Co-funded by the
Erasmus+ Programme
of the European Union

SOFIA UNIVERSITY
"ST. KLIMENT OHRIDSKI
EST. 1888

ICT-TEX

ICT-TEX course on Digital skills

Topic 8: Introduction to Artificial Intelligence and Machine Learning

The course is developed under Erasmus+ Program Key Action 2:
Cooperation for innovation and the exchange of good practices Knowledge Alliance

ICT IN TEXTILE AND CLOTHING HIGHER EDUCATION AND BUSINESS

Project Nr. 612248-EPP-1-2019-1-BG-EPPKA2-KA

The information and views set out in this publication are those of the authors and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.

Co-funded by the
Erasmus+ Programme
of the European Union

Hands-on exercise

CASE STUDY 2 – THE AFRICAN TRADITIONAL MOTIFS FABRIC DESIGN

These slides are part of the Topic 8 on *"Introduction to Artificial Intelligence and Machine Learning"* of the course on Digital skills in Textile and clothing industry.

Check also the main presentation in this topic, as well as the additional reading resources, available in the ICT-TEX platform.

Problem

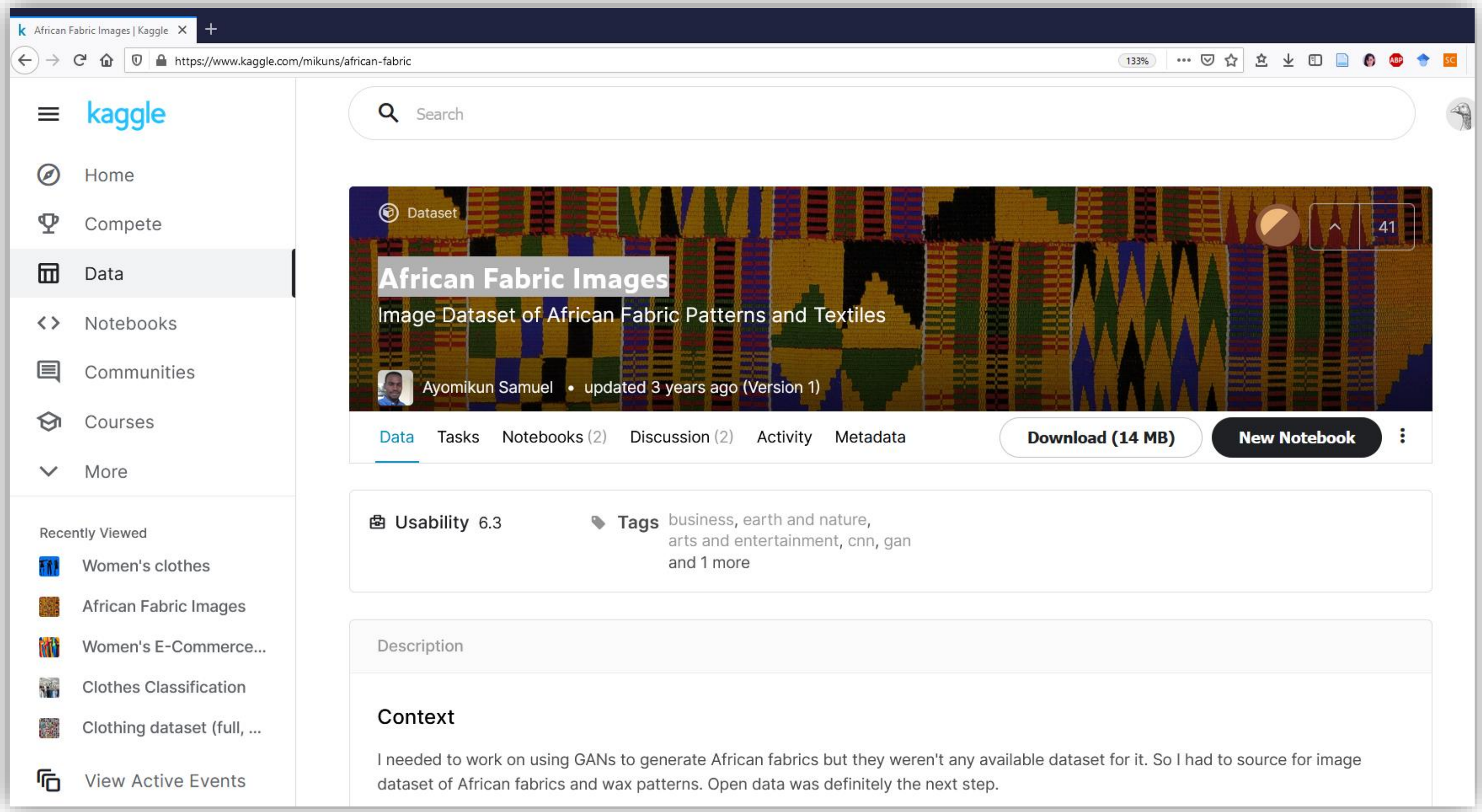
- New fabric design is quite challenging task, especially when you need to fulfill some requirements, like adapting traditional motifs in fabric design, or following the trends in the color schemes for the next fashion season, etc.
- In this use case we need to investigate the African traditional motifs in fabric design and to select different patterns on which base to be designed new "African" fabric designs

Problem

- There is a huge variety of African fabric motifs. We will investigate big data image samples and on which base will be discovered different typical African motifs
- The designer will use the discovered groups of motifs as a basis for new fabric design development

Data

- We need dataset that contain images with Africa fabric motifs
- Searching for some open data sets we identify the **„African Fabric Images”** in Kaggle that perfectly fits the problem needs.
- Please, download the dataset locally on your computer.



African Fabric Images

Image Dataset of African Fabric Patterns and Textiles

Ayomikun Samuel • updated 3 years ago (Version 1)

Usability 6.3

Tags business, earth and nature, arts and entertainment, cnn, gan and 1 more

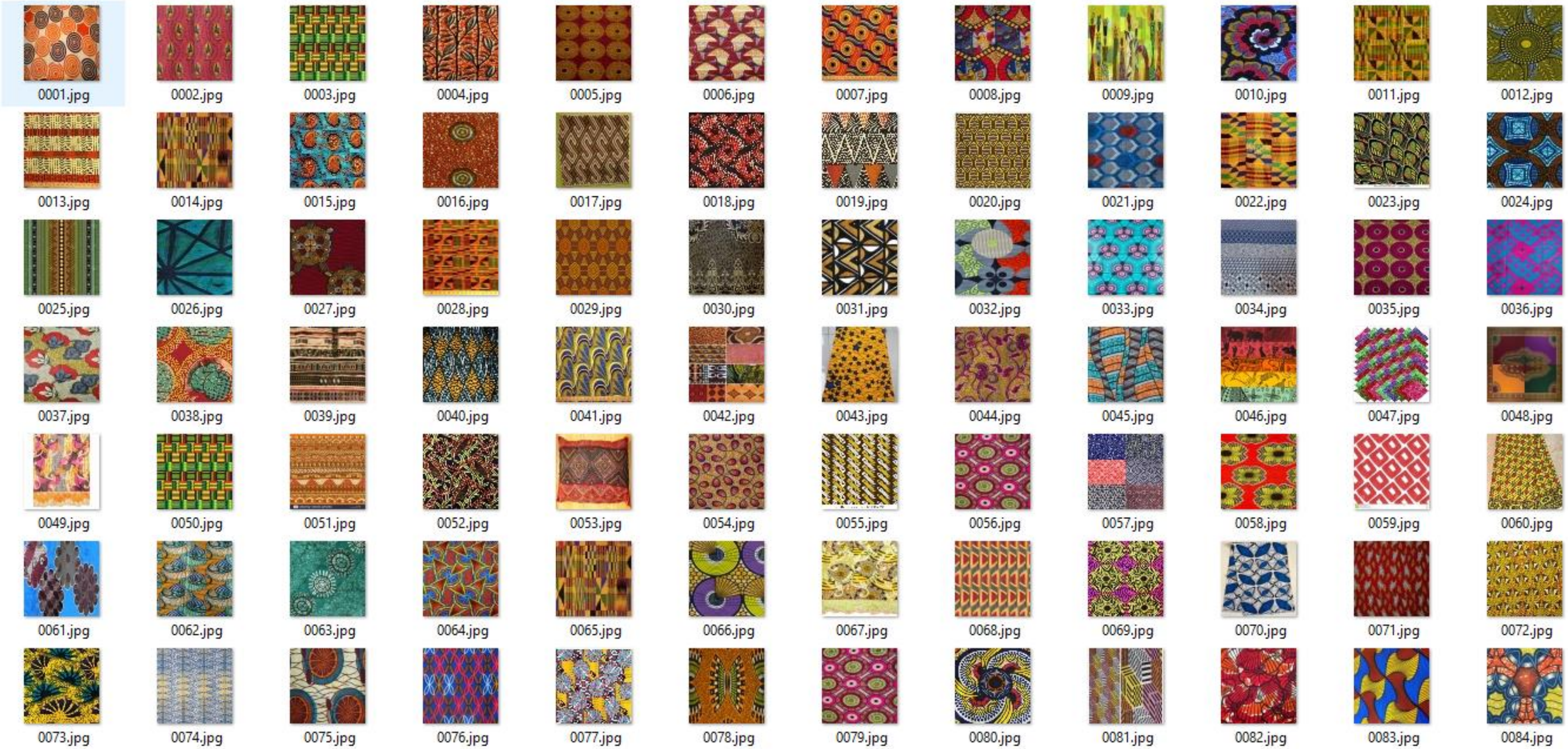
Description

Context

I needed to work on using GANs to generate African fabrics but they weren't any available dataset for it. So I had to source for image dataset of African fabrics and wax patterns. Open data was definitely the next step.

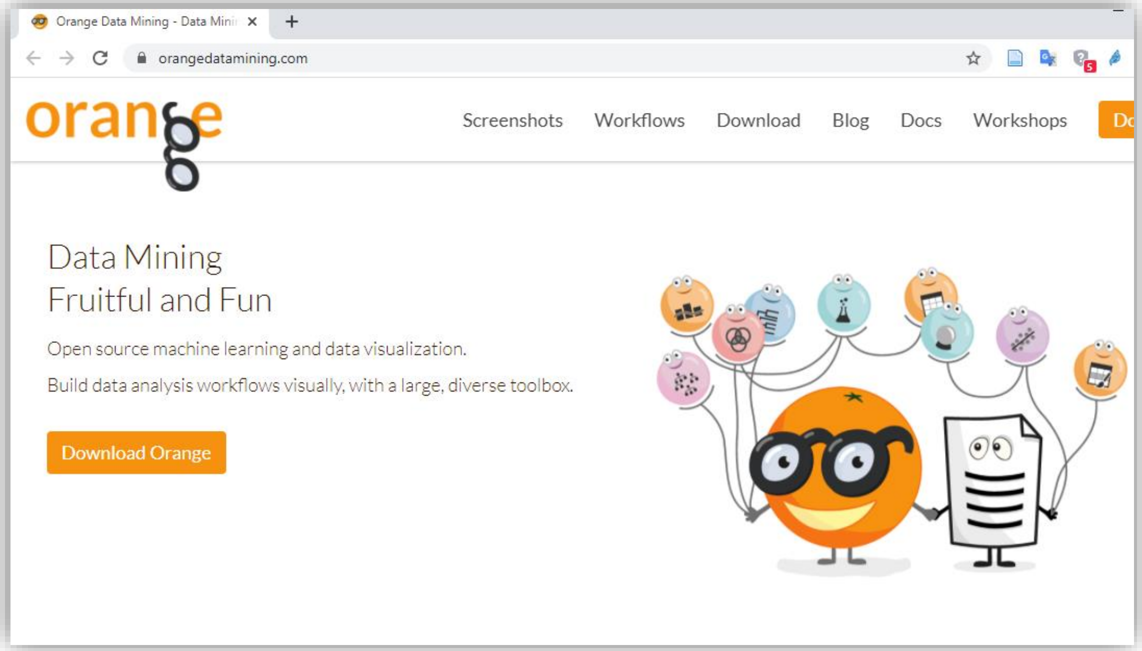
https://www.kaggle.com/mikuns/african-fabric

Sample of the African Fabric images



Framework

- For processing data, we will use Orange Data Mining Framework – free software that provides the basic AI tools in user friendly format that is appropriate for use even from non-technical users.
- (https://orangedatamining.com/)



Data Mining
Fruitful and Fun

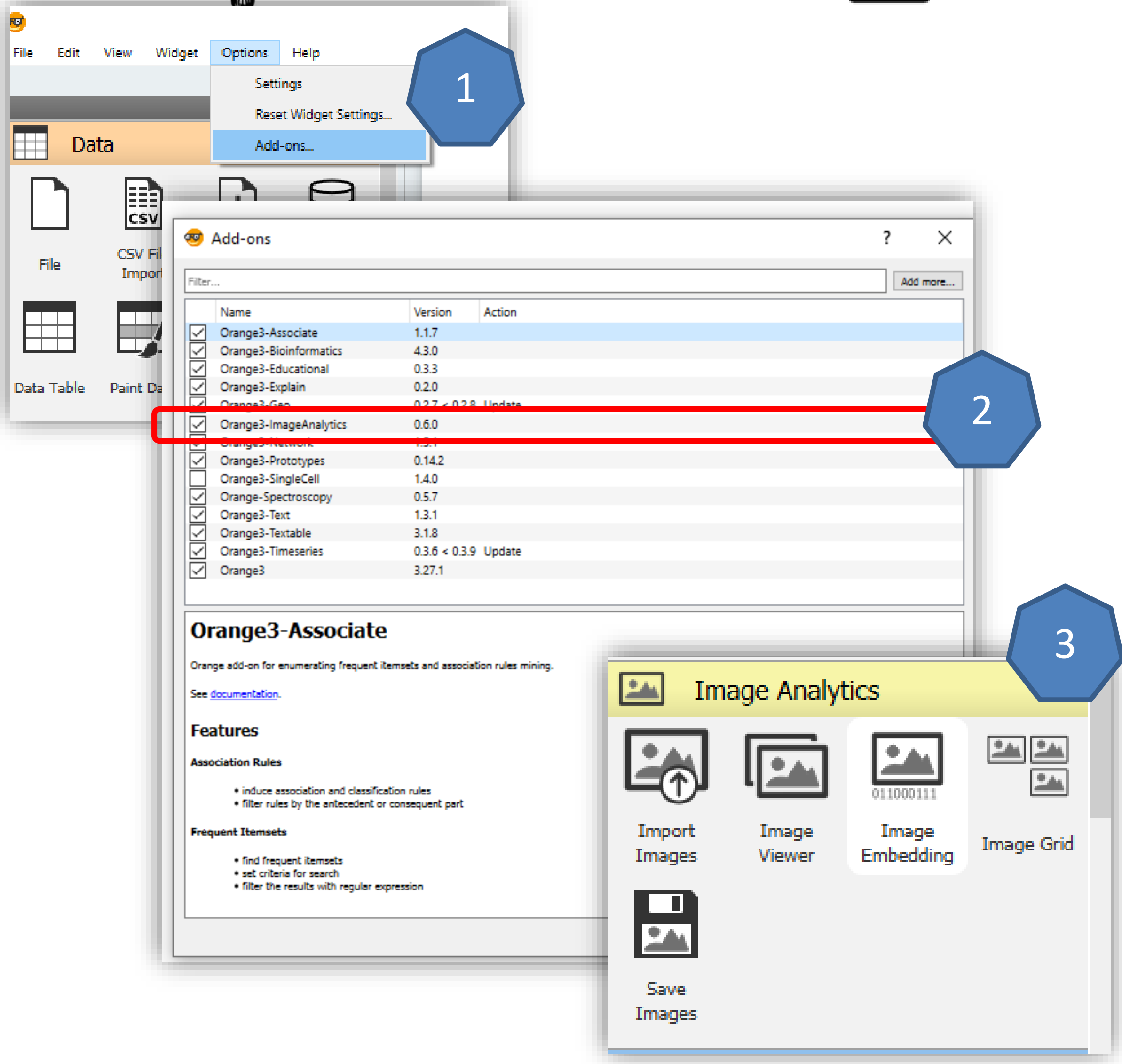
Open source machine learning and data visualization.

Build data analysis workflows visually, with a large, diverse toolbox.

Co-funded by the
Erasmus+ Programme
of the European Union

Setup

- The core version of Orange contains basic functionalities only. You need to install some add-ons for Image processing.
- Please, before start ensure that Image Analytics panel is installed



	Name	Version	Action
☑	Orange3-Associate	1.1.7	
☑	Orange3-Bioinformatics	4.3.0	
☑	Orange3-Educational	0.3.3	
☑	Orange3-Explain	0.2.0	
☑	Orange3-Geo	0.2.7 < 0.2.8	Update
☑	Orange3-ImageAnalytics	0.6.0	
☑	Orange3-Network	1.5.1	
☑	Orange3-Prototypes	0.14.2	
□	Orange3-SingleCell	1.4.0	
☑	Orange-Spectroscopy	0.5.7	
☑	Orange3-Text	1.3.1	
☑	Orange3-Textable	3.1.8	
☑	Orange3-Timeseries	0.3.6 < 0.3.9	Update
☑	Orange3	3.27.1	

Orange3-Associate

Orange add-on for enumerating frequent itemsets and association rules mining.

See documentation.

Features

Association Rules

- induce association and classification rules
- filter rules by the antecedent or consequent part

Frequent Itemsets

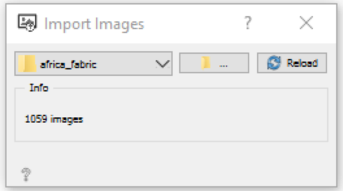
- find frequent itemsets
- set criteria for search
- filter the results with regular expression

Step 1

- Create "New Project"
- Select from "Image Analytics" toolkit the widget "Import Images"
- Set the source folder to be the one that contains your datasets

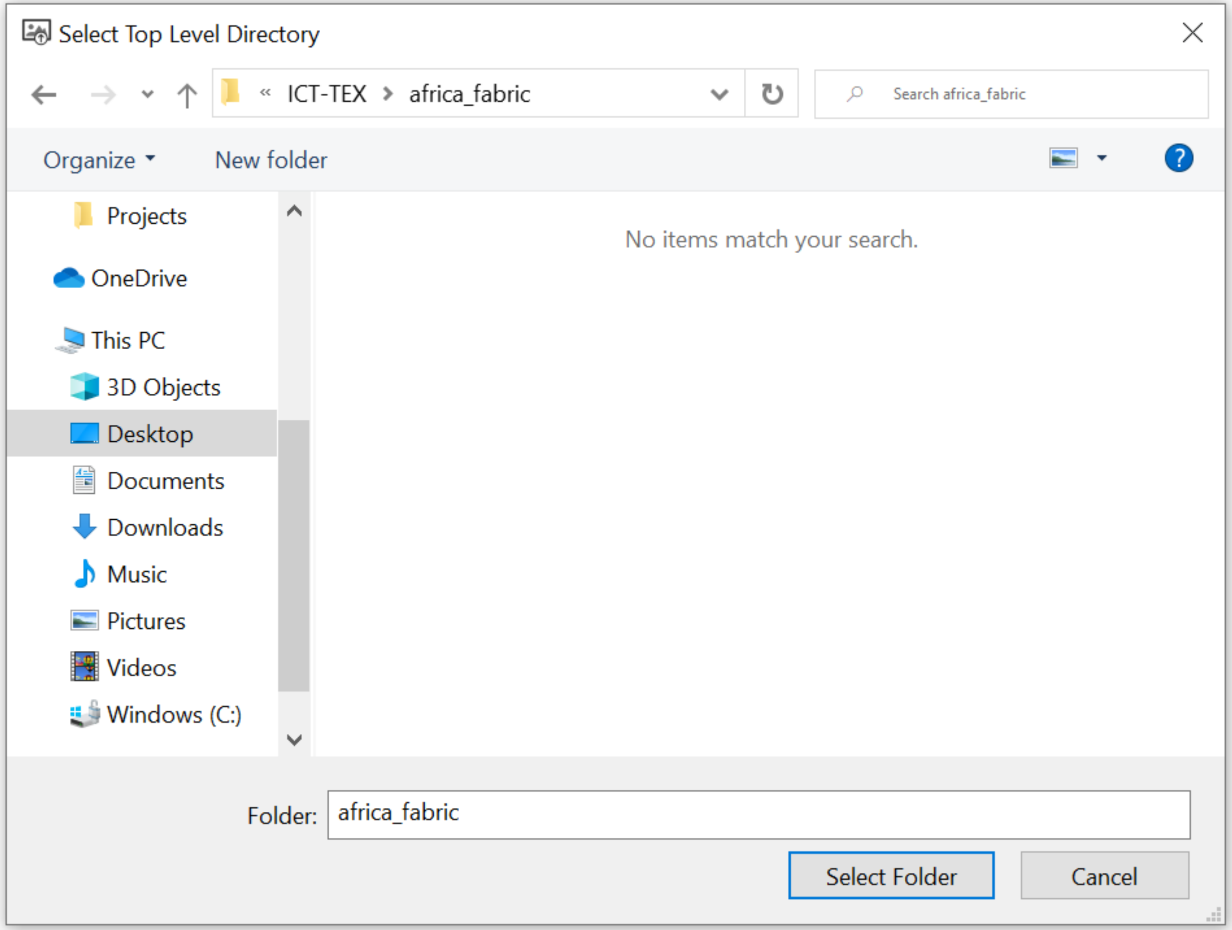
Import Images

Import Images



africa_fabric

1059 images



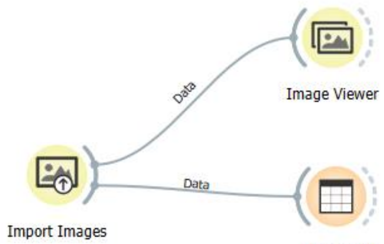
No items match your search.

Folder: africa_fabric

For more information about this widget: https://orangedatamining.com/widget-catalog/image-analytics/importimages/

Step 2

To inspect the content of the loaded dataset can be added widget “Image Viewer” and “Data table” widget from panel Data



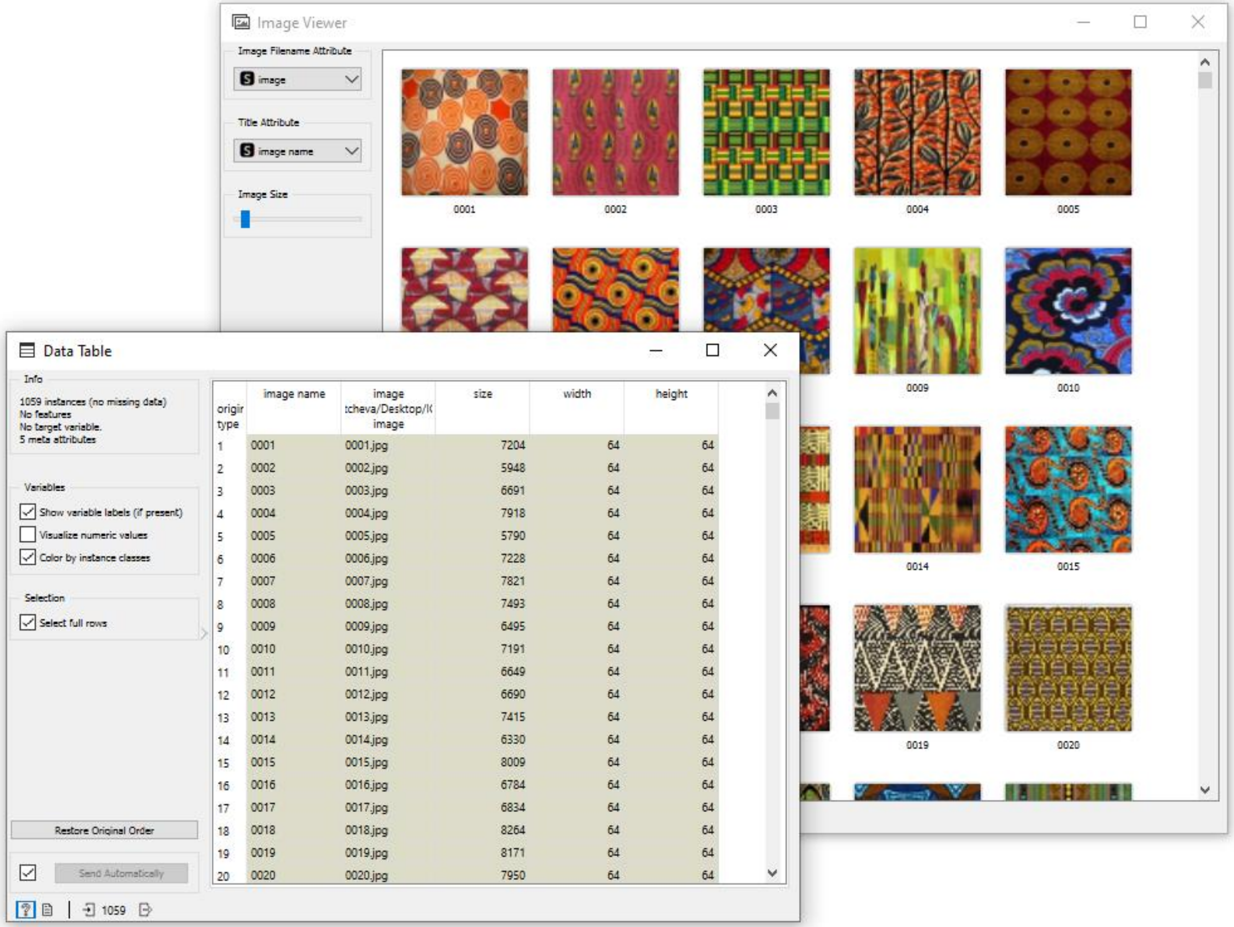


Image Viewer

Image Filename Attribute: image

Title Attribute: image name

Image Size

Data Table

Info

1059 instances (no missing data)
No features
No target variable.
5 meta attributes

Variables

☑ Show variable labels (if present)

□ Visualize numeric values

☑ Color by instance classes

Selection

☑ Select full rows

Restore Original Order

☑ Send Automatically

1059

origir type	image name	image tcheva/Desktop/I(image	size	width	height
1	0001	0001.jpg	7204	64	64
2	0002	0002.jpg	5948	64	64
3	0003	0003.jpg	6691	64	64
4	0004	0004.jpg	7918	64	64
5	0005	0005.jpg	5790	64	64
6	0006	0006.jpg	7228	64	64
7	0007	0007.jpg	7821	64	64
8	0008	0008.jpg	7493	64	64
9	0009	0009.jpg	6495	64	64
10	0010	0010.jpg	7191	64	64
11	0011	0011.jpg	6649	64	64
12	0012	0012.jpg	6690	64	64
13	0013	0013.jpg	7415	64	64
14	0014	0014.jpg	6330	64	64
15	0015	0015.jpg	8009	64	64
16	0016	0016.jpg	6784	64	64
17	0017	0017.jpg	6834	64	64
18	0018	0018.jpg	8264	64	64
19	0019	0019.jpg	8171	64	64
20	0020	0020.jpg	7950	64	64

Step 3

- To identify similarity between images we can used Image Embeddings widget, that uses pretrained Deep Learning models over ImageNet data set. (http://image-net.org/index). For this study we will use **InceptionV3** - Google's deep neural network for image recognition.
- To view Image Embeddings can be used Image Grid widget. Note that more similar images are positioned more closely in the grid
- The Image Embeddings transforms the images features to vectors. Those vectors can be explored through Data Table view. Please, note that there are generated 2048 new columns representing features with numeric values.

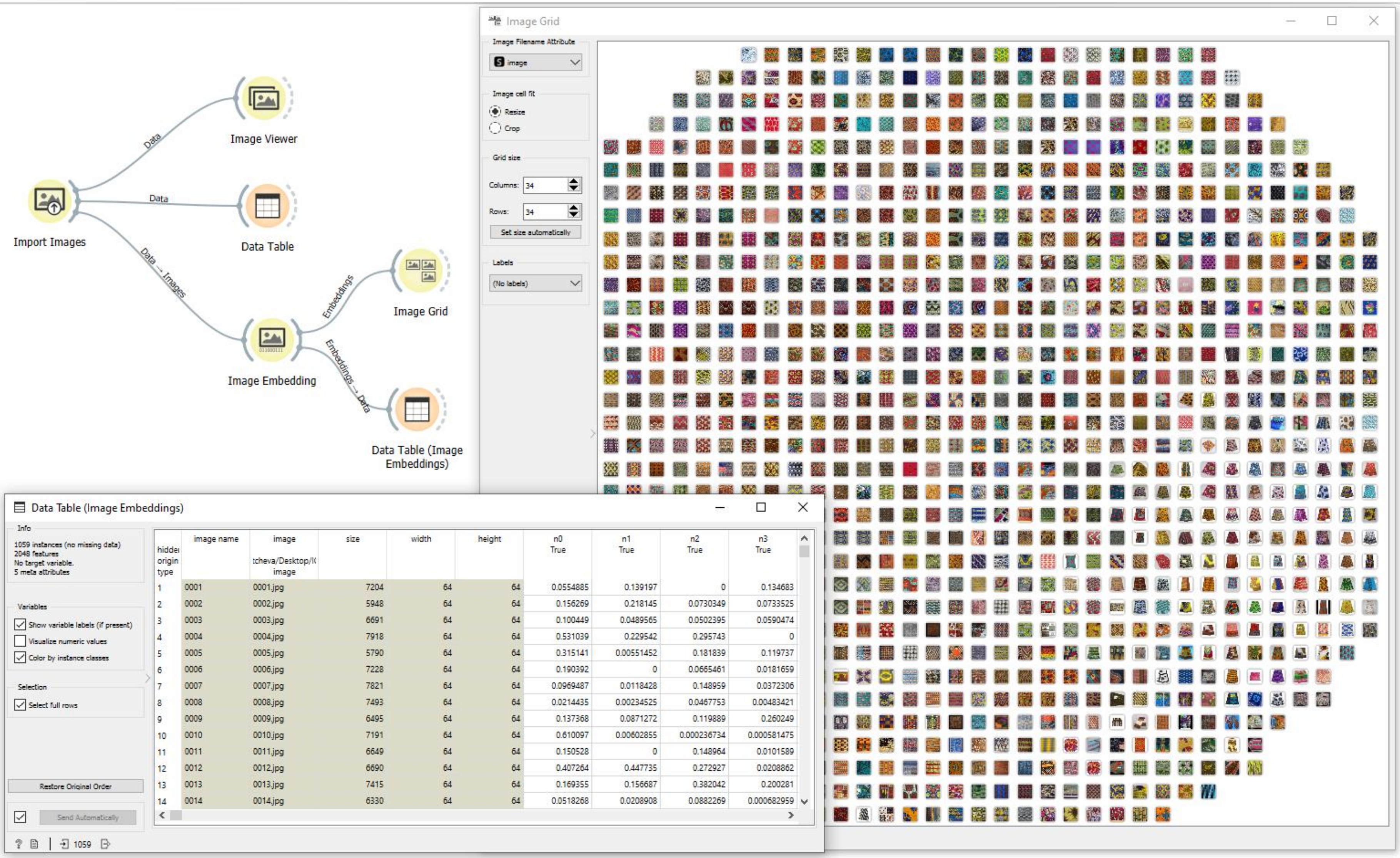


Image Grid

Image Filename Attribute: image

Image cell fit
● Resize
○ Crop

Grid size
Columns: 34
Rows: 34
Set size automatically

Labels: (No labels)

Data Table (Image Embeddings)

Info
1059 instances (no missing data)
2048 features
No target variable.
5 meta attributes

Variables
☑ Show variable labels (if present)
☐ Visualize numeric values
☑ Color by instance classes

Selection
☑ Select full rows

Restore Original Order

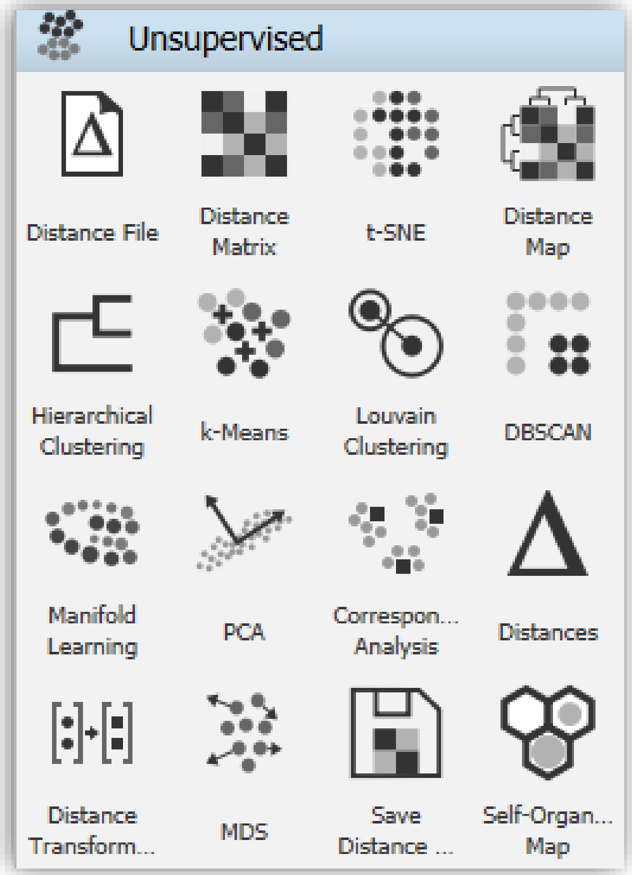
☑ Send Automatically

1059

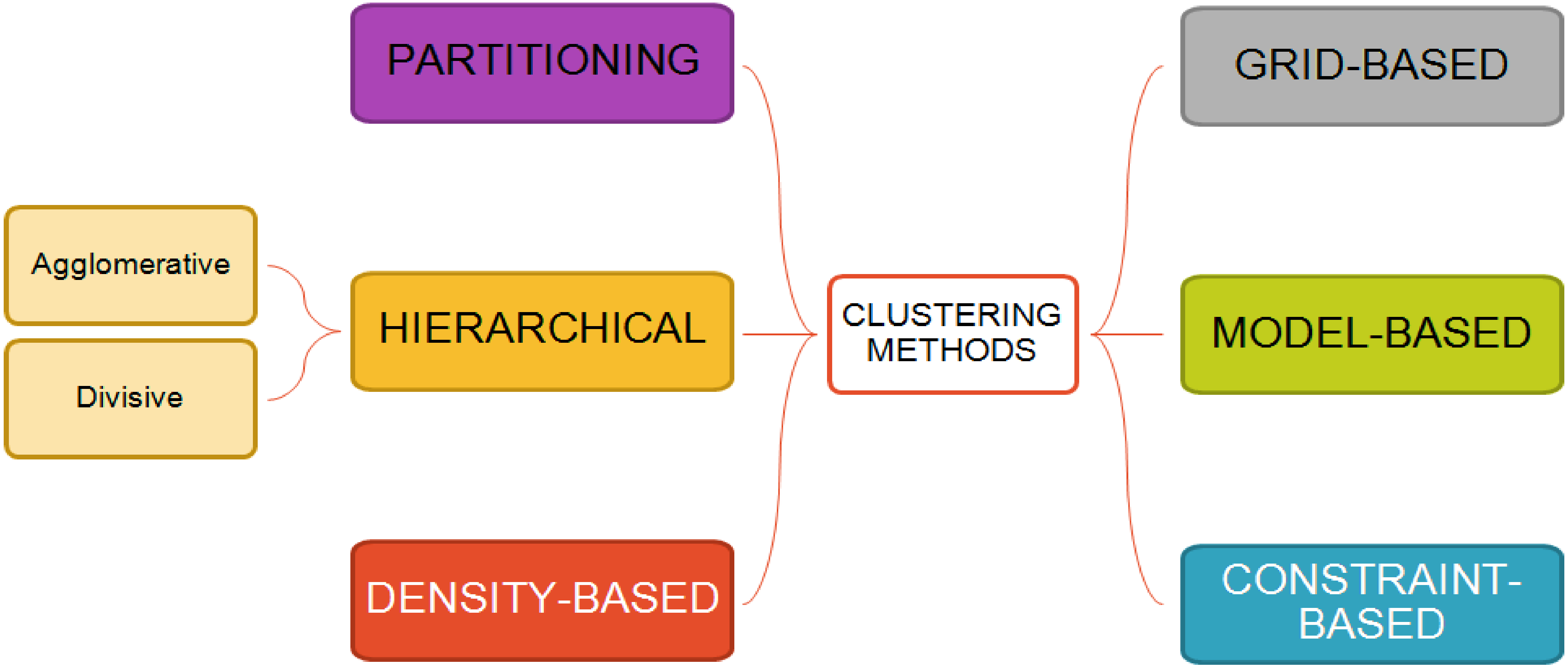
hidden origin type	image name	image :cheva/Desktop/I(image	size	width	height	n0 True	n1 True	n2 True	n3 True
1	0001	0001.jpg	7204	64	64	0.0554885	0.139197	0	0.134683
2	0002	0002.jpg	5948	64	64	0.156269	0.218145	0.0730349	0.0733525
3	0003	0003.jpg	6691	64	64	0.100449	0.0489565	0.0502395	0.0590474
4	0004	0004.jpg	7918	64	64	0.531039	0.229542	0.295743	0
5	0005	0005.jpg	5790	64	64	0.315141	0.00551452	0.181839	0.119737
6	0006	0006.jpg	7228	64	64	0.190392	0	0.0665461	0.0181659
7	0007	0007.jpg	7821	64	64	0.0969487	0.0118428	0.148959	0.0372306
8	0008	0008.jpg	7493	64	64	0.0214435	0.00234525	0.0467753	0.00483421
9	0009	0009.jpg	6495	64	64	0.137368	0.0871272	0.119889	0.260249
10	0010	0010.jpg	7191	64	64	0.610097	0.00602855	0.000236734	0.000581475
11	0011	0011.jpg	6649	64	64	0.150528	0	0.148964	0.0101589
12	0012	0012.jpg	6690	64	64	0.407264	0.447735	0.272927	0.0208862
13	0013	0013.jpg	7415	64	64	0.169355	0.156687	0.382042	0.200281
14	0014	0014.jpg	6330	64	64	0.0518268	0.0208908	0.0882269	0.000682959

Step 5

- The dataset of images for African fabric stains is already preprocessed and ready for application of ML methods
- Separation of the dataset on similar images is a clusterization task, that is a type of unsupervised learning



Clustering Methods



CLUSTERING METHODS

PARTITIONING

HIERARCHICAL

Agglomerative

Divisive

DENSITY-BASED

GRID-BASED

MODEL-BASED

CONSTRAINT-BASED

- Because we do not know in advance to how many groups should be clustered dataset, there are the following options:
  - To find the distances between different images, using some distance metric and then to use hierarchical clustering
  - To explore and evaluate different numbers of clusters using density-based clustering

Distance metric

⍰

$f\colon A \times A \rightarrow \mathbb{R}$ is called **metric** if

$a)\ f(a,b) \geq 0$ for each $a, b \quad A$

$b)\ f(a,a) = 0$

$c)\ f(a,b) = f(b,a)$

$d)\ f(a,b) + f(b,c) \geq f(a,c)$

Hamming distance

Two vectors:

$$a = \{a_1, a_2, \dots \dots .\,.\,, an\}$$

$$b = \{b_1, b_2, \dots \dots .\,.\,, b_n\}$$

Function $\rho(a, b) = \ number\ \ of\ a_i \neq bi$, i=1,2,….,n is called ***Hamming distance.***

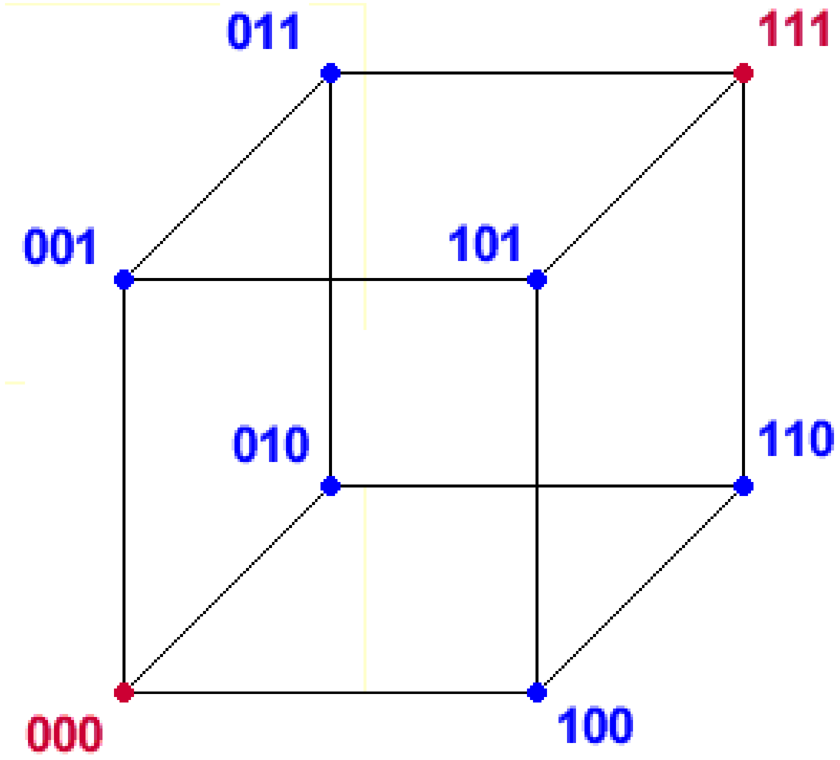
(number of corrections, necessary to be made in order two vectors to be the same)

Example 1

$$a = (\mathbf{0}, \mathbf{1}, 0, 1, \mathbf{1}, \mathbf{0}, 0)$$
$$b = (\mathbf{1}, \mathbf{0}, 0, 1, \mathbf{0}, \mathbf{1}, 0)$$

$$\rho(a, b) = 4$$

Example 2



Applications

- The Hamming distance is named after Richard Hamming, who introduced it in his fundamental paper on Hamming codes *Error detecting and error correcting codes* in 1950.
- It is used in telecommunication to count the number of flipped bits in a fixed-length binary word as an estimate of error, and therefore is sometimes called the signal distance.
- Hamming weight analysis of bits is used in several disciplines including information theory, coding theory, and cryptography.

Euclidean Distance

$$a = (x1, y1)$$

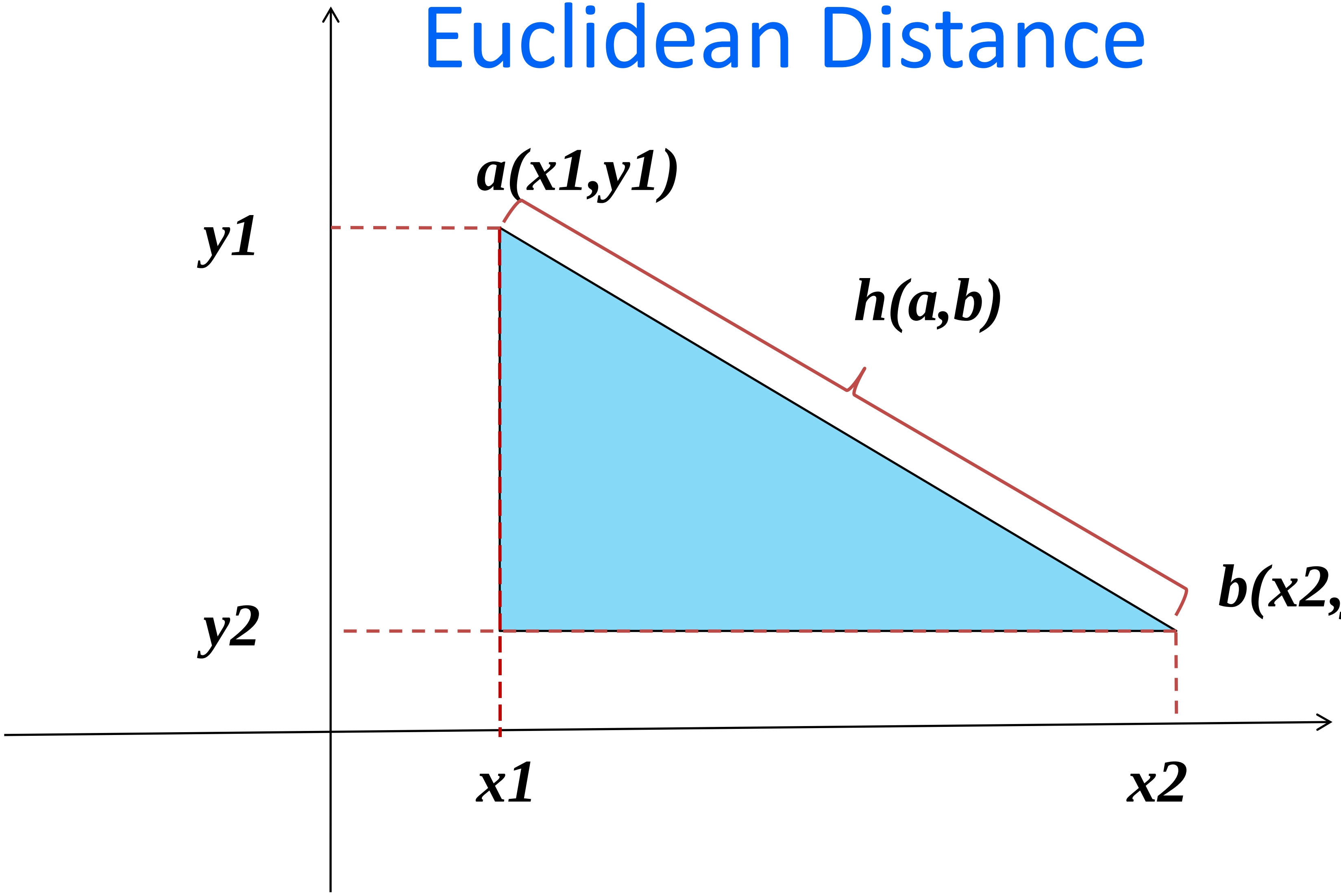
$$b = (x2, y2)$$

points in Cartesian Coordinate system

Euclidean metric

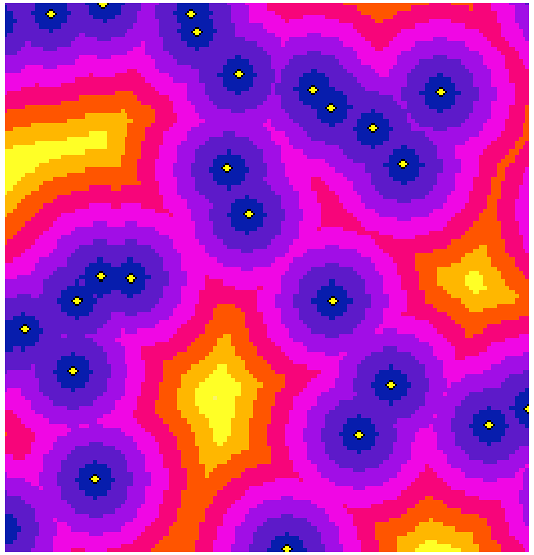
$$\boldsymbol{h}(\boldsymbol{a}, \boldsymbol{b}) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

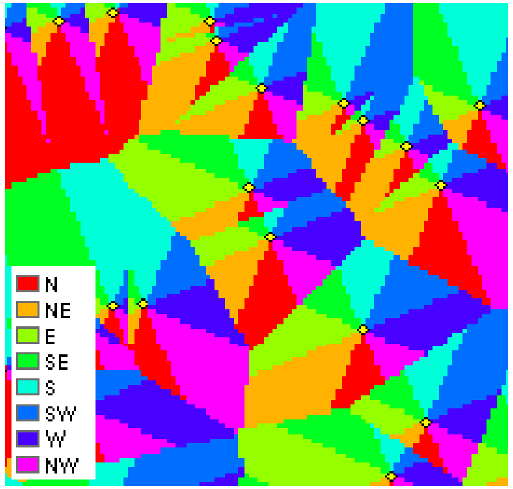
Euclidean Distance

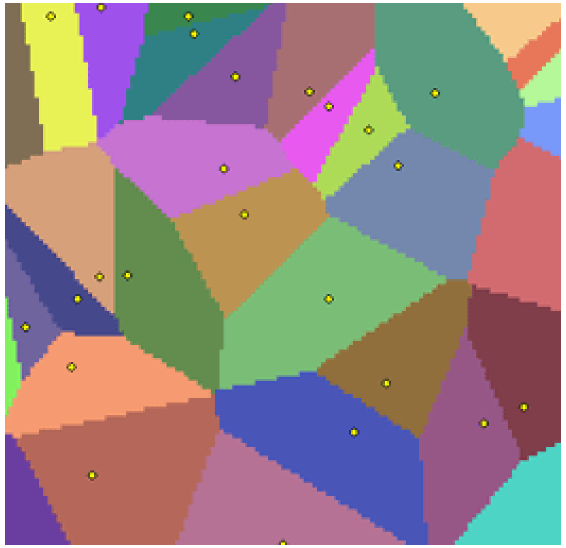


Example

Map showing the direction of the nearest town for each location.







Copyright images. Source: https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/understanding-euclidean-distance-analysis.htm

Examples

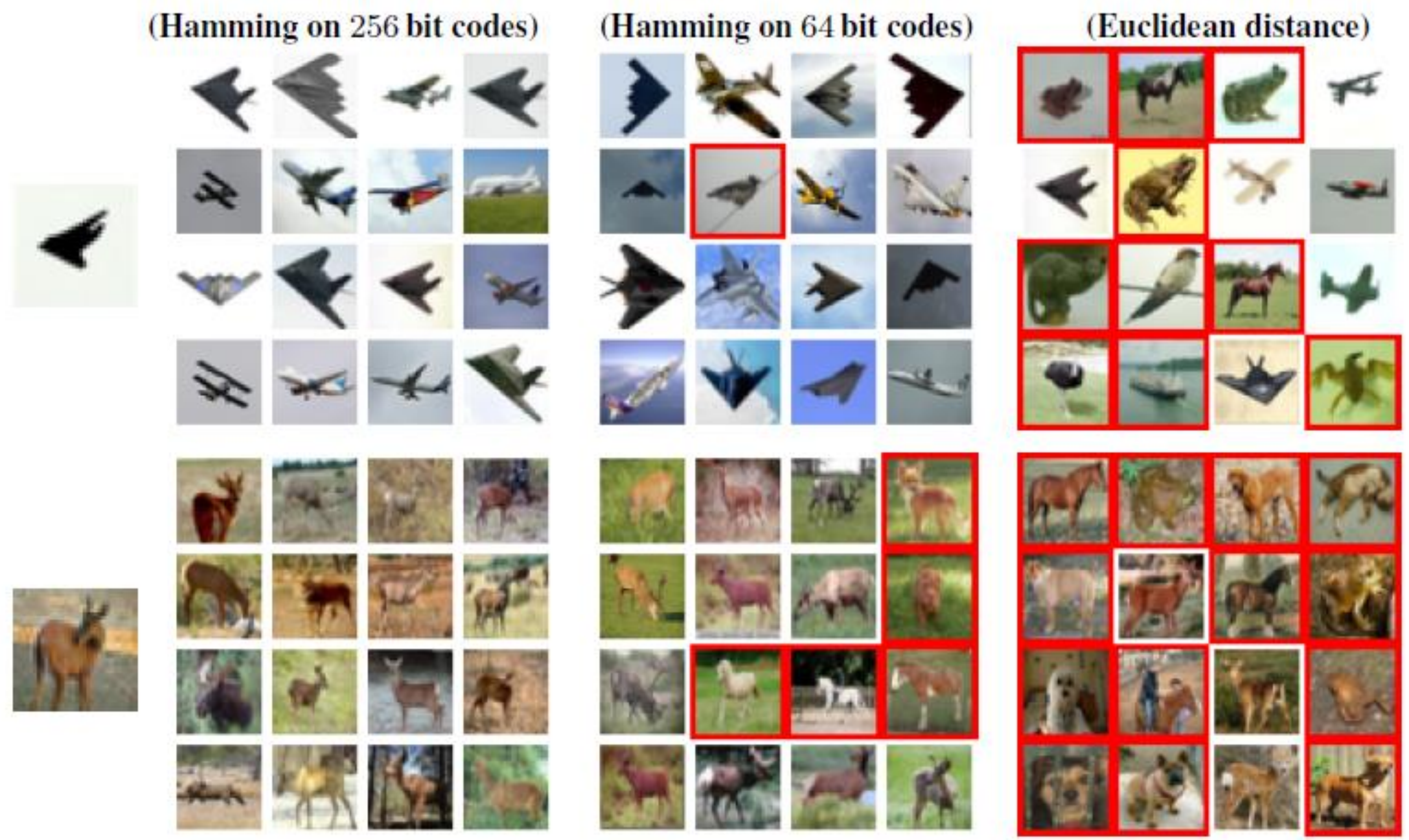


Figure 2: Retrieval results for two CIFAR-10 test images using Hamming distance on 256-bit and 64-bit codes, and Euclidean distance on bag-of-words features. Red rectangles indicate mistakes.

Copyright images. Source: https://www.semanticscholar.org/paper/Hamming-Distance-Metric-Learning-Norouzi-Fleet/1e05247708515d45166ef96a153f4e22811aa2c6/figure/2
Published in: NIPS 2012, Hamming Distance Metric Learning, Mohammad Norouzi, David J. Fleet, R. Salakhutdinov

Manhattan distance

Manhattan distance between two points

$$p1 = (x1, y1)$$
$$p2 = (x2, y2)$$
$$d(p1, p2) = |x1 - x2| + |y1 - y2|$$

Also called city block distance and taxicab geometry

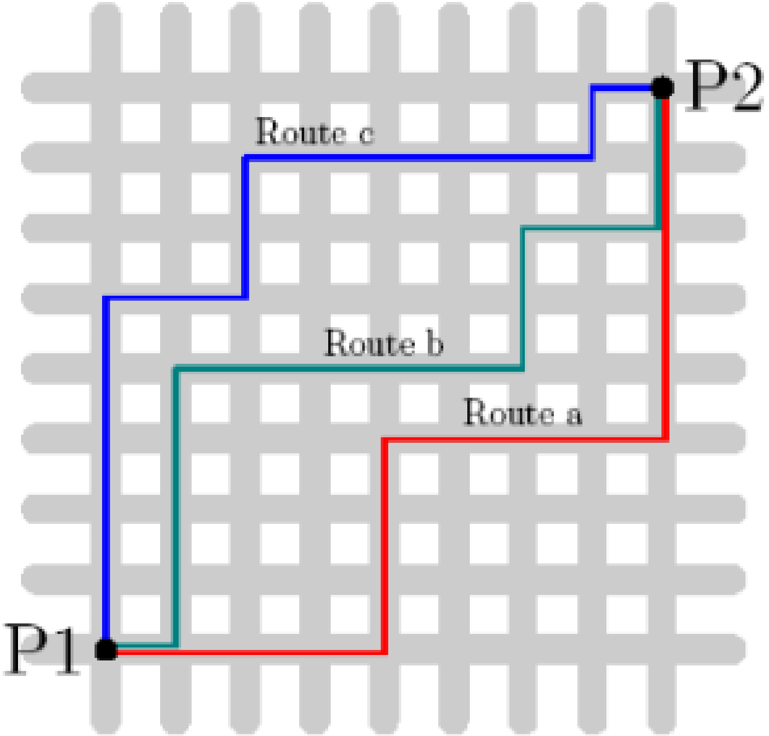
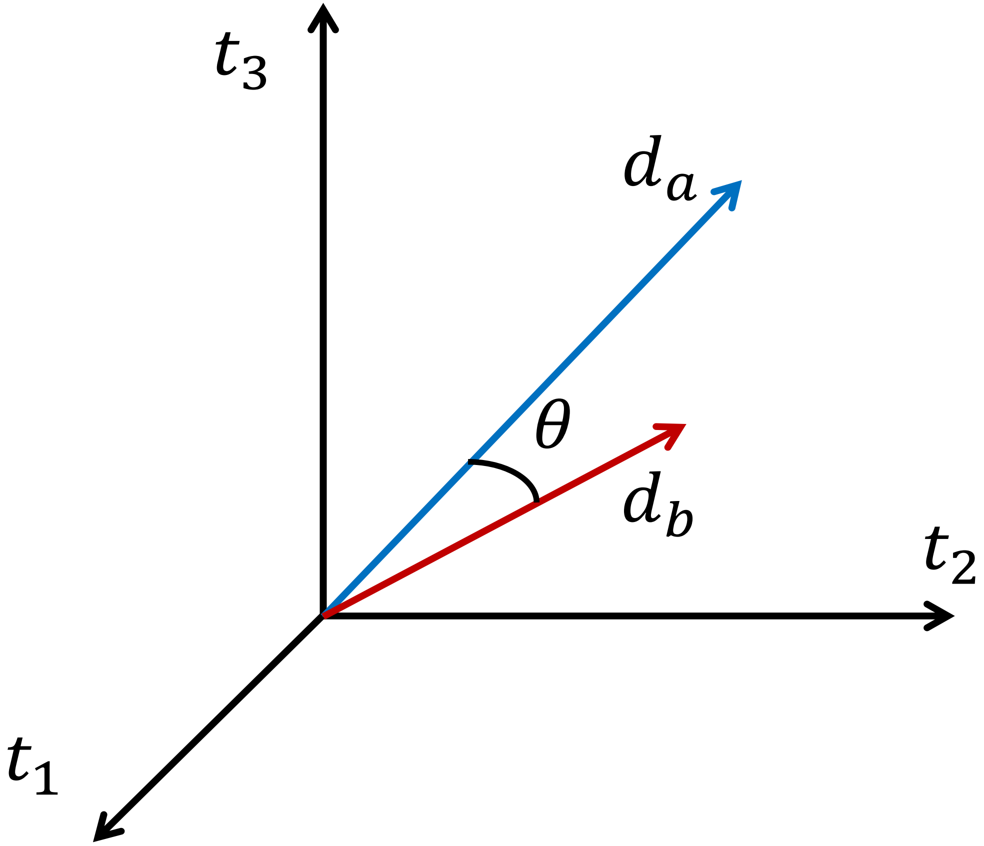


Image source (License - Public Domain): https://en.wikipedia.org/wiki/Taxicab_geometry#/media/File:Manhattan_distance.svg

Cosine distance



The similarity metric between two vectors

$d_a = \langle w_{a1}, w_{a2}, \cdots, w_{an} \rangle$ and

$d_b = \langle w_{b1}, w_{b2}, \cdots, w_{bn} \rangle$

is defined as follows:

$$s(d_a, d_b) = \cos(\theta) = \frac{d_a \cdot d_b}{\|d_a\| \|d_b\|}$$

Where

$$\|d_a\| = \sqrt{w_{a1}^2 + w_{a2}^2 \ \ + \cdots + w_{an}^2}$$

$$d_a \cdot d_b = w_{a1} w_{b1} + w_{a2} w_{b2} + \cdots + w_{an} w_{bn}$$

Thus $\cos(\theta) = 1$ iff both vectors are exactly the same, and $\cos(\theta) = 0$ for totally different vectors.

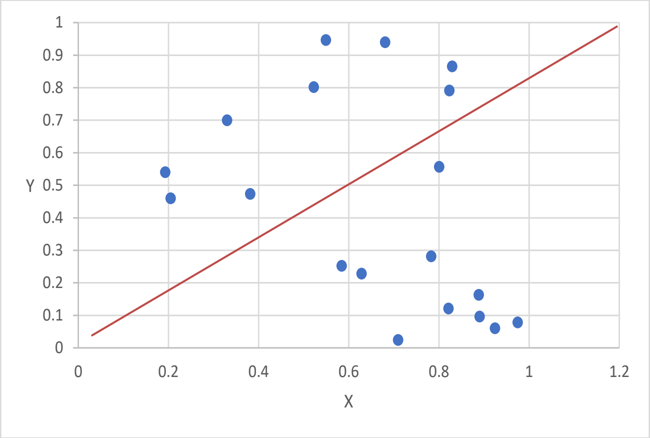
Pearson Correlation

A Pearson correlation is a number between -1 and +1 that indicates to which extent 2 variables X and Y are linearly related.

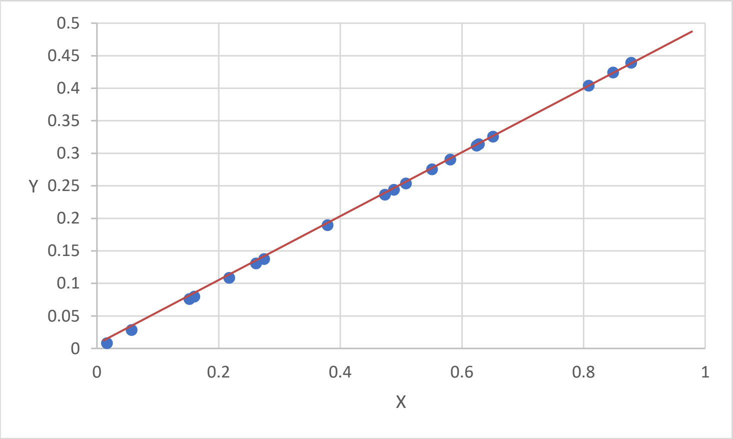
$$r = \frac{\sum(X-\bar{X})\sum(Y-\bar{Y})}{\sqrt{\sum(X-\bar{X})^2}\sqrt{\sum(Y-\bar{Y})^2}}$$

Where $\bar{X}$ is the mean of X variable, and $\bar{Y}$ is the mean of Y variable

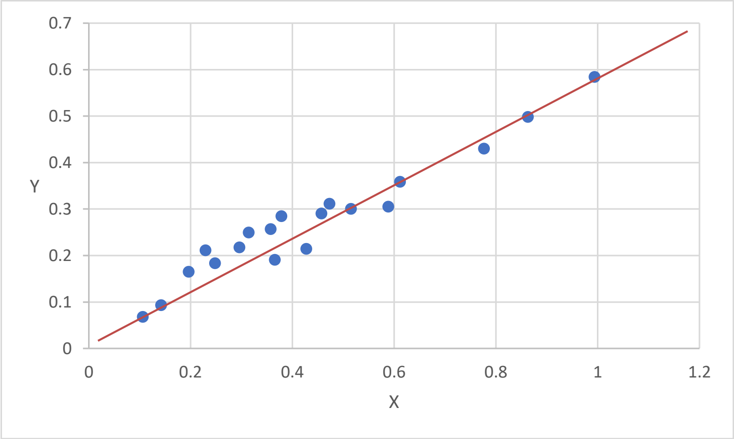
Examples



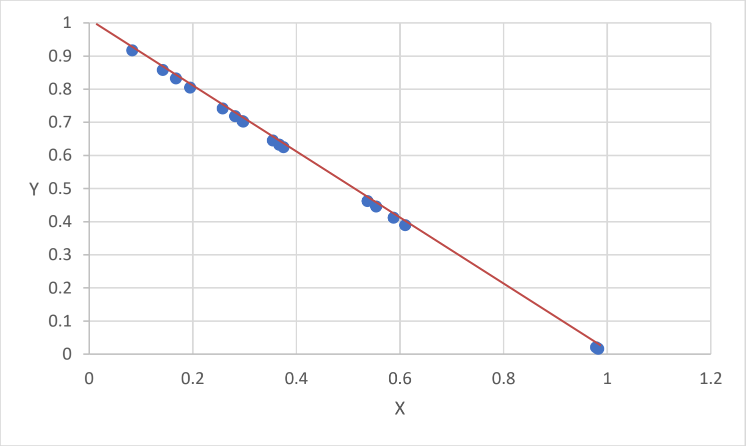
Weak/ no correlation



Large positive correlation



Medium positive correlation



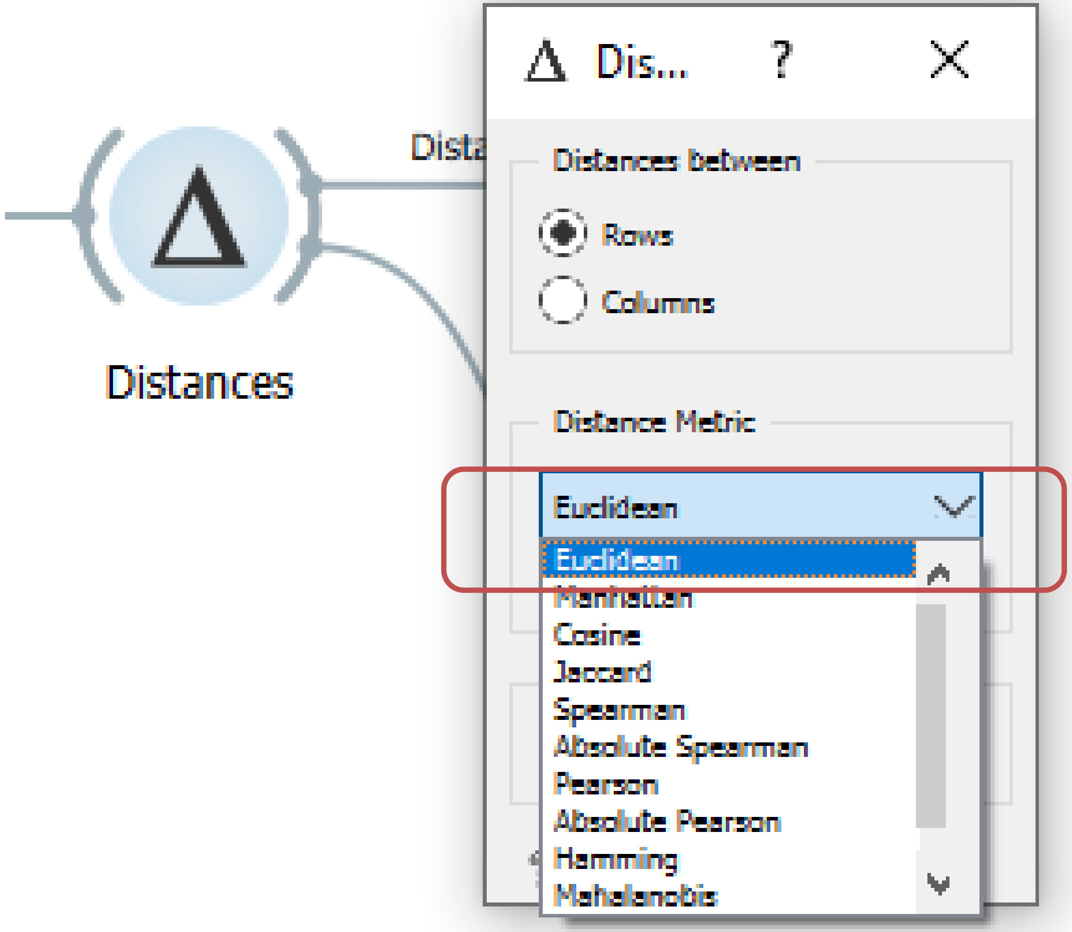
Large negative correlation

Spearman correlation

A Spearman rank correlation is a number between -1 and +1 that indicates to what extent 2 variables X and Y are monotonously related.

Step 6

- For measuring distances, we select the widget Distances and a distance metric
- You can explore different distance metrics
- Please note, that the calculation of distance metrics can take some time.



Distances between

● Rows

○ Columns

Distance Metric

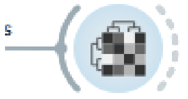
Euclidean

- Euclidean
- Manhattan
- Cosine
- Jaccard
- Spearman
- Absolute Spearman
- Pearson
- Absolute Pearson
- Hamming
- Mahalanobis

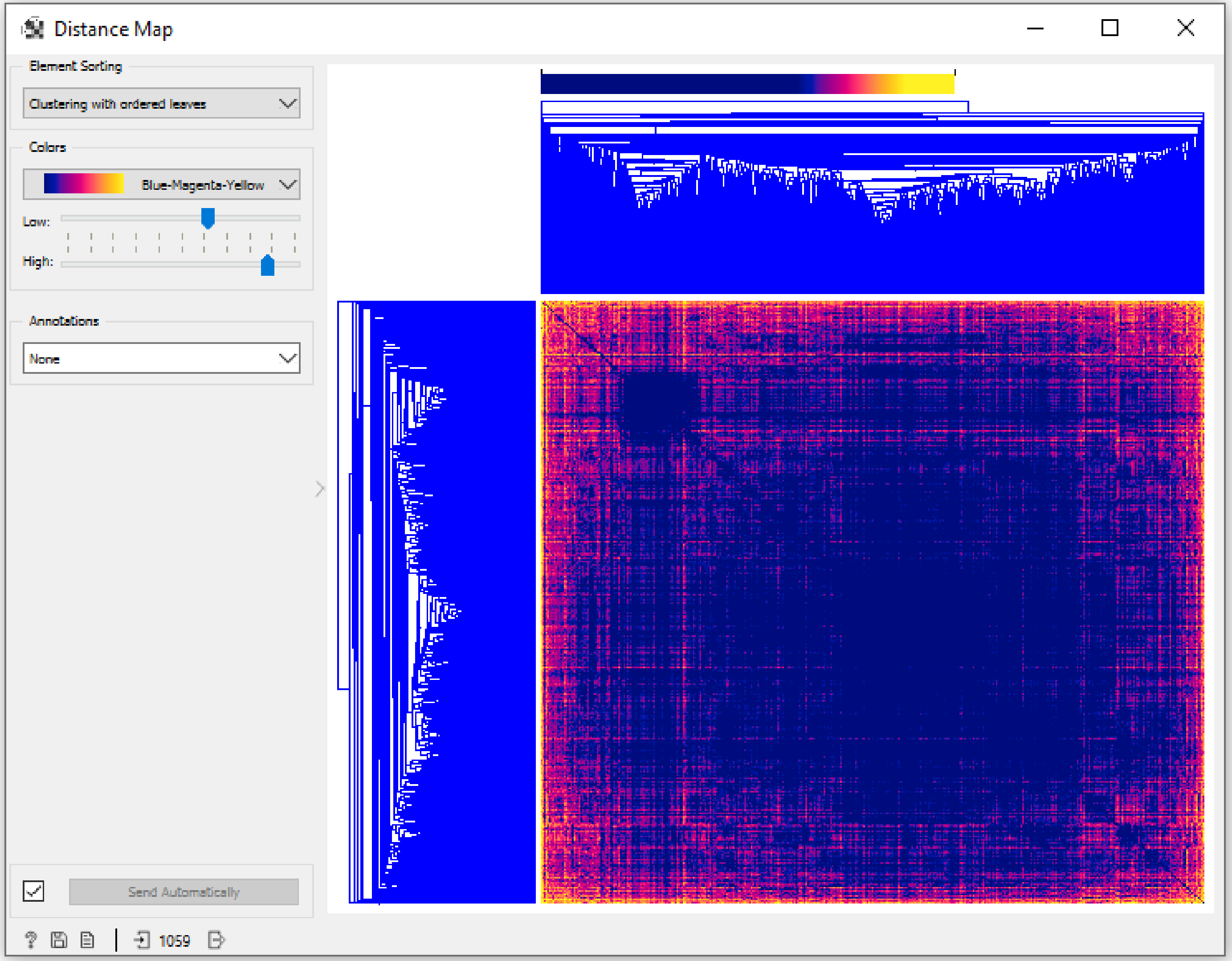
Step 7

The Distance Map widget allows to visualize the distances between images.

It easily can be seen that there is a high variety of image in the dataset and there hardly can be distinguished some clusters.



Distance Map



Distance Map

Element Sorting: Clustering with ordered leaves

Colors: Blue-Magenta-Yellow

Low:

High:

Annotations: None

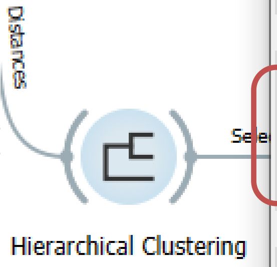
☑ Send Automatically

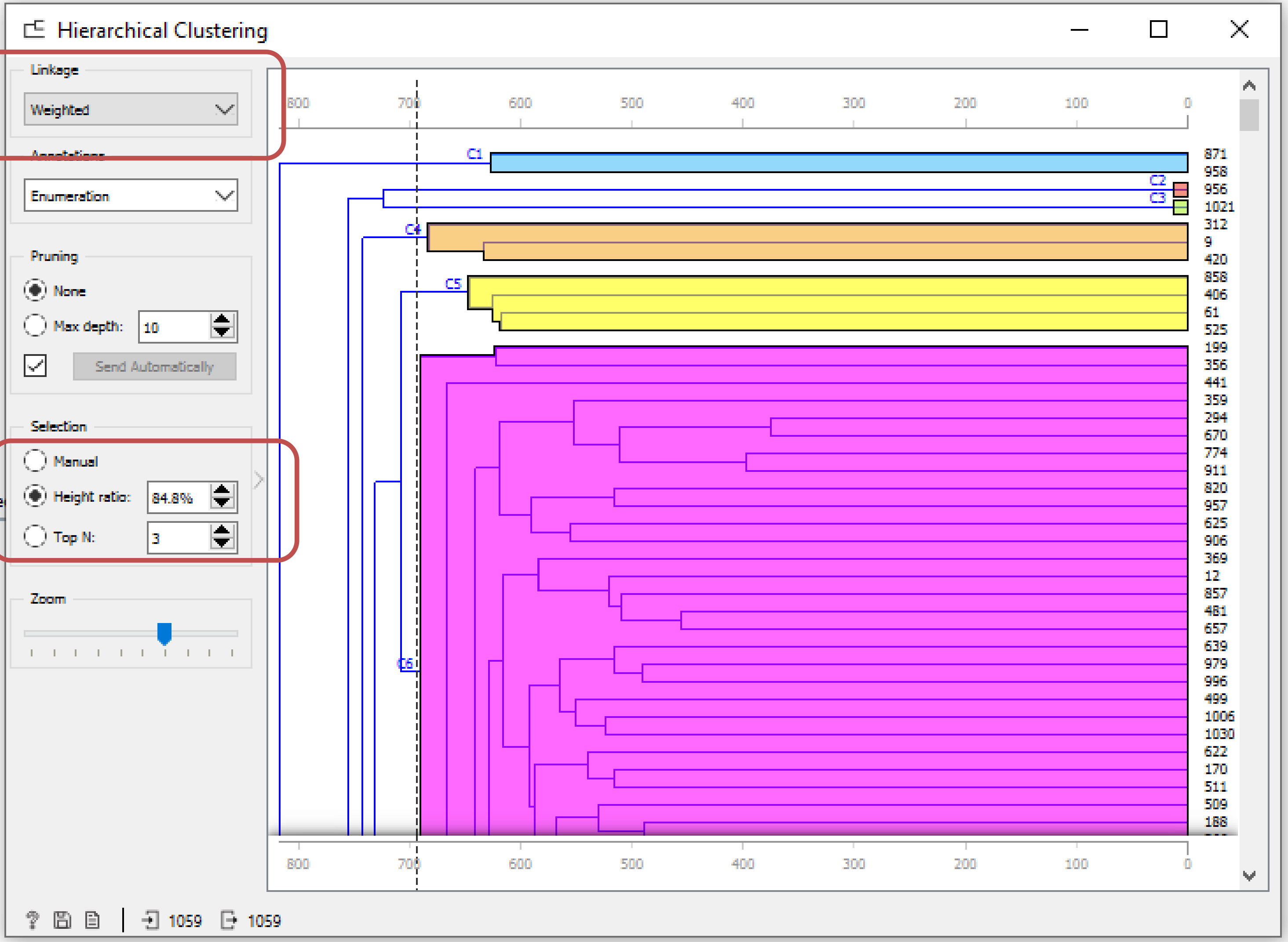
1059

Step 8

Hierarchical clustering widget partitions the images in clusters.

You can explore different Linkage and Selection parameters




Hierarchical Clustering

Linkage: Weighted

Annotations: Enumeration

Pruning
- ● None
- ○ Max depth: 10

☑ Send Automatically

Selection
- ○ Manual
- ● Height ratio: 84.8%
- ○ Top N: 3

Zoom

1059 1059

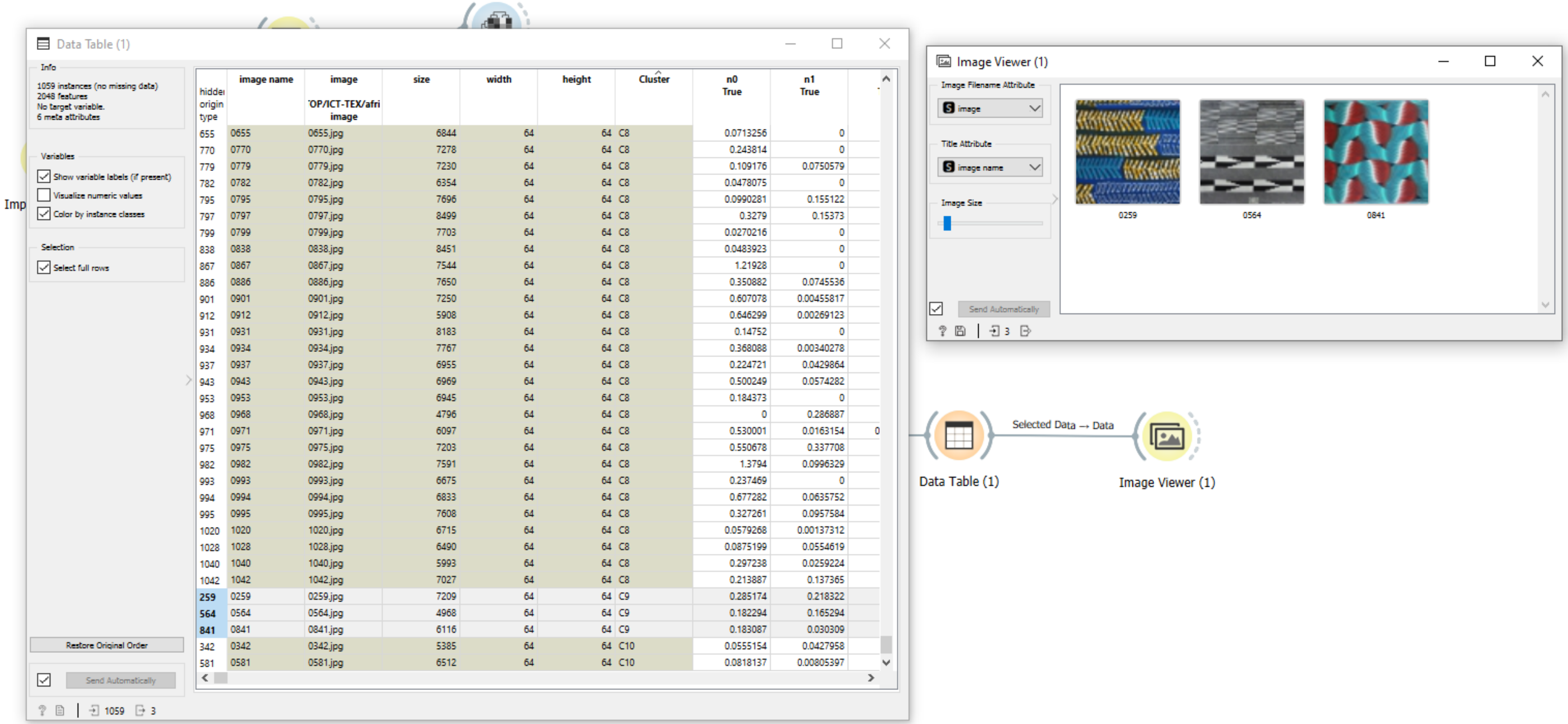
Co-funded by the
Erasmus+ Programme
of the European Union

Step 9

To examine the clustering results can be used Datasets table and Image viewer.

You can select the images from some cluster and visualize only them for manual evaluation of their similarity.

Please note, that you can first click on the "Cluster" column header in order to group by cluster all data rows



Data Table (1)

Info
1059 instances (no missing data)
2048 features
No target variable.
6 meta attributes

Variables
☑ Show variable labels (if present)
☐ Visualize numeric values
☑ Color by instance classes

Selection
☑ Select full rows

Restore Original Order

☑ Send Automatically

1059 3

hidde origin type	image name	image 'OP/ICT-TEX/afri image	size	width	height	Cluster	n0 True	n1 True	
655	0655	0655.jpg	6844	64	64	C8	0.0713256	0	
770	0770	0770.jpg	7278	64	64	C8	0.243814	0	
779	0779	0779.jpg	7230	64	64	C8	0.109176	0.0750579	
782	0782	0782.jpg	6354	64	64	C8	0.0478075	0	
795	0795	0795.jpg	7696	64	64	C8	0.0990281	0.155122	
797	0797	0797.jpg	8499	64	64	C8	0.3279	0.15373	
799	0799	0799.jpg	7703	64	64	C8	0.0270216	0	
838	0838	0838.jpg	8451	64	64	C8	0.0483923	0	
867	0867	0867.jpg	7544	64	64	C8	1.21928	0	
886	0886	0886.jpg	7650	64	64	C8	0.350882	0.0745536	
901	0901	0901.jpg	7250	64	64	C8	0.607078	0.00455817	
912	0912	0912.jpg	5908	64	64	C8	0.646299	0.00269123	
931	0931	0931.jpg	8183	64	64	C8	0.14752	0	
934	0934	0934.jpg	7767	64	64	C8	0.368088	0.00340278	
937	0937	0937.jpg	6955	64	64	C8	0.224721	0.0429864	
943	0943	0943.jpg	6969	64	64	C8	0.500249	0.0574282	
953	0953	0953.jpg	6945	64	64	C8	0.184373	0	
968	0968	0968.jpg	4796	64	64	C8	0	0.286887	
971	0971	0971.jpg	6097	64	64	C8	0.530001	0.0163154	0
975	0975	0975.jpg	7203	64	64	C8	0.550678	0.337708	
982	0982	0982.jpg	7591	64	64	C8	1.3794	0.0996329	
993	0993	0993.jpg	6675	64	64	C8	0.237469	0	
994	0994	0994.jpg	6833	64	64	C8	0.677282	0.0635752	
995	0995	0995.jpg	7608	64	64	C8	0.327261	0.0957584	
1020	1020	1020.jpg	6715	64	64	C8	0.0579268	0.00137312	
1028	1028	1028.jpg	6490	64	64	C8	0.0875199	0.0554619	
1040	1040	1040.jpg	5993	64	64	C8	0.297238	0.0259224	
1042	1042	1042.jpg	7027	64	64	C8	0.213887	0.137365	
259	0259	0259.jpg	7209	64	64	C9	0.285174	0.218322	
564	0564	0564.jpg	4968	64	64	C9	0.182294	0.165294	
841	0841	0841.jpg	6116	64	64	C9	0.183087	0.030309	
342	0342	0342.jpg	5385	64	64	C10	0.0555154	0.0427958	
581	0581	0581.jpg	6512	64	64	C10	0.0818137	0.00805397	

Image Viewer (1)

Image Filename Attribute: image

Title Attribute: image name

Image Size

☑ Send Automatically

3

Examples for clusters for some hyper parameters

- Metric: Spearman
- Linkage: Weighted
- Selection: Height ratio: 75%

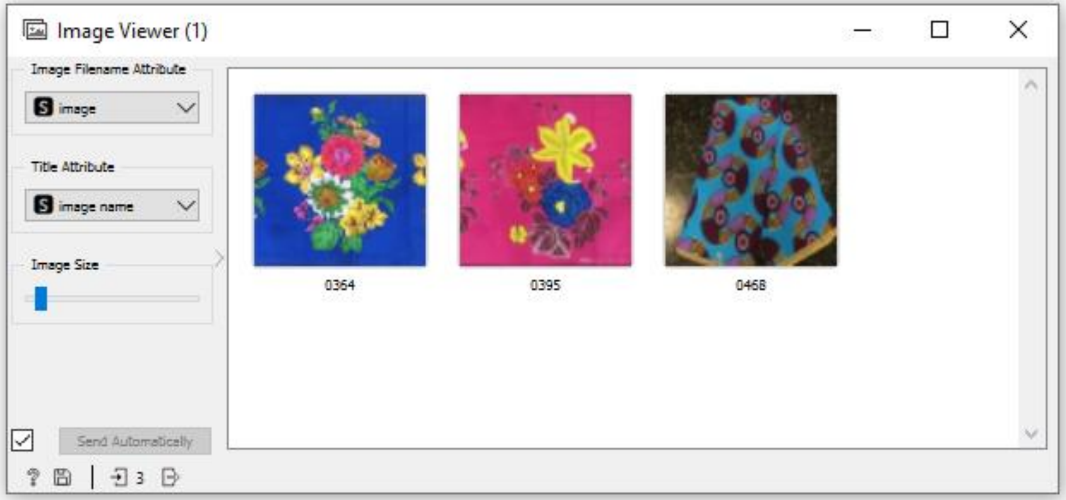


Image Viewer (1)

Image Filename Attribute: image

Title Attribute: image name

Image Size

☑ Send Automatically

- Metric: Absolute Spearman
- Linkage: Weighted
- Selection: Height ratio: 75%

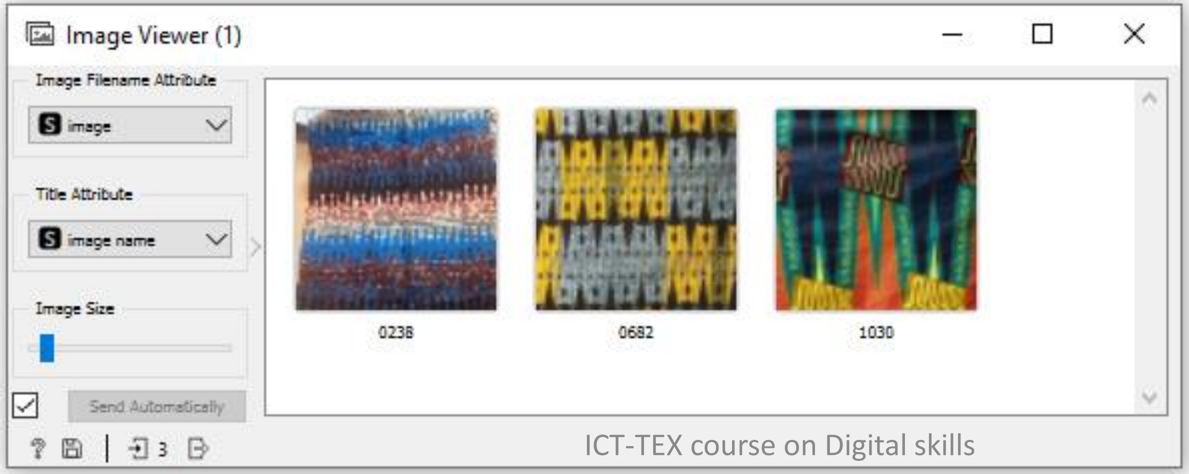


Image Viewer (1)

Image Filename Attribute: image

Title Attribute: image name

Image Size

☑ Send Automatically

Examples for clusters for some hyper parameters

- Metric: Absolute Pearson
- Linkage: Complete
- Selection: Height ratio: 75%

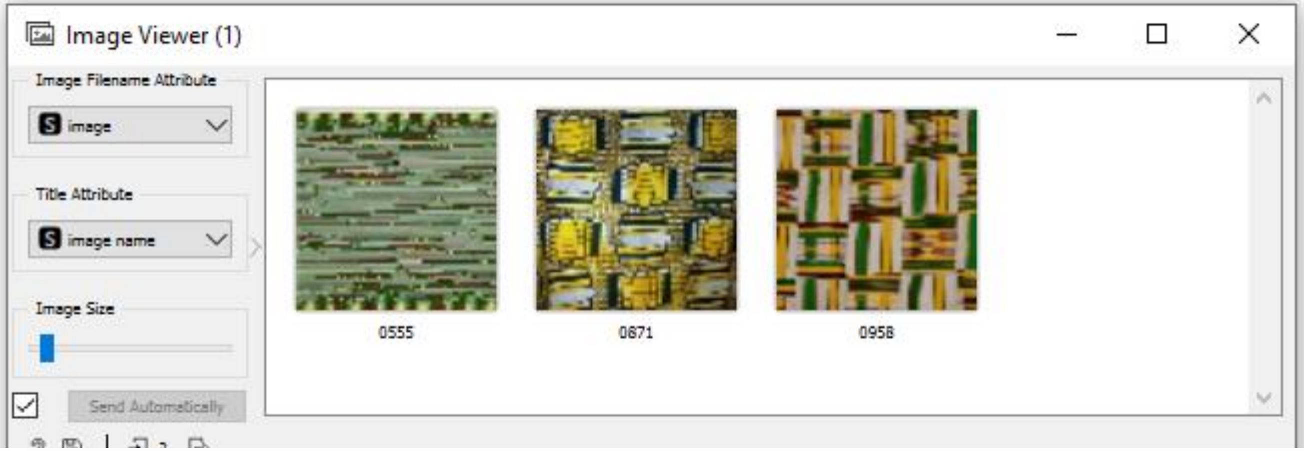


Image Viewer (1)

Image Filename Attribute: image

Title Attribute: image name

Image Size

☑ Send Automatically

0555 0871 0958

- Metric: Pearson
- Linkage: Complete
- Selection: Height ratio: 75%

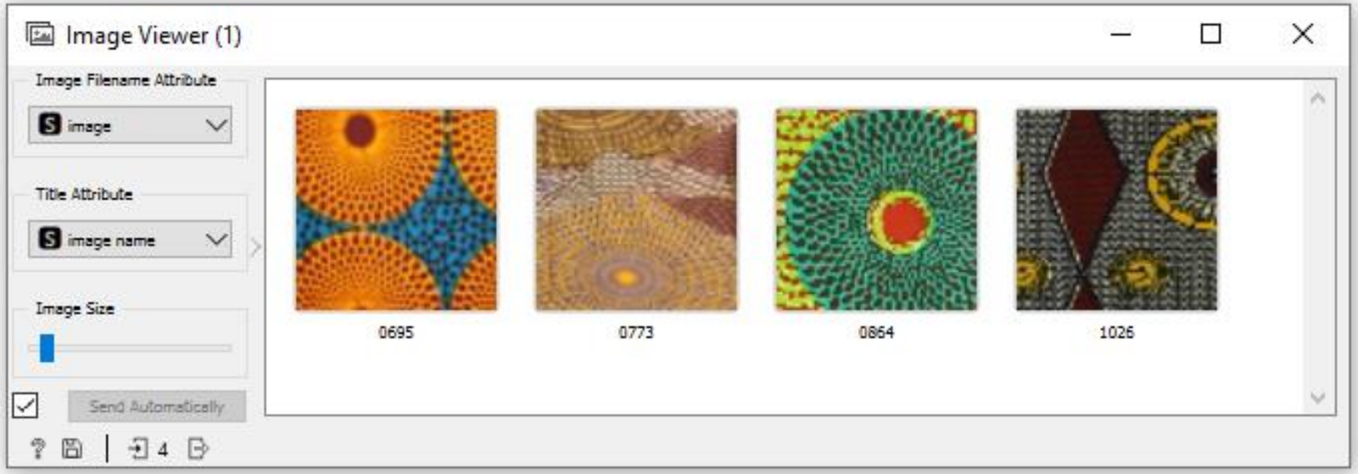


Image Viewer (1)

Image Filename Attribute: image

Title Attribute: image name

Image Size

☑ Send Automatically

0695 0773 0864 1026

4

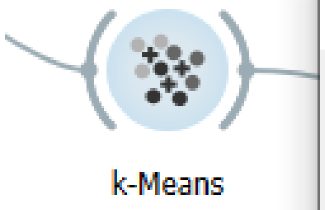
Step 10

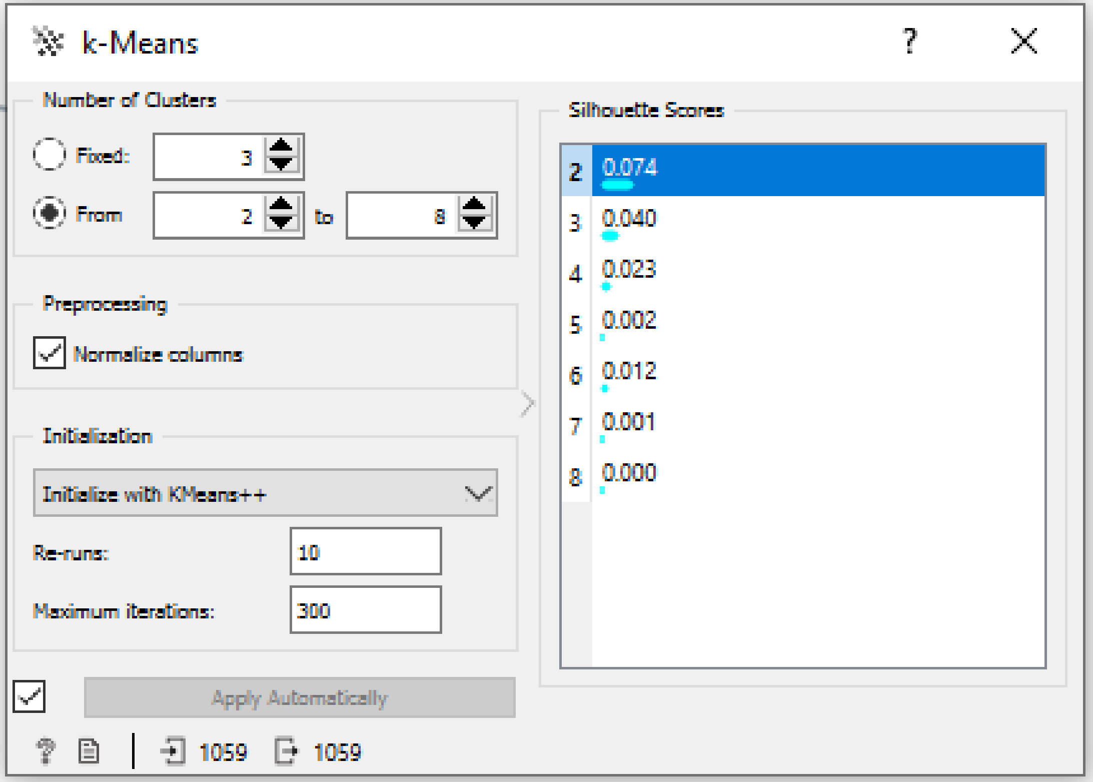
To explore and evaluate different numbers of clusters using density-based clustering we can use k-Means widget

Because we do not know the number of clusters, we can specify a range in which will be explored the value K and the result models will be scored.

The best model is the one with the highest score.

In the current example the best model is for K=2, i.e. 2 clusters




k-Means

Number of Clusters

○ Fixed: 3

● From 2 to 8

Preprocessing

☑ Normalize columns

Initialization

Initialize with KMeans++

Re-runs: 10

Maximum iterations: 300

☑ Apply Automatically

1059 1059

Silhouette Scores

2	0.074
3	0.040
4	0.023
5	0.002
6	0.012
7	0.001
8	0.000

Step 11

Like in Step 9 we add data table and image viewer widgets for manual inspection of clusters.

For such rich dataset, splitting the images in two datasets, means that only some of the features are taken in consideration. It is quite difficult to determine the similarity by the fabrics design in these two clusters.

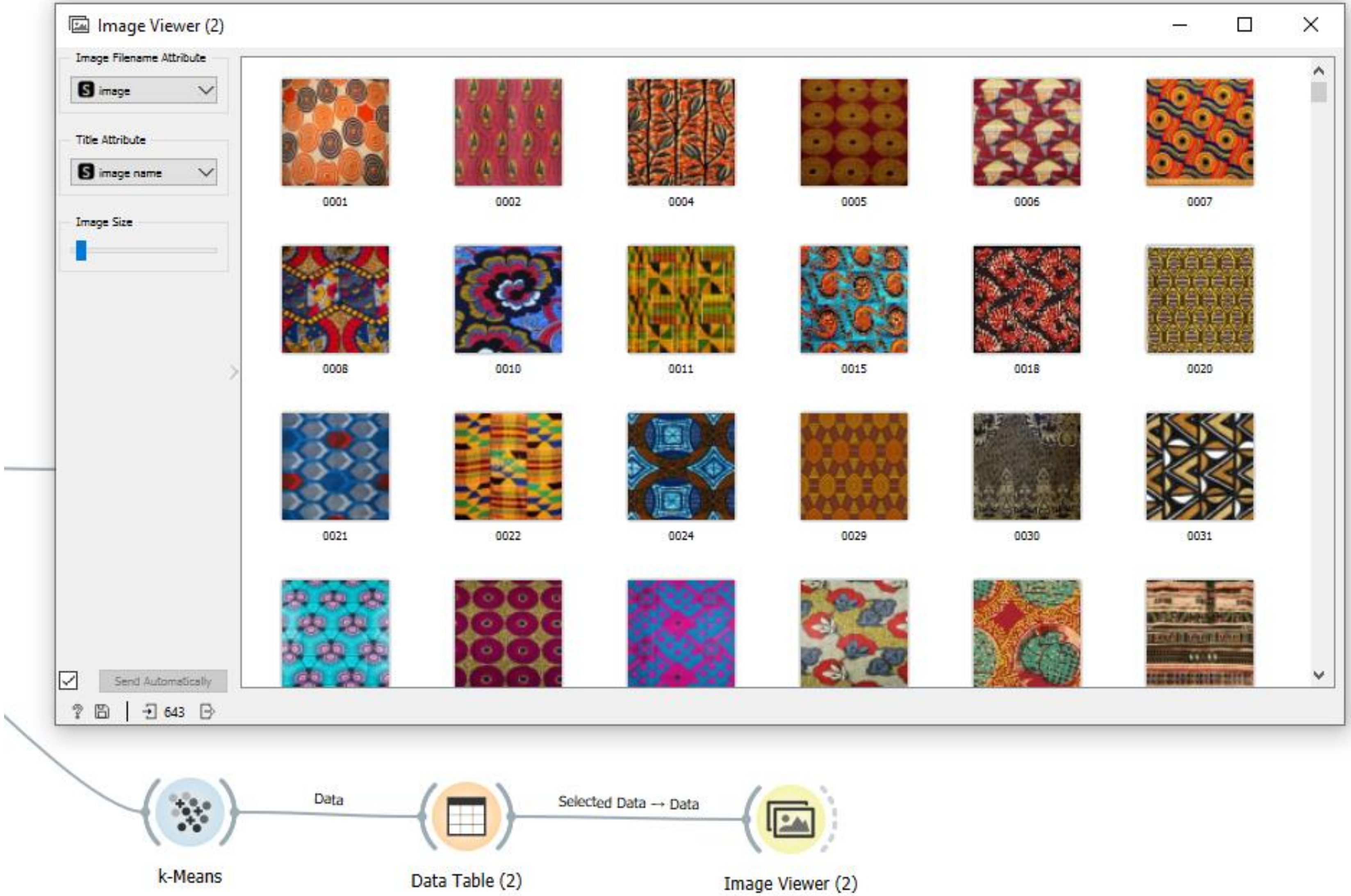


Image Filename Attribute: image

Title Attribute: image name

Discussion

- The results of ML clusterization models show that the best result have:
  - Metric: (Absolute) Pearson
  - Linkage: Complete
  - Selection: Height ratio: 75%
- The huge variety of patterns, needs the dataset to be splitted in too many clusters.
- The results of demonstrated ML classification models are promising and show that the task can be solved with satisfactory precision for stain detection in fabric manufacturing.

References:

- The material of these slides is based on the following resources:
  - Applications of AI in Textile Industry https://frontier.cool/blogposts/importance-machine-learning-textile-industry
  - Orange widget catalog: https://orangedatamining.com/widget-catalog/
  - Orange Data Mining Framework: https://orangedatamining.com/
  - Distance Metrics: https://medium.com/analytics-vidhya/various-types-of-distance-metrics-machine-learning-cc9d4698c2da

CONTACTS

Coordinator:
Technical University of Sofia

Project coordinator:
assoc. prof. Angel Terziev, PhD
aterziev@tu-sofia.bg

Web-site: [ICT-TEX.eu](ICT-TEX.eu)

Co-funded by the
Erasmus+ Programme
of the European Union

KNOWLEDGE ALLIANCE
ICT-TEX
ICT IN TEXTILE AND CLOTHING
HIGHER EDUCATION AND BUSINESS

The information and views set out in this publication are those of the authors and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.