PCA – Dimension Reduction Tool

Principal Components Analysis – for Dimension Reduction

Dataset: Pendigits – features of handwritten digits from 0 to 9
Classification using k-NN on raw data and PCA dimension reduced data
10992 observations and 16 predictors

0 to 9 digits

```
   0    1    2    3    4    5    6    7    8    9
1143 1143 1144 1055 1144 1055 1056 1142 1055 1055
```

PCA Algorithm:

- *Find mean of each predictor and subtract each value with the mean – centering the values of all predictors*
- *Find the variance and covariance of all predictors to form the covariance matrix*
- *Use eigen decomposition of the covariance matrix to find the eigen values and eigen vectors*
- *Sort the eigen values decreasing order*
- *Each eigen value represents each principal component and we can calculate amount of variance each principal component shows in data.*

Covariance Matrix:

```
> head(covart)
          X1         Y1        X2        Y2        X3         Y3         X4
X1 1173.59567  193.22150 253.44664  78.54773 -513.8902  -95.93597 -384.57752
Y1  193.22150  263.04203 -34.24972  66.56460 -171.7717 -111.81609 -158.89211
X2  253.44664  -34.24972 693.95281 223.35558  374.1344  324.96675 -153.14168
Y2   78.54773   66.56460 223.35558 367.24534  127.3335  349.59444  -21.26525
X3 -513.89024 -171.77167 374.13444 127.33346 1162.8451  428.02926  446.69802
Y3  -95.93597 -111.81609 324.96675 349.59444  428.0293  728.82114  153.30756
         Y4         X5         Y5         X6        Y6         X7         Y7
X1 -104.5793  -44.78331  -95.93654  241.68810  -98.5460  103.50000  -25.76604
Y1 -185.8737  -44.80641 -135.52489   87.39745  -10.2186   48.66132   66.75236
X2  233.1224 -129.74223  -26.44254  -22.10404 -331.0837  -95.83208 -412.82794
Y2  154.2405 -102.23194 -147.42568 -129.47560 -380.7851  -38.25472 -411.53865
X3  363.0786 -260.50167   45.72210 -411.05929 -285.0867 -259.09778 -318.08961
Y3  629.6584  -33.19514  114.55686 -151.01130 -463.1122 -109.22641 -711.38080
          X8         Y8
X1 -512.33566   42.17681
Y1  -91.24484   74.04499
X2 -197.23635 -297.17446
Y2   91.32750 -264.05394
X3   75.09877 -152.52009
Y3  105.66053 -577.66297
```

Eigen Decomposition of covariance matrix:

```
> eigval
 [1] 4213.71294 3702.06880 2285.55300 1341.26427  861.92207  718.26285
457.33818
 [8]  397.59184  286.79007  204.27425  129.06142  100.31804   66.15316
58.67900
[15]   27.40546   24.36745
```

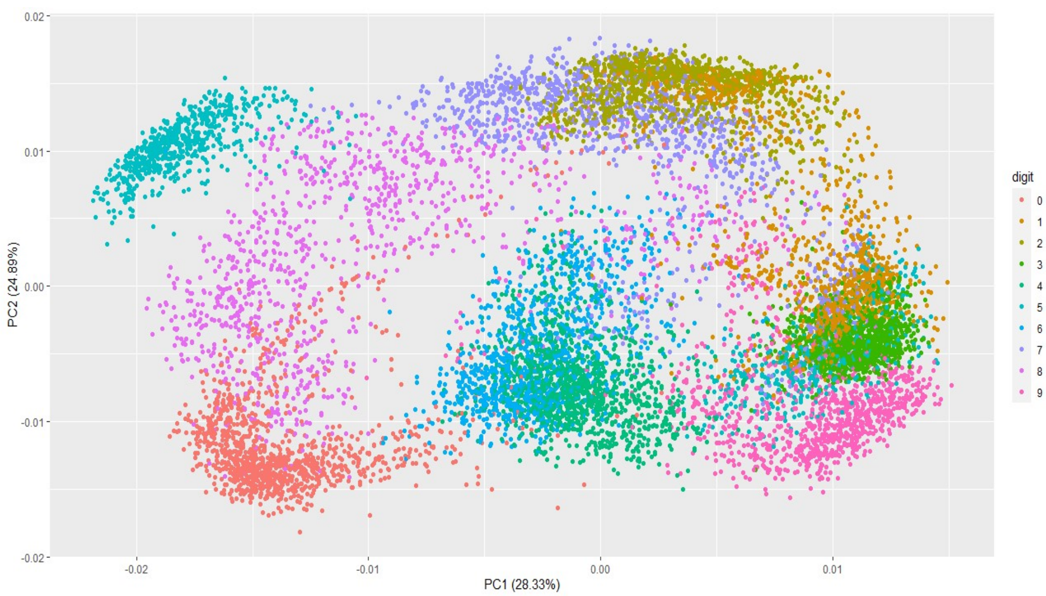
<u>Percentage of variance each eigen value contributes:</u>

```
> round(eigval*100/sum(eigval),2)
 [1] 28.33 24.89 15.37  9.02  5.79  4.83  3.07  2.67  1.93  1.37  0.87
0.67
[13]  0.44  0.39  0.18  0.16
```

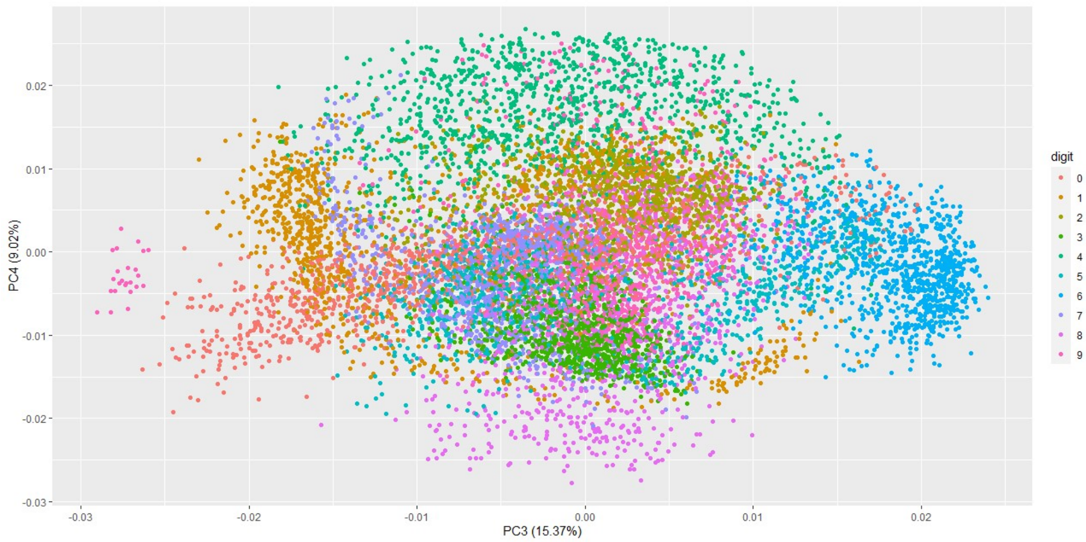
Number of Principal components to be included:

```
Number of PCs to be included: 1  for percentage variation: 28.32793 %
Number of PCs to be included: 2  for percentage variation: 53.21619 %
Number of PCs to be included: 3  for percentage variation: 68.5815 %
Number of PCs to be included: 4  for percentage variation: 77.59854 %
Number of PCs to be included: 5  for percentage variation: 83.39307 %
Number of PCs to be included: 6  for percentage variation: 88.2218 %
Number of PCs to be included: 7  for percentage variation: 91.29639 %
Number of PCs to be included: 8  for percentage variation: 93.96932 %
Number of PCs to be included: 9  for percentage variation: 95.89735 %
Number of PCs to be included: 10  for percentage variation: 97.27065 %
Number of PCs to be included: 11  for percentage variation: 98.1383 %
Number of PCs to be included: 12  for percentage variation: 98.81272 %
Number of PCs to be included: 13  for percentage variation: 99.25745 %
Number of PCs to be included: 14  for percentage variation: 99.65194 %
Number of PCs to be included: 15  for percentage variation: 99.83618 %
Number of PCs to be included: 16  for percentage variation: 100 %
```

Score plot – PCA1 v/s PCA2



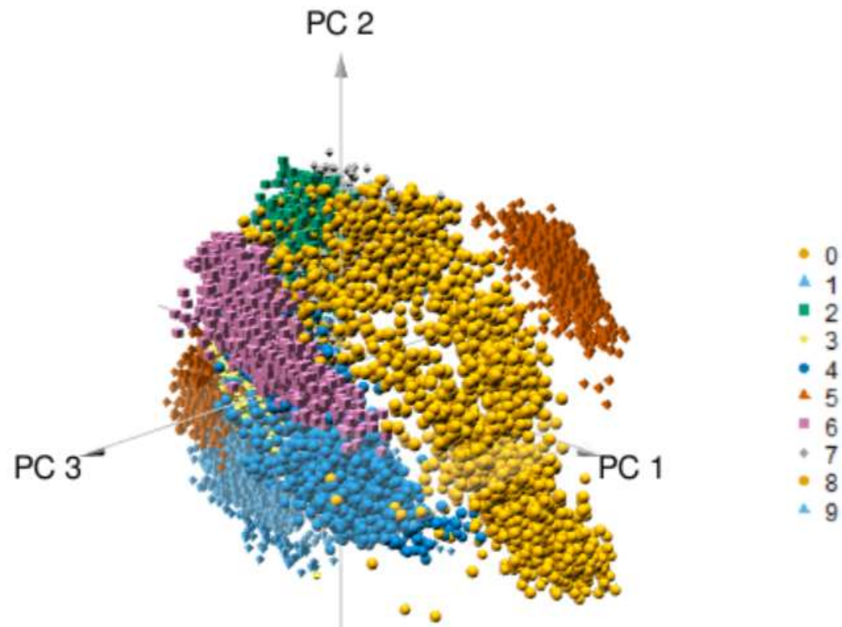
Score plot – PCA3 v/s PCA4

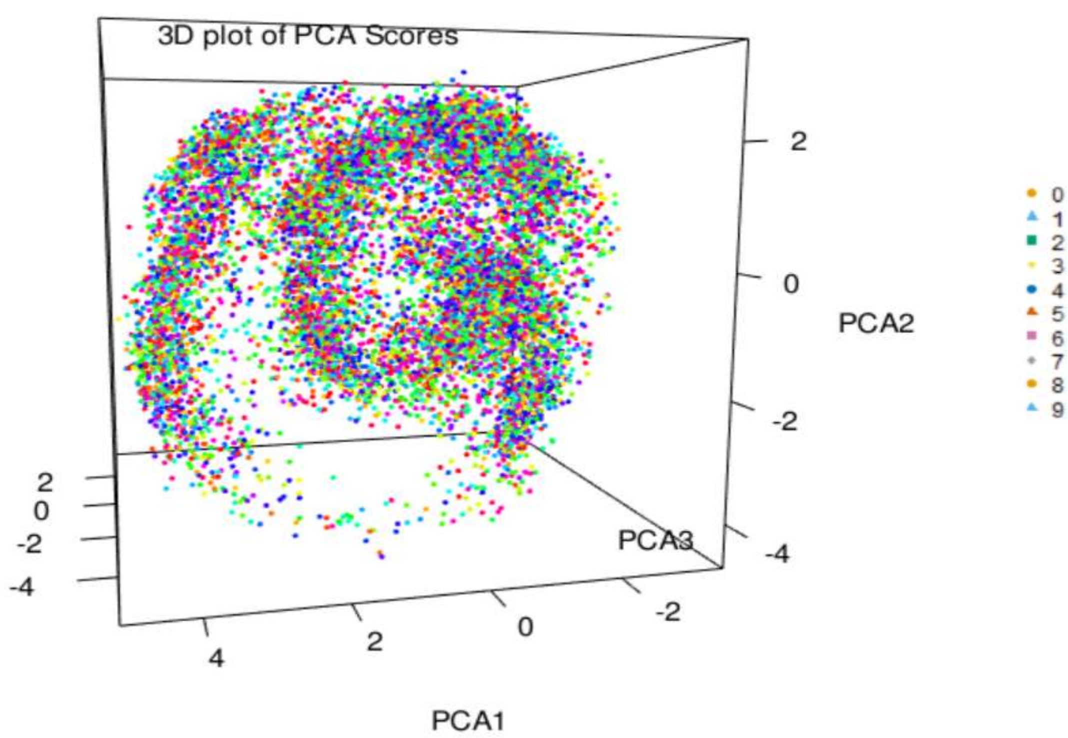


Score Plot- PCA5 v/s PCA6

We see that PCA1 and PCA2 bring more variations in the data points as they spread out there by accuracy of the classification increases by including PCA1 and PCA2 but they just contribute about – 53% variations in the dataset. So at least we need more than 90% variance in the dataset so we have to include first 7 components to correctly classify the data.

3D plot of PCA:





k-NN- Nearest neighbors

Non – parametric method which derives the class or value of the test data based on the nearest neighbors (train data). We have to run multiple trails to find the optimal number of neighbors to be considered for the given dataset. Let’s see how k-NN behaves for both complete raw data and also data with first 7 principal components.

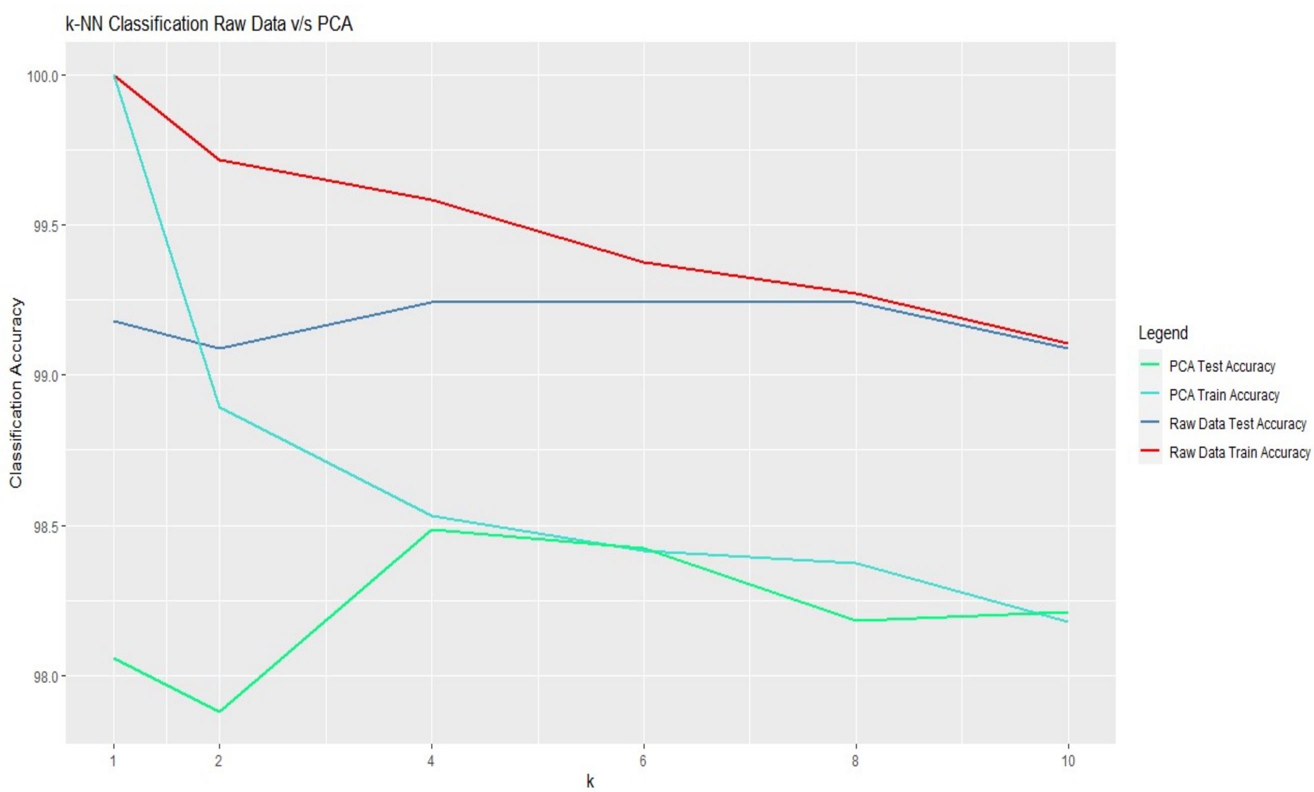
Raw Data:

```
> dim(rtraindat)
[1] 7694   17
> dim(rtestdat)
[1] 3298   17
```

Data – with top 7 principal components

```
> dim(ptraindat)
[1] 7694    7
> dim(ptestdat)
[1] 3298    7
```

k-NN was fit with different values of k – 1,2,4,6,8,10



When k-4, both raw data and principal component data performs well for test data.
Accuracy of test data for both raw data and principal component data:

```
> frtestacc
```

```
[1] 99.30261

> fptestacc
[1] 98.39297
```

Conclusion:

Test accuracy raw data is 99% and principal components data is 98% which is almost same and with 7 principal components which is reduced data from 16 predictors to just 7 predictors and accuracy is almost same. So PCA is very powerful tool for dimension reduction with which we can analyze how data is distributed and reduce data dimension for building the simpler models with reduced data.