

# Discriminately Boosted Clustering

Zarmina Ursino  
[zarmina.ursino@studio.unibo.it](mailto:zarmina.ursino@studio.unibo.it)

# Summary



Data exploration

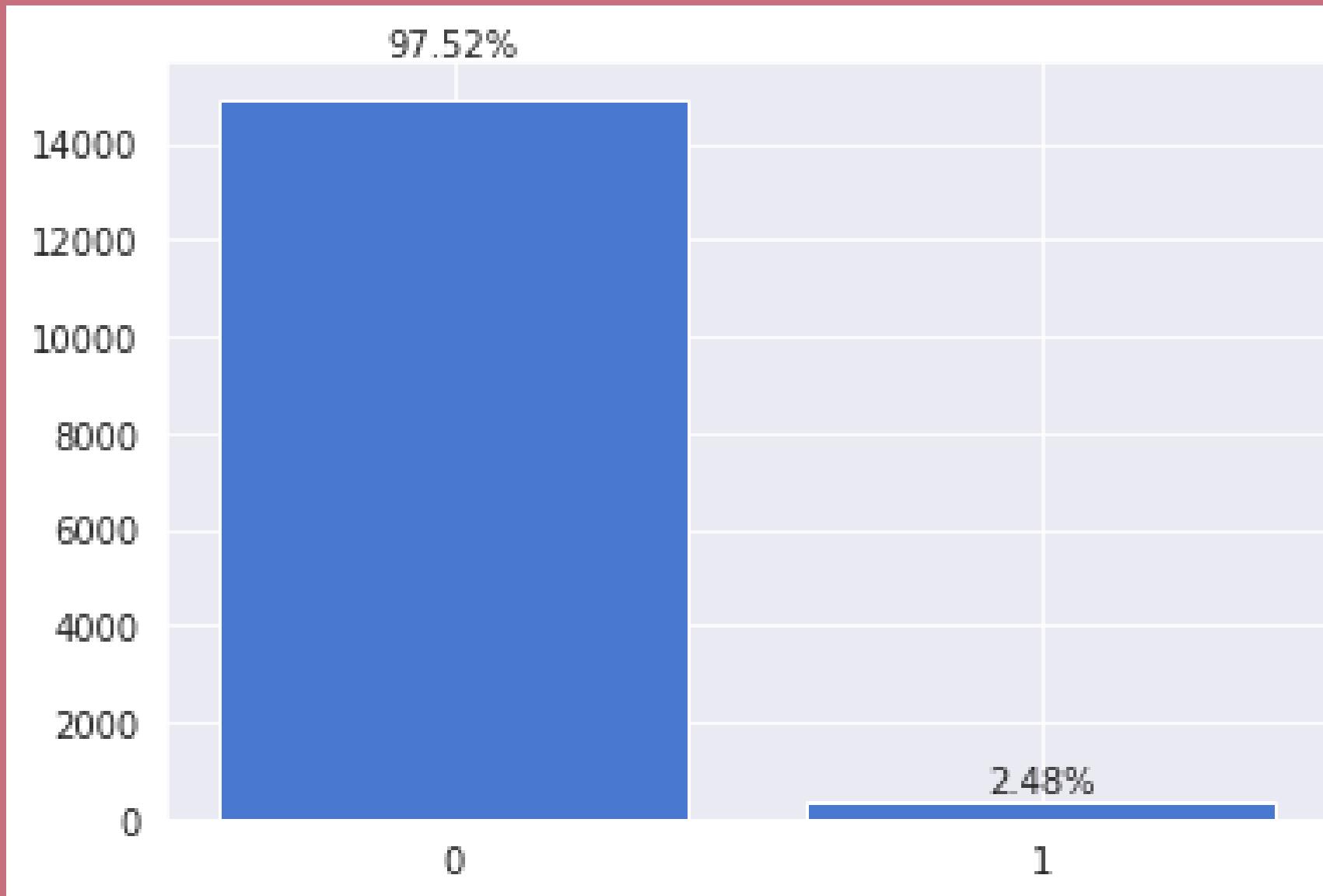
Convolutional Autoencoders

Results

Evaluation metrics

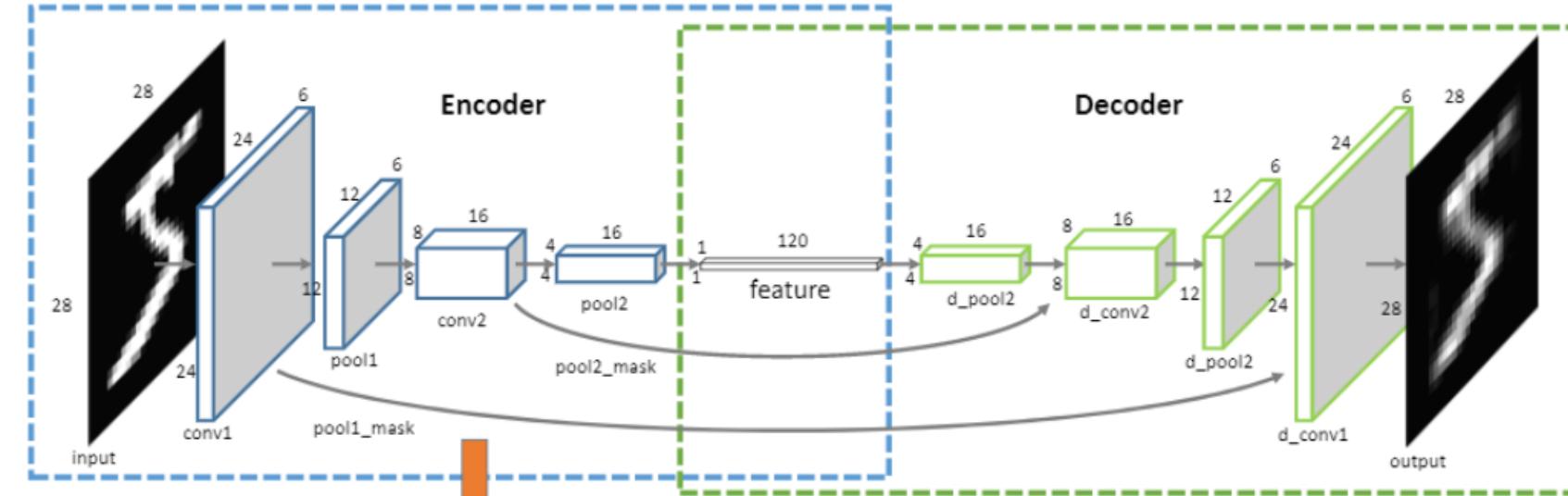
# Data exploration

- Removal of non-significant columns
- Checking missing data
- Cleaning the target
- Splitting into X and y
- Scaling of features

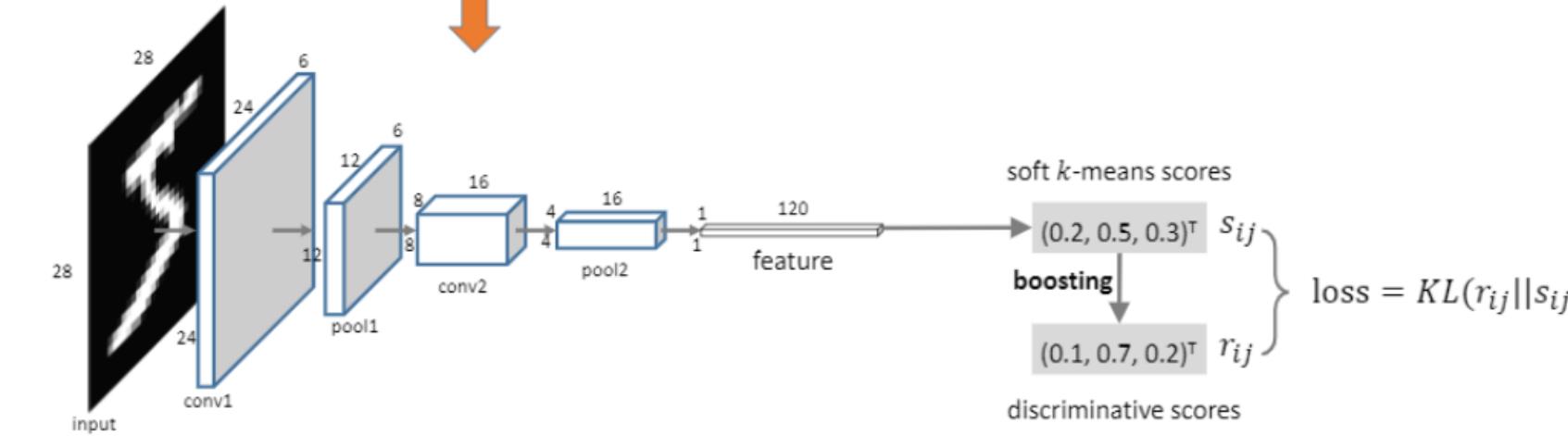


# Model

I. FCAE

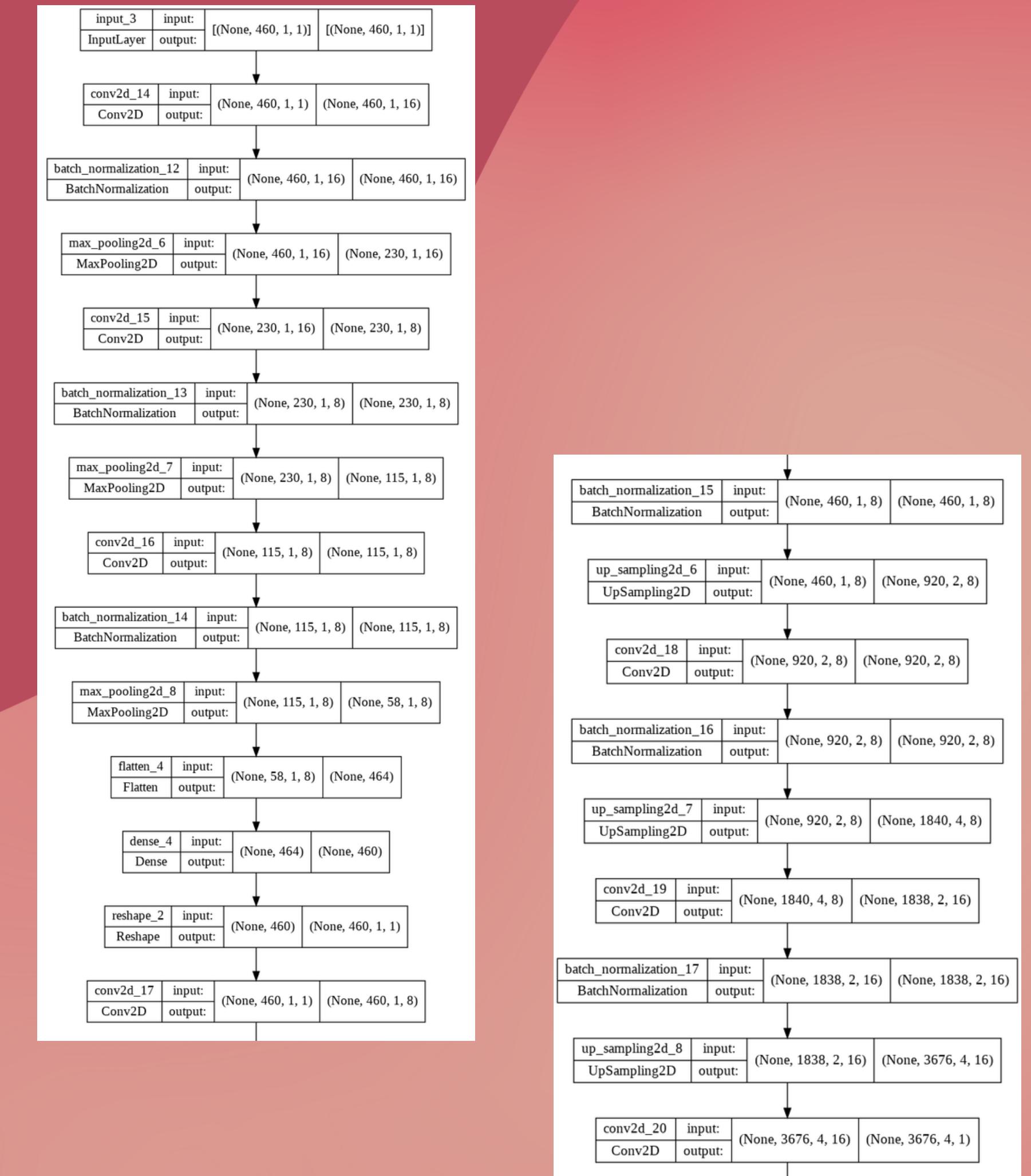


II. DBC

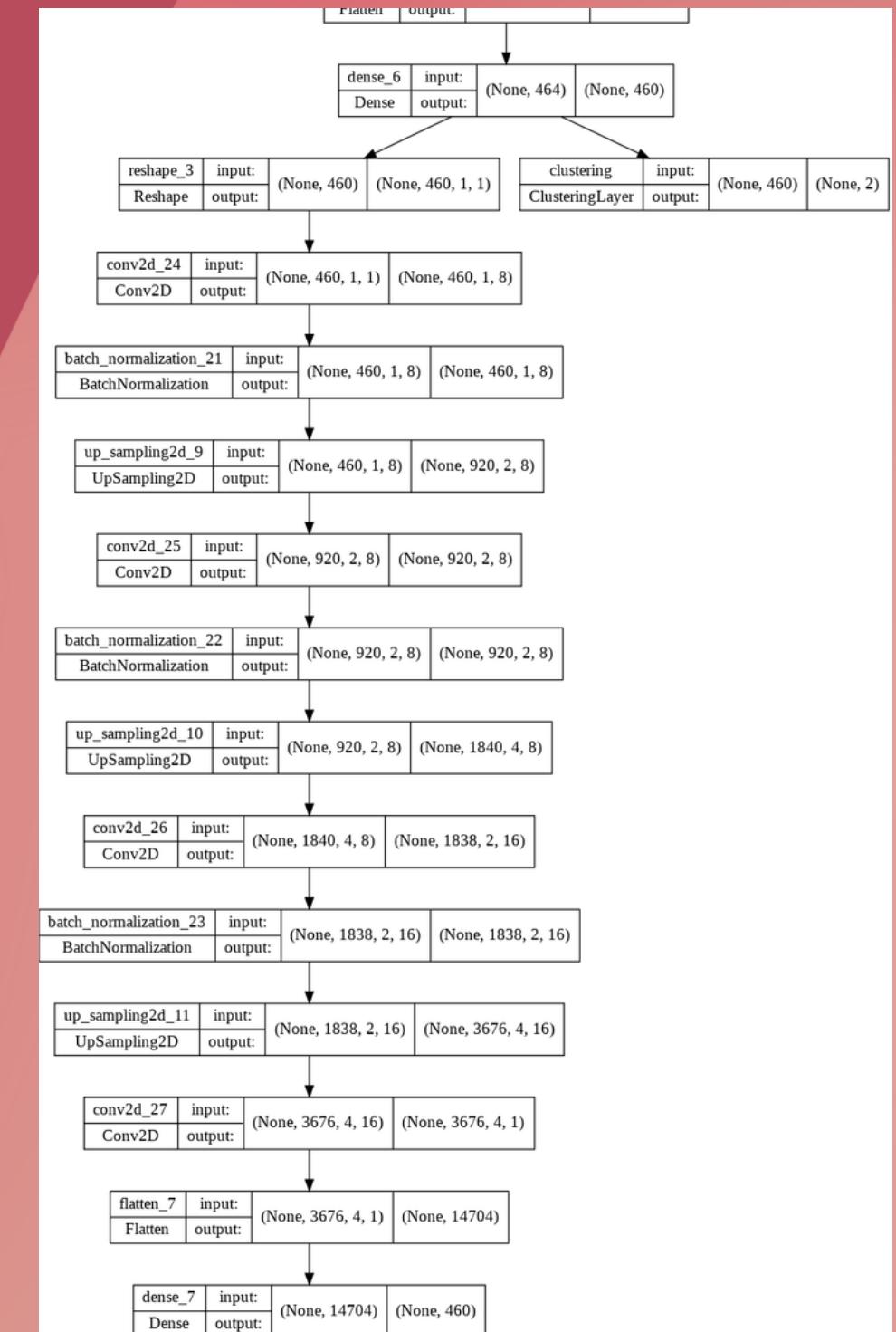
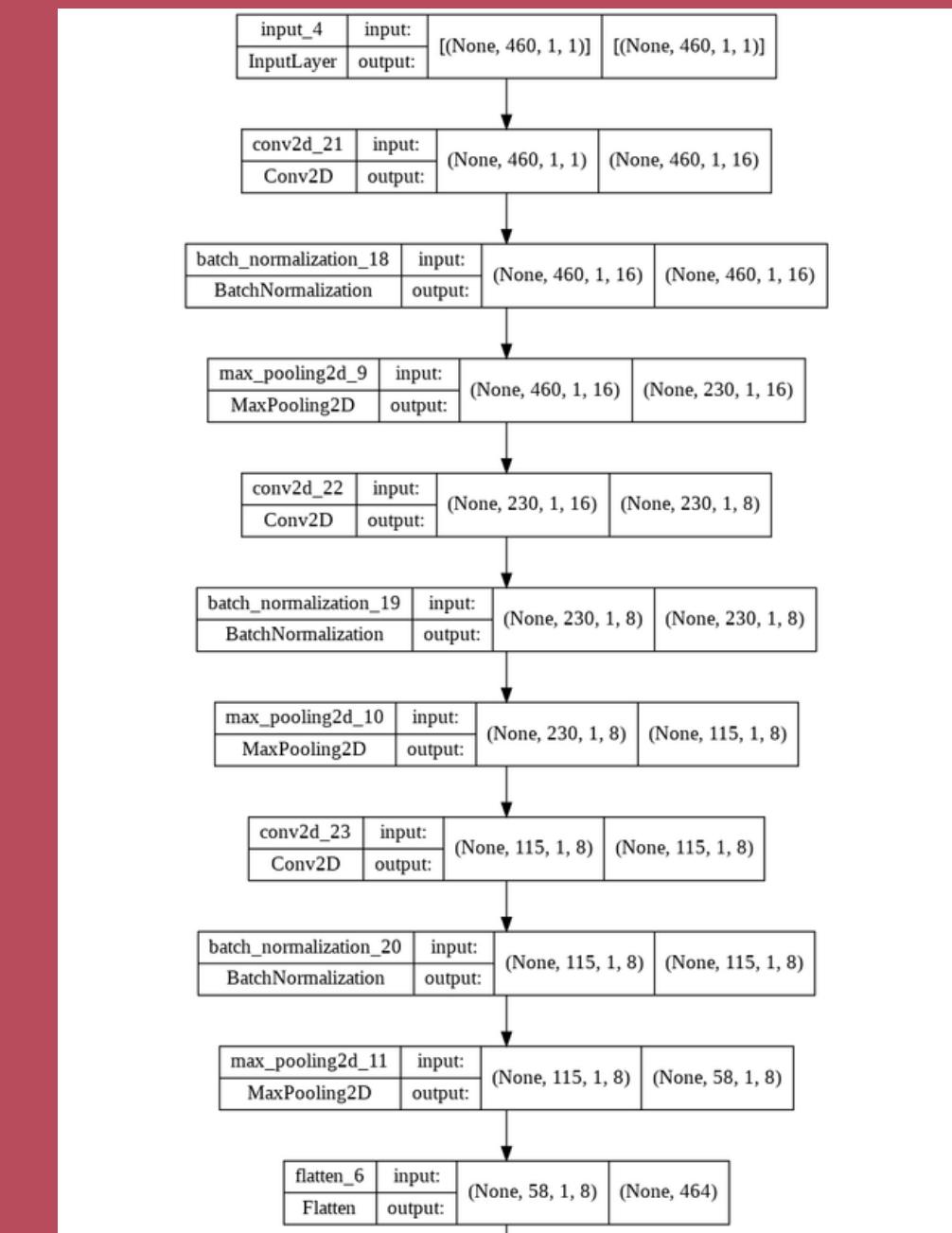


# First step: FCAE

- Fully convolutional autoencoder
- Batch Normalization strategy
- Symmetric architecture



# Second Step: DBC



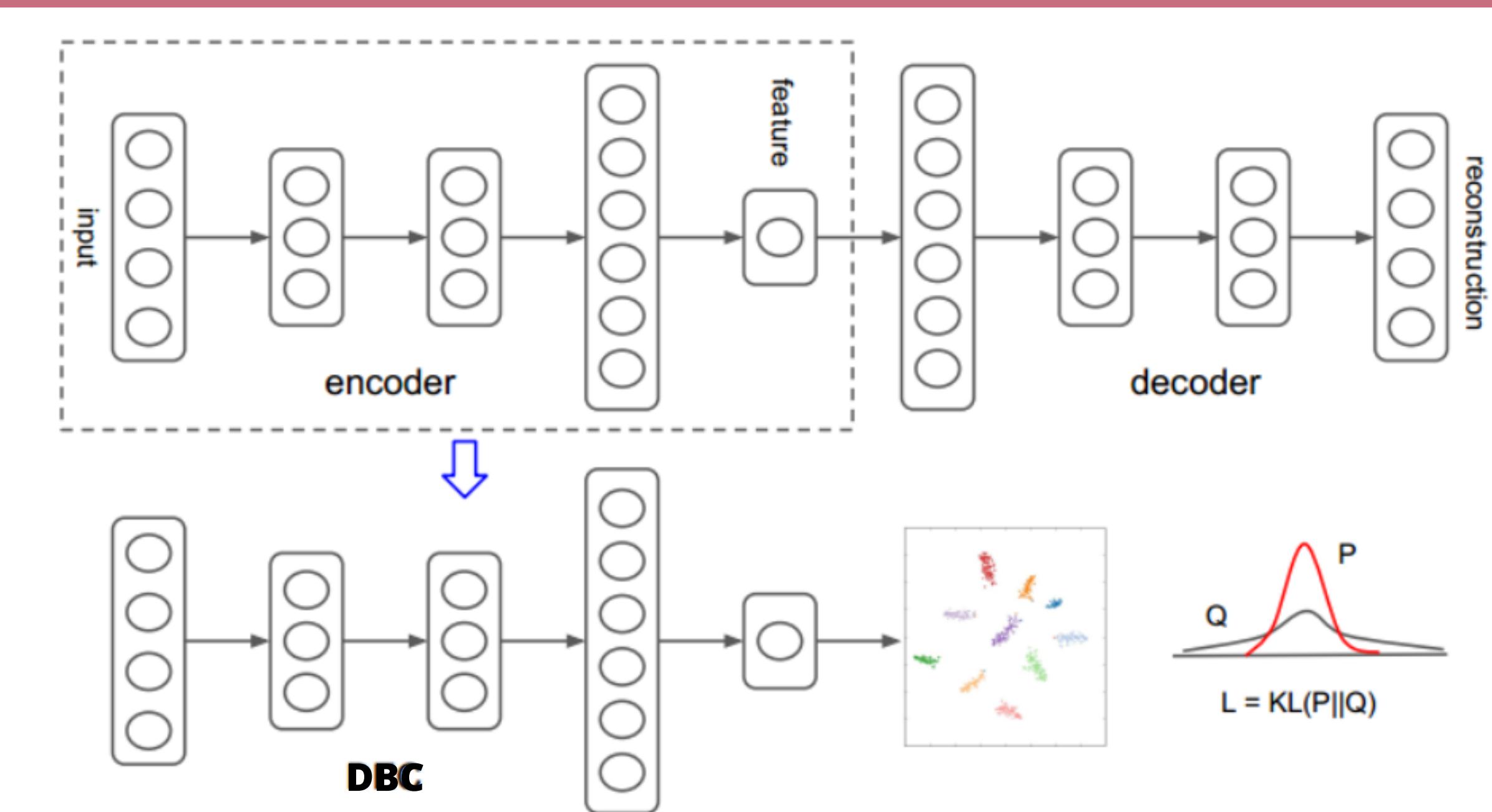
## Clustering Layer:

- `build(input_shape)`
- `call(x)`
- `compute_output_shape(input_shape)`

# Kmeans

Advantages:

- Easy usage
- Fast
- Many uses cases
- Inertia



# Training Clustering Model

I used the target distribution which have the following properties:

- Strengthen predictions
- Put more emphasis
- Prevent distortion

Optimizer: Adam

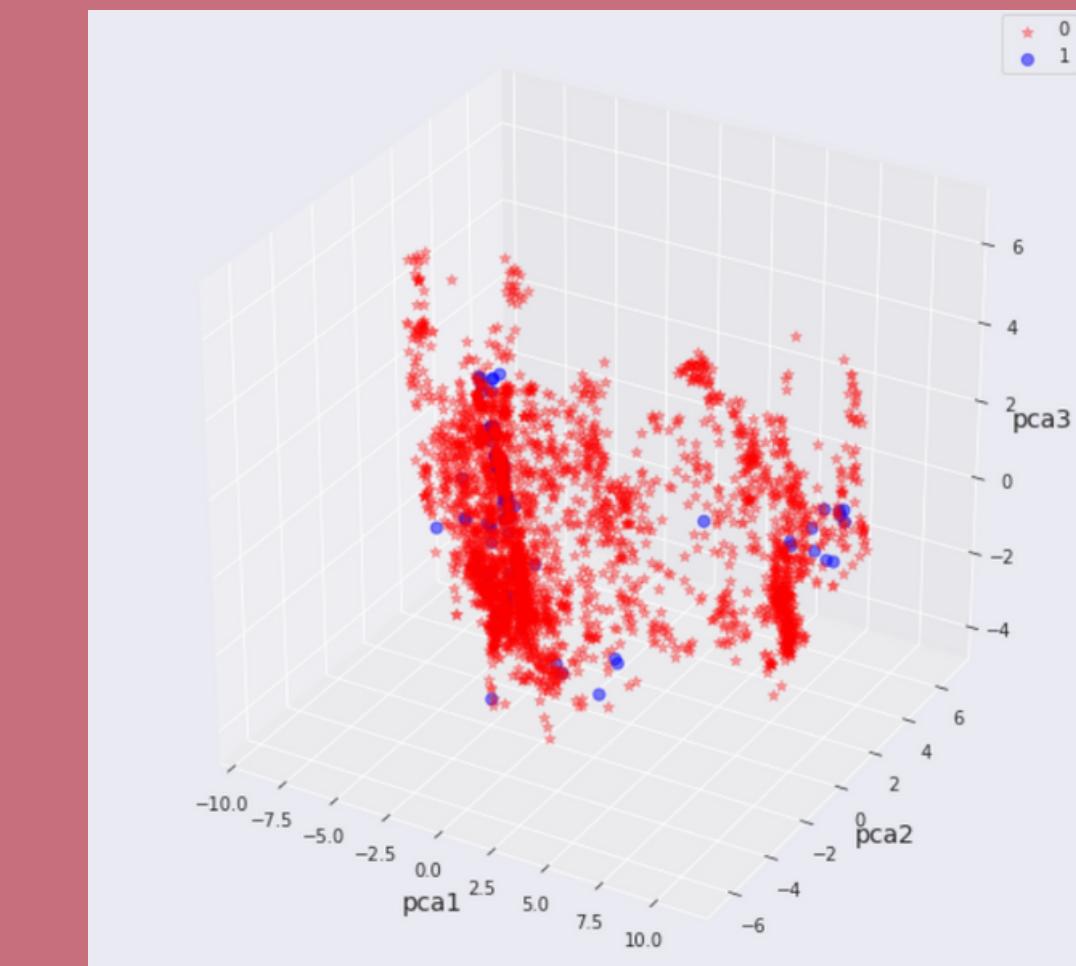
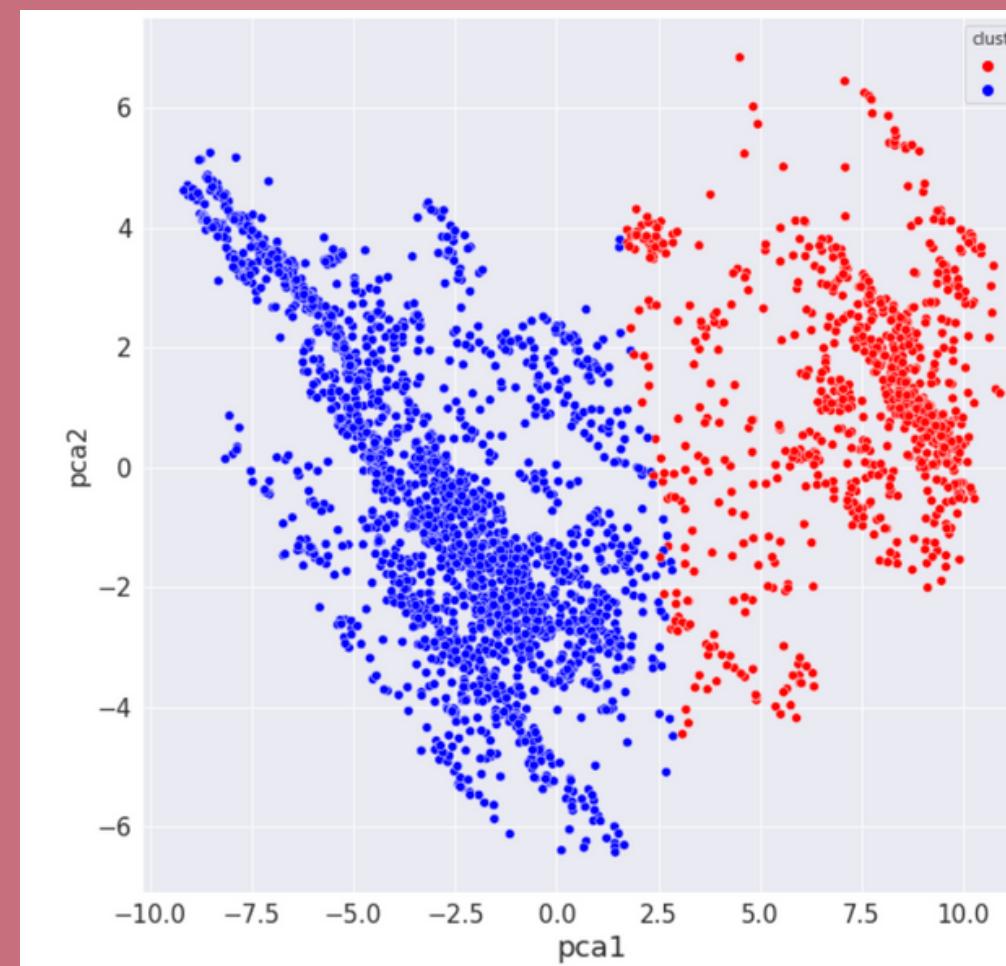
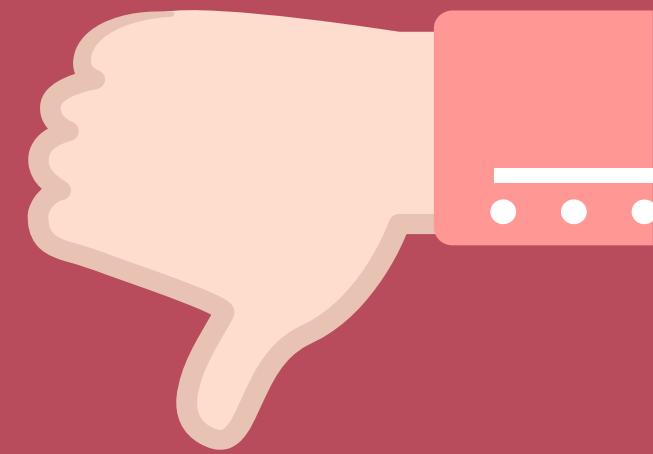
Loss function: Kullback-Leibler divergence (KL divergence)

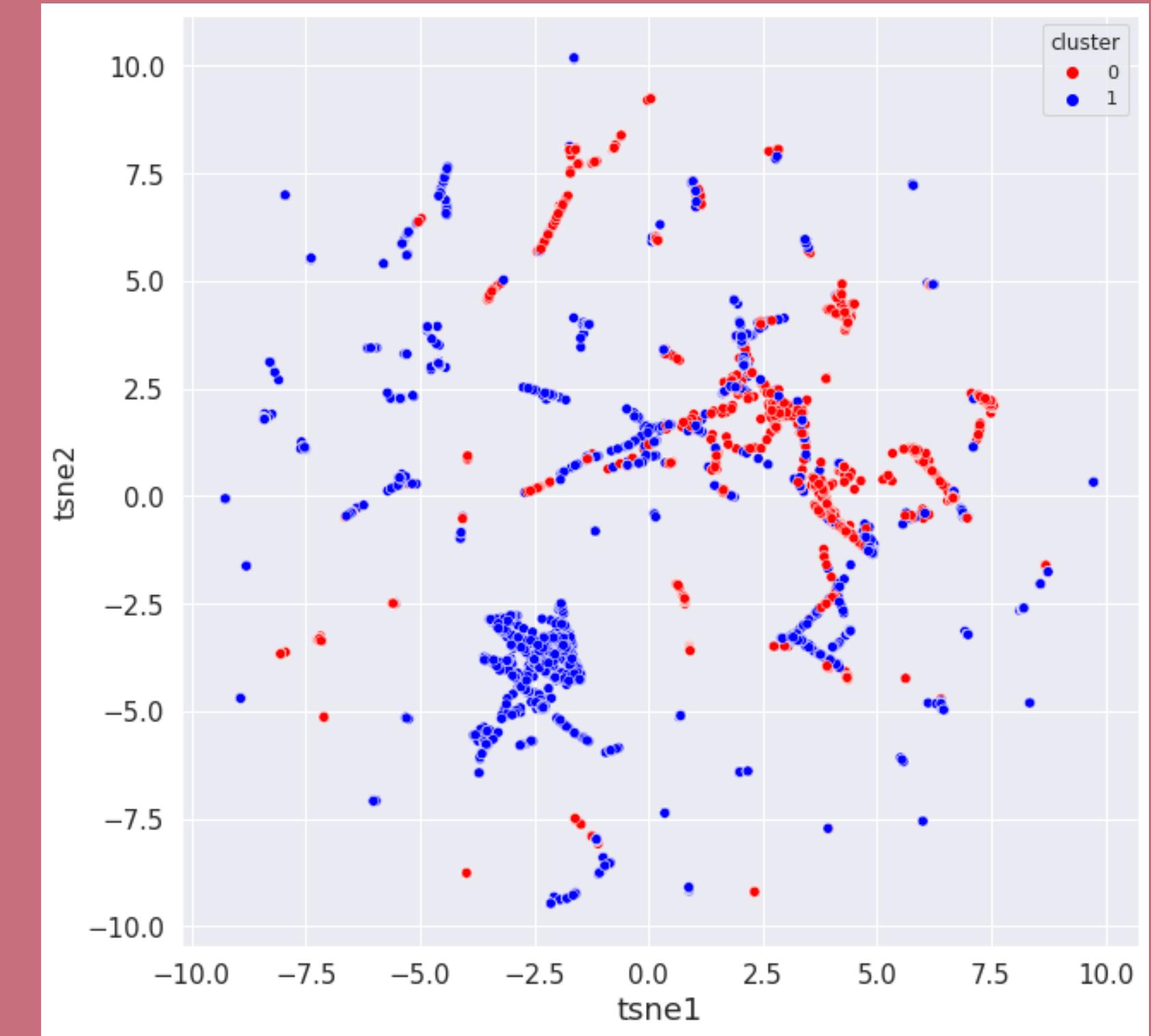
# Results



# PCA

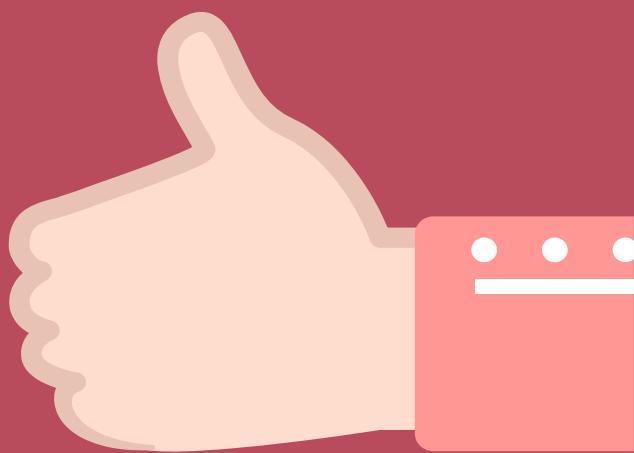
- Removes Correlated Features
- Improves Algorithm Performance
- Reduces Overfitting
- Improves Visualization



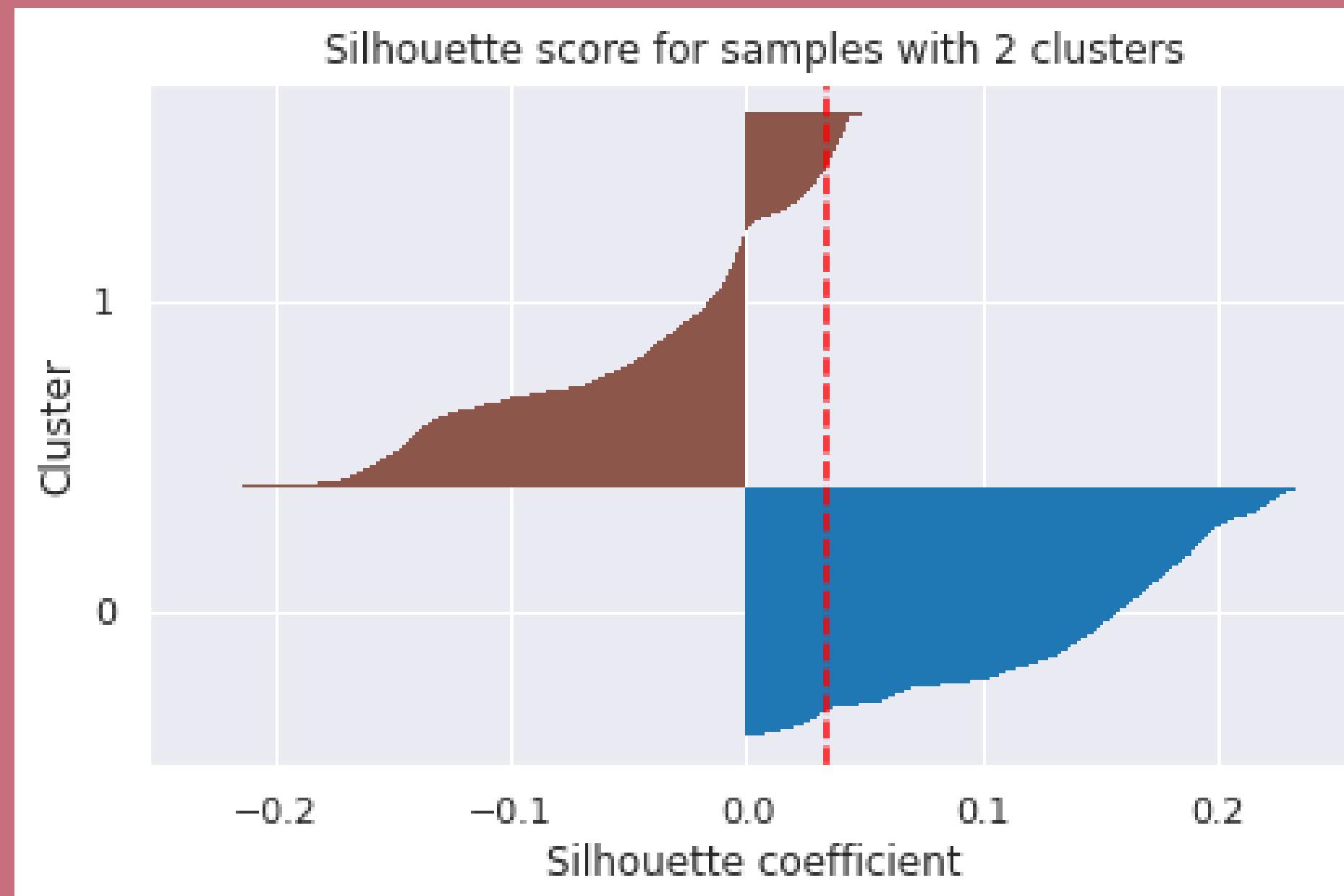


# TSNE

- Handles Non Linear Data Efficiently
- Preserves Local and Global Structure



# Silhouette

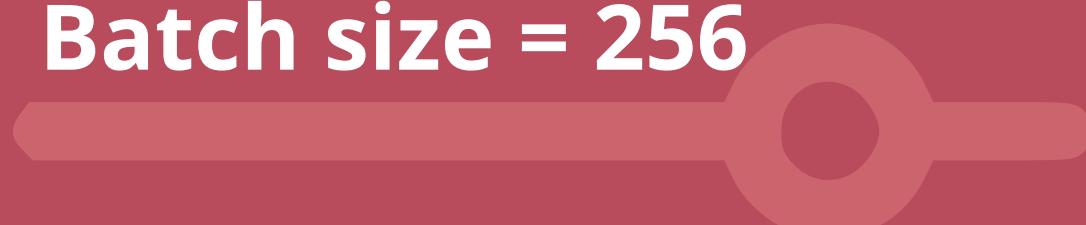


**59.10%**

**Epochs = 100**



**Batch size = 256**



**Validation Batch size = 128**



Accuracy

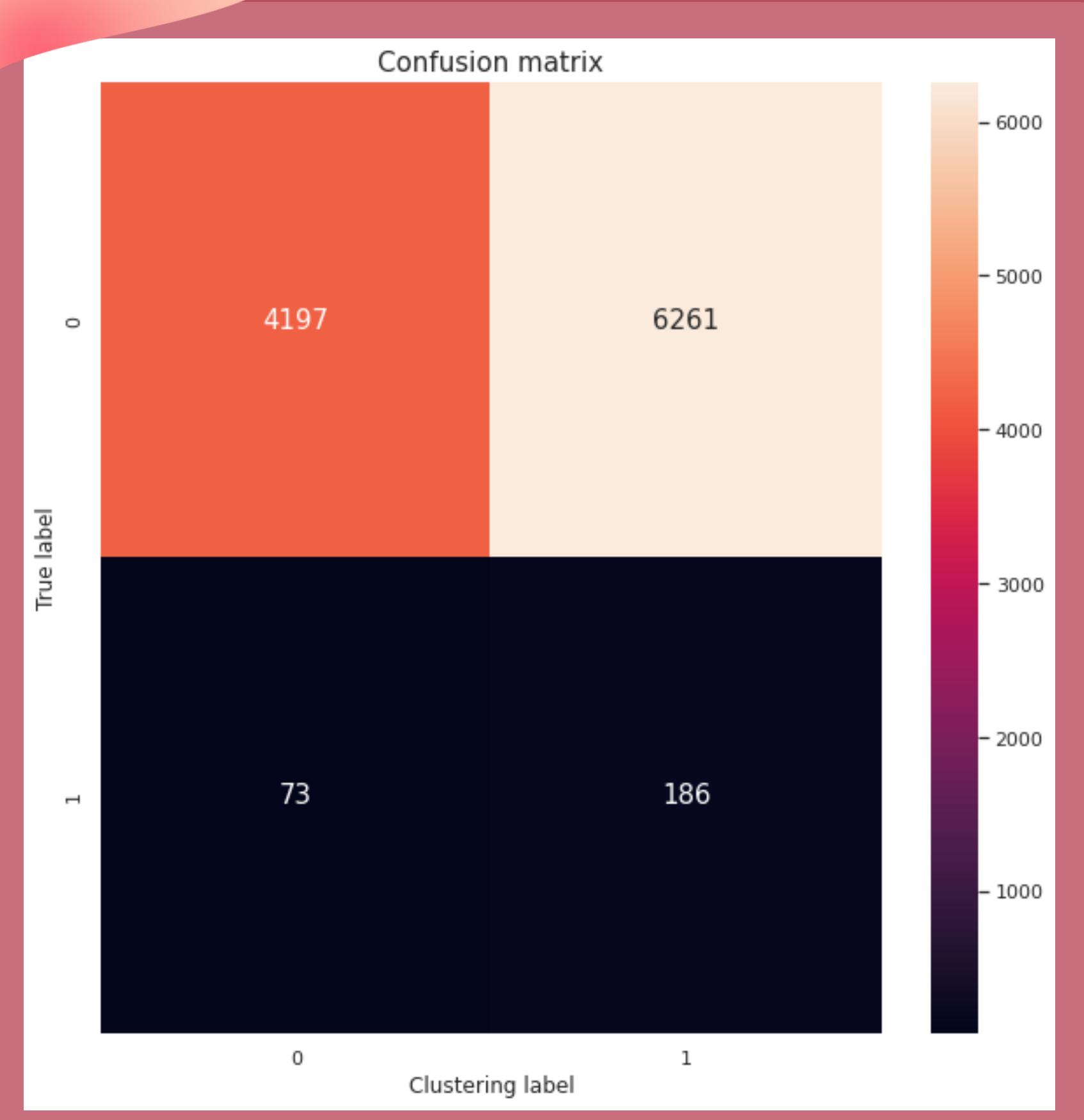
**0.034447**

NMI

**0.002651**

Silhouette score

# Confusion Matrix



*Thank you*