PCA and KNN

Muhammad Banatwalla

October 30, 2020

1 Introduction

For this homework, we were tasked with using the Principal Component Analysis(PCA) to reduce the dimensions of the data we used in the previous homework. After finding the reduced dimension, we will run the K-Nearest Neighbor(KNN) analysis and compare our results to the KNN results before PCA. Because our data in HW 2 was unbalanced, we have chosen new data to work with in this homework.

1.1 Preliminary Treatment of the Data

Again getting our data from the ”University of California Irvine Repository for Machine Learning Datasets”, we chose 3 new fonts to work with. The fonts we chose are Century, Ebrima, and Gill. Before treatment of the data, the data had the following sizes: Century - 7994, Ebrima - 6892, Gill - 5836. After selecting only the cases that are italic and with strength = .4, like we did in HW 2, they had the new following sizes: Century - 1999, Ebrima - 1723, Gill - 1459. We still felt this was not balanced enough so we removed a random 350 cases from only the Century data set. Now, for Class 1 we used the Century font with a new size of 1649, for Class 2 we used the Ebrima font with a size of 1723, and for Class 3 we used the Gill font with a size of 1459. Class 1, Class 2, and Class 3 were unioned into a larger data set we call DATA with size 4831. Further, we then standardized DATA to get a new set called SDATA.

2 HW 2 Results

Because we chose new data sets, we re-ran the methods used in HW 2 on the new data so that we had something to compare our PCA results to.

2.1 KNN Results

We used our 3 classes as stated above and ran KNN on the testing and training sets. Below are the values for the training accuracy and testing accuracy for various values of k:

1

k	$trainperf_k$	$testperf_k$
1	98%	81%

2	89%	74%
3	87%	75%
4	84%	75%
5	82%	74%
10	78%	72%
15	74%	68%
20	72%	67%
30	70%	67%
40	69%	66%
50	68%	64%
100	65%	60%

Table 1: *T rainperf$_k$andT estperf$_k$*

We also plotted the values versus k so that we could visualize.

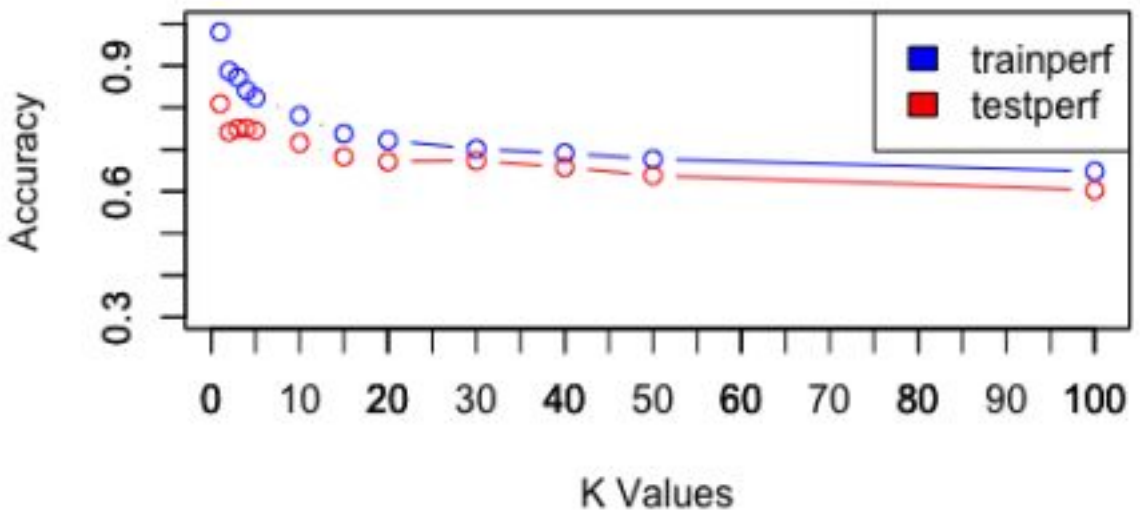


Figure 1: trainperf and testperf curve v. K-values

As you can see from both the table and the plot, the best value for k is k = 1. Below are the confusion matrices for the training and testing set for k = 1:

2

	Predicted: CENTURY	Predicted: EBRIMA	Predicted: GILL
Actual: CENTURY	97%	0%	3%
Actual: EBRIMA	0%	98%	2%
Actual: GILL	0%	0%	100%

Table 2: Confusion Matrix for k = 1 (*T rainingData*)

	Predicted: CENTURY	Predicted: EBRIMA	Predicted: GILL
Actual: CENTURY	84%	8%	8%
Actual: EBRIMA	13%	76%	11%
Actual: GILL	9%	8%	83%

Table 3: Confusion Matrix for k = 1 (*T estingData*)

On the training set, we had near perfect classification for all of the fonts. The accuracy for the testing data is not as good but this is expected. For the training data, 13% of the Ebrima font was classified as Century and 11% of the Ebrima font was classified as Gill. These misclassifications have the biggest impact on the accuracy. Overall, we had the most success with this k value.

3 Principal Component Analysis

We utilized PCA to reduce the dimensions of our data. Right now, the data consists of 400 features. The idea of PCA is to reduce the number of features without losing accuracy. This reduction in dimension allows us to better explore the data. Since we are working with a large data set with 400 features we did data compression by PCA analysis in hopes of eliminating redundant information to hopefully improve KNN automatic classification. PCA works by transforming a large set of features into a smaller one that still contains most of the information in the large set.

3.1 Correlation

Using the standardized data set SDATA, we computed a correlation matrix on the gray level image intensity values for the 400 pixel positions. Here is a sample from that matrix:

```
> cdata
              r0c0          r0c1          r0c2          r0c3          r0c4          r0c5
r0c0  1.000000e+00  0.9350568240  7.972501e-01  0.6336011717  0.4825978621  0.3497170274
r0c1  9.350568e-01  1.0000000000  8.983527e-01  0.7099515308  0.5271943396  0.3672537459
              r0c6          r0c7          r0c8          r0c9         r0c10         r0c11         r0c12
r0c0  0.225553581  0.144307619  0.042535105 -0.0202033836 -4.241540e-02 -0.0218945799  0.041320640
r0c1  0.223237170  0.125510625  0.017536759 -0.0442689010 -6.906481e-02 -0.0469212422  0.023957776
              r0c13         r0c14         r0c15         r0c16         r0c17         r0c18         r0c19
r0c0  0.107542539  0.185900570  2.461333e-01  3.197959e-01  0.3991170292  0.4487893788  0.4849997829
r0c1  0.095938146  0.189665991  2.711047e-01  3.623970e-01  0.4566513565  0.4973503971  0.4747118668
              r1c0          r1c1          r1c2          r1c3          r1c4          r1c5          r1c6
r0c0  0.8260523743  0.7614663979  6.563000e-01  5.062536e-01  0.330818660  0.181691428  0.055114323
r0c1  0.7902421181  0.8334391906  7.458305e-01  5.790651e-01  0.370707956  0.187323468  0.044446803
              r1c7          r1c8          r1c9         r1c10         r1c11         r1c12         r1c13
r0c0 -0.026005428 -8.806310e-02 -0.1019247701 -0.1152600910 -0.125560940 -0.0883995448 -0.013099684
r0c1 -0.050206708 -1.181651e-01 -0.1292388016 -0.1415651238 -0.152680261 -0.1091894609 -0.019211762
             r1c14         r1c15         r1c16         r1c17         r1c18         r1c19          r2c0
r0c0  0.069050847  0.159338967  0.2446747389  0.3006080741  0.3420230063  0.381550577  6.824451e-01
r0c1  0.072094747  0.182799968  0.2874515506  0.3533257380  0.3887148514  0.364307875  6.467814e-01
```

Below are the top 10 highest correlation values:

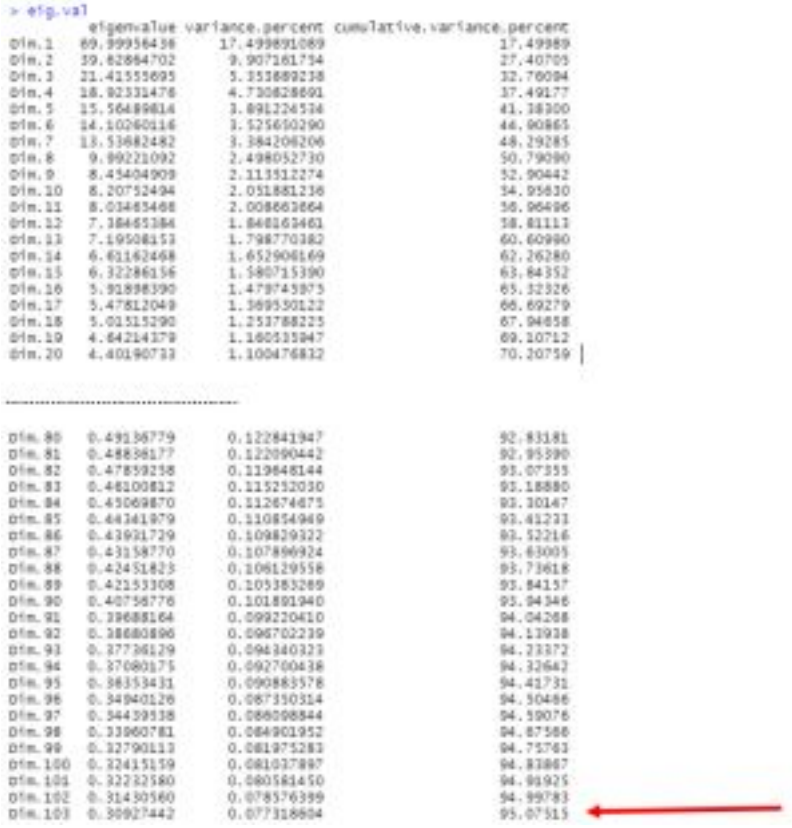
X	Y	Correlation
r0c1	r0c0	.9351
r19c0	r19c1	.9332
r14c1	r13c1	.9300
r19c19	r19c18	.9298
r19c18	r19c17	.9239
r13c1	r12c1	.9224
r12c0	r11c0	.9185
r12c1	r12c0	.9178
r12c1	r11c1	.9177
r19c9	r19c10	.9172

Table 4: Top 10 Correlation Values

These are the pixel positions corresponding to the 10 pairs of features that have the highest correlations. These high correlations likely come from pixels that are close to each other in the image. They may contain redundant information that we can use PCA analysis to reduce.

3.2 Eigenvector and Eigenvalue Computation

PCA works by computing principal components which are new variables that are con structed as linear combinations of the initial variables. PCA tries to put maximum possible information in the first principal component then the maximum remaining in the second principal component and so on. Eigenvalues are the coefficients attached to eigenvectors, which give the amount of variance carried in each Principal Component. Here the eigenvalues are ordered from highest to lowest in order of significance:



```
> eig.val
        eigenvalue variance.percent cumulative.variance.percent
Dim.1   69.99956439     17.499891089                   17.49989
Dim.2   39.62864702      9.907161754                   27.40705
Dim.3   21.41555695      5.353889238                   32.76094
Dim.4   18.92331476      4.730828691                   37.49177
Dim.5   15.56489814      3.891224534                   41.38300
Dim.6   14.10260116      3.525650290                   44.90865
Dim.7   13.53682482      3.384206206                   48.29285
Dim.8    9.99221092      2.498052730                   50.79090
Dim.9    8.45404909      2.113512274                   52.90442
Dim.10   8.20752494      2.051881236                   54.95630
Dim.11   8.03483468      2.008663664                   56.96496
Dim.12   7.38465384      1.846163461                   58.81113
Dim.13   7.19508153      1.798770382                   60.60990
Dim.14   6.61162468      1.652906169                   62.26280
Dim.15   6.32286156      1.580715390                   63.84352
Dim.16   5.91898390      1.479745975                   65.32326
Dim.17   5.47812049      1.369530122                   66.69279
Dim.18   5.01515290      1.253788225                   67.94658
Dim.19   4.64214379      1.160535947                   69.10712
Dim.20   4.40190733      1.100476832                   70.20759

-----------------------------------------

Dim.80   0.49136779      0.122841947                   92.83181
Dim.81   0.48838177      0.122090442                   92.95390
Dim.82   0.47859258      0.119648144                   93.07355
Dim.83   0.46100812      0.115252030                   93.18880
Dim.84   0.45069870      0.112674675                   93.30147
Dim.85   0.44341979      0.110854949                   93.41233
Dim.86   0.43921729      0.109804322                   93.52214
Dim.87   0.43158770      0.107896924                   93.63005
Dim.88   0.42451823      0.106129558                   93.73618
Dim.89   0.42153308      0.105383269                   93.84157
Dim.90   0.40756776      0.101891940                   93.94346
Dim.91   0.39688164      0.099220410                   94.04268
Dim.92   0.38680896      0.096702239                   94.13938
Dim.93   0.37736129      0.094340323                   94.23372
Dim.94   0.37080175      0.092700438                   94.32642
Dim.95   0.36353431      0.090883578                   94.41731
Dim.96   0.34940126      0.087350314                   94.50466
Dim.97   0.34439538      0.086098844                   94.59076
Dim.98   0.33960781      0.084901952                   94.67566
Dim.99   0.32790113      0.081975283                   94.75763
Dim.100  0.32415159      0.081037897                   94.83867
Dim.101  0.32232580      0.080581450                   94.91925
Dim.102  0.31430560      0.078576399                   94.99783
Dim.103  0.30927442      0.077318604                   95.07515
```

Figure 2: Eigenvalues

After computing the eigenvalues, we found r new features that had a proportion of

vari ance explained of 95% to be r=103. We plot the eigenvalues versus dimension:

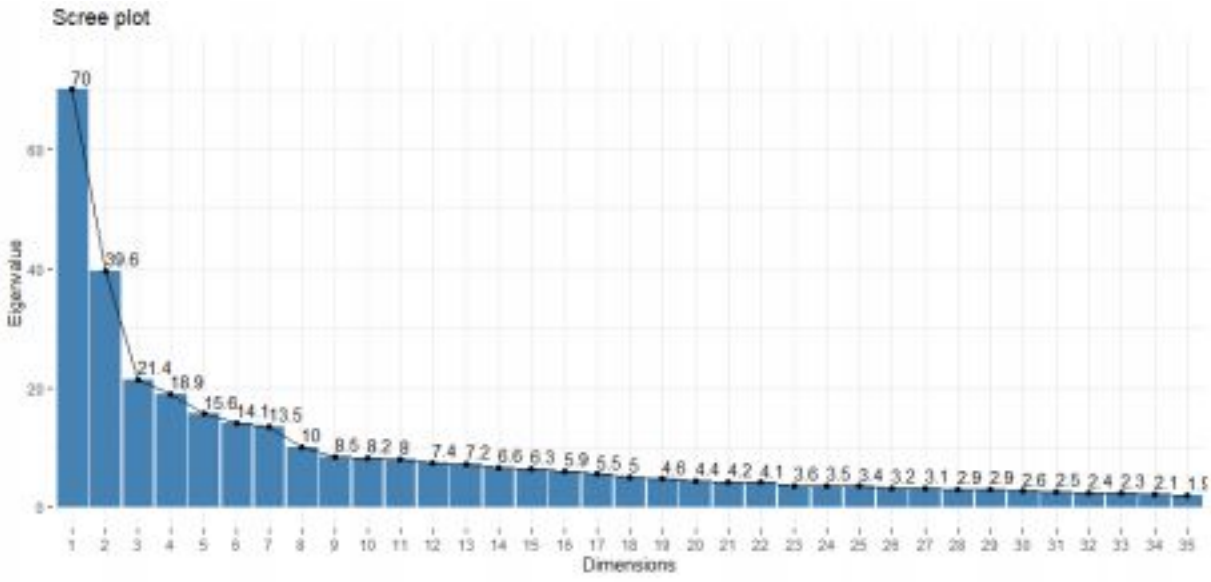


Figure 3: Eigenvalues v. Dimension

An eigenvalue is a number, telling you how much variance there is in the data. Here the eigenvalues are ordered in descending order. This plot verified that we can use PCA with our dataset because it shows that PCA put maximum possible information in the first few principal components. So now if we reduce our dataset's dimensionality, we can focus our analysis on the first few components without too much information loss.

We also plotted the percentage of explained variance by each principal component

verusus dimension:

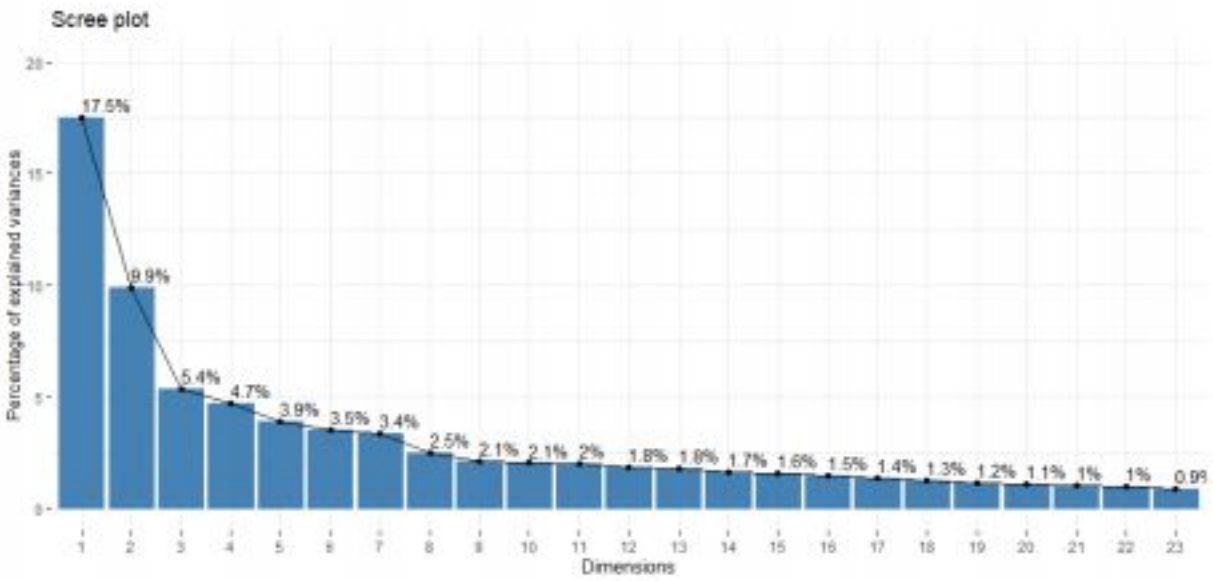


Figure 4: Dimension v. Explained Variance

This plot displays the percent of explained variance explained by each principal component. The percentage of explained variances tells us how much variance in the data is explained by a particular principal component.The first principal component explains most of the variance from our data (17.5%) and the second principal component displays the second most variation (9.9%) and the third principal component explains the third most variation and so on. By dimension 22 only 1% of the variation is being explained by the principal component and it continues to decrease after that. Therefore, after dimension 22 we will not lose too much information.

3.3 Computation of Principal Components

We then computed the principal components for each case:

```
> newdata
          PC1          PC2          PC3          PC4          PC5          PC6
1 -5.247361e-01  1.706141956 -8.824886e+00 -10.150293642 -3.287208e+00  8.153013e-01
2 -2.213185e+00  2.116477102 -9.052385e+00  -8.645547742 -1.869840e+00  9.444039e-01
3 -4.652678e+00 -0.553560436  8.576387e+00   4.053174580 -3.130206e+00 -5.854253e+00
4 -4.231302e+00 -7.715720703  9.004518e+00   1.309275842 -2.492956e+00  5.470143e+00
5 -1.683625e-01  5.305413920  1.567565e+00  -3.065944389 -3.786262e+00 -6.122244e-03
6 -3.752290e+00  6.828679947  3.600730e-01  -5.241876060  6.049994e-01  2.777677e+00
7 -3.389319e+00  5.967004108 -5.136321e+00  -0.367403352 -9.250950e-01  4.063120e+00
8 -6.546439e-02  3.926379543  8.654658e-02  -2.040368402 -1.337580e-01 -1.807838e+00
9 -3.552373e+00  0.243908616  4.009555e+00  10.929288597 -3.014941e+00 -7.749700e-01
```

Figure 5: Principal Components

3.4 Graphic Interpretation

Principal Component Analysis takes a dataset with a lot of dimensions like ours and flattens it to fewer dimensions so we can look at it. It flattens the data by focusing on the things that are different between the features. To visually see the impact of PCA, we display and interpret 6 scatterplots in the plane.

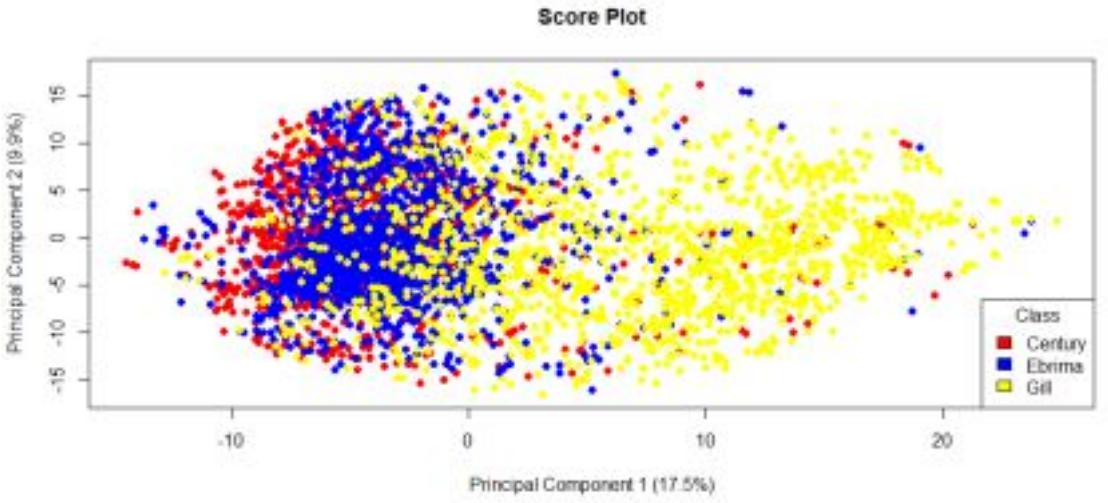


Figure 6: Principal Component 1 v. Principal Component 2

Principal component 1 captures the most amount of variation - 17.5% and principal component 2 captures the second most amount of variation - 9.9%. This plot shows the two directions of greatest variation in the data. Each observation in our data set received a “score” based on how much influence they had on each principal component.

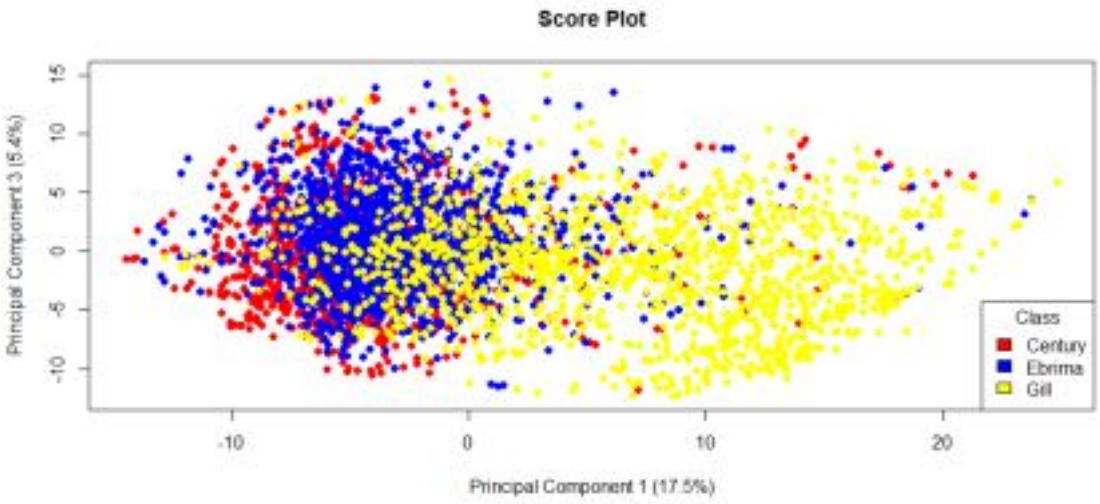


Figure 7: Principal Component 1 v. Principal Component 3

Based on the plot, the cases with the font Gill tend to have higher PC1 scores and are spread out the most. Cases under Ebrima and century have relatively lower scores with century tending to be the lowest. There are several outliers in the PC1 variables which makes our clusters to not be as well defined as we want. There does not seem to be a clear difference between the PC3 scores, though there seems to be a higher concentration of cases with font gill among the lowest scores and barely any font gill amongst the highest scores.

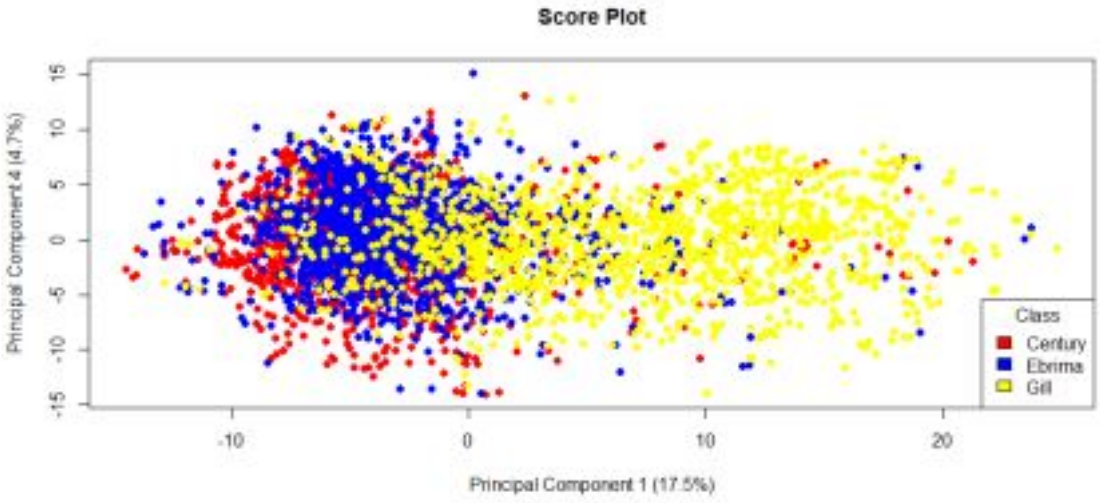


Figure 8: Principal Component 1 v. Principal Component 4

Based on the plot, the cases with the font Gill tend to have higher PC1 scores and are spread out the most. Cases under Ebrima and century have relatively lower scores with

century tending to be the lowest. There are several outliers in the PC1 variables which makes our clusters to not be as well defined as we want. For PC4 scores, there is a high concentration of class Century amongst the lowest scores but for the rest of the scores all Fonts appear approximately equal amount of times.

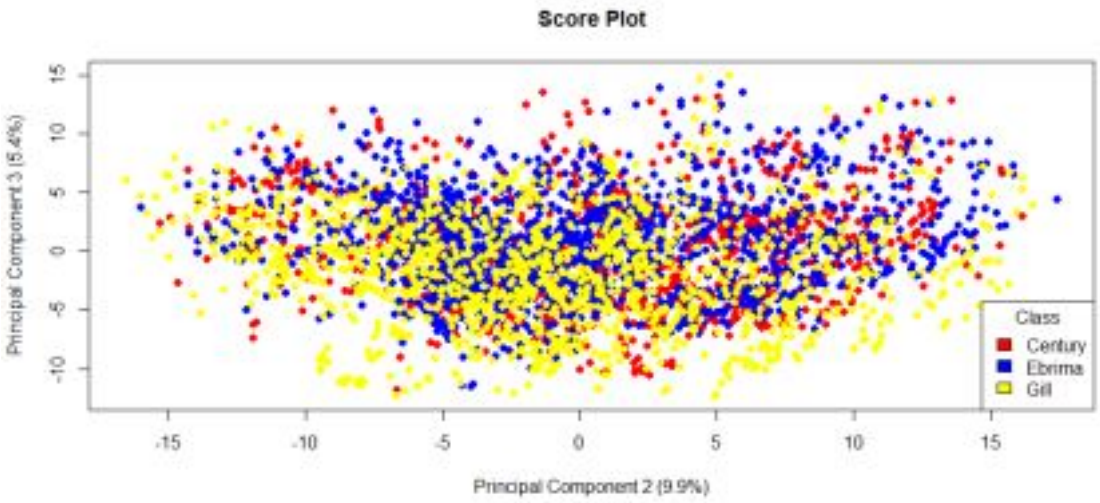


Figure 9: Principal Component 2 v. Principal Component 3

There does not seem to be a clear difference between the PC3 scores, though there seems to be a higher concentration of cases with font gill among the lowest scores and barely any font gill amongst the highest scores. PC2 seems like it would be a particularly bad discriminator of our data as all cases are spread out widely across the classes.

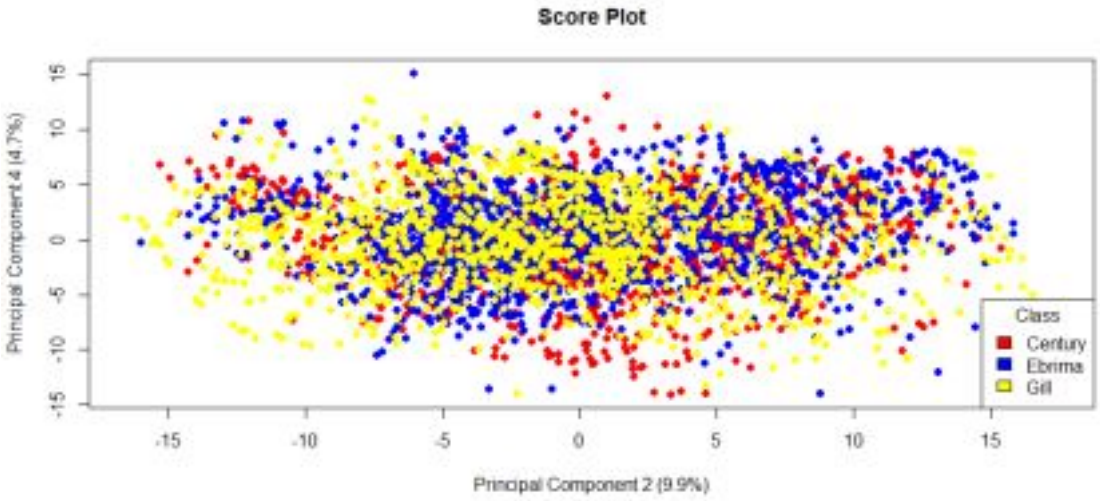


Figure 10: Principal Component 2 v. Principal Component 4

PC2 seems like it would be a particularly bad discriminator of our data as all cases are spread out widely across the classes. For PC4 scores, there is a high concentration of class Century amongst the lowest scores but for the rest of the scores all Fonts appear approximately equal amount of times.

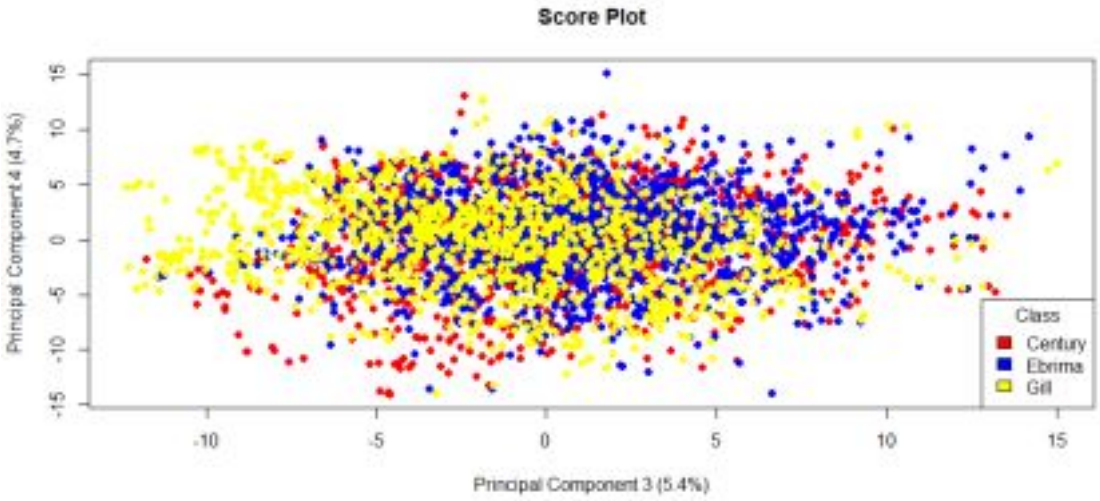


Figure 11: Principal Component 3 v. Principal Component 4

There does not seem to be a clear difference between the PC3 scores, though there seems to be a higher concentration of cases with font gill among the lowest scores and barely any font gill amongst the highest scores. For PC4 scores, there is a high concentration of class Century amongst the lowest scores but for the rest of the scores all Fonts appear approximately equal amount of times.

4 K-Nearest Neighbor After PCA

Through PCA we went from 400 features to 103 features. We now rerun the KNN algorithm on our new reduced data. Here are the trainperf and testperf for various k-values:

k	$trainperf_k$	$testperf_k$
1	98%	86%
2	88%	78%
3	87%	78%
4	84%	77%
5	82%	77%
6	81%	76%
7	80%	75%
8	78%	74%

9	77%	74%
10	76%	74%
11	75%	72%
12	74%	72%
13	74%	72%
14	73%	71%
15	72%	71%
16	712%	70%
17	71%	70%
18	71%	70%
19	70%	70%
20	70%	69%

Table 5: $T\ rainperf_k and T\ estperf_k$

We then plotted these values versus k to visualize:

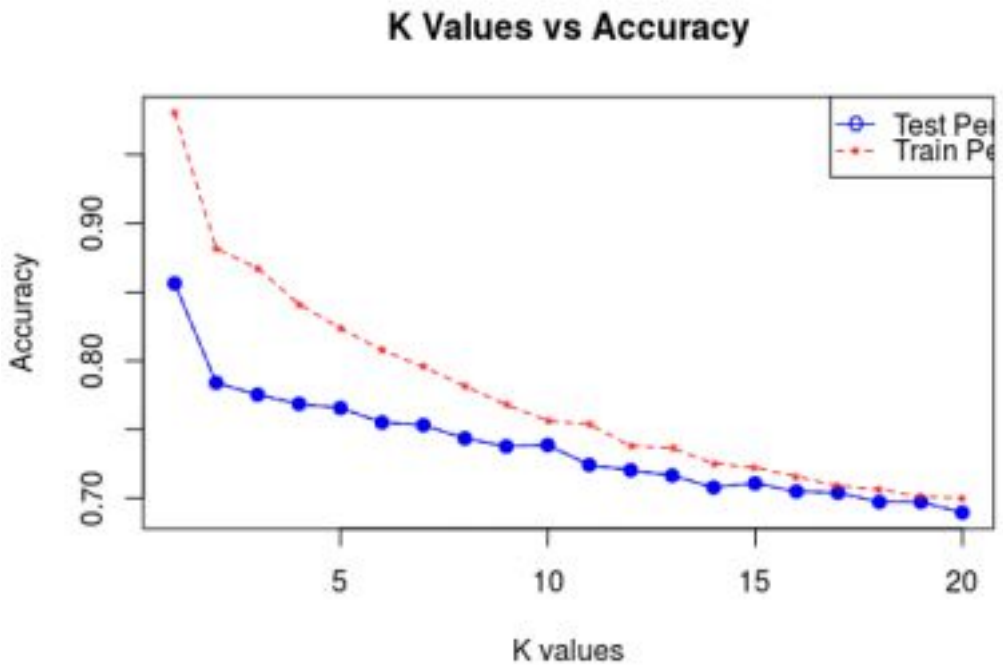


Figure 12: testperf and trainperf v. k-value

Again, our best performance came from k = 1. Below are the confusion matrices for the training and testing set:

	Predicted: CENTURY	Predicted: EBRIMA	Predicted: GILL
Actual: CENTURY	100%	0%	0%
Actual: EBRIMA	2%	98%	0%
Actual: GILL	5%	0%	95%

Table 6: Confusion Matrix for k = 1 (*T rainingData*)

	Predicted: CENTURY	Predicted: EBRIMA	Predicted: GILL
Actual: CENTURY	92%	4%	4%
Actual: EBRIMA	13%	79%	8%
Actual: GILL	10%	6%	84%

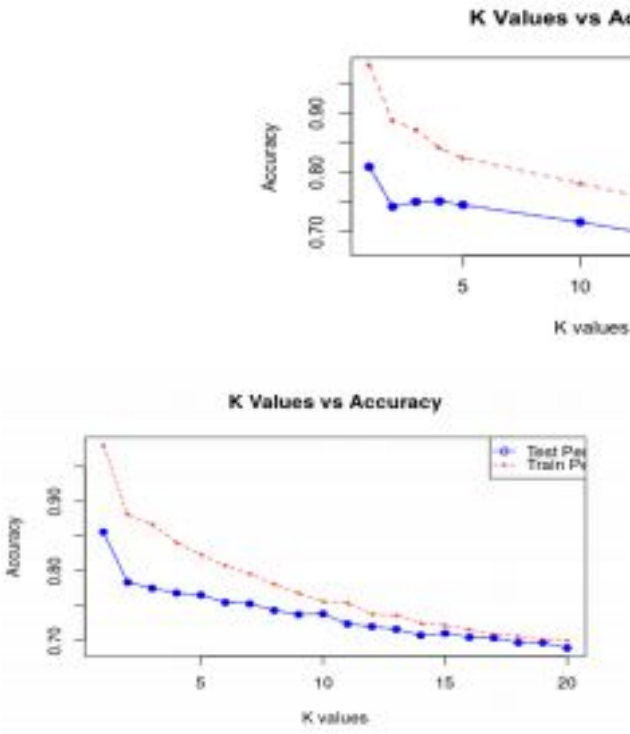
Table 7: Confusion Matrix for k = 1 (*T estingData*)

Overall, for the training data, the performance was nearly perfect for each font. In the testing data, for Ebrima, 13% true class Ebrima were wrongly predicted as century by our

KNN classifier. Also, 10% of true class Gill were wrongly predicted as Century. These two misclassifications had the largest effects on the accuracy for the test set.

5 Before and After Dimension Reduction Comparison

We will look at the KNN performance before and after implementing PCA. Here we will look at the graphs of the trainperf and testperf curve versus k.



(a) KNN Accuracy Before PCA (b) KNN Accuracy After PCA

One thing that we can visually notice from these graphs is that after PCA the train and test accuracy curves are closer together. Additionally, we can see the increase in accuracy on the test curve after PCA.

Since we got k = 1 as our best k even after PCA, we look at the actual performance values and compare:

Data Set	Accuracy Before PCA	Accuracy After PCA
TRAIN	98%	98%

TEST	81%	86%

Table 8: KNN Perfomance Before and After PCA (k = 1)

Overall, our accuracy improved signifcantly for the test set after PCA. The accuracy did not change for the train set.

We also look at the 95% confidence intervals for the test set before and after PCA:

Font	Confidence Int. Before PCA	Confidence Int. After PCA
CENTURY	(80%, 88%)	(89%, 94%)
EBRIMA	(71%, 88%)	(75%, 84%)
GILL	(80%, 88%)	(80%, 88%)

Table 9: Confidence Intervals for k =1 Testing Confusion Matrices Before and After PCA

The confidence intervals are disjoint for Century. This means that there is a difference in the performance for Century if we reduce the dimensions of our data set beforehand

accuracy will be better. There is a wide overlap between the two performance intervals for Gill with and without dimension reduction. Their confidence intervals are also the same width. This indicates that the unknown true accuracy for both with and without dimension reduction would be very close to each other. For Ebrima there is some overlap in the two confidence intervals but the confidence interval after dimension reduction is narrower. This means that with dimension reduction we can be more confident about our estimate.

6 Conclusion and Further Suggestions

Overall, we saw a slight improvement in our accuracy after implementing PCA dimension reduction. Specifically, we saw this improvement in the testing data. Ideally, we do not want our results from training to testing to widely differ, so this can be considered a success. For further improvements, we could implement weights on the features after reduction. Addi tionally, it seems that our graphs from our PCA did not yield any distinct groups like we would have hoped. This is probably because the largest sources of variation are similar in all our fonts. One thing we can try to remedy this is to use some form of preprocessing, scaling or transformation for example. We can also find good discriminators with alternate methods such as PLS-DA or PC-DA. Overall we should understand that PCA does not give us the best discriminators of our data, rather it removes the features that contribute very little to the variance. While using PCA, we

should be aware of the effects of the high variance features on the results. This feature or features may have no information on them and may mislead us and our classifier to nowhere. Imagine the noise in the data (signal noise) that will come up as the most important feature by PCA but, actually, it is worthless!