Multivariate Analysis of User Knowledge

Isabella Chittumuri
Professor Iordan Slavov
STAT 717: Multivariate Analysis
City University of New York, Hunter College

Abstract

In this project we study the statistical analysis of user knowledge to determine what methods of data reduction is better as well as what models produce the lowest misclassification.

1. Introduction

1.1 Data

This project used the "User Knowledge Modeling Data Set" created by H. T. Kahraman, S. Sagiroglu, I. Colak, donated in October, 2009 to University of California (UCI) Machine Learning Repository. The dataset contained real data about the students' knowledge status about the subject of Electrical DC Machines.

User modeling is a form of human-computer interaction where the computer customizes and adapts its system to the specific needs of a user. In this case, researchers used students'/users' model to customize subjects according to their knowledge. The goal of a user modeling system is to provide enough or suitable knowledge for students/users [2].

The dataset was available in excel format with three sheets: information, training data, and test data. The training data consisted of 258 observations, while the test data consisted of 145 observations. We combined both data sheets into one sheet with 403 observations.

The dataset was consistent in the measurement of 6 attributes and 403 observations. There were no missing values, which makes the data complete with equal probability. There were five independent attributes and one dependent attribute:

Independent Attributes:

1. STG (The degree of study time for goal object materials)
2. SCG (The degree of repetition number of user for goal object materials)
3. STR (The degree of study time of user for related objects with goal object)
4. LPR (The exam performance of user for related objects with goal object)
5. PEG (The exam performance of user for goal objects)

Dependent Attribute:

1. UNS (The knowledge level of user)

In this report, we explore and classify attributes that model the knowledge of users. All of the analysis was done using R 3.6.3.

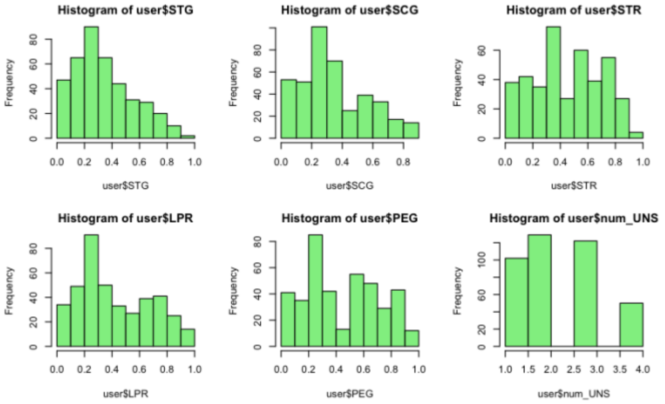
2. Exploratory Analysis

2.1 Data Observations

The first thing we did was call the summary function on the dataset. We saw that the five number summaries for all independent attributes are within the range of [0,10]. We also saw that the dependent attribute UNS contained knowledge levels observations of 102 High, 129 Low, 122 Middle, and 50 Very Low. We then created a new column vector called num_UNS, resulting in the numeric output of UNS where 1=High, 2=Low, 3=Middle, and 4=Very Low.

We created frequency histogram plots for all of the attributes. **Figure 1** depicts six frequency histograms. STG, SCG, LPR, PEG produced a right-skewed distribution. STR produced a normal distribution. UNS showed how many of each knowledge levels are in the dataset. We saw that the most frequent is Low and least frequent is Very Low.

Figure 1. Frequency of Attributes



2.2 Density Plots

Next we plotted the independent attributes against each other and found that there are linearly independent. Then we made density plots of each independent attribute in relation to the dependent attribute UNS.

Figure 2. Density plots of five attributes in reaction to one attribute

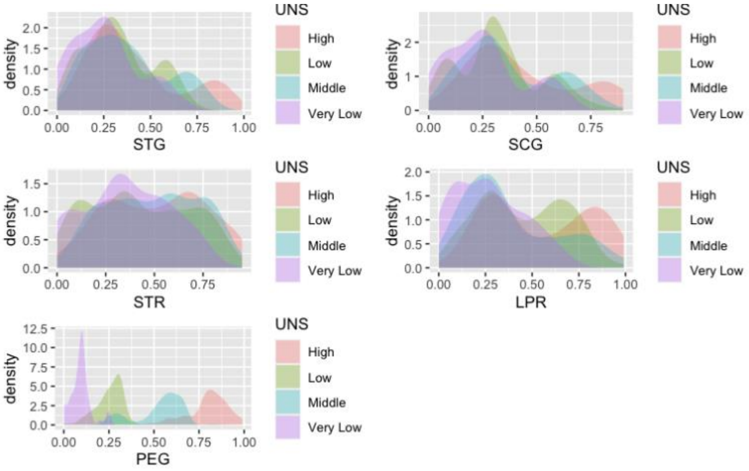


Figure 2 shows the each attribute's distribution of the four knowledge levels of UNS. Attributes STG, SCG, STR, and LPR had similar density distributions for each level. This showed no relationship between those four attributes and UNS. However, PEG showed a distinct relationship to UNS. PEG was the exam performance of user for goal objects. We saw that the lower a user performs on the exam, the lower their knowledge level is, and vice versa.

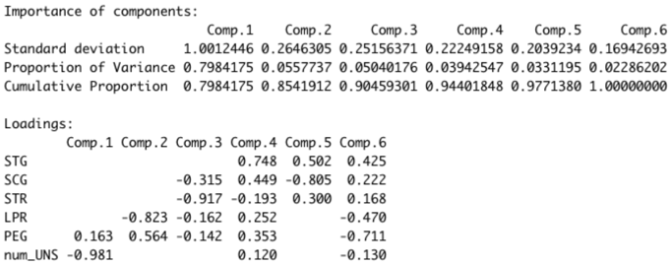
3. Principal Component Analysis

Principal Component Analysis (PCA) is a method used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set

3.1 Covariance

First, we computed the covariance of all six variables of the dataset. We used the covariance matrix to understand how the variables of the data set were varying from the mean with respect to each other, and to see if there is any relationship between them. After computing the PCA of the covariance matrix, we get six principal components. Principal components (PC) are new variables that are constructed as linear combinations of the initial variable. **Figure 3** depicts the importance of these components and their loadings.

Figure 3. Proportion of variance explained



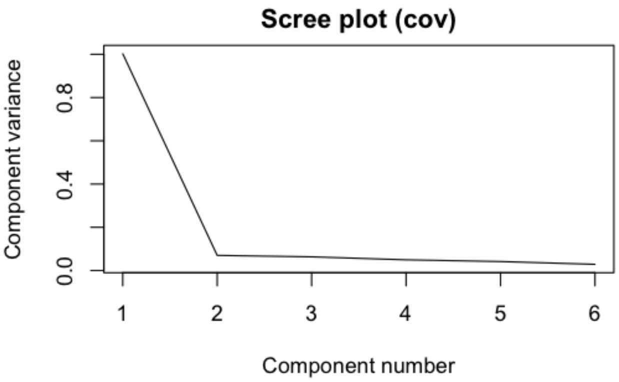
```
Importance of components:
                          Comp.1    Comp.2     Comp.3     Comp.4    Comp.5     Comp.6
Standard deviation     1.0012446 0.2646305 0.25156371 0.22249158 0.2039234 0.16942693
Proportion of Variance 0.7984175 0.0557737 0.05040176 0.03942547 0.0331195 0.02286202
Cumulative Proportion  0.7984175 0.8541912 0.90459301 0.94401848 0.9771380 1.00000000

Loadings:
        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
STG                           0.748  0.502  0.425
SCG                   -0.315  0.449 -0.805  0.222
STR                   -0.917 -0.193  0.300  0.168
LPR            -0.823 -0.162  0.252        -0.470
PEG      0.163  0.564 -0.142  0.353        -0.711
num_UNS -0.981                0.120        -0.130
```

We wanted to retain just enough components to explain a large percentage of the total variation of the original variables. The values between 70% and 90% were sufficient enough. In this case, the first and second PC accounted for 85% of the total variance of the observed variables. If we add the third PC, this percentage up will increase to 90%. This was understandable since the first three PCs encompass most of proportion of variance of the data. We saw in the loadings that component 1 contains variance from only PEG and num_UNS.

We computed the eigenvalues of the covariance matrix. We computed the mean of those eigenvalues which resulted in 0.209. We can excluded the PCs whose eigenvalues are less than the average eigenvalue. In this case, only the first PC eigenvalue of 1 was greater than the average eigenvalue. All other PC eigenvalues were below the mean.

We then plotted a scree plot, which how much variation each PC captures from the data. Seen in **Figure 4**, our scree plot has a steep curve that bends quickly and flattens out. This suggests that the first two PCs are sufficient to describe the essence of the data, Based on our entire analysis, we should retain the first two PCs.

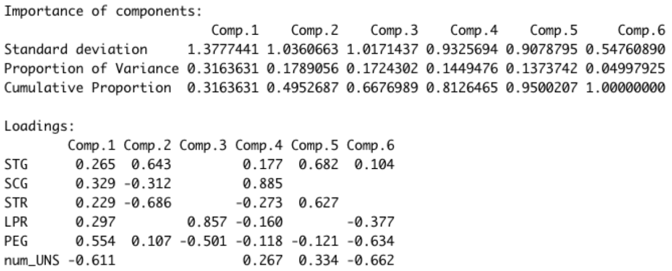
Figure 4. Covariance Scree Plot



3.2 Correlation

Next, we computed the correlation of all six variables of the dataset. We use the correlation matrix to understand is a method of the strength and direction of the relationships between the variables. A correlation is a function of the covariance, but a correlation matrix standardizes each of the variables, with a mean of 0 and a standard deviation of 1. **Figure 5** depicts the importance of the correlation components and their loadings.

Figure 5. Proportion of variance explained



```
Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4    Comp.5     Comp.6
Standard deviation     1.3777441 1.0360663 1.0171437 0.9325694 0.9078795 0.54760890
Proportion of Variance 0.3163631 0.1789056 0.1724302 0.1449476 0.1373742 0.04997925
Cumulative Proportion  0.3163631 0.4952687 0.6676989 0.8126465 0.9500207 1.00000000

Loadings:
        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
STG      0.265  0.643         0.177  0.682  0.104
SCG      0.329 -0.312         0.885
STR      0.229 -0.686        -0.273  0.627
LPR      0.297         0.857 -0.160        -0.377
PEG      0.554  0.107 -0.501 -0.118 -0.121 -0.634
num_UNS -0.611                0.267  0.334 -0.662
```

After computing the PCA of the correlation matrix, we get six principal components. In this case, the first three PCs account for 67% of the total variance of the observed variables. If we add the fourth PC, this percentage up will increase to 81%. We can see in the loadings that component 1 contains variance from all variables.

Similarly with the covariance matrix, we computed the eigenvalues of the correlation matrix and the mean of those values, which resulted in 1. Only the first three PCs are greater than the average eigenvalue, meaning that these are the PCs to include.

We then plotted a scree plot, which how much variation each PC captures from the data. Seen in **Figure 6**, scree plot is doesn't have really steep curve. The line actually bends and flattens out slowly, suggesting that we would need more PCs than normal. Based on our entire analysis, we should retain the first three PCs.

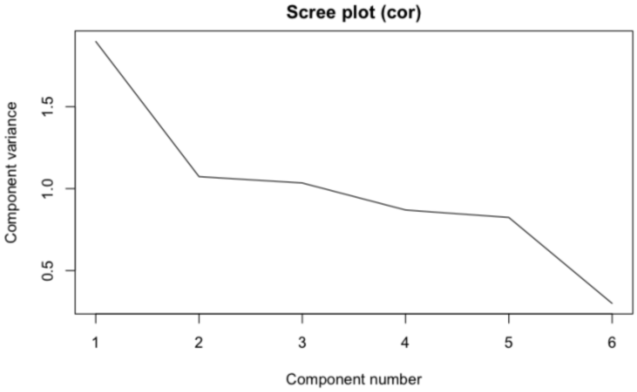


Figure 6. Correlation Scree Plot

4. Clustering

Clustering involves finding subgroups of objects such that the objects in a group will be related to one another and unrelated to the objects in other groups.

4.1 K-means clustering

This type of clustering involves seeking to partition the observations into a prespecified number of clusters. To do so, we first standardized the dataset. Then we computed the optimal number of clusters using the “wss” method, seen in **Figure 7**. We can use the elbow method, where the bend indicates the optimal number of clusters. Here we see that the optimal number of clusters is 4; clusters past 4 have little value.

We then performed k-means clustering on the data matrix, using four clusters. **Figure 8** depicts the result of the cluster sizes and means. This shows the variation of each variable that belongs to a specific clustering group.

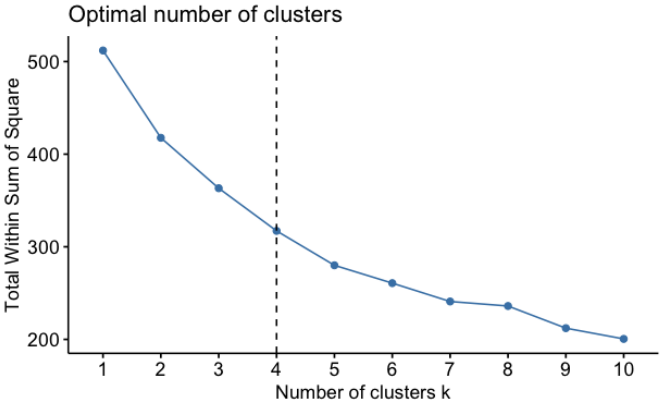


Figure 7. Optimal number of k-clusters

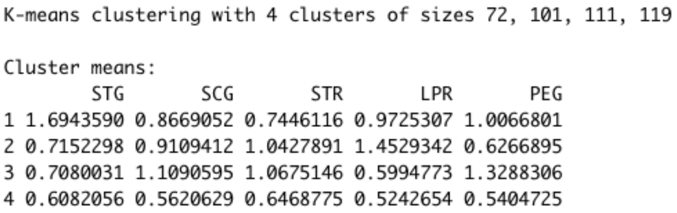


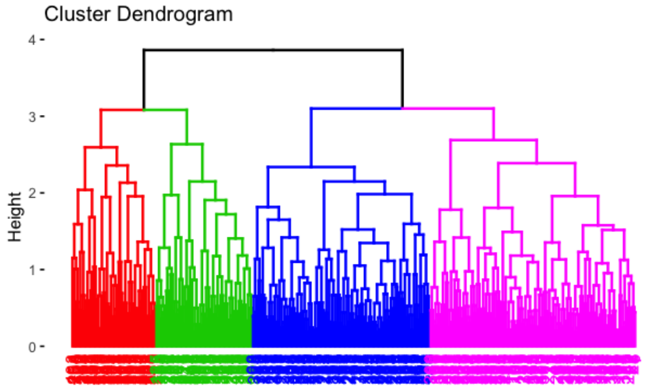
Figure 8. K-means clustering size and mean

```
K-means clustering with 4 clusters of sizes 72, 101, 111, 119

Cluster means:
        STG       SCG       STR       LPR       PEG
1 1.6943590 0.8669052 0.7446116 0.9725307 1.0066801
2 0.7152298 0.9109412 1.0427891 1.4529342 0.6266895
3 0.7080031 1.1090595 1.0675146 0.5994773 1.3288306
4 0.6082056 0.5620629 0.6468775 0.5242654 0.5404725
```

4.2 Hierarchical Clustering

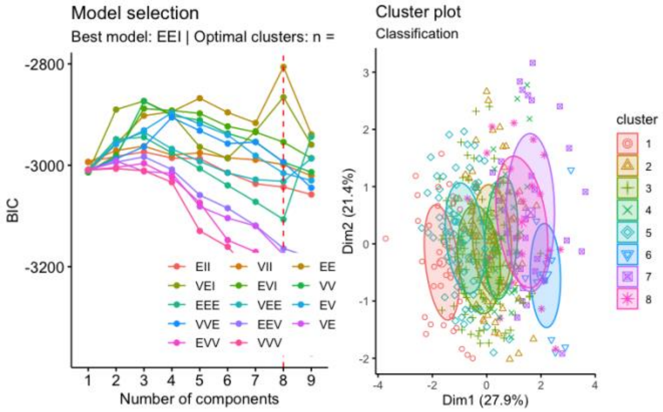
In type of clustering, we do not know the numbers of clusters in advance. Hierarchical clustering starts with each point in its own cluster, identifies the closest clusters and merges them. To do so, we computed the distance matrix using the standardized data with an Euclidean method. Then we used the hclust function on the dissimilarity structure produced by the distance matrix. **Figure 9** depicts the result of this cluster dendrogram.

Figure 9. Hierarchical cluster dendrogram



Then we computed the Gaussian mixture modelling for model-based clustering. We plotted a visualization of the BIC values and classification, seen in **Figure 10**. Looking at the left plot, according to the model selection, our best model is EEI. EEI. is the model that uses diagonal, equal volume and shape. The right plot shows that the first principle component accounts for 27.9% of variation. The second principle component accounts for 21.4% of the variation. So together they account for 49.3% of the variation.

Figure 10. BIC and cluster classification



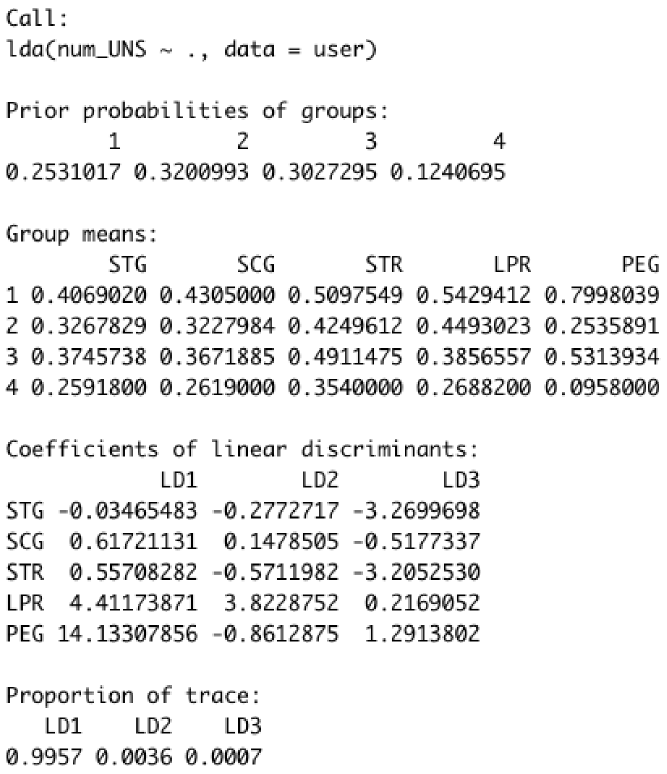
5. Classification Analysis

Classification refers to a predictive modeling problem where a class label is predicted for a given example of input data.

5.1 Linear Classification

Linear classifiers classify data into labels based on a linear combination of input features. We executed a linear discriminant analysis (LDA) on the original dataset, where UNS was the response and the other five attributes were predictors. From the LDA, the group means, coefficients of linear discriminants and proportion of trace are seen in **Figure 11**. The discriminant functions are linear combinations of the four variables. Percentage separations achieved by the first discriminant function is 99.57% Percentage separations achieved by the second discriminant function is 0.36%

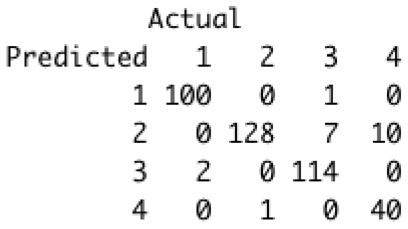
Figure 11. LDA results



```
Call:
lda(num_UNS ~ ., data = user)

Prior probabilities of groups:
        1         2         3         4 
0.2531017 0.3200993 0.3027295 0.1240695 

Group means:
        STG       SCG       STR       LPR       PEG
1 0.4069020 0.4305000 0.5097549 0.5429412 0.7998039
2 0.3267829 0.3227984 0.4249612 0.4493023 0.2535891
3 0.3745738 0.3671885 0.4911475 0.3856557 0.5313934
4 0.2591800 0.2619000 0.3540000 0.2688200 0.0958000

Coefficients of linear discriminants:
            LD1        LD2        LD3
STG -0.03465483 -0.2772717 -3.2699698
SCG  0.61721131  0.1478505 -0.5177337
STR  0.55708282 -0.5711982 -3.2052530
LPR  4.41173871  3.8228752  0.2169052
PEG 14.13307856 -0.8612875  1.2913802

Proportion of trace:
   LD1    LD2    LD3 
0.9957 0.0036 0.0007 
```

We created predictions based on the linear classification functions and produced it's corresponding classification table. **Figure 12** depicts the predicted values against the actual values. After, we found the misclassification error to be 5.2%.

Figure 12. LDA classification table

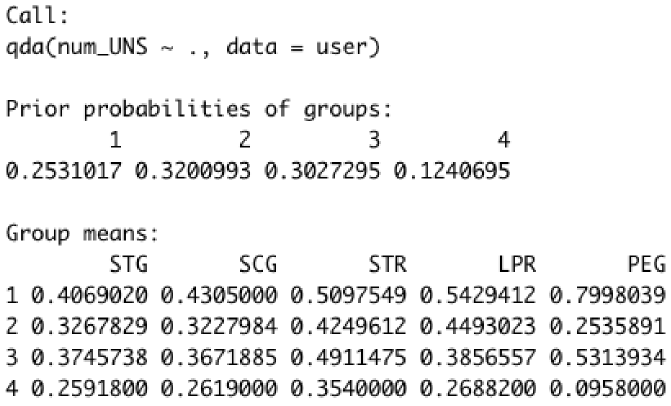


```
          Actual
Predicted   1   2   3   4
        1 100   0   1   0
        2   0 128   7  10
        3   2   0 114   0
        4   0   1   0  40
```

5.2 Quadratic Classification

A quadratic classifier is statistical classifier that uses a quadratic decision surface to separate measurements of two or more classes of objects or events. Similarly to LDA, we executed a quadratic discriminant analysis (QDA) on the original dataset, where UNS was the response and the other five attributes were predictors. **Figure 13** shows the QDA group means.

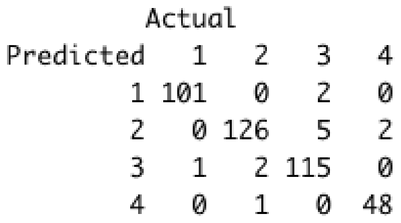
Figure 13. QDA results



```
Call:
qda(num_UNS ~ ., data = user)

Prior probabilities of groups:
        1         2         3         4 
0.2531017 0.3200993 0.3027295 0.1240695 

Group means:
        STG       SCG       STR       LPR       PEG
1 0.4069020 0.4305000 0.5097549 0.5429412 0.7998039
2 0.3267829 0.3227984 0.4249612 0.4493023 0.2535891
3 0.3745738 0.3671885 0.4911475 0.3856557 0.5313934
4 0.2591800 0.2619000 0.3540000 0.2688200 0.0958000
```

Again, we created predictions based on the quadratic classification functions and produced it’s corresponding classification table. **Figure 14** depicts the predicted values against the actual values. After, we found the misclassification error to be 3.2%.

Figure 14. QDA classification table



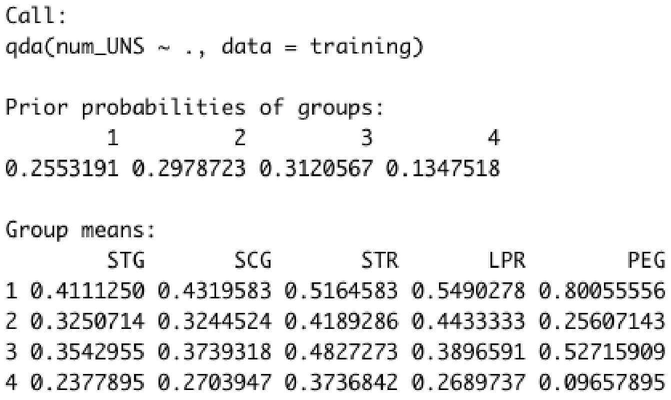
```
          Actual
Predicted   1   2   3   4
        1 101   0   2   0
        2   0 126   5   2
        3   1   2 115   0
        4   0   1   0  48
```

In comparison, the quadratic classification functions seemed to better predict the data than the linear classification functions, because it produced a lower misclassification error.

5.3 Hold-out method

The hold-out method is when you split up your dataset into a 'train' and 'test' set. The training set is what the model is trained on, and the test set is used to see how well that model performs on unseen data. Recall the original dataset split the data 70% to 30% where 258 was in training and 145 was in test. We used this split probability with QDA, since it has a lower misclassification error than LDA. **Figure 15** depicts the group means of our train QDA.

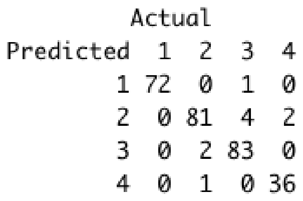
Figure 15. Hold-out QDA results



```
Call:
qda(num_UNS ~ ., data = training)

Prior probabilities of groups:
        1         2         3         4 
0.2553191 0.2978723 0.3120567 0.1347518 

Group means:
        STG       SCG       STR       LPR        PEG
1 0.4111250 0.4319583 0.5164583 0.5490278 0.80055556
2 0.3250714 0.3244524 0.4189286 0.4433333 0.25607143
3 0.3542955 0.3739318 0.4827273 0.3896591 0.52715909
4 0.2377895 0.2703947 0.3736842 0.2689737 0.09657895
```

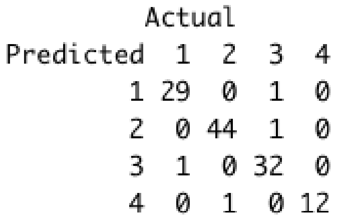
We created predictions using the train model against the training data and produced it’s corresponding classification table, seen in Figure 16. After, we found the misclassification error on train to be 3.55%. Then we created predictions using the train model against the testing data and produced it's corresponding classification table, seen in Figure 17. After, we found the misclassification error on test to be 5.78%.

Figure 16. Train classification



```
         Actual
Predicted  1  2  3  4
        1 72  0  1  0
        2  0 81  4  2
        3  0  2 83  0
        4  0  1  0 36
```

Figure 17. Test classification



```
         Actual
Predicted  1  2  3  4
        1 29  0  1  0
        2  0 44  1  0
        3  1  0 32  0
        4  0  1  0 12
```

6. Results and Conclusion

When doing PCA, component 1 from the covariance matrix loadings only includes variances for PEG and UNS while that of the correlation matrix includes variance from all six variables. Based on this difference, the correlation matrix may seem like the best choice. However, as we mentioned during our exploratory analysis, PEG is the only variables that is directly related to UNS. In addition, we saw that first three components of the covariance matrix account for 90% of the variance, which that of the correlation matrix only accounts for 67% of the variance. Therefore, it is more appropriate to use the covariance matrix to use for data reduction.

We found that from both the K-means clustering and hierarchical clustering, that the ideal number of clusters is four. Looking at the BIC model selection, we found that EEI was our best model. This type of model is diagonal and has equal volume and shape. The cluster classification plot showed us that the 1st and 2nd PC account for 49.3% of the variance.

During our classification analysis, we found that the LDA had a misclassification error of 5.2% while QDA has a misclassification error of 3.2%. Because of this result, we used QDA to test the hold-out method. From this method, we found that the misclassification error on the train data was 3.55% while the misclassification error on the test data was 5.78%.

As we result, we found that PEG was the most included variance in all of the models, especially for the covariance PCA. Likewise, QDA did a good job of classifying knowledge level of the user.

References

[1] *UCI Machine Learning Repository: User Knowledge Modeling Data*. H. Tolga Kahraman, Seref Sagiroglu, Ilhami Colak. 2009. Web. http://archive.ics.uci.edu/ml/datasets/User+Knowledge+Modeling

[2] Kahraman, H. Tolga, Sagiroglu, Seref, and Colak, Ilhami. *The development of intuitive knowledge classifier and the modeling of domain dependent data*. Research Paper. 2012. https://www.academia.edu/3100709/The_development_of_intuitive_knowledge_classifier_and_the_modeling_of_domain_dependent_data