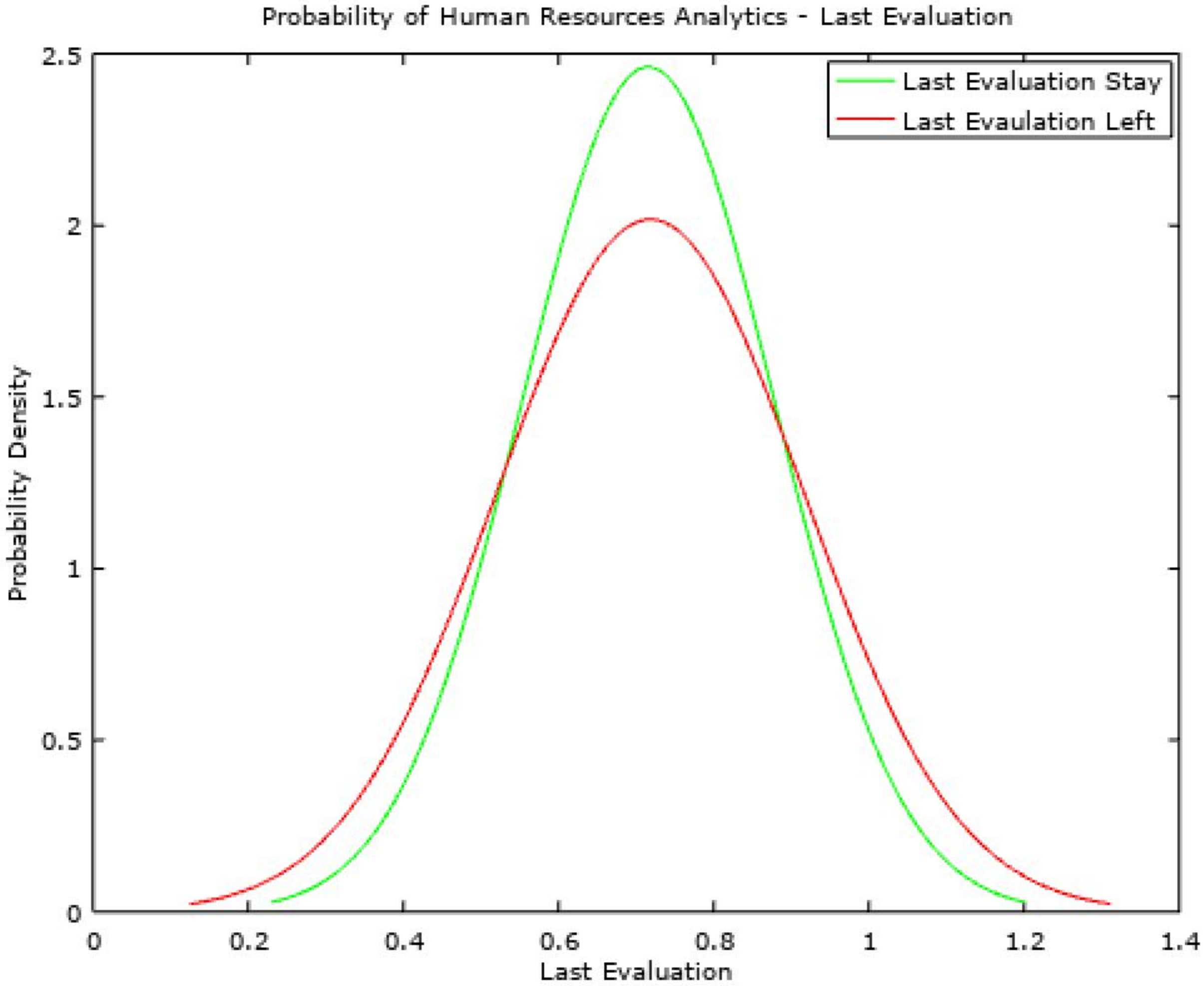


Figure 2: Last Employee Evaluation PDF

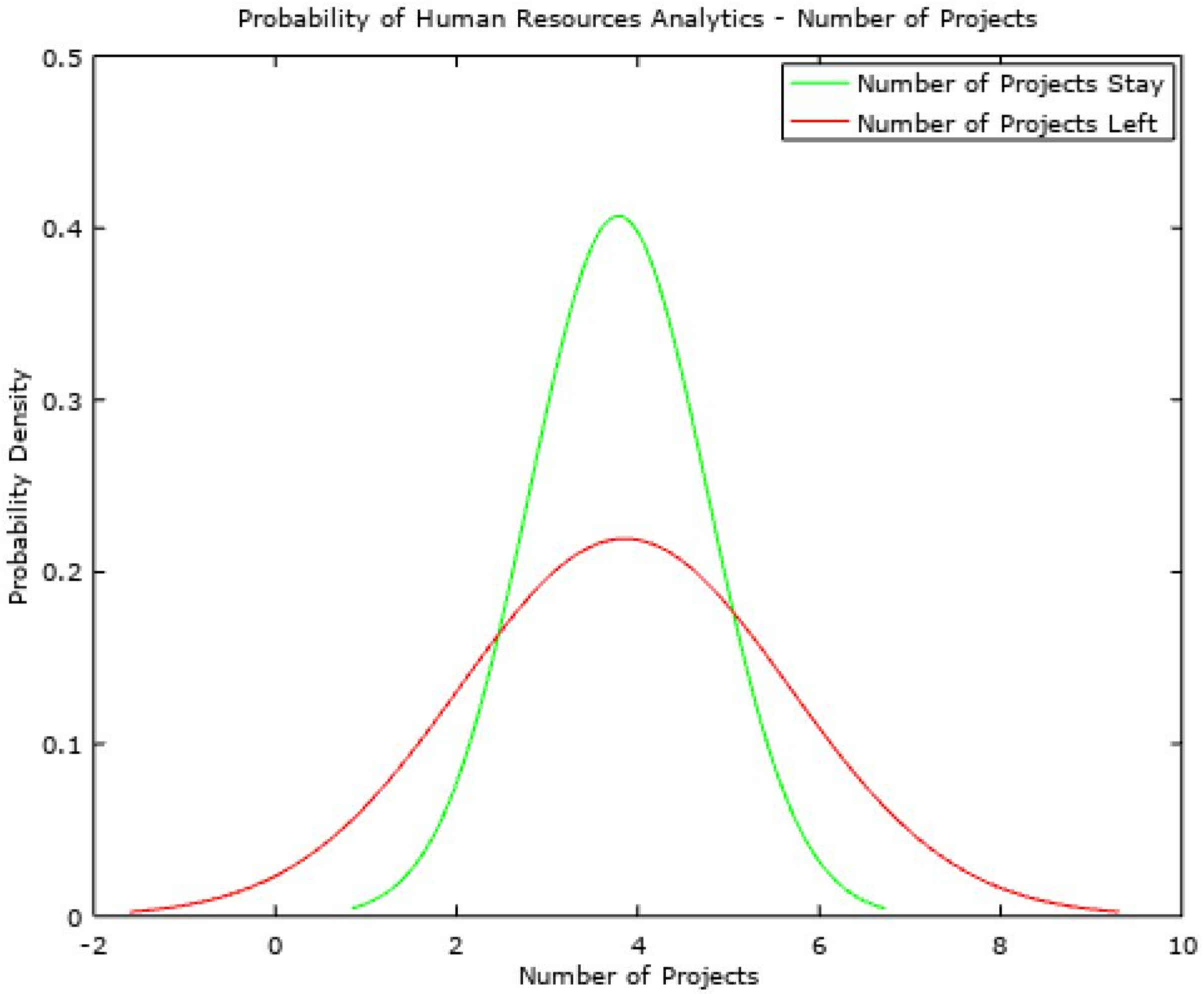


Figure 3: Number of Employee Projects PDF

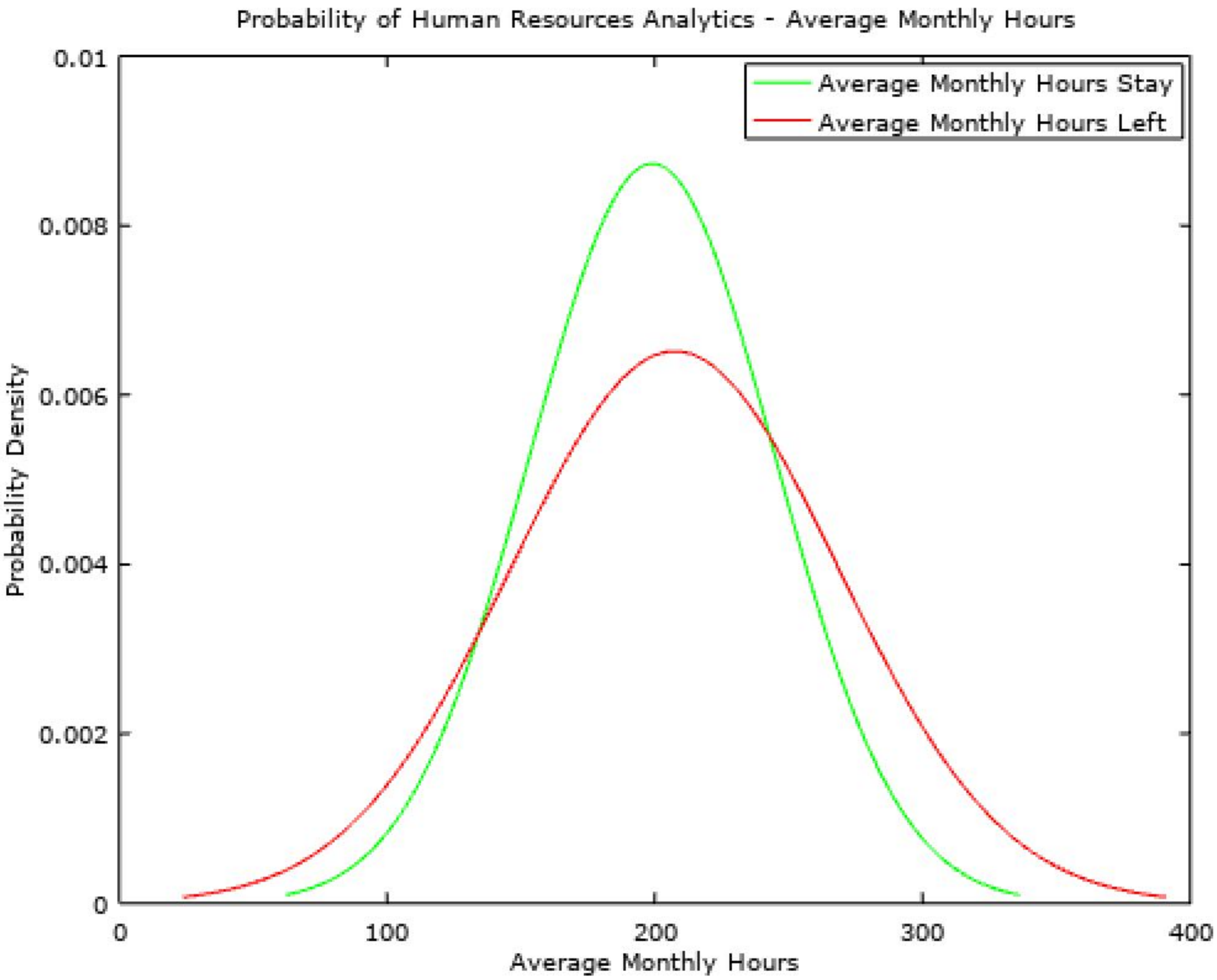


Figure 4: Average Monthly Hours Worked PDF

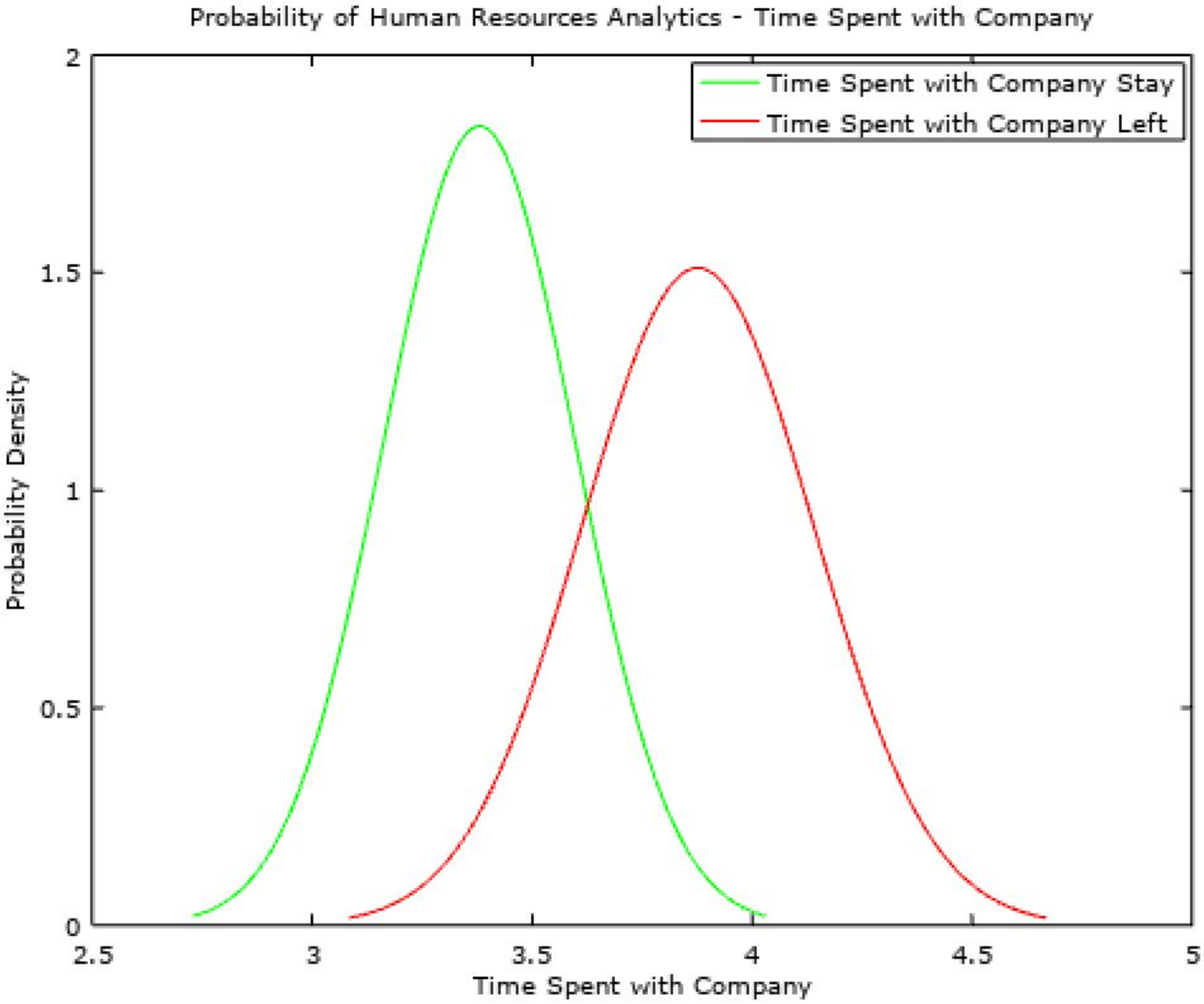


Figure 5: Time Employee Spent with Company PDF

Analysis Section

Note

Here is the order of the attributes from left to right (index 1 to 9)

Satisfaction level' 'Last eval' '# Projects' 'Avg Monthly Hours' 'Time Spent' 'Work Accident' 'Promotion' ‘Salary’ ‘Department’

Mean, standard deviation and covariance of the original

Mean of the original data set

```
 0.6128    0.7161    3.8031  201.0503    3.4982    0.1446    0.0213    0.5316    6.9363
```

Looking at the mean and comparing to the data set, I did not find any particular odd things about the result. They all match with the range of value that we saw in the original data set. For example, the first and second column has the range from 0-1.0. Here are some information that can be drawn from the mean

Most people are satisfied with their jobs (0.6). And most people also did good on their jobs as well (0.7). On average, each person did a total of 4 projects and they spend roughly 200 hours each month

Standard Deviation of the original Data set

```
 0.2486    0.1712    1.2326   49.9431    1.4601    0.3517    0.1443    0.2124    2.7473
```

Looking at the standard deviation, I also did not find anything odd about the result as well. The remaining is also similar to the mean where the range of value match with what we saw in the data set

Covariance matrix of the original data set

```
covariance =

   1.0e+03 *

    0.0001    0.0000   -0.0000   -0.0002   -0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0001    0.0029    0.0000   -0.0000   -0.0000   -0.0000    0.0000
   -0.0000    0.0001    0.0015    0.0257    0.0004   -0.0000   -0.0000   -0.0000    0.0001
   -0.0002    0.0029    0.0257    2.4943    0.0093   -0.0002   -0.0000   -0.0000    0.0011
   -0.0000    0.0000    0.0004    0.0093    0.0021    0.0000    0.0000    0.0000   -0.0001
```

```
    0.0000   -0.0000   -0.0000   -0.0002    0.0000    0.0001    0.0000    0.0000    0.0000
    0.0000   -0.0000   -0.0000   -0.0000    0.0000    0.0000    0.0000    0.0000   -0.0000
    0.0000   -0.0000   -0.0000   -0.0000    0.0000    0.0000    0.0000    0.0000   -0.0000
    0.0000    0.0000    0.0001    0.0011   -0.0001    0.0000   -0.0000   -0.0000    0.0075
```

Looking at this matrix, we can conclude some of the following:

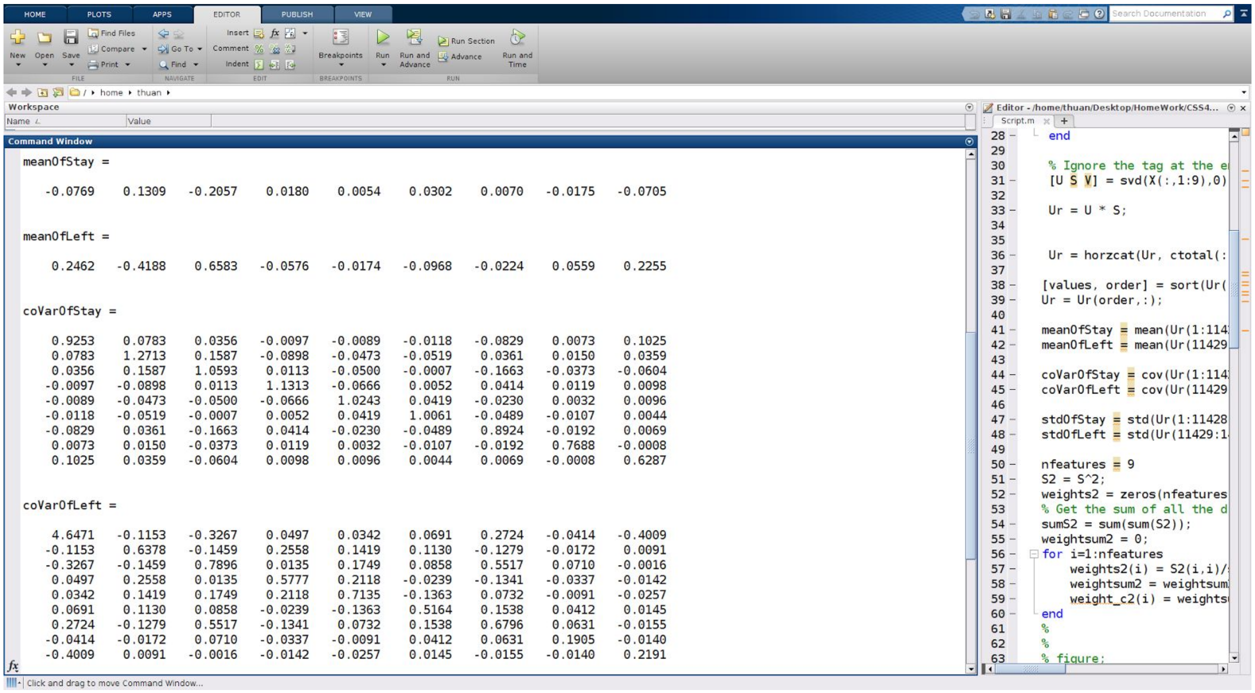
- Satisfaction level and Last Eval has no relationship (since the coefficient is 0)
- Satisfaction level and # Projects has no relationship
- Satisfaction level and Time Spent has no relationship
- Satisfaction level and Work Accident has no relationship
- Satisfaction level and Promotion has no relationship
- Last Eval and Time Spent has no relationship
- Last Eval and Work Accident has no relationship
- Last Eval and Promotion has no relationship
- # Projects and Work Accident has no relationship
- # Projects and Promotion has no relationship
- Avg Monthly Hours and Promotion has no relationship
- Time Spent has no relationship with Work Accident and Promotion

We can also look for places where the coefficient is positive that indicate there is a positive relationship between those two features. And negative relationship when there is a negative coefficient .

Mean ,standard deviation and covariance in the new space

All of the results generated was by using methods from others assignment like PCA and Homework 3. We just changed the data set and the index of the features. Like normalizing the original data set first before using svd

Mean and Covariance of the new space



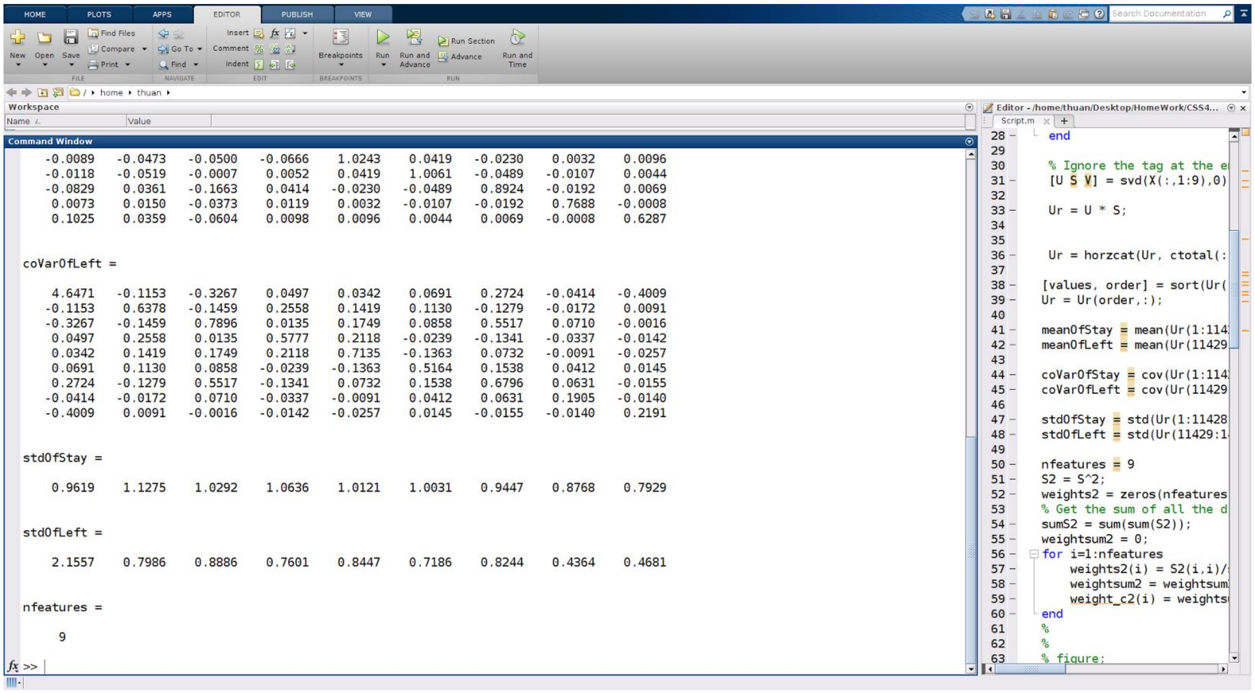
```
meanOfStay =

   -0.0769    0.1309   -0.2057    0.0180    0.0054    0.0302    0.0070   -0.0175   -0.0705


meanOfLeft =

    0.2462   -0.4188    0.6583   -0.0576   -0.0174   -0.0968   -0.0224    0.0559    0.2255


coVarOfStay =

    0.9253    0.0783    0.0356   -0.0097   -0.0089   -0.0118   -0.0829    0.0073    0.1025
    0.0783    1.2713    0.1587   -0.0898   -0.0473   -0.0519    0.0361    0.0150    0.0359
    0.0356    0.1587    1.0593    0.0113   -0.0500   -0.0007   -0.1663   -0.0373   -0.0604
   -0.0097   -0.0898    0.0113    1.1313   -0.0666    0.0052    0.0414    0.0119    0.0098
   -0.0089   -0.0473   -0.0500   -0.0666    1.0243    0.0419   -0.0230    0.0032    0.0096
   -0.0118   -0.0519   -0.0007    0.0052    0.0419    1.0061   -0.0489   -0.0107    0.0044
   -0.0829    0.0361   -0.1663    0.0414   -0.0230   -0.0489    0.8924   -0.0192    0.0069
    0.0073    0.0150   -0.0373    0.0119    0.0032   -0.0107   -0.0192    0.7688   -0.0008
    0.1025    0.0359   -0.0604    0.0098    0.0096    0.0044    0.0069   -0.0008    0.6287


coVarOfLeft =

    4.6471   -0.1153   -0.3267    0.0497    0.0342    0.0691    0.2724   -0.0414   -0.4009
   -0.1153    0.6378   -0.1459    0.2558    0.1419    0.1130   -0.1279   -0.0172    0.0091
   -0.3267   -0.1459    0.7896    0.0135    0.1749    0.0858    0.5517    0.0710   -0.0016
    0.0497    0.2558    0.0135    0.5777    0.2118   -0.0239   -0.1341   -0.0337   -0.0142
    0.0342    0.1419    0.1749    0.2118    0.7135   -0.1363    0.0732   -0.0091   -0.0257
    0.0691    0.1130    0.0858   -0.0239   -0.1363    0.5164    0.1538    0.0412    0.0145
    0.2724   -0.1279    0.5517   -0.1341    0.0732    0.1538    0.6796    0.0631   -0.0155
   -0.0414   -0.0172    0.0710   -0.0337   -0.0091    0.0412    0.0631    0.1905   -0.0140
   -0.4009    0.0091   -0.0016   -0.0142   -0.0257    0.0145   -0.0155   -0.0140    0.2191
```

```
  end

   % Ignore the tag at the e
   [U S V] = svd(X(:,1:9),0)

   Ur = U * S;


   Ur = horzcat(Ur, ctotal(:

  [values, order] = sort(Ur(
  Ur = Ur(order,:);

  meanOfStay = mean(Ur(1:114
  meanOfLeft = mean(Ur(11429

  coVarOfStay = cov(Ur(1:114
  coVarOfLeft = cov(Ur(11429

  stdOfStay = std(Ur(1:11428
  stdOfLeft = std(Ur(11429:1

  nfeatures = 9
  S2 = S^2;
  weights2 = zeros(nfeatures
  % Get the sum of all the d
  sumS2 = sum(sum(S2));
  weightsum2 = 0;
  for i=1:nfeatures
      weights2(i) = S2(i,i)/
      weightsum2 = weightsum
      weight_c2(i) = weights
  end
  %
  %
  % figure;
```

Figure 6: Mean and Covariance of the new space

Looking at the mean and covariance in the new space, I can see that all of them are now normalized. All of the value of the means are now in the range +-1. Furthermore, the covariance matrix also shows that all of the new features are now correlated. Which make sense since each new axis in the new space is comprised of features in the old space

Standard Deviation of the new space



```
   -0.0089   -0.0473   -0.0500   -0.0666    1.0243    0.0419   -0.0230    0.0032    0.0096
   -0.0118   -0.0519   -0.0007    0.0052    0.0419    1.0061   -0.0489   -0.0107    0.0044
   -0.0829    0.0361   -0.1663    0.0414   -0.0230   -0.0489    0.8924   -0.0192    0.0069
    0.0073    0.0150   -0.0373    0.0119    0.0032   -0.0107   -0.0192    0.7688   -0.0008
    0.1025    0.0359   -0.0604    0.0098    0.0096    0.0044    0.0069   -0.0008    0.6287


coVarOfLeft =

    4.6471   -0.1153   -0.3267    0.0497    0.0342    0.0691    0.2724   -0.0414   -0.4009
   -0.1153    0.6378   -0.1459    0.2558    0.1419    0.1130   -0.1279   -0.0172    0.0091
   -0.3267   -0.1459    0.7896    0.0135    0.1749    0.0858    0.5517    0.0710   -0.0016
    0.0497    0.2558    0.0135    0.5777    0.2118   -0.0239   -0.1341   -0.0337   -0.0142
    0.0342    0.1419    0.1749    0.2118    0.7135   -0.1363    0.0732   -0.0091   -0.0257
    0.0691    0.1130    0.0858   -0.0239   -0.1363    0.5164    0.1538    0.0412    0.0145
    0.2724   -0.1279    0.5517   -0.1341    0.0732    0.1538    0.6796    0.0631   -0.0155
   -0.0414   -0.0172    0.0710   -0.0337   -0.0091    0.0412    0.0631    0.1905   -0.0140
   -0.4009    0.0091   -0.0016   -0.0142   -0.0257    0.0145   -0.0155   -0.0140    0.2191


stdOfStay =

    0.9619    1.1275    1.0292    1.0636    1.0121    1.0031    0.9447    0.8768    0.7929


stdOfLeft =

    2.1557    0.7986    0.8886    0.7601    0.8447    0.7186    0.8244    0.4364    0.4681


nfeatures =

     9

fx >>
```

```
    end

  % Ignore the tag at the e
  [U S V] = svd(X(:,1:9),0)

  Ur = U * S;


  Ur = horzcat(Ur, ctotal(:

[values, order] = sort(Ur(
Ur = Ur(order,:);

meanOfStay = mean(Ur(1:114
meanOfLeft = mean(Ur(11429

coVarOfStay = cov(Ur(1:114
coVarOfLeft = cov(Ur(11429

stdOfStay = std(Ur(1:11428
stdOfLeft = std(Ur(11429:1

nfeatures = 9
S2 = S^2;
weights2 = zeros(nfeatures
% Get the sum of all the d
sumS2 = sum(sum(S2));
weightsum2 = 0;
for i=1:nfeatures
    weights2(i) = S2(i,i)/
    weightsum2 = weightsum
    weight_c2(i) = weights
end
%
%
% figure;
```

Figure 7: Standard deviation of the new space

The new standard deviation is also like the new mean in which all of its value are now within a range to each other. This is good because this suggest that there is no outliners in the new space

In addition, the ratio of the standard deviation and the new mean for each class also suggest that none of the features are useless. Because none of the results are 0. Suggesting that there are variance within the data

Scree Plots

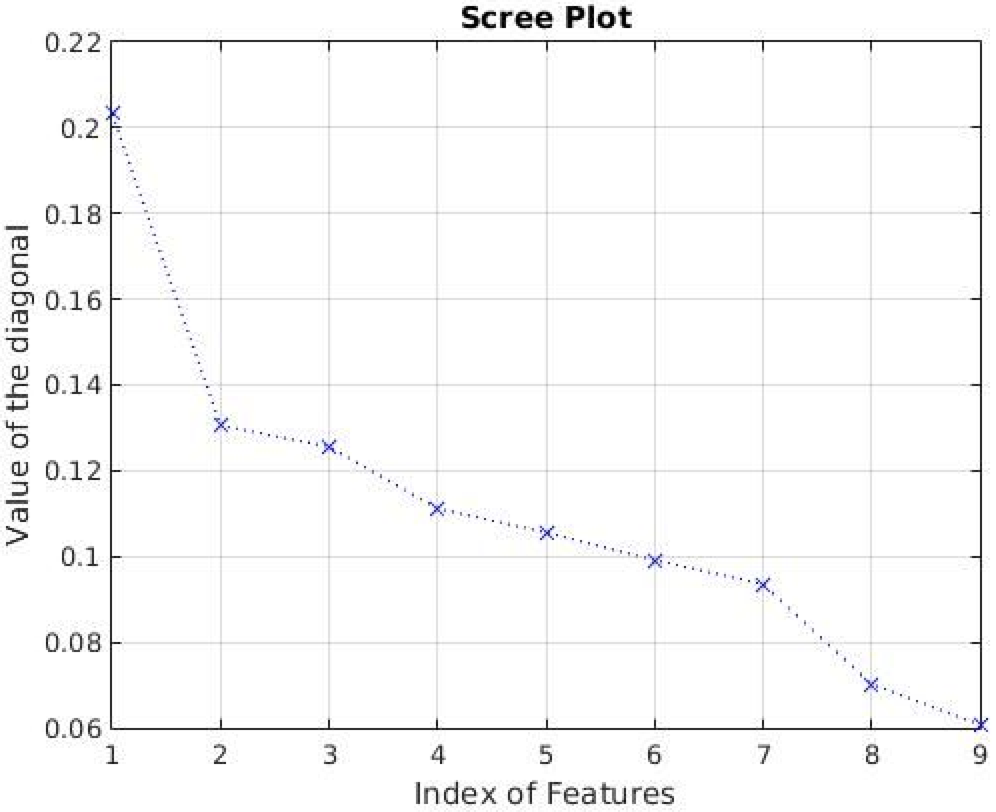


Figure 8: Scree Plot of the new space

Looking at the scree plot, we can determine that we only need the first 7 features because after the 7th feature, the 8th and 9th feature does not capture enough information (8th feature capture less than 0.08 and 9th feature capture around 0.06). Note that the features are in the new space

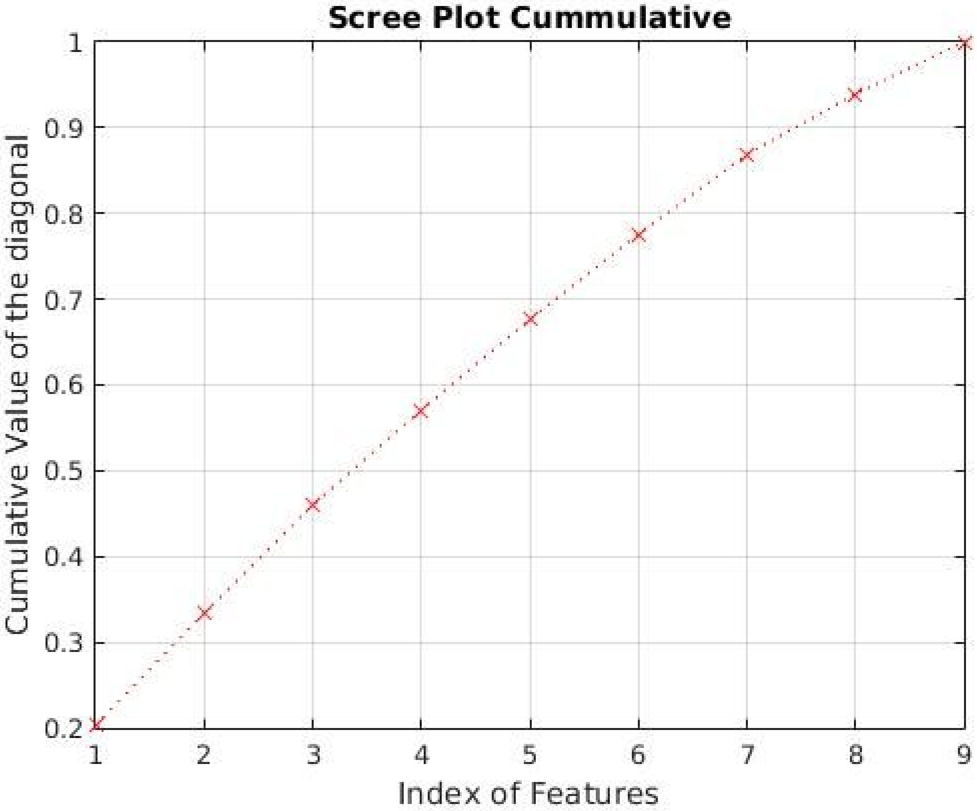


Figure 9: Cumulative Scree Plot

The same conclusion can also be made from looking at the cumulative scree plot where only the first 7 features are needed. At the 7th feature, we have captured around 87% percent of information already and we should stop here. Adding more features will increase the percent that we capture the information, but it will lead to overfitting

Loading Vectors

Below are the illustration of the loading vector of the new space. In my opinion, only loading vectors 1 to 7 are important because of their value in the scree plot

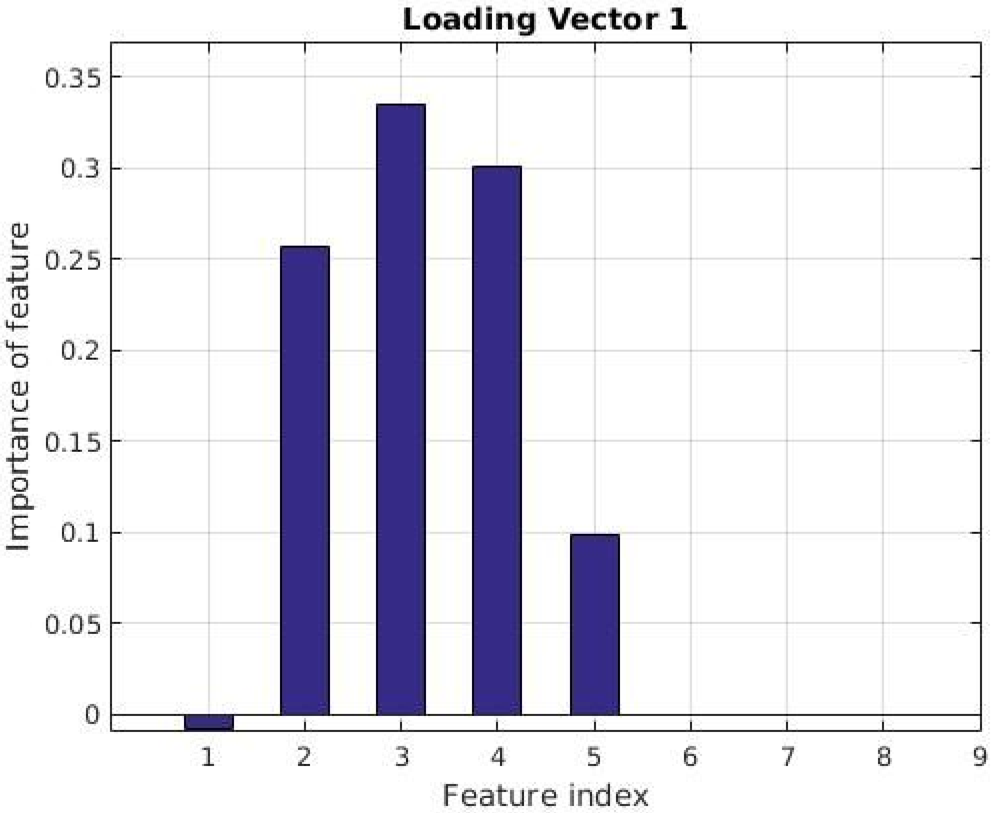


Figure 10: Loading Vector 1

Loading vector 1 suggest that the feature at index 1,6,7,8,9 in the old space does not contribute much to the first loading vector in the new space. All of the features that contribute to the new space are 2,3,4,5 and they all have positive impact.

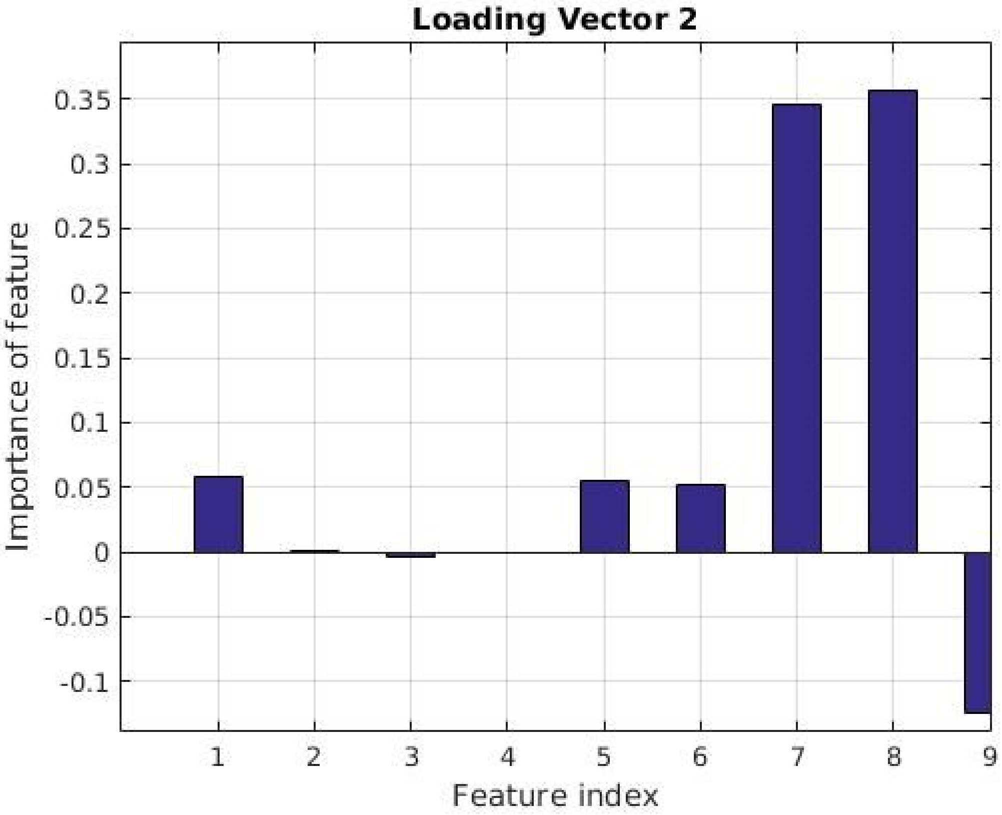


Figure 11: Loading vector 2

Looking at the loading vector 2, only feature at index 7 and 8 contribute to this loading vector(positive) whereas others do not contribute much like feature at index 1,5 and 6 and feature at index 9 (negatively). The feature as index 2,3,4 do not contribute to the loading vector

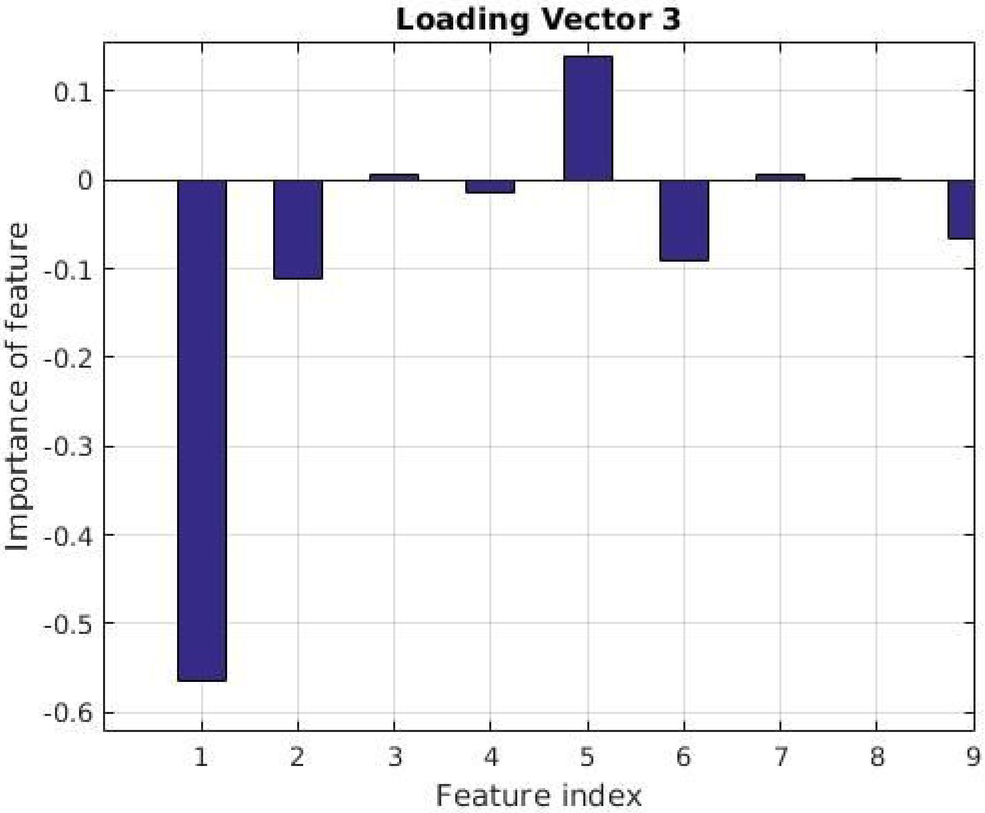


Figure 12: Loading Vector 3

Looking at loading vector 3, we can see that feature 3,4,7,8 do not contribute to the loading vector. Index 1 contribute the most (negatively), And index 2 and 6 seems to be correlated (having the same contribution) since their height are roughly the same

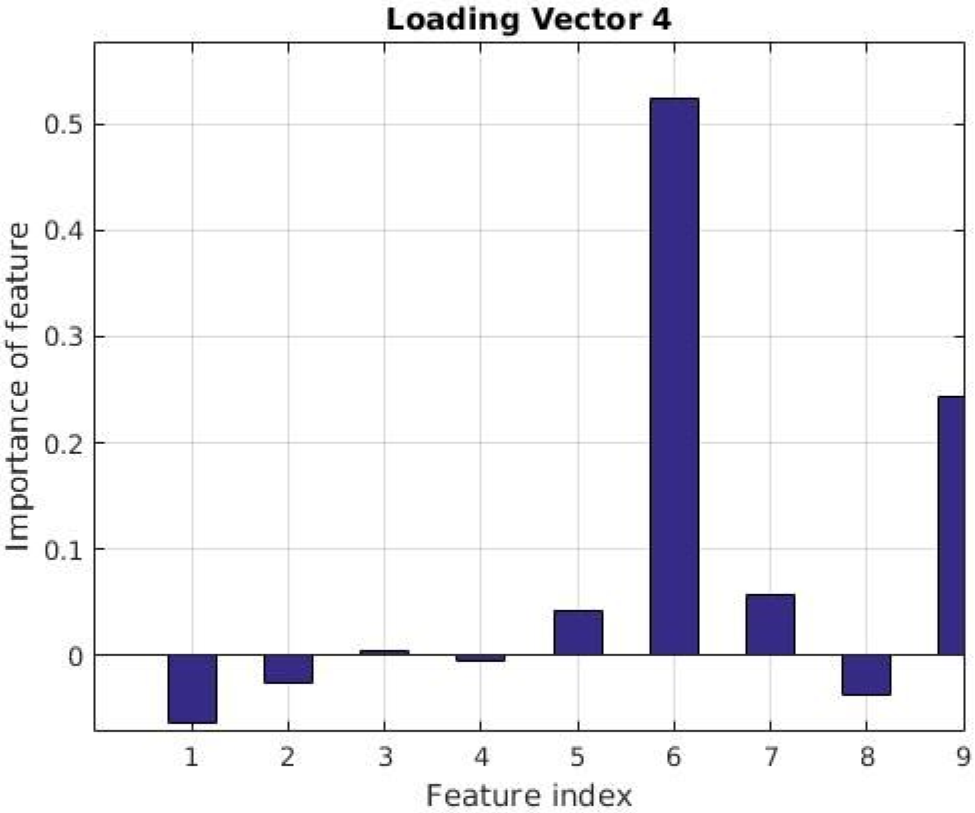


Figure 13: Loading Vector 4

Loading vector 4 shows that features at index 6 and 9 contribute the most to the loading vector. The remaining features contribute a tiny portion of the entire loading vector (<0.1 importance)

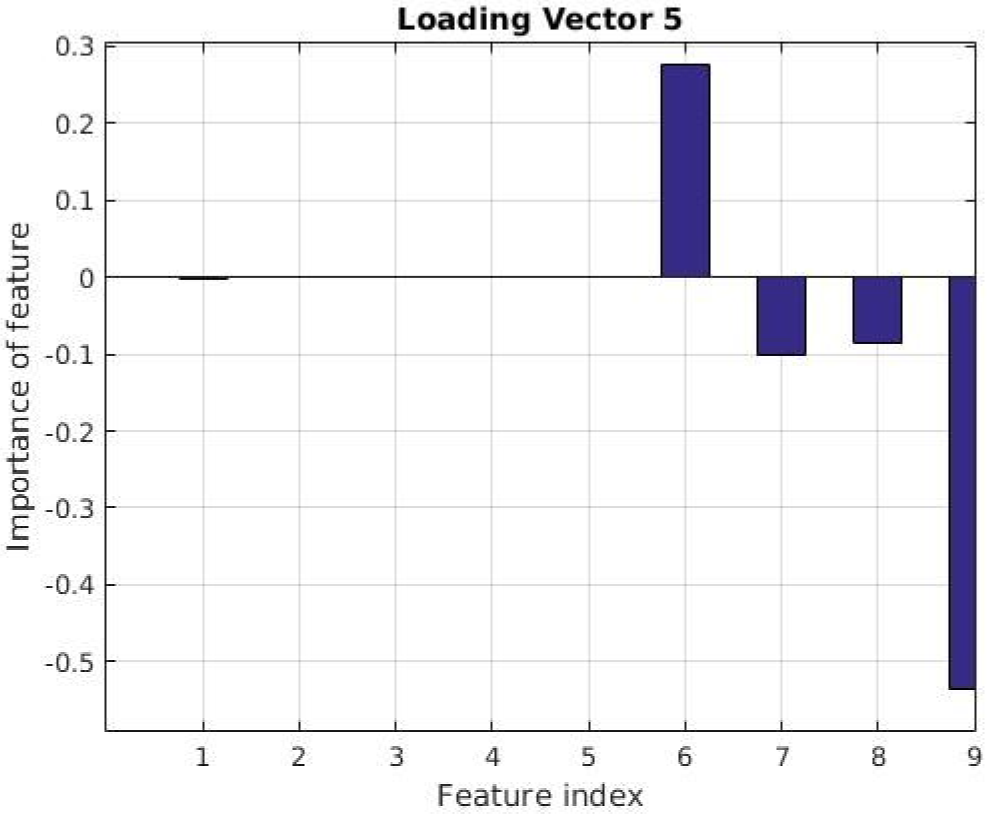


Figure 14: Loading Vector 5

Loading vector 5 shows that features 1 to 5 in the old space did not have any contribution at all. The most contribution is the feature 9 and feature 6 (negatively and positively). Feature 7 and 8 have roughly the same height which might indicate correlation

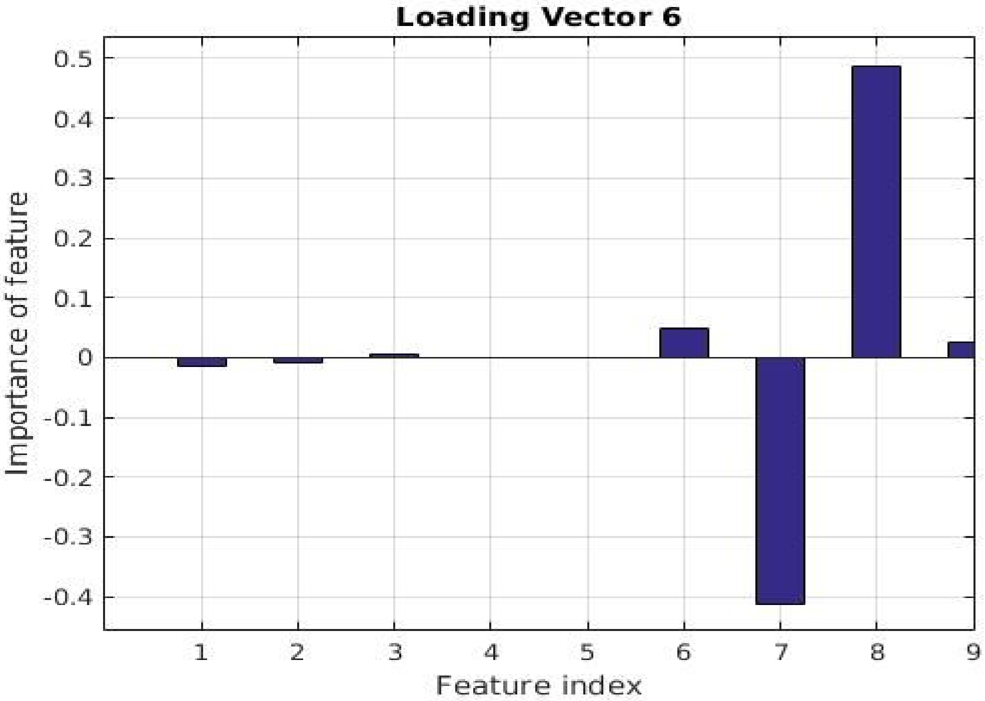


Figure 15:

Loading Vector 6

Loading vector 6 shows that feature 1,2,3,4,5 do not contribute anything to the new space. Only the feature at index 7 and 8 contribute to the new space. The remaining two features 6 and 9 contribute a little

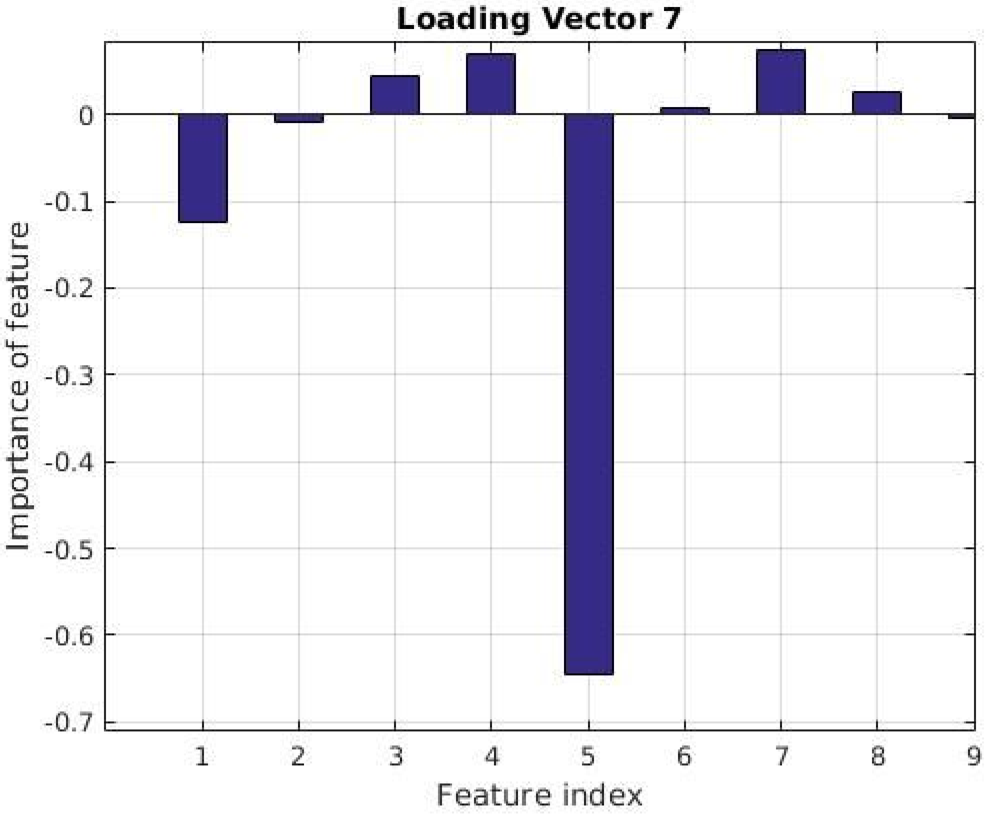


Figure 16: Loading Vector 7

Loading vector 7 this time shows that all feature contributes to the vector. However, only feature 5 have the most significant impact on the loading vector whereas others only have minor impact (Most of the remaining features do not have more than 0.1 importance)

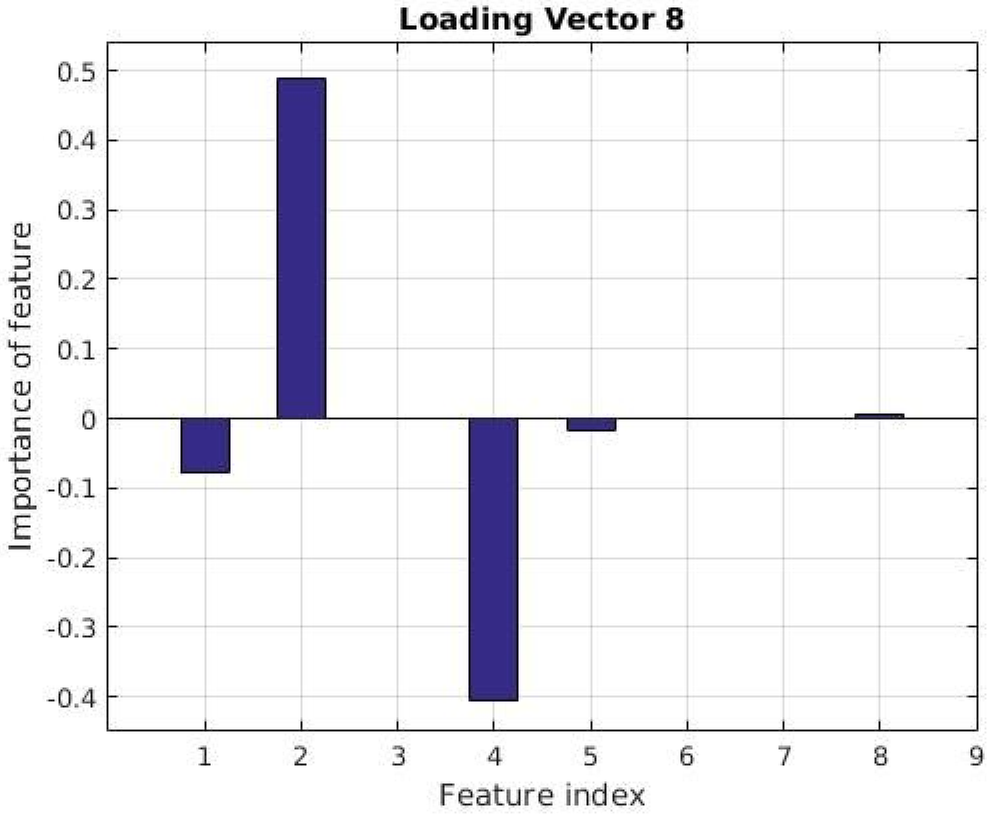


Figure 17: Loading Vector 8

Loading vector 8 shows that feature at index 2 and 4 contribute the most to the loading vector. Whereas feature 1,5 and 8 contribute a little and feature 3,6,7,9 do not contribute at all

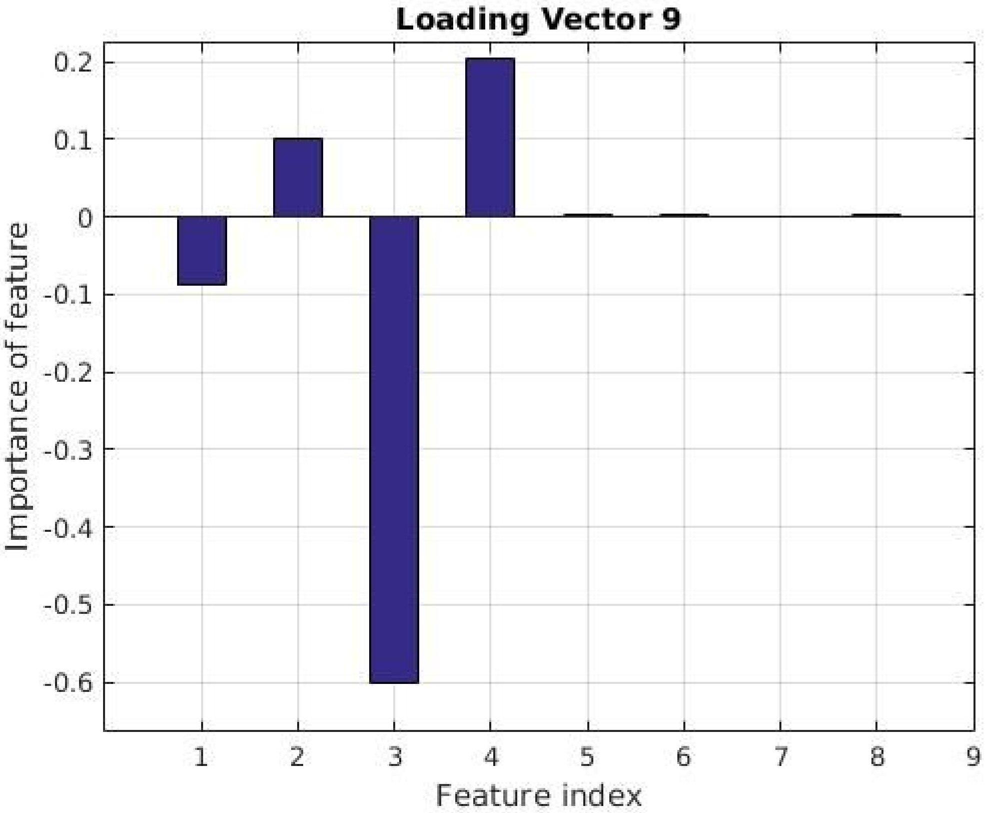


Figure 18: Loading Vector 9

Loading vector 9 shows that feature at index 3 contribute the most. The second contributing feature is feature at index 4. Feature at index 1 and 2 contribute a little whereas the feature at index 5,6,7,8,9 contribute too little or do not contribute at all

Looking at the heights of each index in each loading vectors, I can see that they all contribute
differently to the new axis of each loading vectors. Depending on the loading vectors, some of the indexes contribute more than the other.
I also did not see any consistent pattern of the height of the indexes in these loading vectors. This indicate that there is no correlation between any indexes so that we can eliminate one and only use the other. Thus, all the indexes are necessary

3D Scatter Plots

The following scatter plot is of the original space using the features which appeared to have the widest breadth of quantitative data. An employee's satisfaction level, the number of projects they were working on, and the average monthly hours they spent at work all appeared to be better data to use than some of the other features which consisted of mainly 1’s or 0’s.

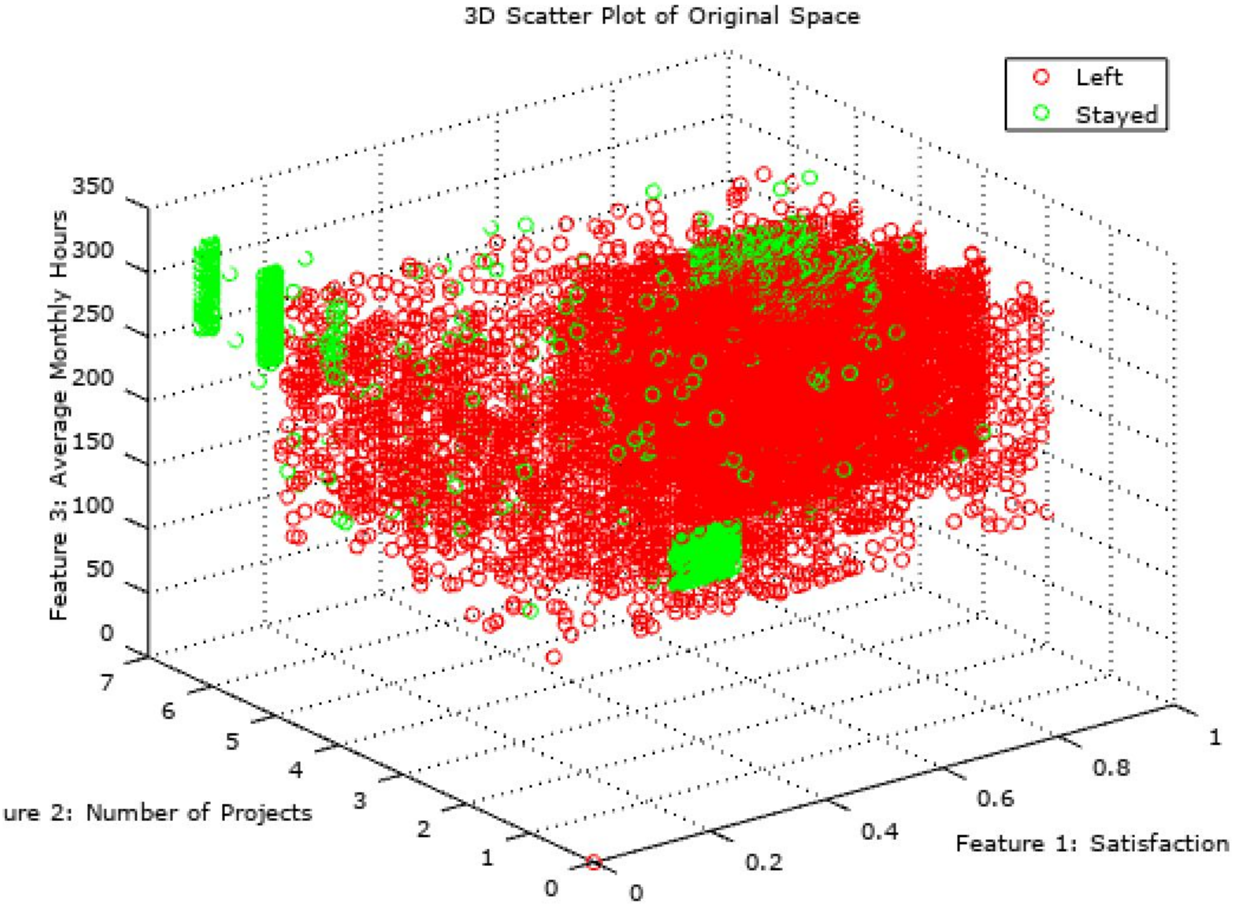


Figure 19: 3D Scatter Plot of Original Space

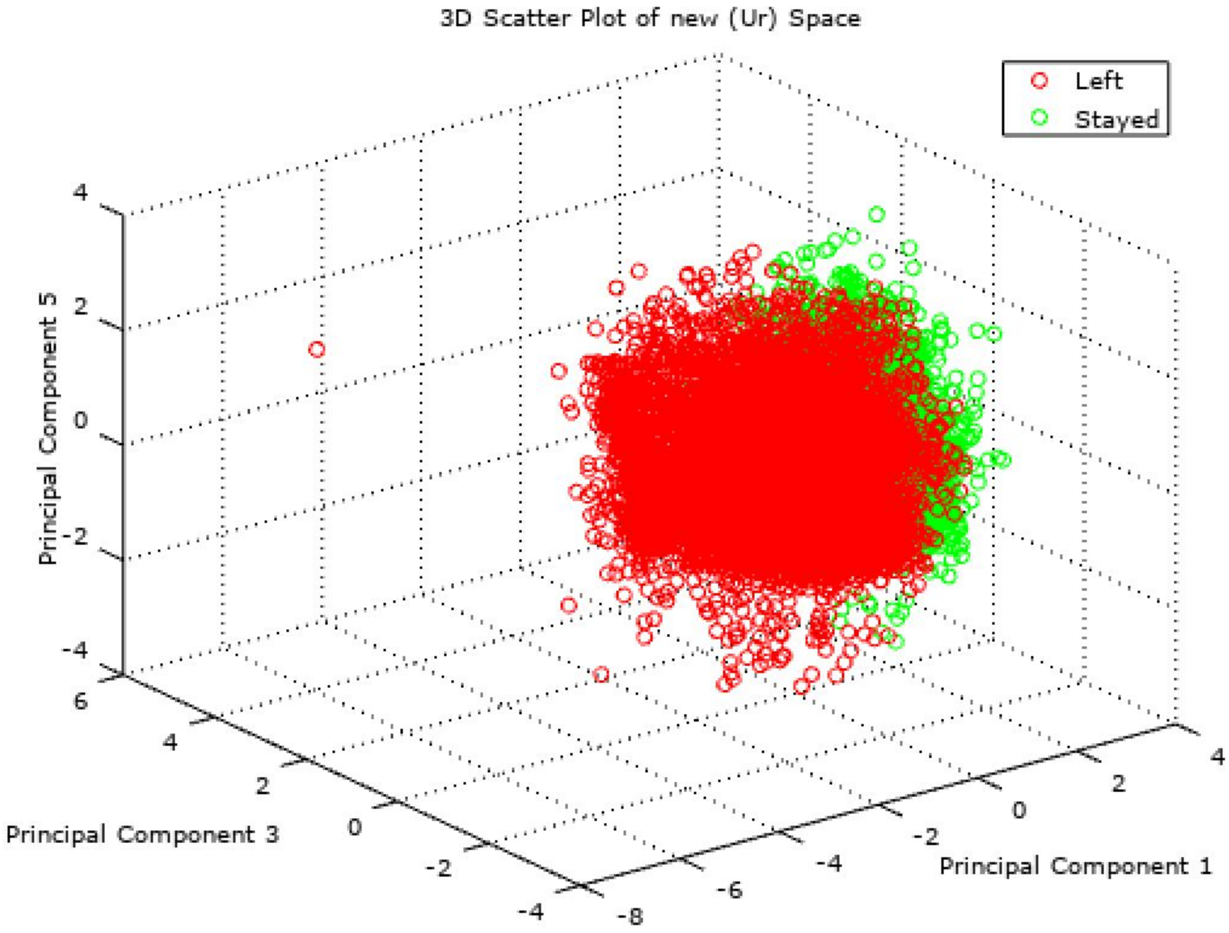


Figure 20: 3D Scatter Plot of New Space

By running our data through PCA, we are able to plot this new space (Figure 15) which shows a much better separation of data. This means that our new components should work well for use in a classifier. The Principal Component Analysis is very useful in the way it takes bits and pieces of the original observations and gives you back a new space with better correlated grouping.

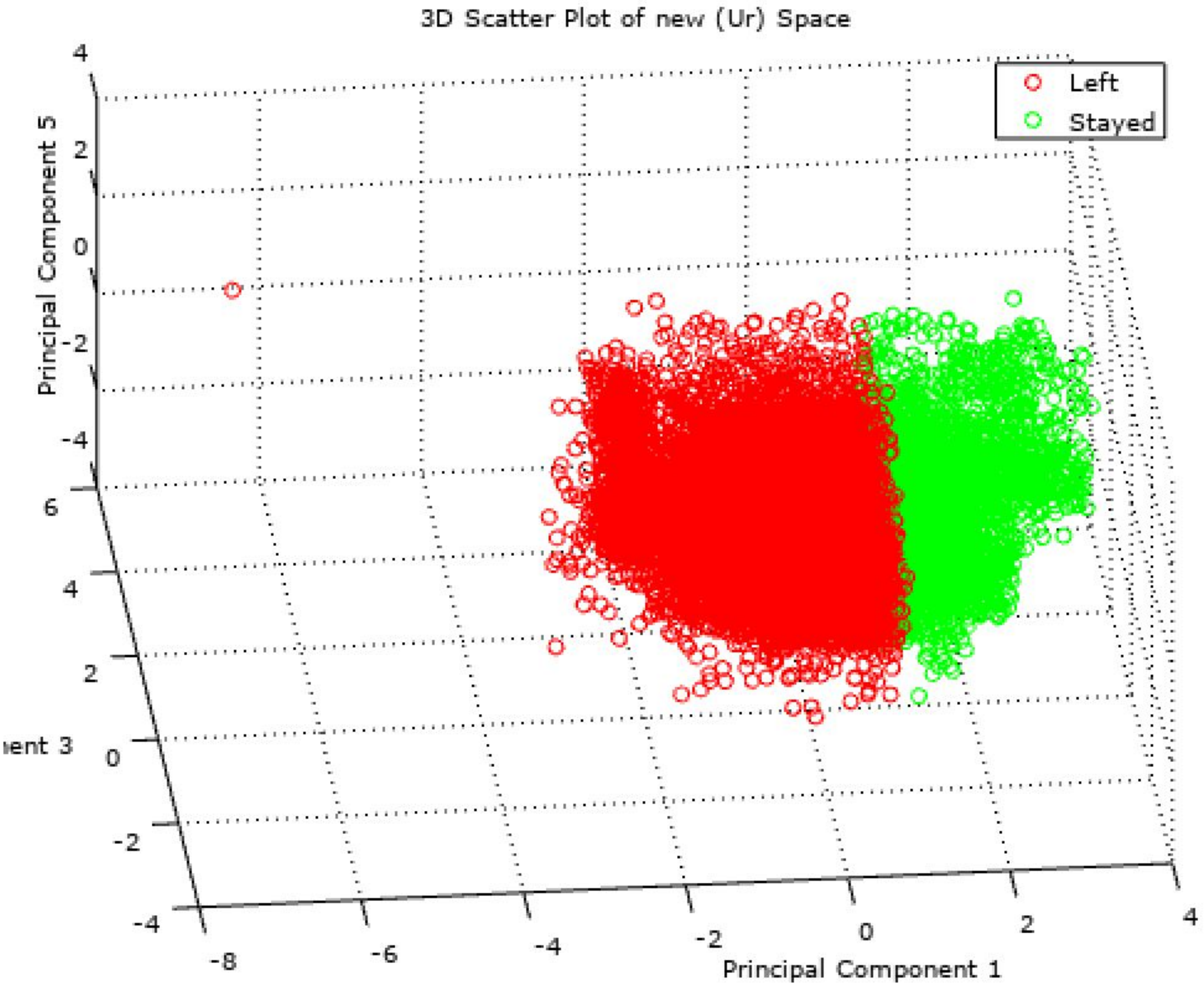


Figure 21: 3D plot of New Space rotated to show separation

Here, Figure 21 shows the same scatter plot as in Figure 20. This plot has been rotated to get a better view of the separation.

You can see that while they are grouped quite well, there is still some overlap right in the center. This indicates the area where error may occur when using these components in our classifier.

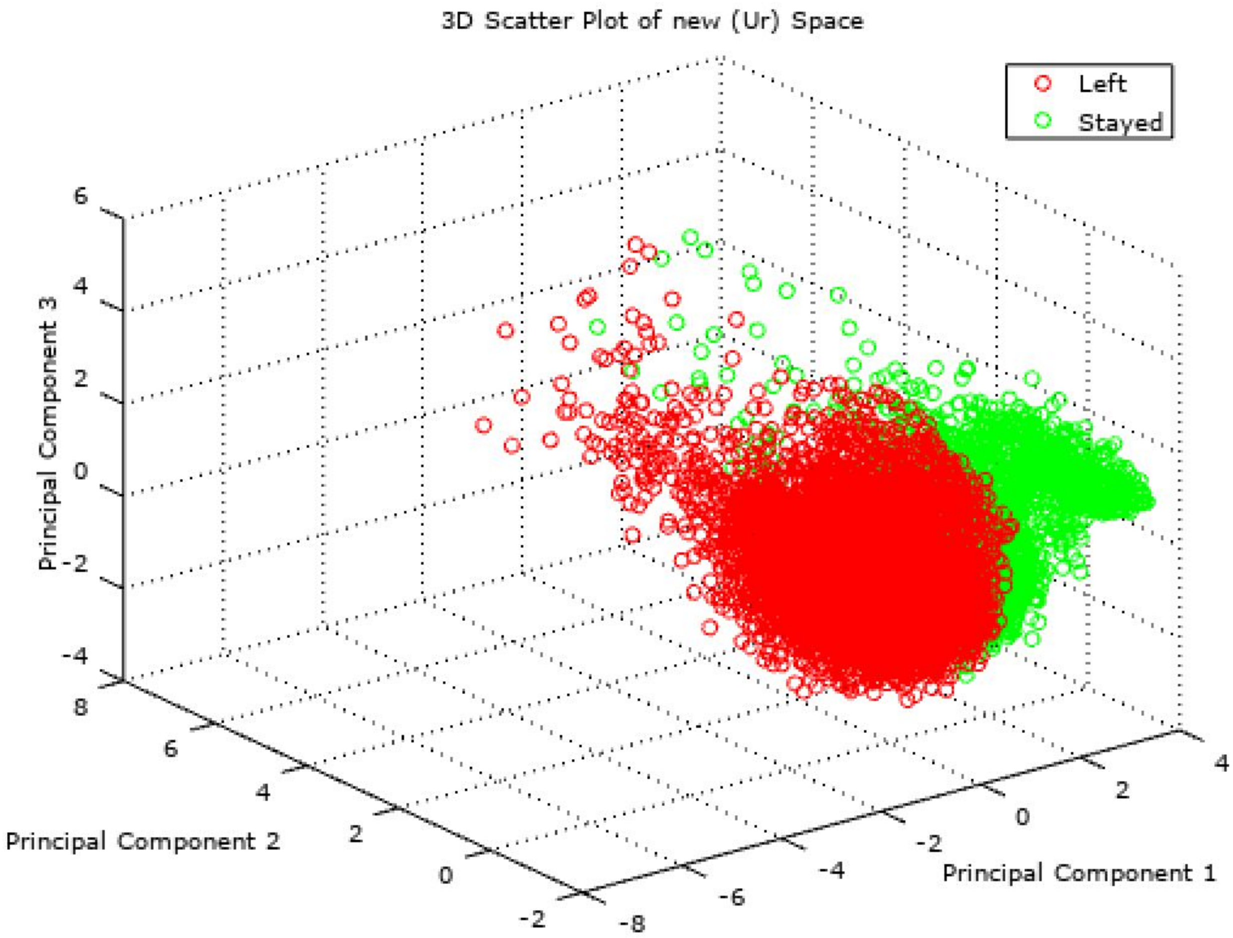


Figure 22: 3D Scatter Plot using Principal Components 1, 2, and 3 from New (Ur) Space

We also wanted to see how the grouping would look by using the first three principal components from the new space, because these are the principal components which are multiplied by the highest S values. However, this doesn't mean they produce the best results. As you can see, the points are still separated, but they spread out more than in Figure 20 which uses the principal components we determined to have the highest values from our loading vectors.

Below, Figure 23 shows the same vector as Figure 22, but rotated to show the grouping better.

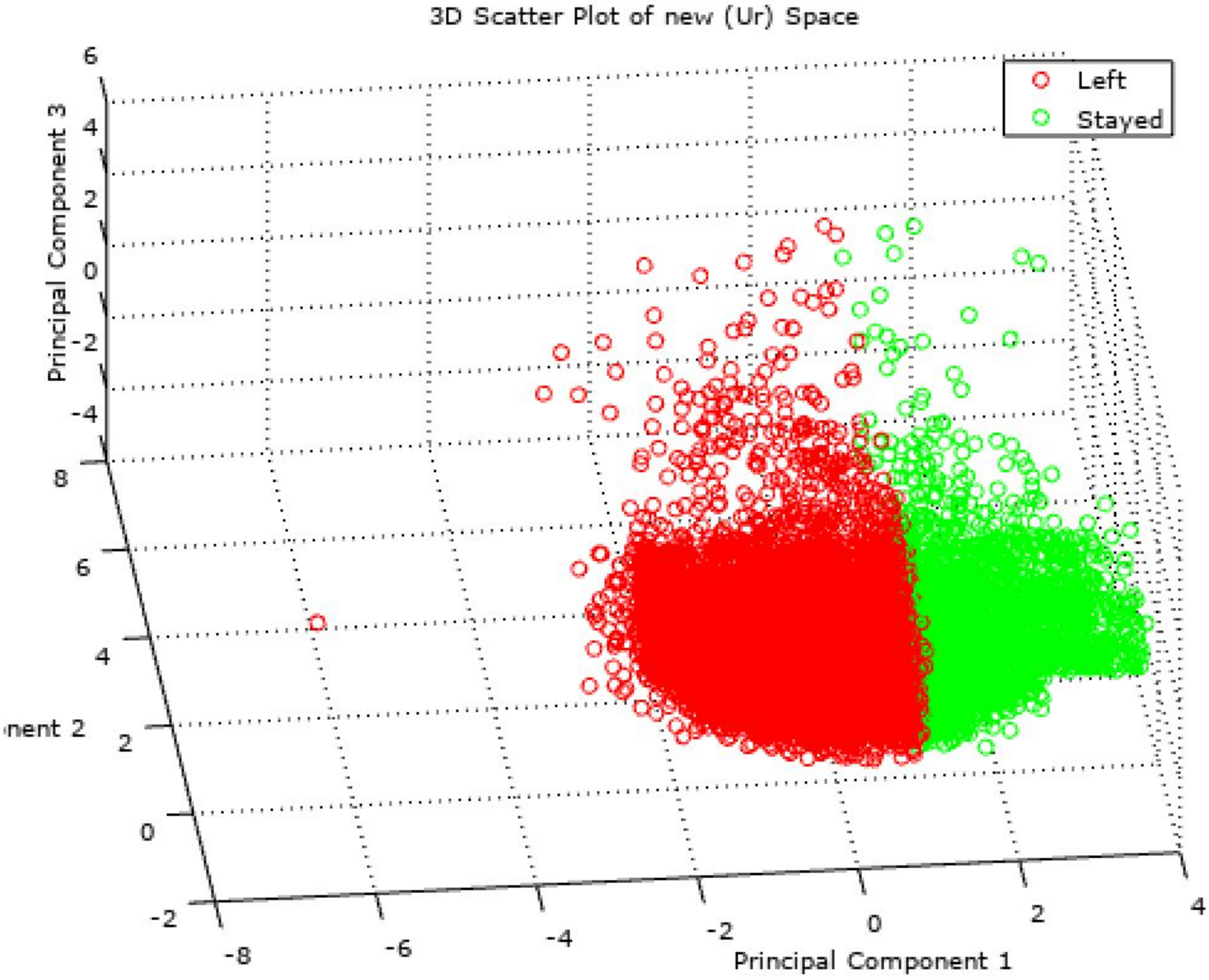


Figure 23: 3D Scatter plot of New Space (Ur) using Principal Components 1, 2, and 3 rotated to show detail.

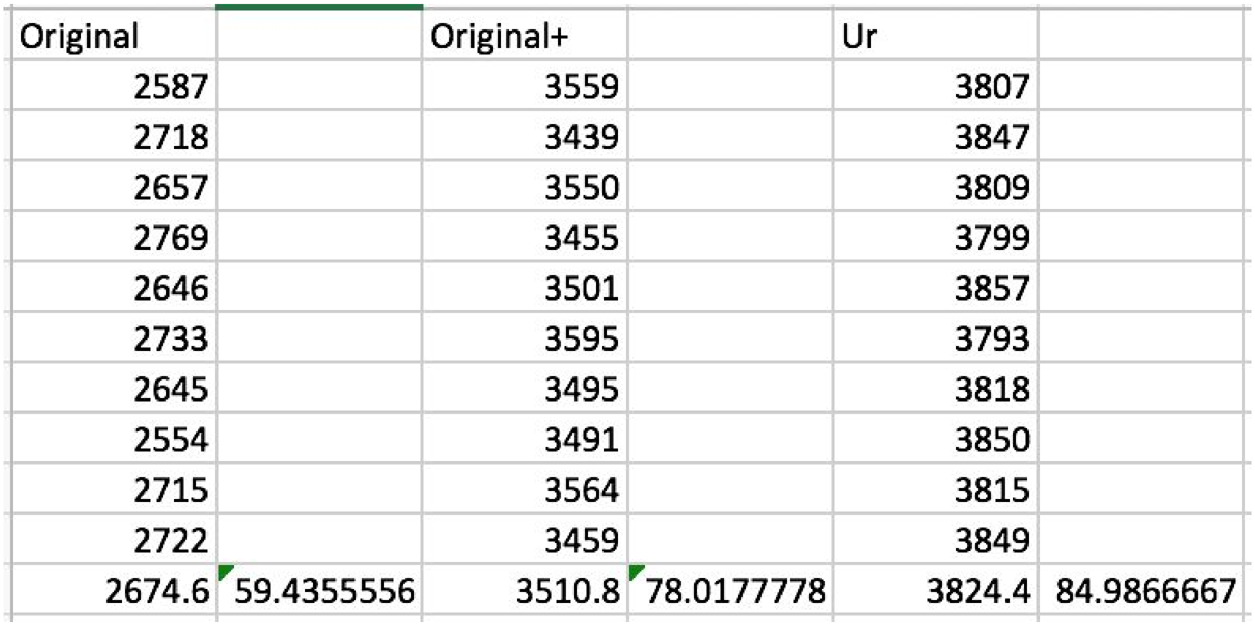
Classification

For classification, we decided to try using the KNN and Bayesian algorithms. We wanted to see how well the classifiers were able to perform using three data set spaces.

The first set we want to test with is the original dataset. This will give us a base score for the performance of the classifiers. The next space is our transformed Ur space that we created in our PCA. We expected the results to be much better than the original, because through PCA we now have a space which is better suited to determine one class from another based on observations. From the loading vectors created in our PCA, we found that there was a redundant feature in the original space that we would feel comfortable removing. That feature is a measure of the last time an employee received a performance review. We determined from the loading plots that between features 2, 3, and 4, that there was some redundancy happening. Out of those three features, “last evaluation” was the lowest performer. From this observation, we came to the conclusion that we should attempt to try the classification again on the original dataset, only without the “last evaluation” column of data. What we believed we would see is an increase in the correct classification results, but more importantly we wanted to see by how much it would raise the percentage of correct classifications.

Bayesian Classifier

When running a Naive Bayes Classifier on our three data sets we expected to see an increase in performance with the transformed dataset. What we didn’t know was how much better the original data set might perform once we removed the redundant feature of “Last Evaluation”. To get a clear picture, we decided to run the classifier 10 times, each time using a random test dataset that consists of 4500 samples, the rest of the set being used as training data for the classifier.



Original		Original+		Ur	
2587		3559		3807	
2718		3439		3847	
2657		3550		3809	
2769		3455		3799	
2646		3501		3857	
2733		3595		3793	
2645		3495		3818	
2554		3491		3850	
2715		3564		3815	
2722		3459		3849	
2674.6	59.4355556	3510.8	78.0177778	3824.4	84.9866667

Figure 24: Test Results for Bayes Classifier on the three data spaces; the Original, the Original minus the “Last Evaluation” column, and the transformed Ur space.

The original data set was our ground floor for expectations. After completing the ten tests and recording the results, the average of the performance was computed. Here we see an average of 59.4%. This is a low number, but to be expected with a regular, unprocessed data set.

The original data set minus the “Last Evaluation” column, which we found to be redundant in PCA, performs with an average of 78.0% correct classification rating. This is much higher than the original set, and is quite surprising that it would perform with a 18.6% increase. It really shows how valuable PCA can be. It could save a lot of time and computation, if this is all that you need to see an increase of in classification results. By simply leaving out this “Last Evaluation” column from any new data, the Naive Bayes Classifier would be able to perform with ~78.0% rating of if an employee would be on route to leave their job or not. This is pretty good, and may be all that’s needed for an employer to determine if they should be concerned about an employee or not. However, with a 22% error range, there’s still good chance of a misclassification.

With the Ur data set, we got a performance rating of 85.0%. This is really much better than the previous versions. So, we know that the singular value decomposition is working. We just need to be sure we are interpreting the results properly, and reducing the correct unwanted features. By reducing the correct features, we can see an increase in our classification results like when we removed “Last Evaluation”, because there is less noise for the classifier to deal with.

Although this turns out to be a reasonable solution for improving the results of even the simplest classifiers, we wanted to see how much better results we could get with a more complex classifier. The next section will discuss our findings when using the KNN classifier.

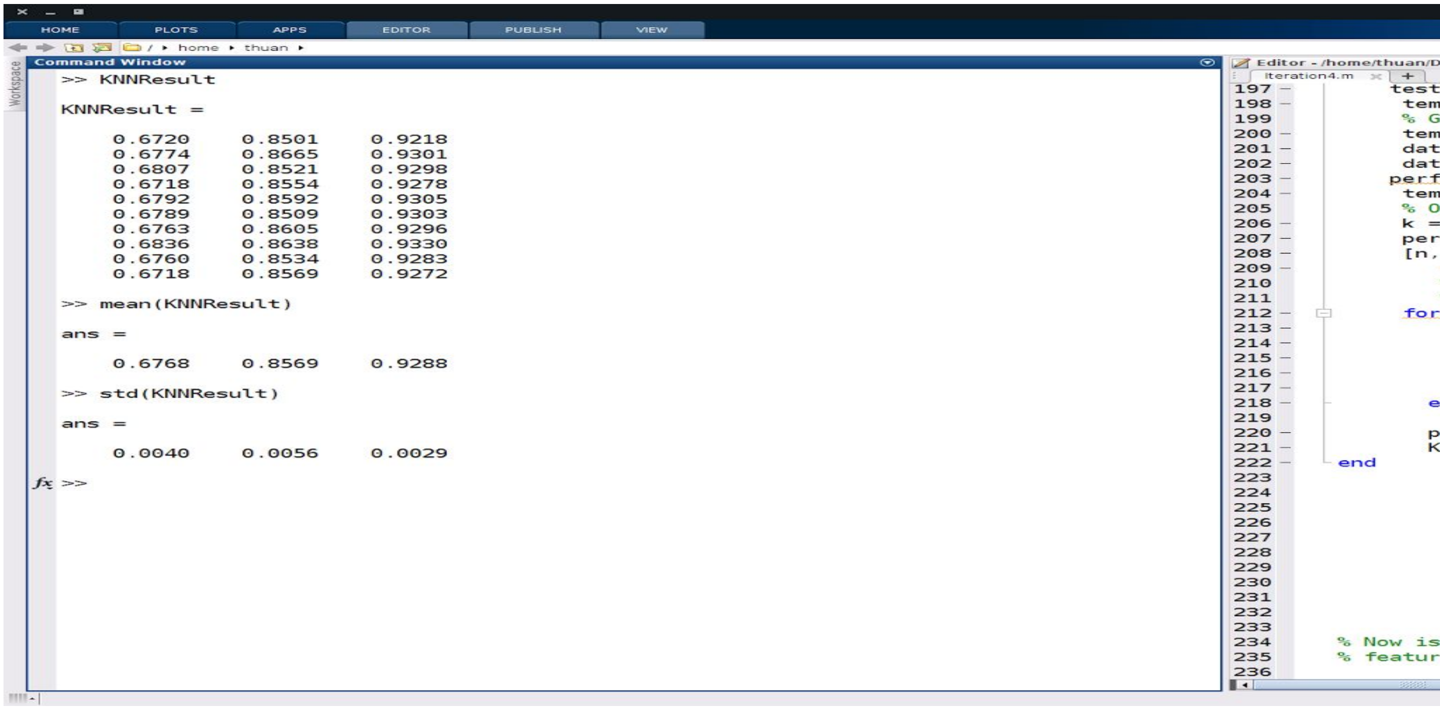
K-Nearest Neighbor

When running the KNN classifier, we can see that the performance improves a lot compared to the Naives Bayes Classifier. One reason could be because of the nature of our data set, which represents human’s behavior. The Bayes Classifier is a conditional probability model that depends on the probability of occurrences, and human behaviors do not have an exact probability model.

In that case, KNN is a better choice because it is a classifier that predicts based on similarities,which is more familiar with human behaviors. We also decided to split the data set into 70% training and 30% testing. Running the classification 10 times for each space (the Ur space, the original data set, the original data set after removing redundant feature). We also choose k = 150 as well.

Below is a figure that represent our result

Figure 25: Performance result of original data set, original data set with the removal of redundant feature and the new Ur space.



```
>> KNNResult

KNNResult =

    0.6720    0.8501    0.9218
    0.6774    0.8665    0.9301
    0.6807    0.8521    0.9298
    0.6718    0.8554    0.9278
    0.6792    0.8592    0.9305
    0.6789    0.8509    0.9303
    0.6763    0.8605    0.9296
    0.6836    0.8638    0.9330
    0.6760    0.8534    0.9283
    0.6718    0.8569    0.9272

>> mean(KNNResult)

ans =

    0.6768    0.8569    0.9288

>> std(KNNResult)

ans =

    0.0040    0.0056    0.0029

fx >>
```

```
      test
       tem
       % G
       tem
       dat
       dat
      perf
       tem
       % O
       k =
       per
       [n,


       for





         e

         p
         K
  end











  % Now is
  % featur
```

After calculating the performance using KNN, we found that the result is much better than the Bayes classifier. In the original data set, we can see that the average performance is only 67%, whereas the original data set with the removal of redundant feature has the average performance of 86%. And the Ur space have have average performance of 93%. This is a significant improvement compared to the Bayes Classifier where the performance are 59% , 78% and 85%

I also calculate the standard deviation in the 10 runs as well and the results shows that there are not much difference in the performance. Hence I can conclude that the result we got are reliable and do not change much given different run.

In addition, we also conclude the same thing with the Bayes Classifier in which the PCA Transformation has helped to increase the performance of classification. And by using the PCA to identify redundant feature, we were able to get rid of the “Last Evaluation” feature to increase the performance of the original data to 86%, an increase of nearly 20%

Calculating the optimal K

In the script we submitted, there is a section that we commented out that will plot the plot above. Basically , we are determining the best K value that give the highest performance

Figure 26: Plot that show the performance of using different K value (Google Drive is odd when adding picture, I can add the pictures with the submission of the script)

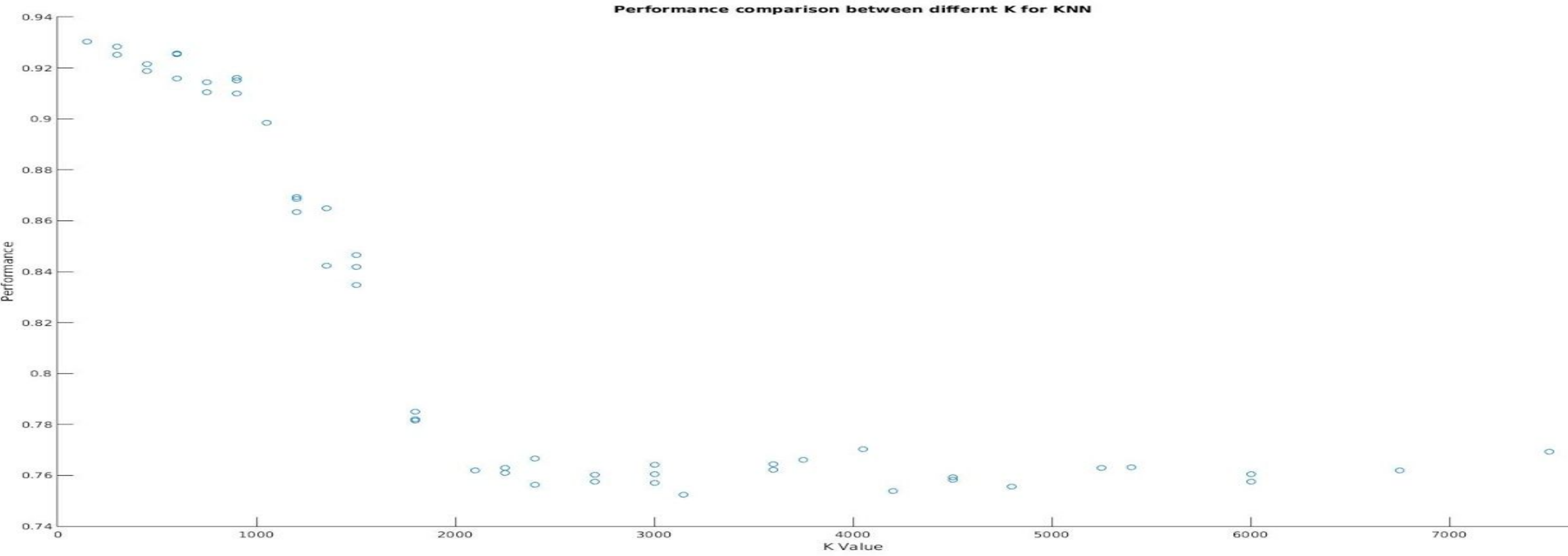


Figure 27: Numeric result of using different K and their performance

Before doing the KNN, we just picked a random number to be the K value. And ran the KNN using it. Even when receiving a high performance, we were still not satisfied and decided to use a more systematic approach in determining the K value. So we decided to create a test (toy) script that run through the KNN classification of the Ur space and recorded the performance result of using different K. And use that K for the classification above.

We were quite surprised that the data set only need 150 neighbors to classify new observations. We thought that it should be something like 5% or 10% of the entire data set (750 or 1500), When we think about it, 150 seems reasonable enough. Considering it like an actual human being, classifying a new observations by looking at 150 similar observations surrounding it should be plenty.

Team Assessment

- Thuan Tran: Both me and Chayse performed research on which dataset to use. After various discussions of the pros and cons, we decided to choose this data set. I also initially created the document and added initial draft ideas into it. For Iteration 2, I created the scatter plot matrix and calculated the covariance matrix. For Iteration 3, I did the first half of PCA which is the normalization, scree plots and loading vectors. I also did the analysis of the first half as well. For Iteration 4 I performed KNN Classifier on the data set

- Chayse Summers: Thuan has been an asset, very engaging and helpful. We've come to work well together and I look forward to seeing what we can discover with this data set. We each researched data sets and found that this one worked well for both of our interests. Thuan has been able to get the documents started and I have come through revised and discussed aspects with him working out a final document for submission. For Iteration 3, I was able to complete the 3D scatter plots and analysis of those plots by building off Thuan's work with normalizing the data, creating the scree plots, and loading vectors. For Iteration 4, I was able to help work on deciding what classifiers to use and which features should be left out. I helped put together the report and worked on determining the classification results from the Bayes classification section.

Measure of Success:

- Be able to **apply pattern recognition** to others problem.

- Be more **proficient in Statistics.**
- The Project need to have a high accuracy when we test it. In short, it need to **create reliable results**.
- Be able to **use different computation tools and be exposed to different library (Matlab, Octave) for analysis**.

References:

"Human Resources Analytics | Kaggle." *Human Resources Analytics* | *Kaggle*. Medium, n.d. Web. 02 Apr. 2017. https://www.kaggle.com/ludobenistant/hr-analytics

Duda, Richard O. Hart, Peter E. Stork, David G. "Pattern Classification". New York: Wiley-Interscience, 2001.