**ASSIGNMENT-3**

**Dimensionality Reduction**

library(dplyr)

library(ggplot2)

require(ggfortify)

data\_jobs <- read.csv("C:/Users/ASHISH/OneDrive/Desktop/jobs\_in\_data.csv")

# Copy the dataset

-> saving the dataset in another variable helps to retain the data if there is any problem during the processing of data

data\_jobs\_copy <- data\_jobs

# check null values

-> Remove null values if any in the dataset so that there is no wrong information.

sum(is.na(data\_jobs\_copy))

# Converting all categorical and ordinal columns to dummy variables

exp\_ <- data\_jobs\_copy$experience\_level

emp\_type\_ <- data\_jobs\_copy$employment\_type

work\_type\_ <- data\_jobs\_copy$work\_setting

size\_com\_ <- data\_jobs\_copy$company\_size

one\_hot\_encoded <- model.matrix(~ exp\_ - 1, data = data\_jobs\_copy)

data\_jobs\_copy<- cbind(data\_jobs\_copy, one\_hot\_encoded)

one\_hot\_encoded <- model.matrix(~ emp\_type\_ - 1, data = data\_jobs\_copy)

data\_jobs\_copy<- cbind(data\_jobs\_copy, one\_hot\_encoded)

one\_hot\_encoded <- model.matrix(~ work\_type\_ - 1, data = data\_jobs\_copy)

data\_jobs\_copy<- cbind(data\_jobs\_copy, one\_hot\_encoded)

one\_hot\_encoded <- model.matrix(~ size\_com\_ - 1, data = data\_jobs\_copy)

data\_jobs\_copy<- cbind(data\_jobs\_copy, one\_hot\_encoded)

# Selecting only required columns

data\_jobs\_selected\_col <- select(data\_jobs\_copy,c(1,5,13:26))

# Scaling the data

scaled\_data <- scale(x=data\_jobs\_selected\_col)

# View(scaled\_data)

# Performing principal component analysis

prcomp\_scaled\_data <- prcomp(scaled\_data,center = TRUE,rank = 2)

prcomp\_scaled\_data

Standard deviations (1, .., p=16):

[1] 1.714839e+00 1.399192e+00 1.356221e+00 1.354873e+00 1.083688e+00 1.056982e+00 1.014798e+00

[8] 9.889024e-01 9.606870e-01 9.131104e-01 8.440032e-01 8.112995e-01 1.283275e-14 9.034232e-15

[15] 2.578603e-15 9.330031e-16

Rotation (n x k) = (16 x 2):

PC1 PC2

work\_year -0.32160324 0.082896038

salary -0.21789625 -0.118271119

exp\_Entry-level 0.23263080 0.098256825

exp\_Executive 0.01868173 0.024237512

exp\_Mid-level 0.17858021 0.313335181

exp\_Senior -0.28135845 -0.336243356

emp\_type\_Contract 0.18440938 -0.049993666

emp\_type\_Freelance 0.14091991 -0.022906650

emp\_type\_Full-time -0.30026754 0.046783113

emp\_type\_Part-time 0.19103882 -0.005026241

work\_type\_Hybrid 0.29444504 0.085293236

work\_type\_In-person -0.23385828 0.587580646

work\_type\_Remote 0.14997629 -0.618908432

size\_com\_L 0.31324511 0.098560877

size\_com\_M -0.40704948 -0.082563030

size\_com\_S 0.27451614 -0.017812790

biplot(prcomp\_scaled\_data)

A black and red text on a white background

Description automatically generated

pca\_scores <- as.data.frame(prcomp\_scaled\_data$x[, 1:2])

ggplot(pca\_scores, aes(x = PC1, y = PC2,color = data\_jobs$salary)) +

geom\_point() +theme\_minimal()

A graph with numbers and lines

Description automatically generated with medium confidence

autoplot(

object = prcomp\_scaled\_data,

data = data\_jobs\_copy,

colour = data\_jobs$salary

)

A graph of different colored dots

Description automatically generated with medium confidence

factoextra::fviz\_eig(prcomp\_scaled\_data)

-> The plot tells us that at max 5 principal components will be useful for the overall estimation of data.

A graph with a line going up

Description automatically generated

n <- nrow(data\_jobs\_selected\_col)

# sampled\_rows <- sample(1:n, 350, replace = FALSE, prob = NULL)

# data\_jobs\_selected\_col\_sample <- data\_jobs\_selected\_col[sampled\_rows,]

# data\_jobs\_selected\_col\_sample

set.seed(823)

X <- data.frame(data\_jobs\_selected\_col[,-1])

X <- X[,sapply(X = X,FUN = max) > 0]

Rtsne\_1 <- Rtsne::Rtsne(

X = unique(X)

)

plot(

Rtsne\_1$Y,

col = data\_jobs$salary,

main = "Scatterplot of MNIST T-SNE two dimensions"

)

A graph of a scatterplot

Description automatically generated

nmf\_data <- data\_jobs\_selected\_col[rowSums(data\_jobs\_selected\_col) > 0,colSums(data\_jobs\_selected\_col) > 0]

nmf\_result <- NMF::nmf(

x = data\_jobs\_selected\_col,

rank = 5

)

nmf\_result

<Object of class: NMFfit>

# Model:

<Object of class:NMFstd>

features: 9355

basis/rank: 2

samples: 16

# Details:

algorithm: brunet

seed: random

RNG: 10403L, 548L, ..., -200570180L [483c731af26bf166b98a11fa8000aae1]

distance metric: 'KL'

residuals: 75661.07

Iterations: 820

Timing:

user system elapsed

0.55 0.02 0.95

basis\_vals <- NMF::basis(

object = nmf\_acq

)

coef\_vals <- NMF::coef(

object = nmf\_acq

)

coef\_vals

work\_year salary exp\_Entry-level exp\_Executive exp\_Mid-level exp\_Senior

[1,] 0.0012099719 0.31745531 2.220446e-16 5.447758e-08 2.220446e-16 1.053404e-06

[2,] 0.0005580412 0.20034928 2.220446e-16 4.230910e-08 2.220446e-16 5.935745e-07

[3,] 0.0031138429 0.02167448 1.082631e-07 4.506969e-09 4.513551e-07 5.170838e-07

[4,] 0.0005377346 0.10246349 2.220446e-16 3.088846e-08 2.220446e-16 3.765323e-07

[5,] 0.0028064338 0.02277727 1.003065e-07 2.220446e-16 3.338273e-07 5.235426e-07

emp\_type\_Contract emp\_type\_Freelance emp\_type\_Full-time emp\_type\_Part-time

[1,] 2.220446e-16 2.220446e-16 6.128715e-07 2.220446e-16

[2,] 2.220446e-16 2.220446e-16 2.874135e-07 2.220446e-16

[3,] 5.838413e-09 3.464001e-09 1.512494e-06 4.350685e-09

[4,] 1.201622e-09 2.220446e-16 2.663787e-07 2.220446e-16

[5,] 9.476715e-10 1.143458e-09 1.372299e-06 1.938555e-09

work\_type\_Hybrid work\_type\_In-person work\_type\_Remote size\_com\_L size\_com\_M

[1,] 2.220446e-16 5.145335e-07 1.288251e-07 3.190322e-08 6.270251e-07

[2,] 2.220446e-16 3.149572e-07 3.645220e-09 7.609705e-09 3.056308e-07

[3,] 5.512667e-08 7.682049e-07 6.654789e-07 1.509838e-07 1.294784e-06

[4,] 2.220446e-16 2.069726e-07 7.519074e-08 7.698364e-09 2.758194e-07

[5,] 2.496096e-08 7.266488e-07 5.943666e-07 1.223866e-07 1.190314e-06

size\_com\_S

[1,] 2.220446e-16

[2,] 2.220446e-16

[3,] 3.902128e-08

[4,] 2.220446e-16

[5,] 2.776534e-08

basis\_vals

1 1.250942e+05 1.098760e+05 543855.78361 1.379248e+05 15533.1916

2 4.439353e+05 9.915854e+04 313837.84264 1.501209e+05 132596.3999

3 1.254534e+04 1.392901e+05 336485.83534 3.614545e+05 245071.9746

4 9.117812e+03 7.698122e+05 188647.38223 4.354217e+05 270921.1432

5 1.439308e+05 1.024572e+05 194880.70785 1.349968e+05 396245.4007

6 2.844713e+05 9.483658e+04 109424.17478 8.043620e+04 442512.5541

7 1.725660e+05 9.275713e+04 173333.27776 1.320479e+05 410379.6916

8 3.191087e+05 4.349356e+05 41866.81298 2.526613e+05 402207.6235

9 4.479169e+04 4.941415e+05 158219.69918 1.254238e+05 403851.8973

10 2.625776e+05 4.355261e+05 371533.49190 2.941647e+05 52339.8625

11 3.577214e+05 1.171427e+05 260296.09166 1.991355e+05 216201.6458

12 3.956165e+05 2.528281e+05 121403.50952 3.792854e+05 292463.4238

13 1.136946e+05 3.759368e+05 214607.97976 1.474432e+05 330579.3929

14 3.367199e+04 2.342833e+04 503383.03059 5.548055e+04 132651.4837

15 4.230749e+04 1.834791e+02 346259.53078 1.807157e+04 315042.7084

16 1.616457e+05 1.139789e+05 287241.46737 7.710057e+04 295164.1637

17 1.837814e+05 1.027294e+03 258909.17859 2.823949e+04 348872.0596

18 9.770792e+04 1.045772e+06 244264.79585 5.058462e+05 102627.7621

19 2.698854e+05 3.584218e+05 145180.99993 6.615255e+05 245208.9459

20 2.108812e+05 2.456431e+05 359642.82886 1.212439e+05 158999.5097

[ reached getOption("max.print") -- omitted 9155 rows ]

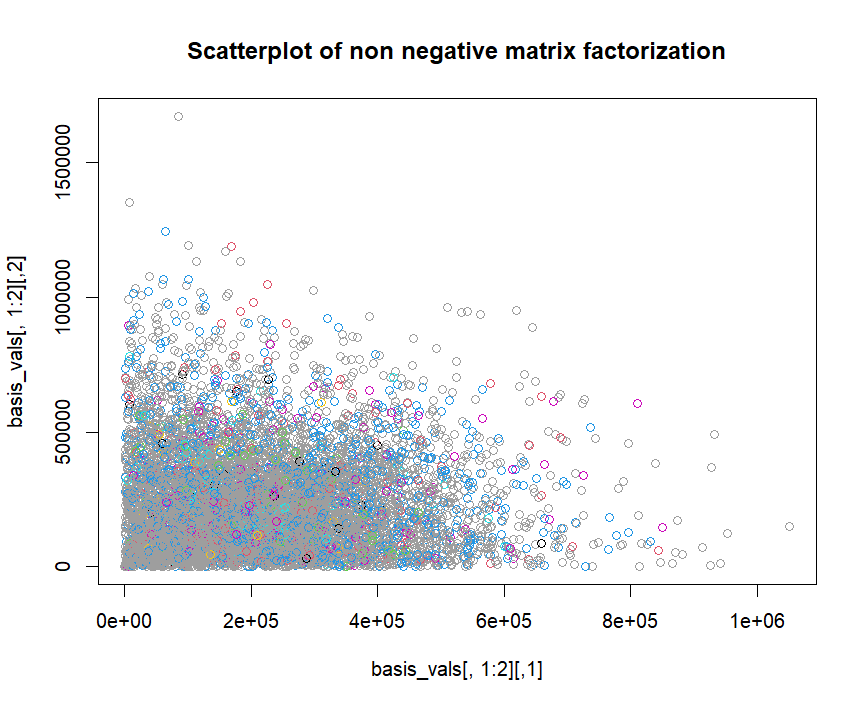
plot(

basis\_vals[,1:2],

col = data\_jobs$salary,

main = "Scatterplot of MNIST PCA two dimensions"

)



RESULT :

After performing the specified analysis and instructions, using PCA I was able to identify outliers very easily and at max 5 principal components are sufficient to retain the important information in the data. I also found that t-SNE is more advanced that is compared to PCA, t-SNE helps us to differentiate or group the non-linear data by reducing the dimensionality while keeping the similar data points together. The graph generated using t-SNE has different shapes of groups but the group contains different class values making it ineffective. Increase the performance values merging all the values into a single group. The non negative matrix factorization is not very useful since the overall residuals for the data is very high. Usually low residual value indicates a better fit.