# Exploring fashion MNIST dataset

ADVANCED DIMENSIONALITY REDUCTION IN R

**Federico Castanedo**
Data Scientist at DataRobot

# What is Fashion MNIST?

- **70.000** grayscale images of **10** clothing categories
- **28x28** pixels
- Identical format to traditional MNIST
- Released by Zalando
- With the goal of replacing MNIST, because:
  - MNIST is easy to predict
  - MNIST is overused
  - MNIST does not represent modern computer vision tasks

# Data exploration

## Dimensionality

```
dim(fashion_mnist)
```

```
60000   785
```

## Target class distribution

```
table(fashion_mnist$label)
```

```
   0    1    2    3    4    5    6    7    8    9
6000 6000 6000 6000 6000 6000 6000 6000 6000 6000
```

# Summary statistics

- Summary statistics of the first 4 pixels from class 0 (t-shirt)

```
summary(fashion_mnist[label==0, 2:5])
```

```
    pixel1              pixel2             pixel3            pixel4
 Min.    :0.000000   Min.    : 0.00000   Min.    : 0.0000   Min.    :  0.0000
 1st Qu.:0.000000   1st Qu.: 0.00000   1st Qu.: 0.0000   1st Qu.:  0.0000
 Median :0.000000   Median : 0.00000   Median : 0.0000   Median :  0.0000
 Mean   :0.001333   Mean   : 0.01583   Mean   : 0.1438   Mean   :  0.3327
 3rd Qu.:0.000000   3rd Qu.: 0.00000   3rd Qu.: 0.0000   3rd Qu.:  0.0000
 Max.    :7.000000   Max.    :11.00000   Max.    :78.0000   Max.    :132.0000
```

# Data visualization

Class names

```
class_names <- c('T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat',
                 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot')
```

Auxiliary data frame

```
xy_axis <- data.frame(x = expand.grid(1:28, 28:1)[,1],
                      y = expand.grid(1:28, 28:1)[,2])
```

# Data visualization

Generate a data frame with `x` , `y` , and the pixel value

```
plot_data <- cbind(xy_axis, fill = as.data.frame(t(fashion_mnist[1, -1]))[,1])
```

Calling ggplot

```
ggplot(plot_data, aes(x, y, fill = fill)) +
    ggtitle(class_names[as.integer(fashion_mnist[1,1])+1]) +
    plot_theme
```

# Custom ggplot theme

- Helps to plot the images

```
plot_theme <- list(
  raster = geom_raster(hjust = 0, vjust = 0),
  gradient_fill = scale_fill_gradient(low = "white",
                                      high = "black", guide = FALSE),
  theme = theme(axis.line = element_blank(),
                axis.text = element_blank(),
                axis.ticks = element_blank(),
                axis.title = element_blank(),
                panel.background = element_blank(),
                panel.border = element_blank(),
                panel.grid.major = element_blank(),
                panel.grid.minor = element_blank(),
                plot.background = element_blank())
)
```

Pullover

# Practical exercises!

ADVANCED DIMENSIONALITY REDUCTION IN R

# Generalized Low Rank Models (GLRM)

ADVANCED DIMENSIONALITY REDUCTION IN R

**Federico Castanedo**
Data Scientist at DataRobot

# Benefits of GLRMs

- Reduces the required storage
- Enables data visualization
- Removes noise
- Imputes missing data
- Simplifies data processing

# Low rank structure

$$\boxed{m\left\{\overbrace{\begin{bmatrix} \quad & & \\ & A & \\ & & \quad \end{bmatrix}}^{n}\right.} \approx m\left\{\overbrace{\begin{bmatrix} \\ X \\ \\ \end{bmatrix}}^{k}\right. \overbrace{\begin{bmatrix} \quad & Y & \quad \end{bmatrix}}^{n}\Big\}\, k$$

# Low rank structure

$$m\left\{\overbrace{\begin{bmatrix} \quad & \quad & \quad \\ & A & \\ & & \end{bmatrix}}^{n}\right. \approx m\boxed{\left\{\overbrace{\begin{bmatrix} \\ X \\ \\ \end{bmatrix}}^{k}\right.} \left.\overbrace{\begin{bmatrix} \quad & Y & \quad \end{bmatrix}}^{n}\right\}k$$

# Low rank structure

$$m\left\{\overbrace{\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix}}^{n}\right. \approx m\left\{\overbrace{\begin{bmatrix} \\ X \\ \\ \end{bmatrix}}^{k}\right. \boxed{\overbrace{\begin{bmatrix} & Y & \end{bmatrix}}^{n}\Big\}\, k}$$

# Generalized low rank models (GLRM)

- Parallelized dimensionality reduction algorithm
- Categorical columns are transformed into binary columns

$$m\left\{\overbrace{\begin{bmatrix} \quad & A & \quad \end{bmatrix}}^{n}\right. \approx m\left\{\overbrace{\begin{bmatrix} X \end{bmatrix}}^{k}\right. \left.\overbrace{\begin{bmatrix} \quad & Y & \quad \end{bmatrix}}^{n}\right\}k$$

# Generalized low rank models (GLRM)

- Each row of X is an example projected in the new low-dimensional space
- Each row of Y is an archetypal feature formed from the columns of A

$$
m\left\{ \overbrace{\begin{bmatrix} \quad & A & \quad \end{bmatrix}}^{n} \right. \approx m\left\{ \overbrace{\begin{bmatrix} X \end{bmatrix}}^{k} \right. \overbrace{\begin{bmatrix} \quad & Y & \quad \end{bmatrix}}^{n} \left. \vphantom{Y} \right\} k
$$

# GLRM in R with H2O

- `H2O` is an open source machine learning framework with R interfaces
- Has a good parallel implementation of GLRM
- Steps: (1) initialize the cluster and (2) store the input data

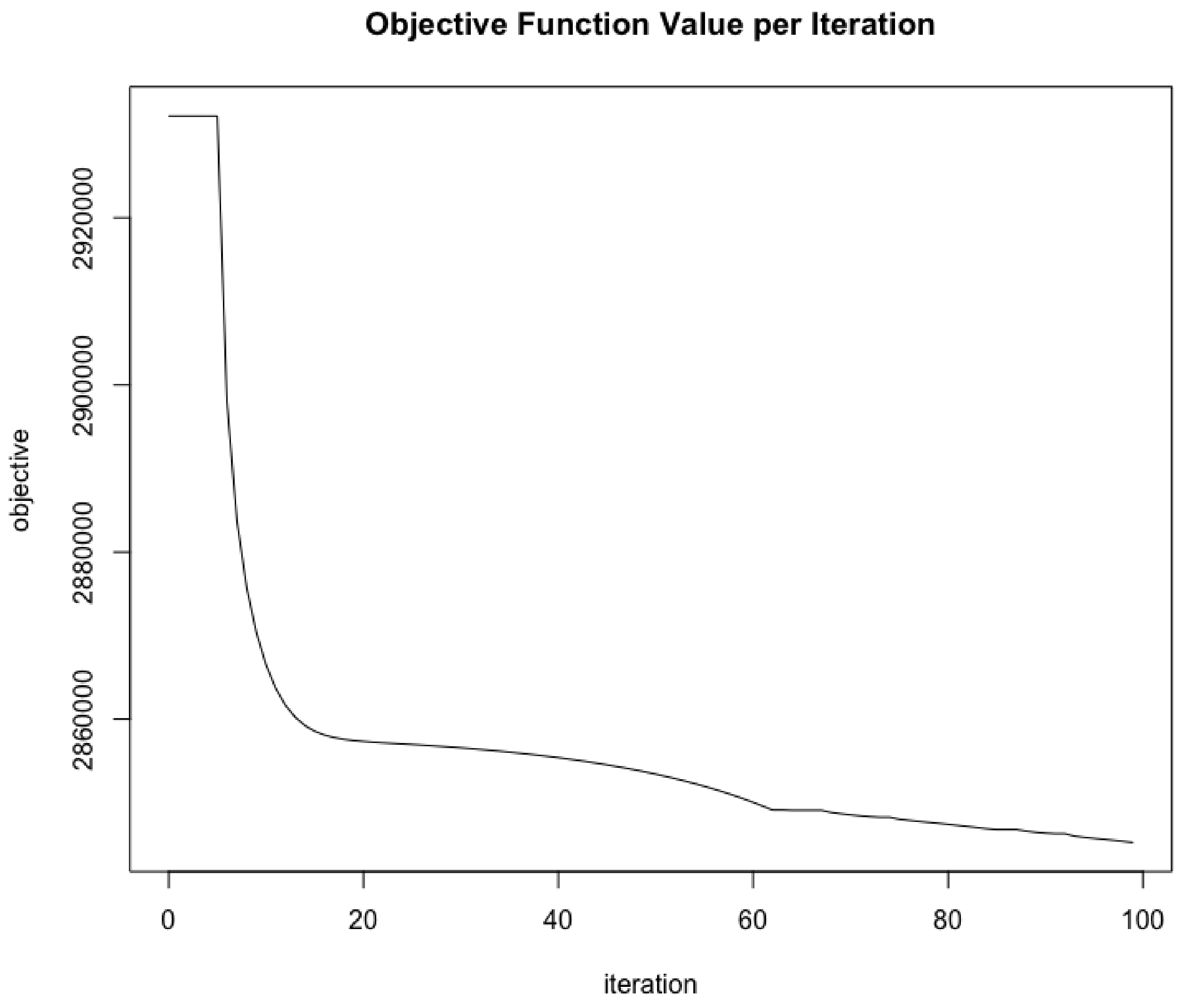
```
# Start a connection with the h2o cluster
h2o.init()

# Store the data into h2o cluster
fashion_mnist.hex <- as.h2o(fashion_mnist, "fashion_mnist.hex")
```

- Build a GLRM model

```
model_glrm <- h2o.glrm(training_frame = fashion_mnist.hex,
                       cols = 2:ncol(fashion_mnist), k = 2,
                       max_iterations = 100)
```

# Objective function value per iteration

```
plot(model_glrm)
```



# Lets practice!

ADVANCED DIMENSIONALITY REDUCTION IN R

# Visualizing a GLRM model

ADVANCED DIMENSIONALITY REDUCTION IN R

**Federico Castanedo**
Data Scientist at DataRobot

# XY decomposition

$$m\left\{\overbrace{\begin{bmatrix} \quad & & \quad \\ & A & \\ & & \end{bmatrix}}^{n}\right. \approx m\left\{\overbrace{\begin{bmatrix} \\ X \\ \\ \end{bmatrix}}^{k}\right. \left.\overbrace{\begin{bmatrix} \quad & Y & \quad \end{bmatrix}}^{n}\right\} k$$

# Getting the XY decomposition

X low-dimensional representation

```
X <- as.data.table(h2o.getFrame(model_glrm@model$representation_name))
```

```
head(X)
```

```
        Arch1      Arch2
1  0.05700855 -0.1639649
2 -0.38297093 -0.4796468
3 -0.04675919  0.5104198
4  0.50123594 -0.3073703
5  0.12971048  0.1678937
6 -0.41766714 -0.3275673
```

# Getting the XY decomposition

Y matrix

```
Y <- model_glrm@model$archetypes
dim(Y)
```

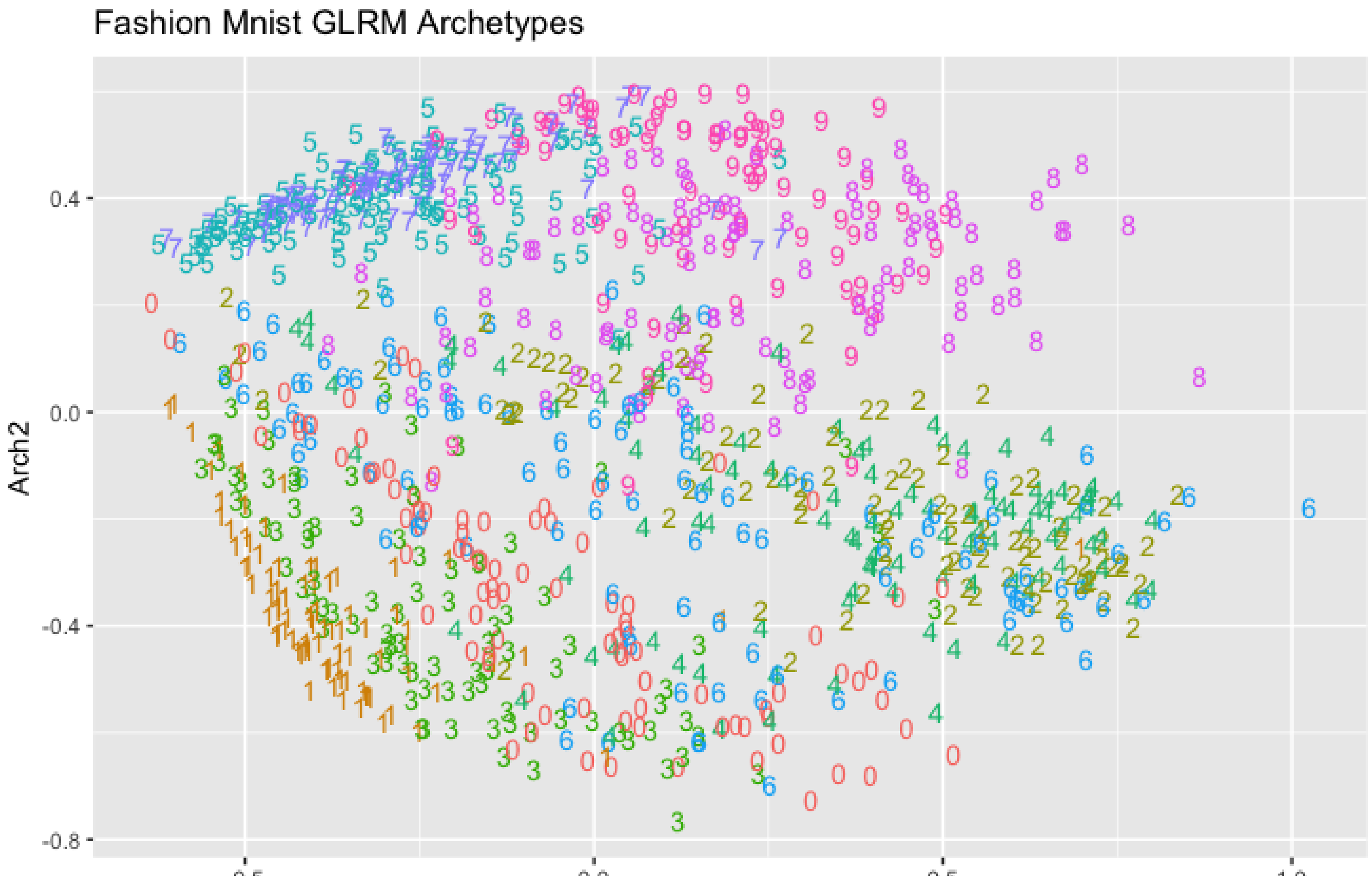
```
2 784
```

```
head(Y[,1:5])
```

```
      pixel1       pixel2        pixel3        pixel4        pixel5
Arch1      0  0.001267437 -0.0004790154 -0.0015502976  0.0013502380
Arch2      0 -0.002971832  0.0003699268 -0.0003715971 -0.0008029028
```

# Visualizing the obtained archetypes

```
ggplot(X, aes(x= Arch1, y = Arch2, color = fashion_mnist$label)) +
  ggtitle("Fashion Mnist GLRM Archetypes") +
  geom_text(aes(label = fashion_mnist$label)) + theme(legend.position="none")
```



# Visualizing the centroids of each class

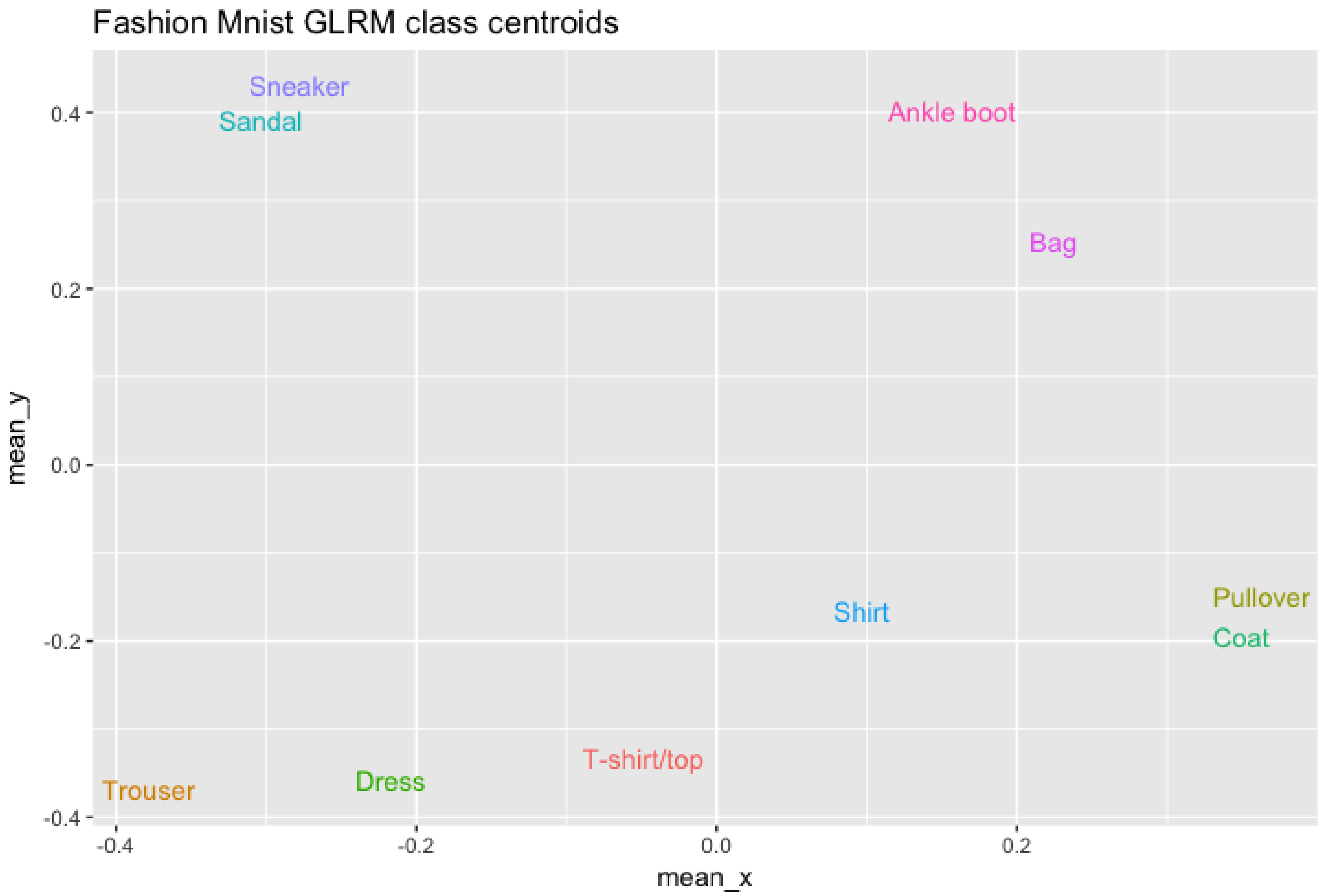
## Computing the centroids

```
X[, label := as.numeric(fashion_mnist$label)]
X[, mean_x := mean(Arch1), by = label]
X[, mean_y := mean(Arch2), by = label]
X_mean <- unique(X, by = "label")
```

```
class_names = c('T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat',
                'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot')
```

## Plotting the values

```
ggplot(X_mean, aes(x = mean_x, y = mean_y, color = as.factor(X_mean$label))) +
        ggtitle("Fashion Mnist GLRM class centroids") +
        geom_text(aes(label = class_names[label])) +
        theme(legend.position="none")
```



Fashion Mnist GLRM class centroids
Sneaker
Sandal
Ankle boot
Bag
Shirt
Pullover
Coat
T-shirt/top
Dress
Trouser
0.4
0.2
0.0
-0.2
-0.4
-0.4
-0.2
0.0
0.2
mean_y
mean_x

# Reconstruction of the original data

Computing X*Y

```
fashion_pred <- predict(model_glrm, fashion_mnist.hex)
```

Obtained dimensions

```
dim(fashion_pred)
```

```
1000  784
```

# First 4 pixels

First 4 pixels of the first two records

```
head(fashion_pred[1:2, 1:4])
```

```
  reconstr_pixel1 reconstr_pixel2 reconstr_pixel3 reconstr_pixel4
1               0    0.0005595307 -0.000087962973  -0.00002745136
2               0    0.0009400381  0.000006014762   0.00077195427
```

# Visualizing the reconstruction error

- Reconstructed input

```
xy_axis <- data.frame(x = expand.grid(1:28,28:1)[,1],
                      y = expand.grid(1:28,28:1)[,2])
```

```
data_reconstructed <- cbind(xy_axis,
  fill = as.data.frame(t(fashion_pred[1000,]))[,1])

plot_reconstructed <- ggplot(plot_data, aes(x, y, fill = fill)) +
  ggtitle("Reconstructed Pullover (K=2)") +
  plot_theme
```

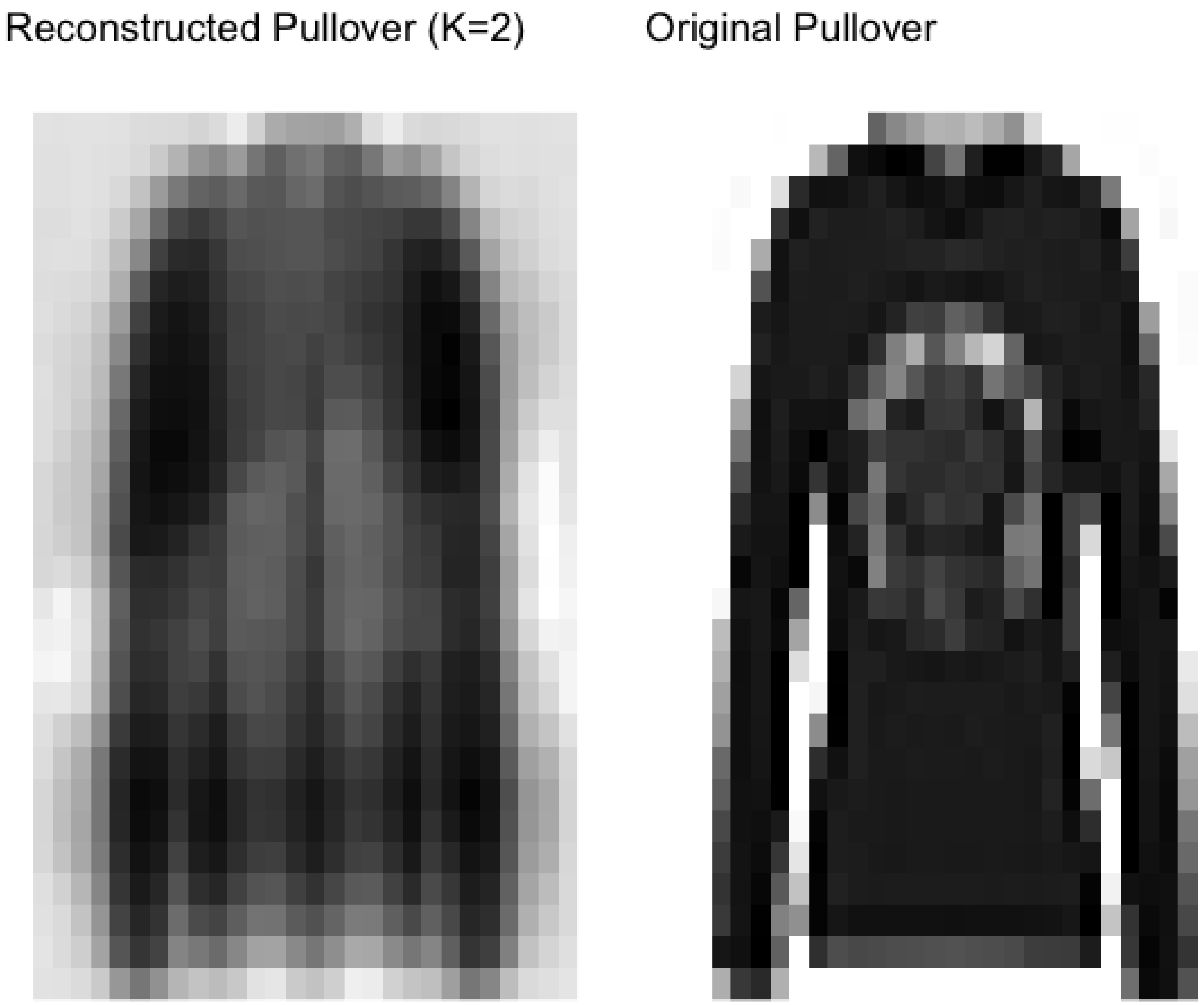
# Visualizing the reconstruction error

Original input

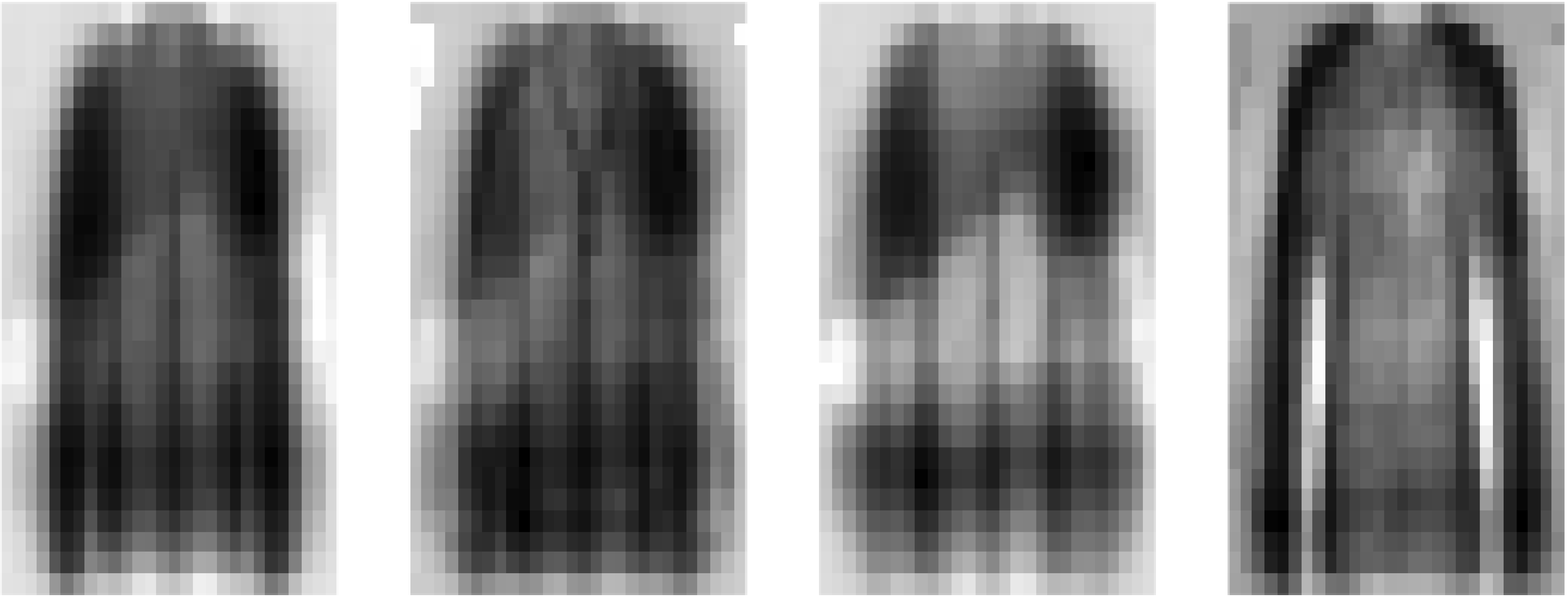
```
data_original <- cbind(xy_axis,
  fill = as.data.frame(t(fashion_mnist[1000, -1]))[,1])

plot_original <- ggplot(plot_data_2, aes(x, y, fill = fill)) +
  ggtitle("Original Pullover") +
  plot_theme
```

Plotting together

```
grid.arrange(plot_reconstructed, plot_original, nrow = 1)
```


Reconstructed Pullover (K=2)
Original Pullover

K=2 K=4 K=8 K=16



K=32 K=64 K=128 Original

# Let's dig into some examples!

ADVANCED DIMENSIONALITY REDUCTION IN R

# Dealing with missing data and speeding-up models

ADVANCED DIMENSIONALITY REDUCTION IN R

Federico Castanedo
Data Scientist at DataRobot

# Missing data

- Common in real-world datasets
  - Intentionally not provided
  - Due to an error
- With GLRM we can impute missing data and assign an estimation

# What to do with missing data

Example: randomly generate missing data

```
fashion_mnist_miss.hex <- h2o.insertMissingValues(fashion_mnist.hex[,-1],
                          fraction = 0.2, seed = 1234)
```

We now have missing values

# What to do with missing data

Example: randomly generate missing data

```
summary(fashion_mnist_miss[,781:784])
```

```
    pixel781          pixel782           pixel783           pixel784
 Min.   :  0.00    Min.   :  0.000    Min.   : 0.0000    Min.   :0
 1st Qu.:  0.00    1st Qu.:  0.000    1st Qu.: 0.0000    1st Qu.:0
 Median :  0.00    Median :  0.000    Median : 0.0000    Median :0
 Mean   :  8.29    Mean   :  2.342    Mean   : 0.3806    Mean   :0
 3rd Qu.:  0.00    3rd Qu.:  0.000    3rd Qu.: 0.0000    3rd Qu.:0
 Max.   :204.00    Max.   :171.000    Max.   :63.0000    Max.   :0
 NA's   :103       NA's   :97         NA's   :98         NA's   :98
```

# Filling missing data

Building a GLRM

```
model_glrm <- h2o.glrm(training_frame =
                       fashion_mnist_miss.hex, transform = "NORMALIZE",
                       ignore_const_cols = FALSE, k = 64,
                       max_iterations = 200, seed = 123)
```

Imputing missing data

```
fashion_pred <- h2o.predict(model_glrm, fashion_mnist_miss.hex)
```

# Observing the result

- Summary of the last 3 pixels

```
summary(fashion_pred[,782:784])
```

```
reconstr_pixel782   reconstr_pixel783   reconstr_pixel784
  Min.   :-0.130872   Min.   :-0.154723   Min.   :0
  1st Qu.:-0.032020   1st Qu.:-0.027012   1st Qu.:0
  Median :-0.007367   Median : 0.001272   Median :0
  Mean   : 0.001873   Mean   : 0.002914   Mean   :0
  3rd Qu.: 0.020030   3rd Qu.: 0.025293   3rd Qu.:0
  Max.   : 0.822162   Max.   : 0.821948   Max.   :0
```

# Speeding up machine learning models

- Another advantage of GLRM
- Training machine learning models is faster using a low-dimensional representation
- Key is to have a good compressed representation

# Training a random Forest and measuring the time

```
time_start <- proc.time()
```

```
rf_model <- randomForest(x = fashion_mnist[, -1],
                         y = fashion_mnist$label,
                         ntree = 20)
```

```
time_end <- timetaken(time_start)
```

# Experiments with Fashion MNIST

- Trained several `h2o` random forests, 4-Fold Cross-Validation
- Fashion MNIST (60.000) was compressed with GLRM and changing the value of K from 2 to 256
- We measure the accuracy and the required time

```
perf_metrics
```

```
   k_values   mean_acc time_taken
1:        0 0.88098335   00:52:17
2:        2  0.5134107   00:02:37
3:        4 0.61005294   00:03:07
4:        8  0.7339327   00:03:34
5:       16 0.80530137   00:05:17
6:       32 0.86116403   00:07:26
7:       64 0.85694784   00:18:21
8:      128  0.8648633   00:16:37
9:      256 0.86634624   00:32:41
```

# Practice!

ADVANCED DIMENSIONALITY REDUCTION IN R

# Summary of the course

ADVANCED DIMENSIONALITY REDUCTION IN R

**Federico Castanedo**
Data Scientist at DataRobot

# Advanced dimensionality reduction

- **Algorithms**: t-SNE and GLRM.
- **Ability**: extract useful representation in low-dimensional space.
- **Advantages**: simplify data processing, ability to visualize high dimensional data, space and time reduction, a way of doing feature selection and in the case of GLRM it can also impute missing data.

# Congratulations!

ADVANCED DIMENSIONALITY REDUCTION IN R