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 7: 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

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

SOFIA UNIVERSITY
"ST. KLIMENT OHRIDSKI
EST. 1888

ICT-TEX

Hands-on exercise

CASE STUDY:
FABRIC STAIN DEFECT CLASSIFICATION

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

SOFIA UNIVERSITY
"ST. KLIMENT OHRIDSKI
EST. 1888

ICT-TEX

These slides are part of the Topic 7 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

- In textile industry one of the most common important tasks is quality control and monitoring of fabric production
- Several problems for fabric damage can occur like – stains, holes, pattern damage, etc.
- Manual inspection is almost impossible due to the huge endless area that should be inspected
- AI technologies can help for this task – for automatic optic inspection of fabric production, by training ML model with different categories of images to identify problems

Problem

- In textile industry one of the most common application of AI is monitoring of fabric production and detection of different defects, like stains, holes, texture, pattern print, etc.
- In this case study we will investigate methods for binary classification, i.e. classification with two classes:
  - Fabrics without defects and
  - Fabrics with different stains.

Challenges

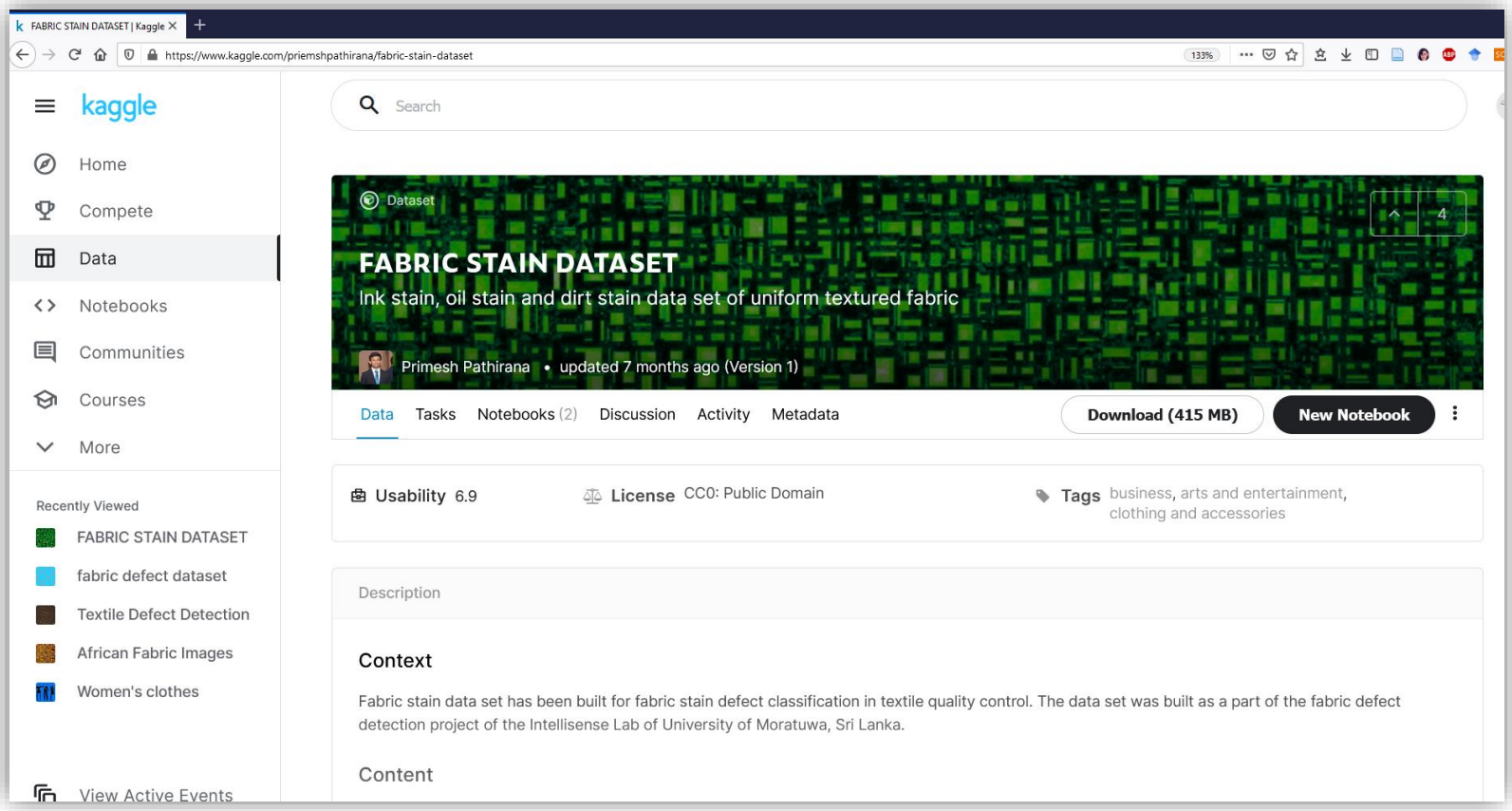
- There is a huge variety of characteristics of stains:
  - size, location, type, shape, number, etc.
- In case the fabric has some printed pattern – the task of stain detection is quite challenging. Thus, for simplicity we will consider only plain fabrics (without pattern).
- For additional simplification and generalization – we will not consider the color of fabric, i.e. we will use grayscale images

Data

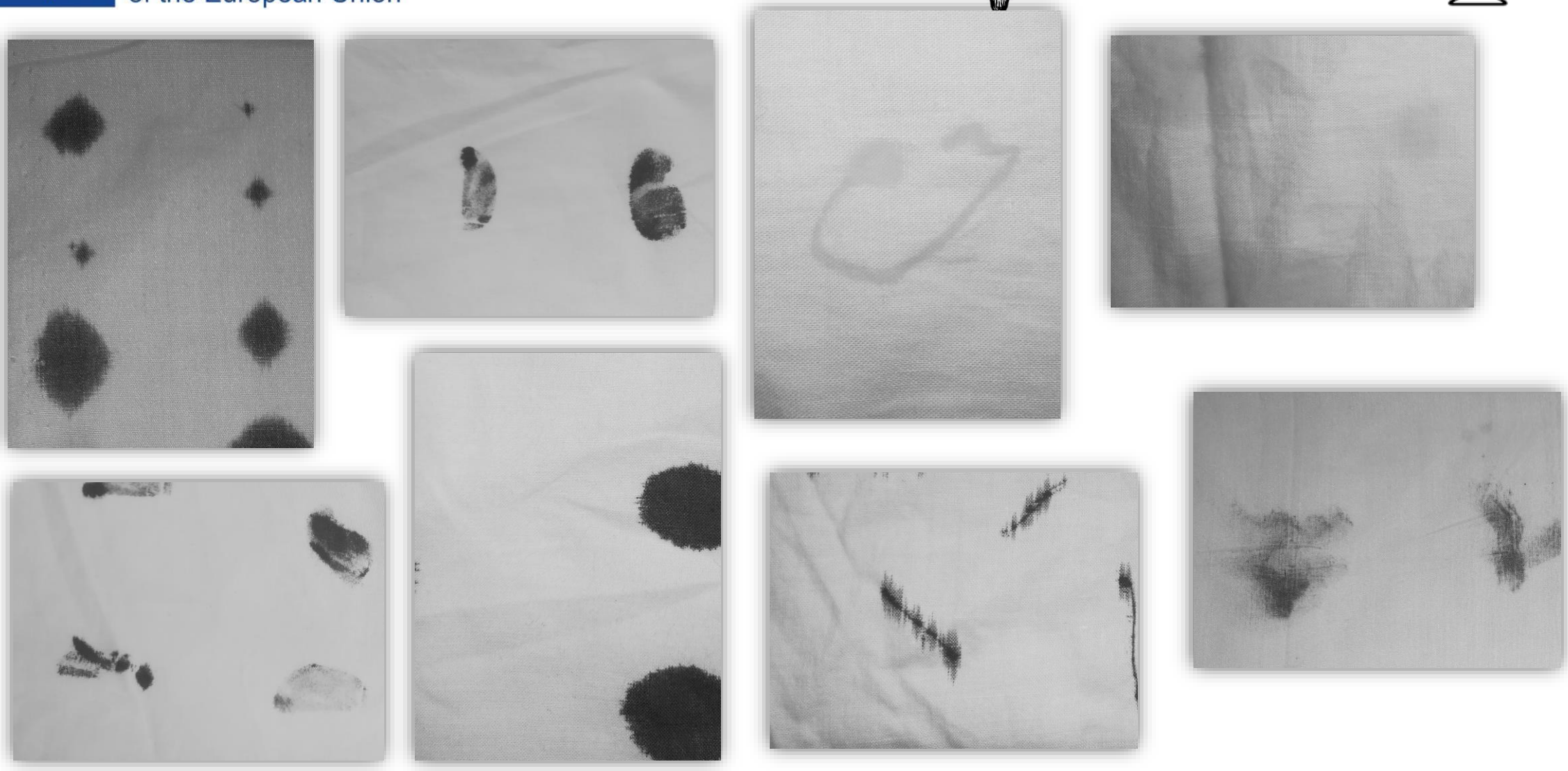
- We need dataset that contain enough classified images of:
  - Fabrics without defects
  - Fabrics with stains
- There is a huge variety of characteristics of stains:
  - size, location, type, shape, number, etc.
- In addition, in case the fabric has some printed pattern – the task of stain detection is quite challenging

Data

- Searching for some open data sets we identify the **„Fabric stain dataset"** in Kaggle that perfectly fits the problem needs.
- Please, download the dataset locally on your computer.



https://www.kaggle.com/priemshpathirana/fabric-stain-dataset



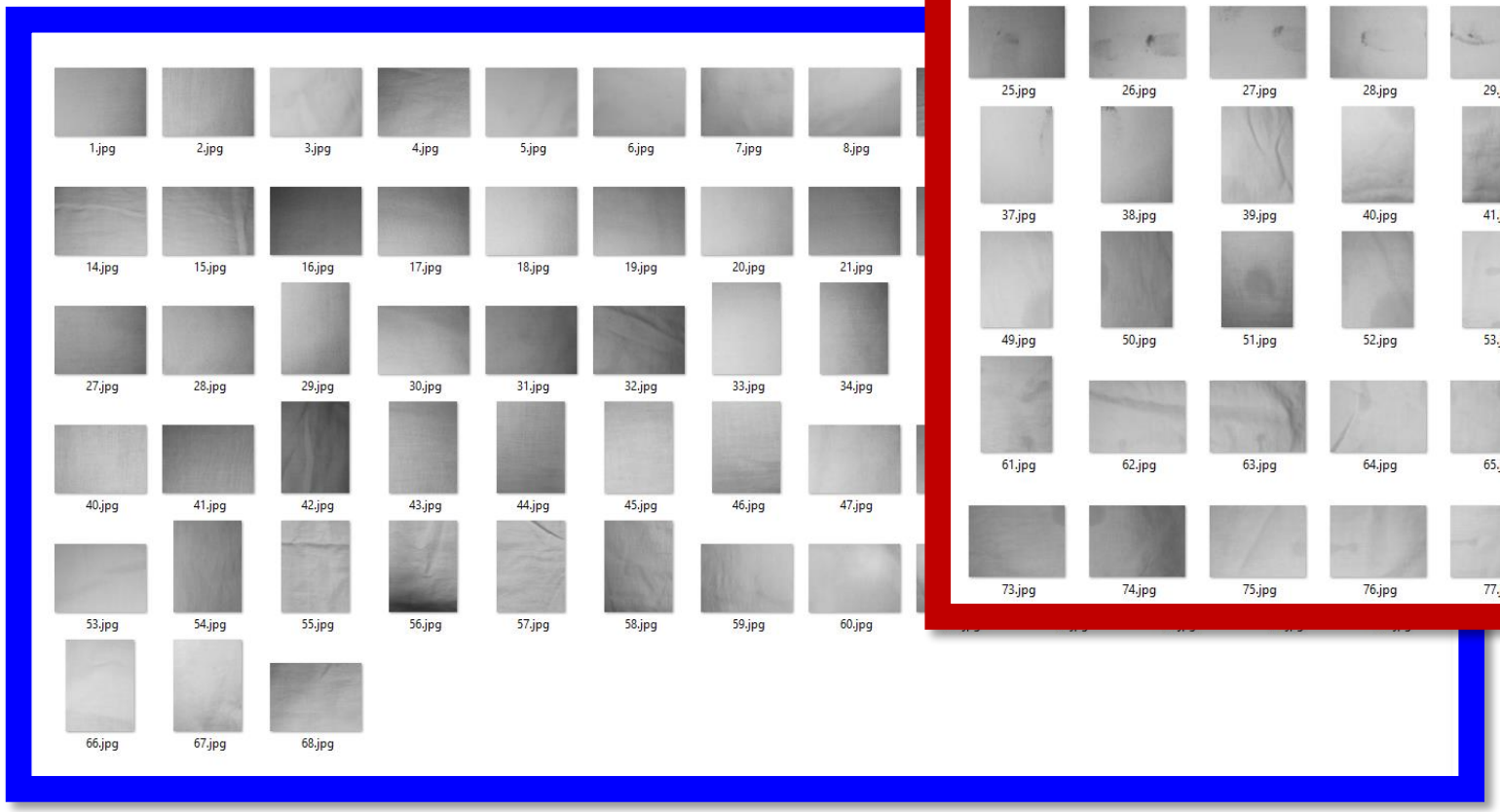
Different types of stains

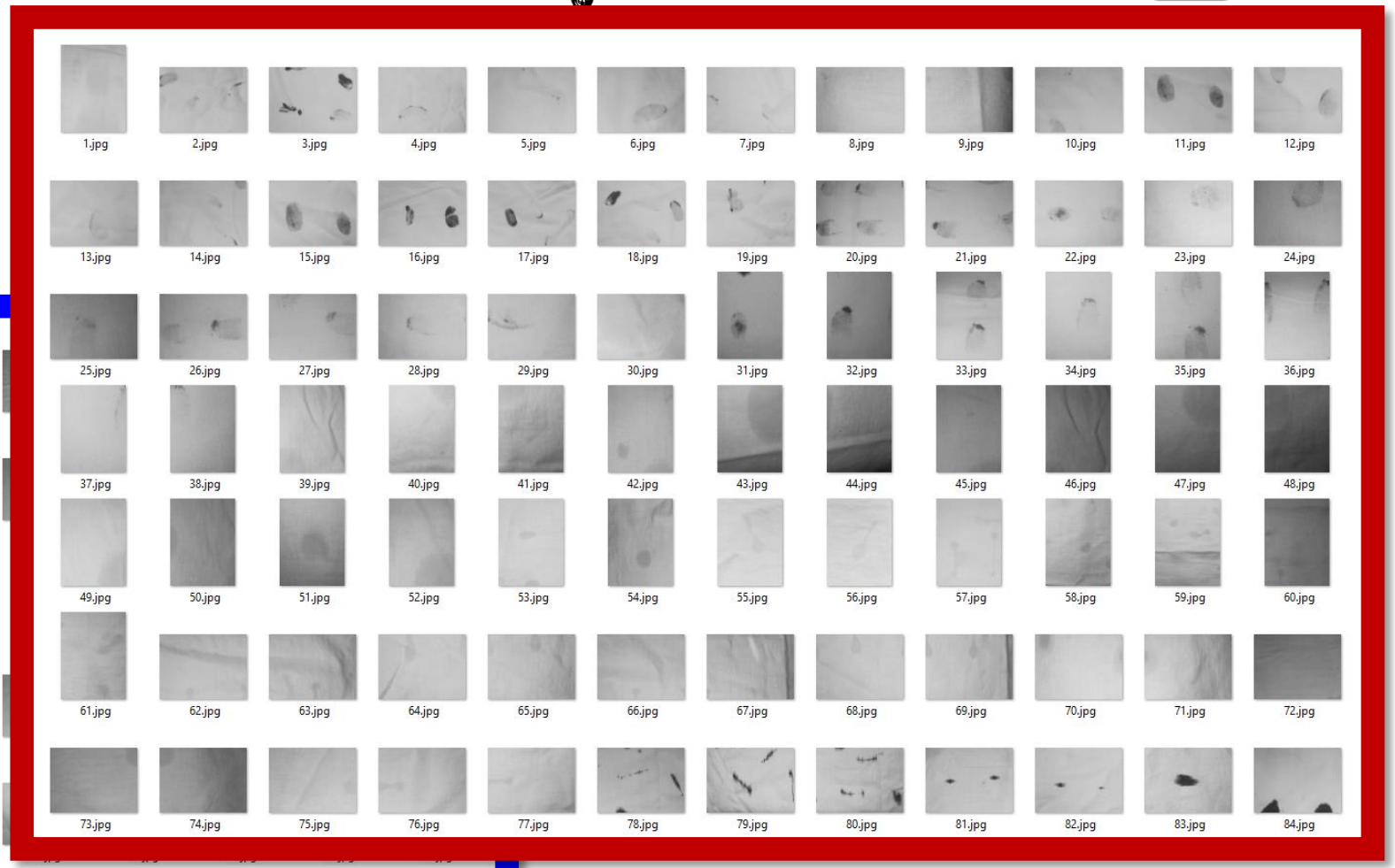
Co-funded by the
Erasmus+ Programme
of the European Union

SOFIA UNIVERSITY
"ST. KLIMENT OHRIDSKI
EST. 1888

ICT-TEX

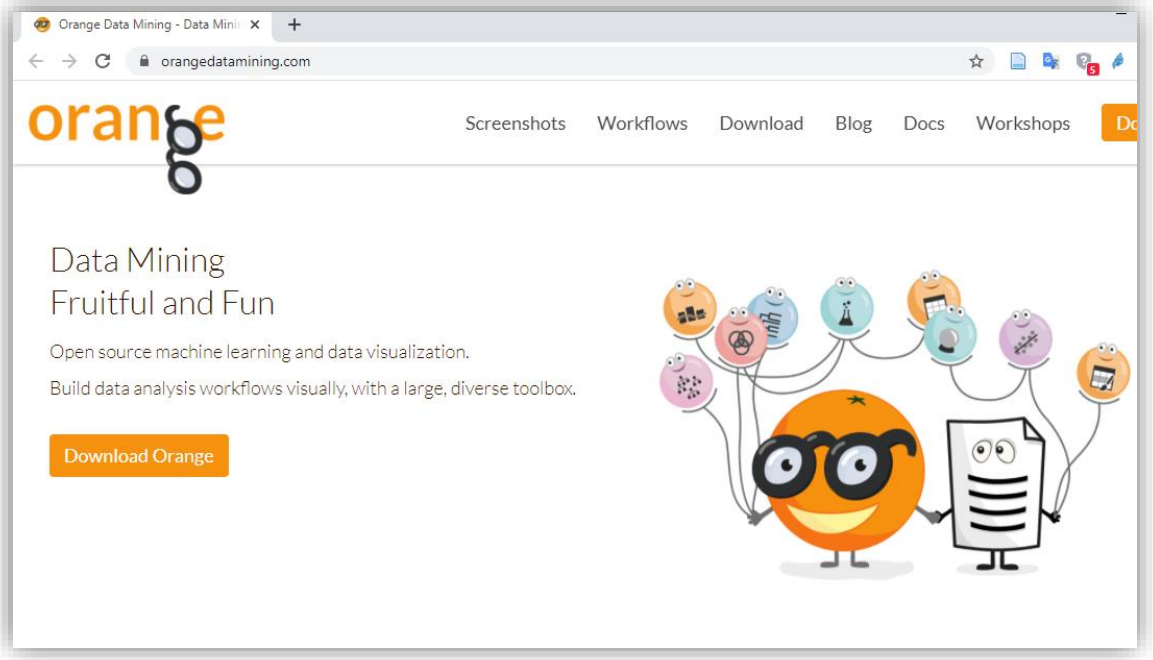
- The dataset contains:
  - 68 images: **"defect free"**
  - 398 images: **"stain"**





Framework

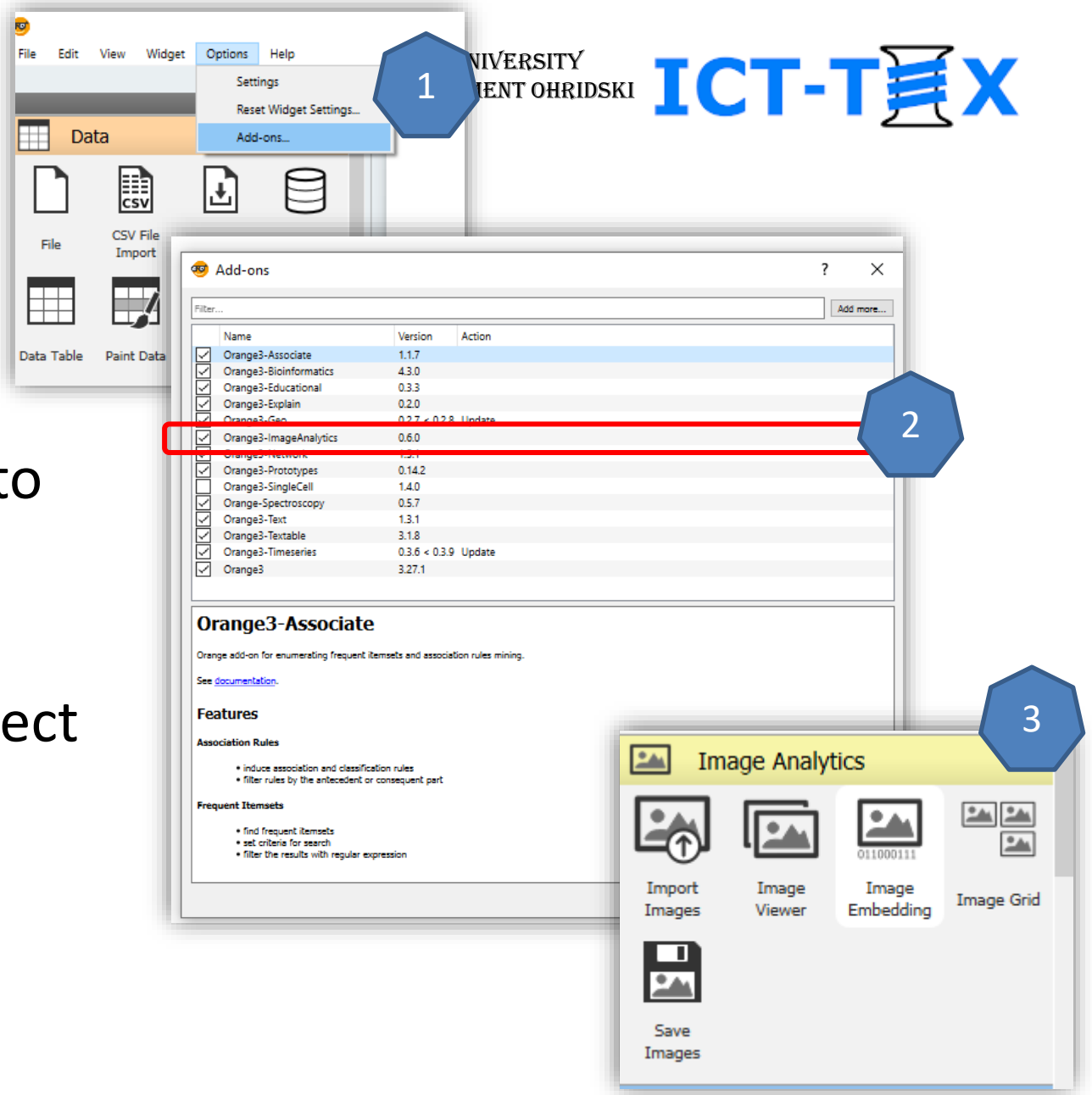
- For processing data, we will use Orange Data Mining Framework – free software that provides the basic AI tools in user friendly format that is appropriate for use even from non-technical users.
- Please, install Orange (https://orangedatamining.com/)



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

Setup

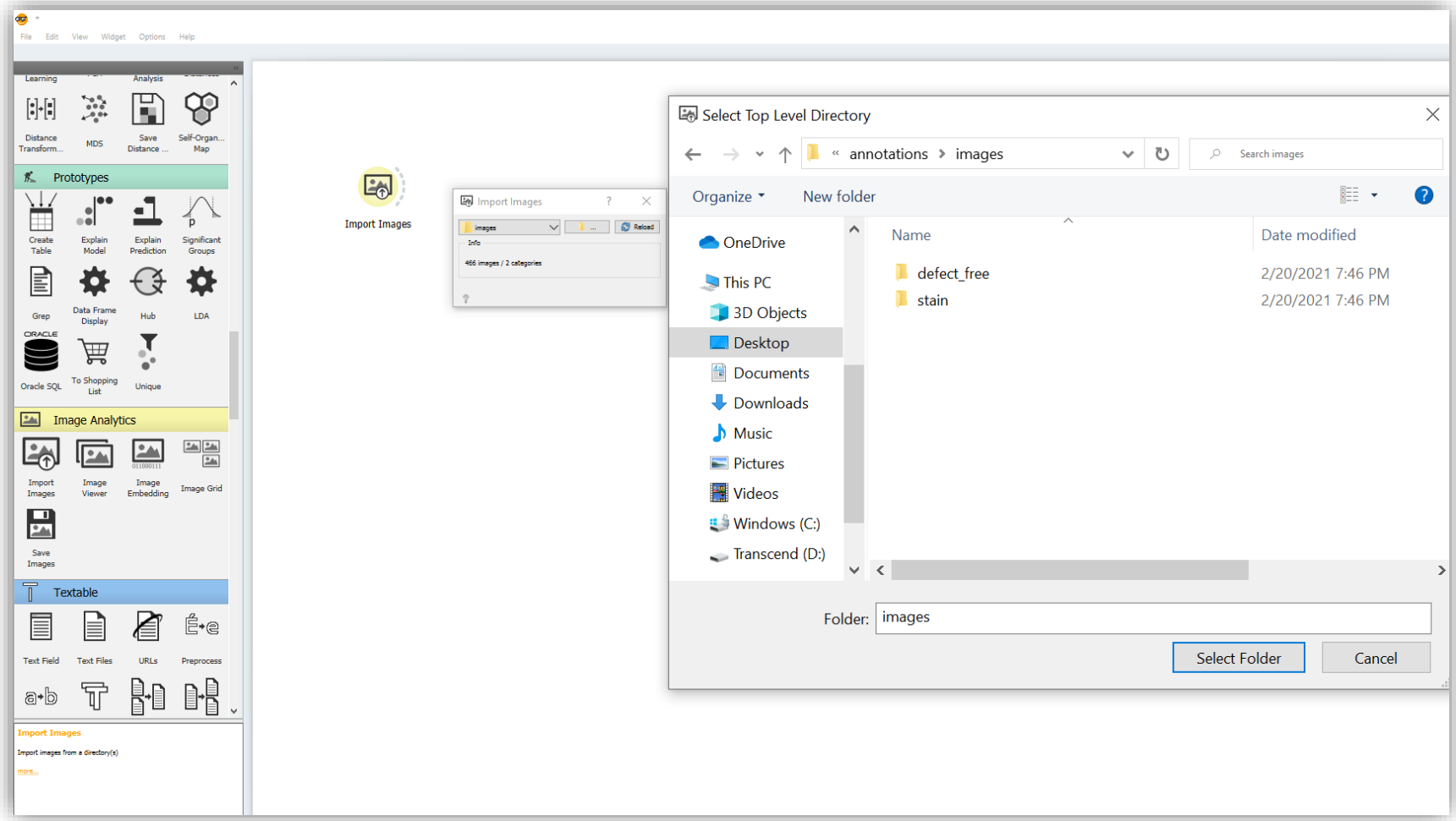
- The core version of Orange contains basic functionalities only. You need to install some add-ons for Image processing.
- To install Image Analytics, please select from the menu Options->Add-ons
- After installation In the Toolbox will appear Image Analytics panel



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



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

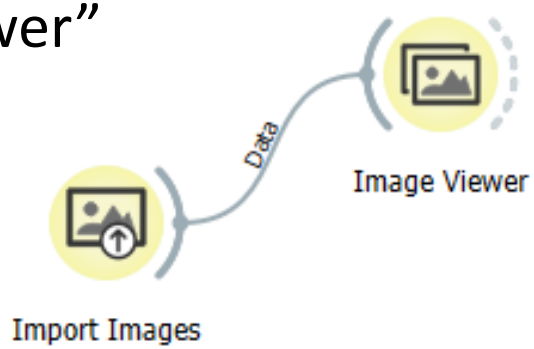
Co-funded by the Erasmus+ Programme of the European Union

SOFIA UNIVERSITY "ST. KLIMENT OHRIDSKI" EST. 1888

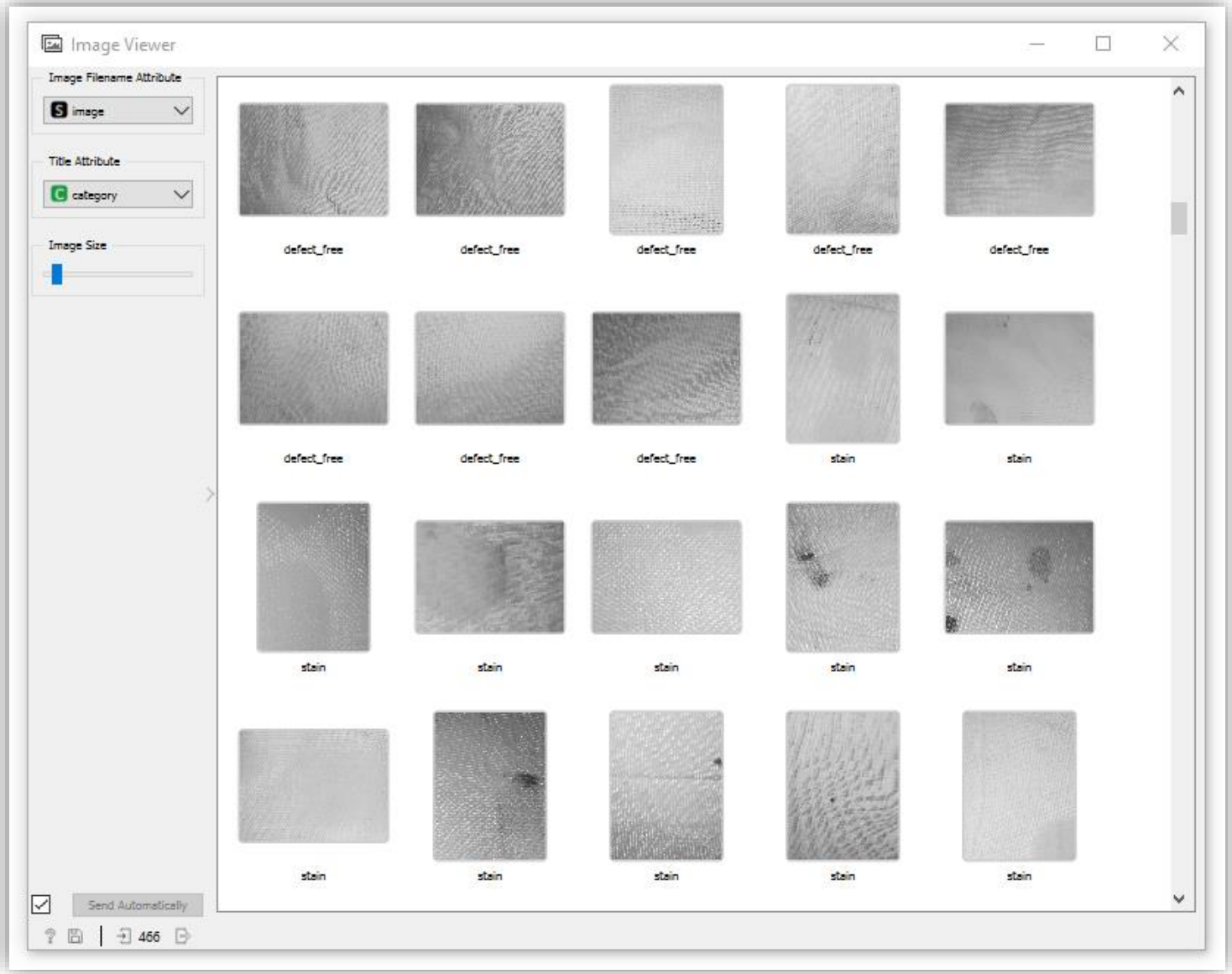
ICT-TEX

Step 2

To inspect the content of the loaded dataset can be added widget "Image Viewer"

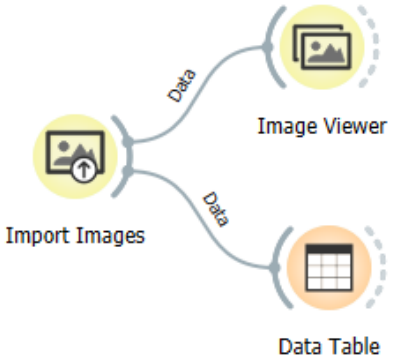


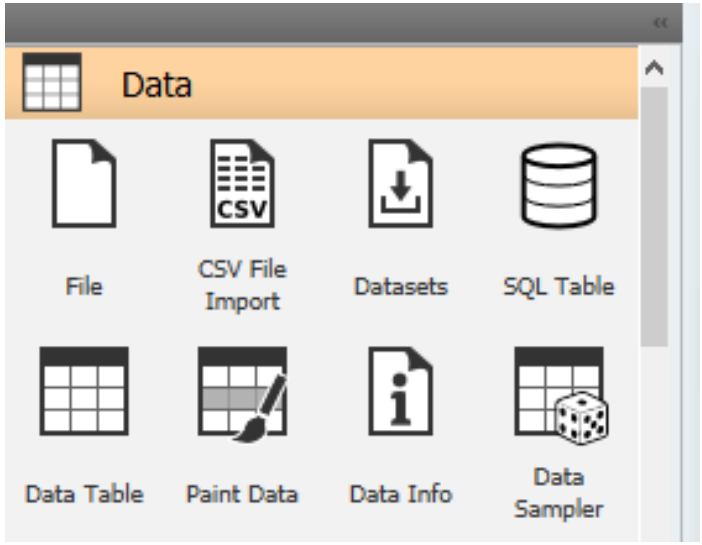
Note that all images from the folder "stain" are classified with category "stain", and those from "defect_free" have category "defect_free"

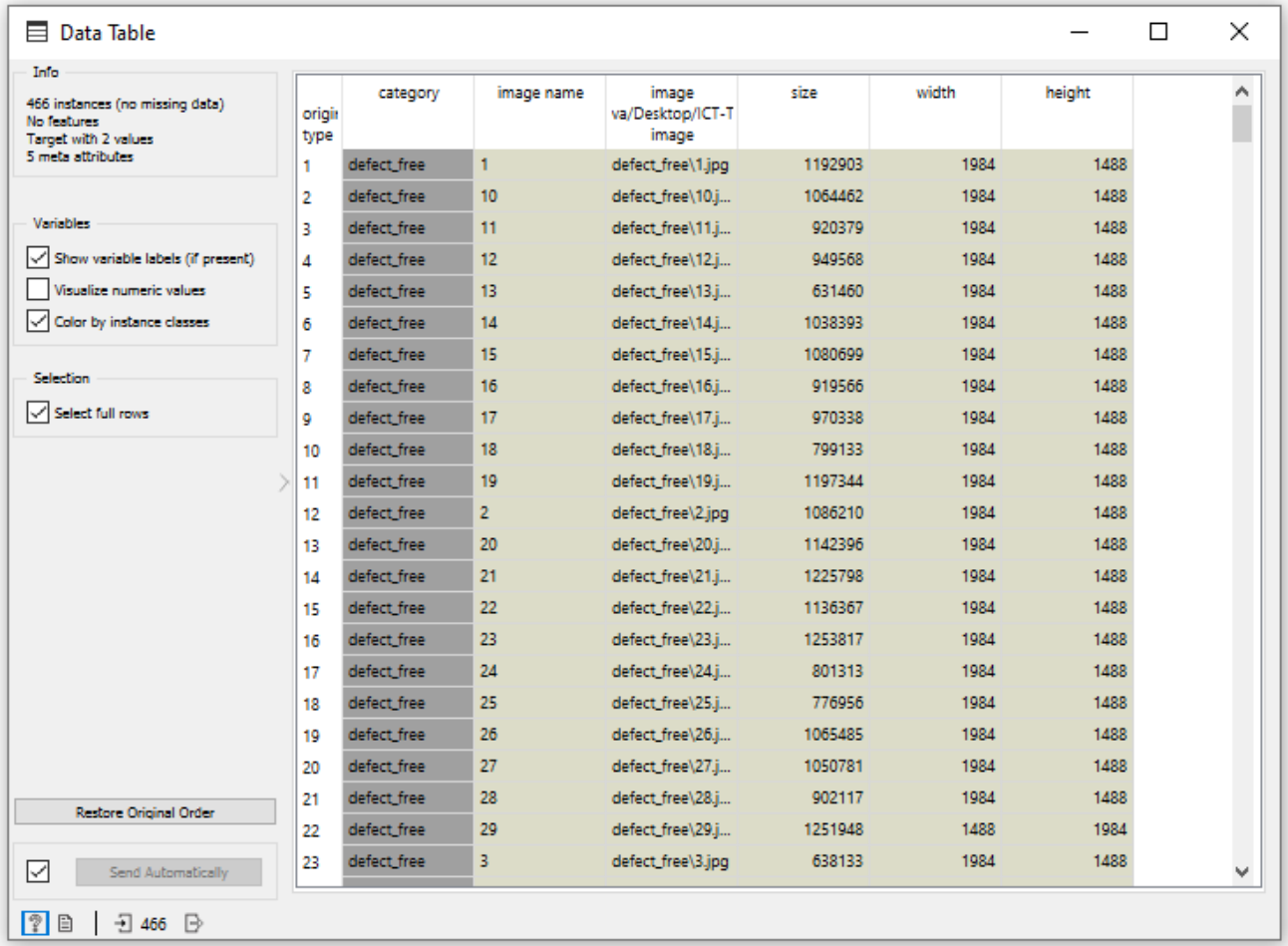


Step 3

- Another view for data inspection that we can be used is “Data table” widget from panel Data





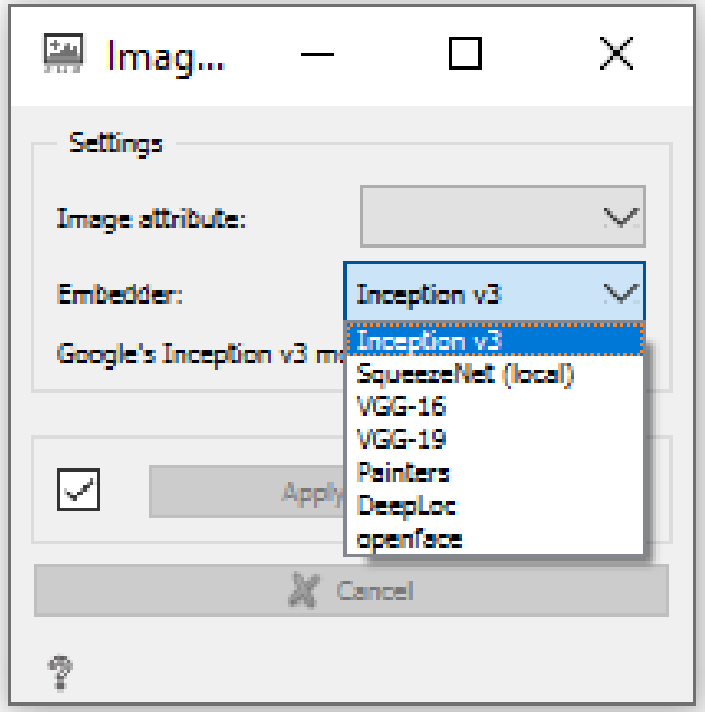


origi type	category	image name	image va/Desktop/ICT-T image	size	width	height
1	defect_free	1	defect_free\1.jpg	1192903	1984	1488
2	defect_free	10	defect_free\10.j...	1064462	1984	1488
3	defect_free	11	defect_free\11.j...	920379	1984	1488
4	defect_free	12	defect_free\12.j...	949568	1984	1488
5	defect_free	13	defect_free\13.j...	631460	1984	1488
6	defect_free	14	defect_free\14.j...	1038393	1984	1488
7	defect_free	15	defect_free\15.j...	1080699	1984	1488
8	defect_free	16	defect_free\16.j...	919566	1984	1488
9	defect_free	17	defect_free\17.j...	970338	1984	1488
10	defect_free	18	defect_free\18.j...	799133	1984	1488
11	defect_free	19	defect_free\19.j...	1197344	1984	1488
12	defect_free	2	defect_free\2.jpg	1086210	1984	1488
13	defect_free	20	defect_free\20.j...	1142396	1984	1488
14	defect_free	21	defect_free\21.j...	1225798	1984	1488
15	defect_free	22	defect_free\22.j...	1136367	1984	1488
16	defect_free	23	defect_free\23.j...	1253817	1984	1488
17	defect_free	24	defect_free\24.j...	801313	1984	1488
18	defect_free	25	defect_free\25.j...	776956	1984	1488
19	defect_free	26	defect_free\26.j...	1065485	1984	1488
20	defect_free	27	defect_free\27.j...	1050781	1984	1488
21	defect_free	28	defect_free\28.j...	902117	1984	1488
22	defect_free	29	defect_free\29.j...	1251948	1488	1984
23	defect_free	3	defect_free\3.jpg	638133	1984	1488

Step 4

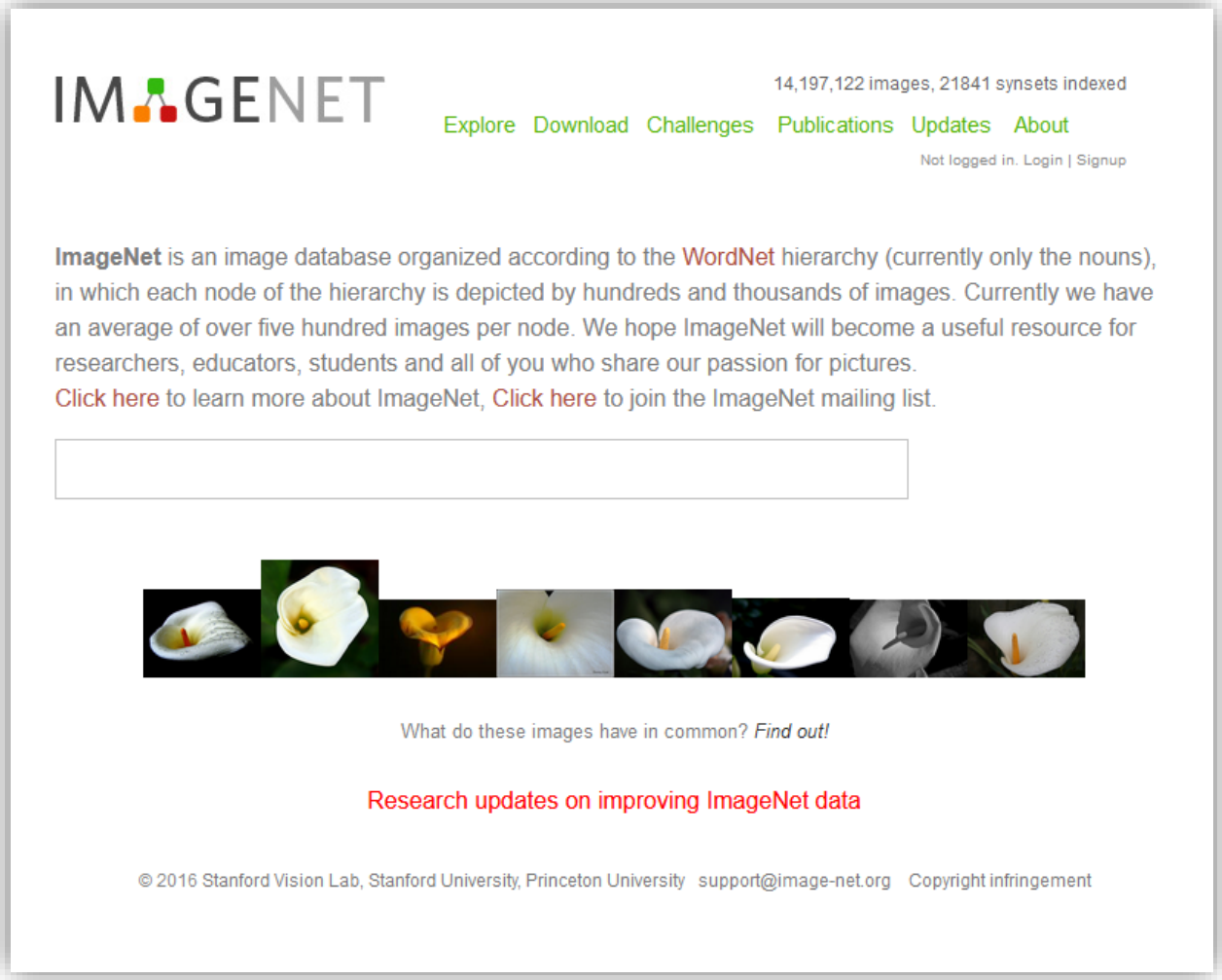
- Image Embedding uses pretrained Deep Learning models to calculate features vectors for each image.
- There are several available DL models that you can choose ("Embedder").
- Please, not that the majority of them will require availability of Internet connection in order to connect to the server for evaluation.

Image Embedding



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

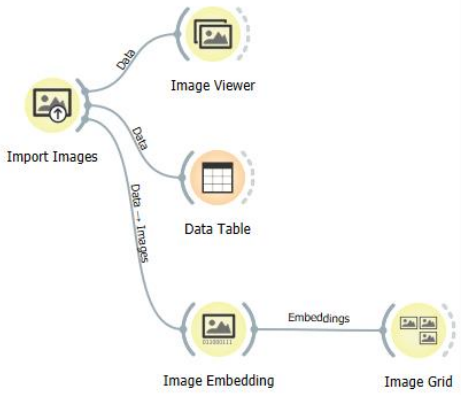
- For this case study we will use
  - **InceptionV3** is Google's deep neural network for image recognition. It is trained on the ImageNet data set. (http://image-net.org/index)

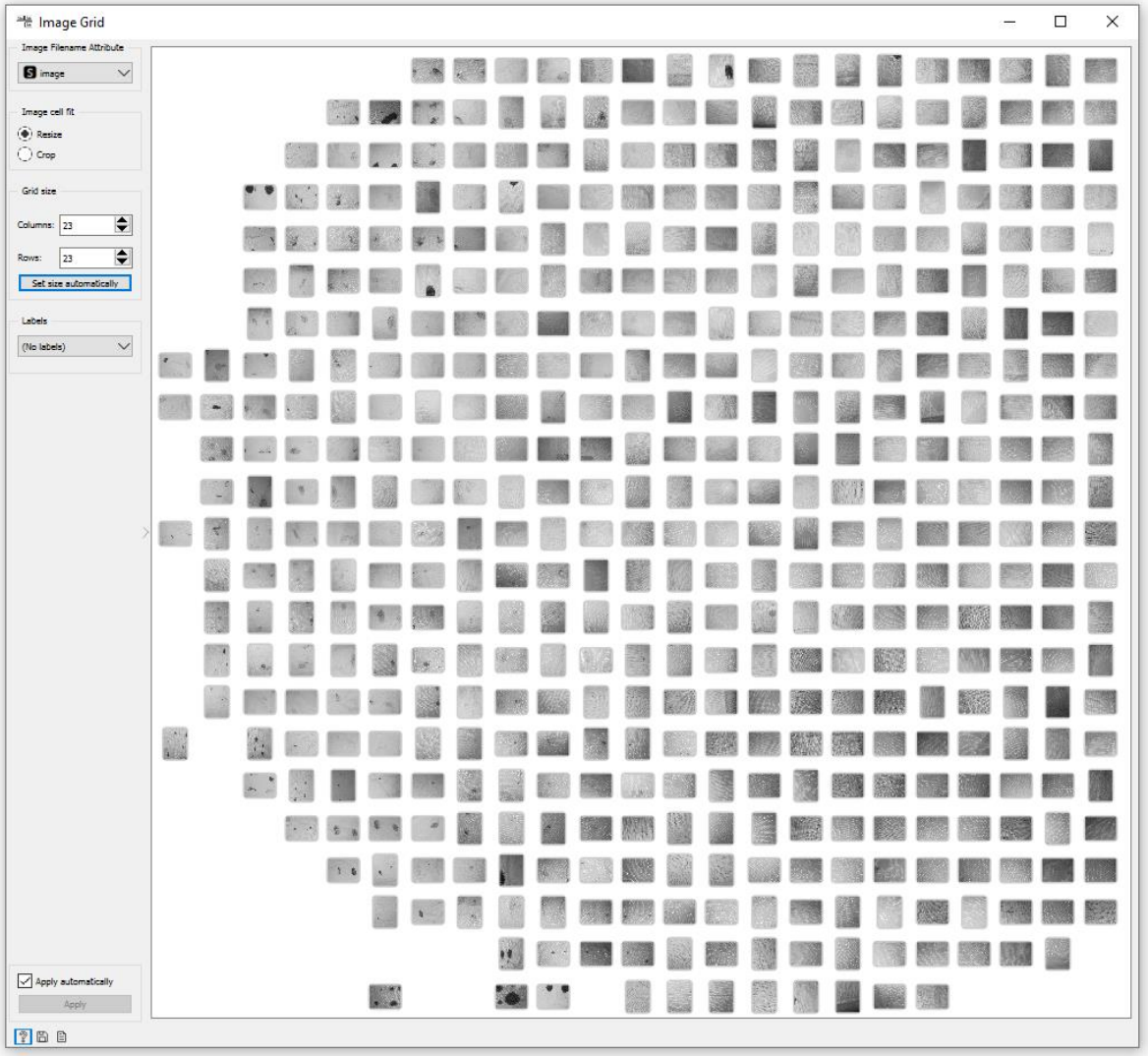


[1] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016. https://arxiv.org/abs/1512.00567

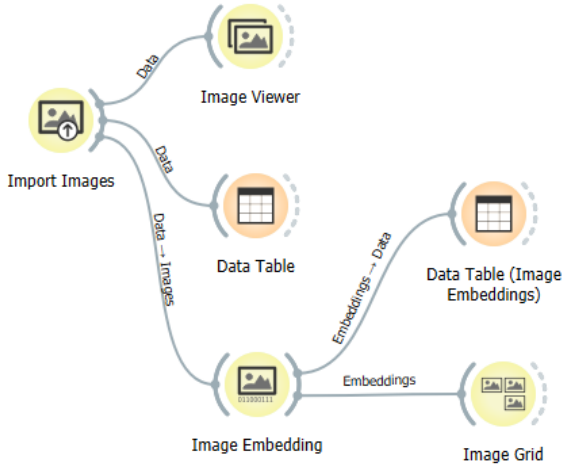
Step 5

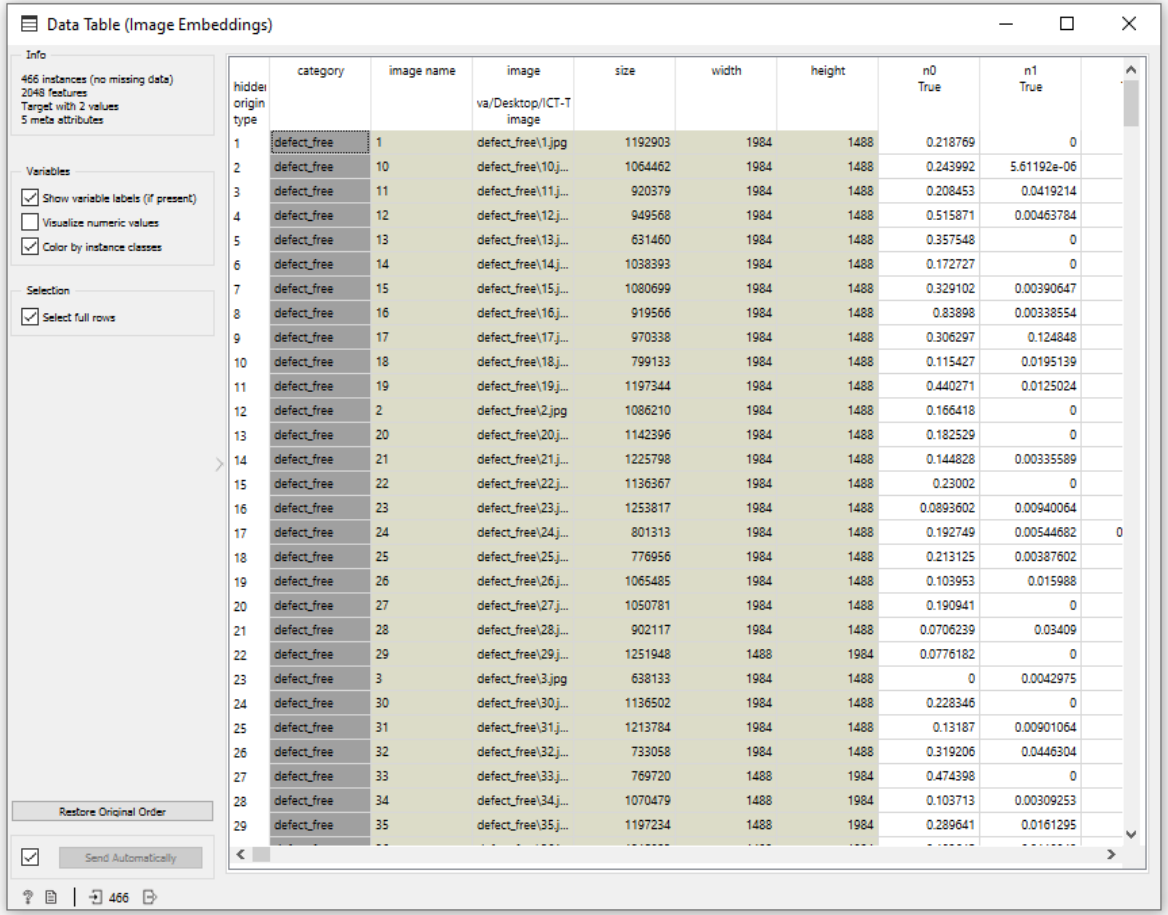
- Select Image Grid widget to view Image Embeddings





Step 6



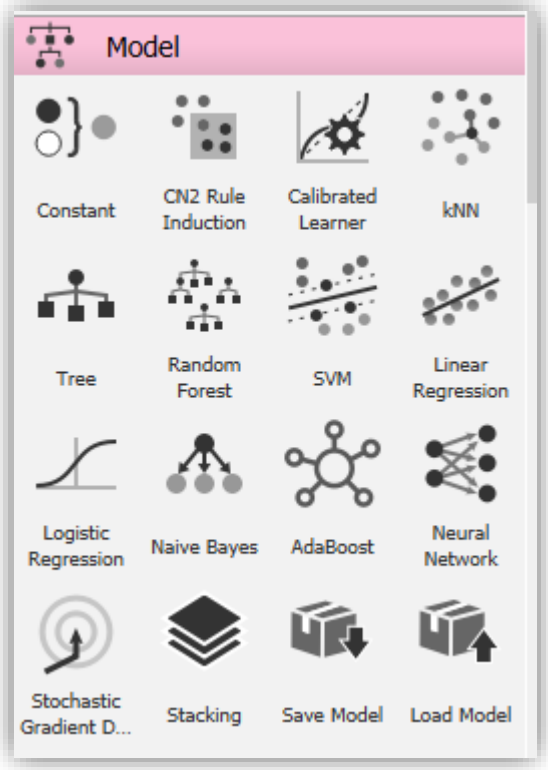


hidden origin type	category	image name	image va/Desktop/ICT-T image	size	width	height	n0 True	n1 True
1	defect_free	1	defect_free\1.jpg	1192903	1984	1488	0.218769	0
2	defect_free	10	defect_free\10.j...	1064462	1984	1488	0.243992	5.61192e-06
3	defect_free	11	defect_free\11.j...	920379	1984	1488	0.208453	0.0419214
4	defect_free	12	defect_free\12.j...	949568	1984	1488	0.515871	0.00463784
5	defect_free	13	defect_free\13.j...	631460	1984	1488	0.357548	0
6	defect_free	14	defect_free\14.j...	1038393	1984	1488	0.172727	0
7	defect_free	15	defect_free\15.j...	1080699	1984	1488	0.329102	0.00390647
8	defect_free	16	defect_free\16.j...	919566	1984	1488	0.83898	0.00338554
9	defect_free	17	defect_free\17.j...	970338	1984	1488	0.306297	0.124848
10	defect_free	18	defect_free\18.j...	799133	1984	1488	0.115427	0.0195139
11	defect_free	19	defect_free\19.j...	1197344	1984	1488	0.440271	0.0125024
12	defect_free	2	defect_free\2.jpg	1086210	1984	1488	0.166418	0
13	defect_free	20	defect_free\20.j...	1142396	1984	1488	0.182529	0
14	defect_free	21	defect_free\21.j...	1225798	1984	1488	0.144828	0.00335589
15	defect_free	22	defect_free\22.j...	1136367	1984	1488	0.23002	0
16	defect_free	23	defect_free\23.j...	1253817	1984	1488	0.0893602	0.00940064
17	defect_free	24	defect_free\24.j...	801313	1984	1488	0.192749	0.00544682
18	defect_free	25	defect_free\25.j...	776956	1984	1488	0.213125	0.00387602
19	defect_free	26	defect_free\26.j...	1065485	1984	1488	0.103953	0.015988
20	defect_free	27	defect_free\27.j...	1050781	1984	1488	0.190941	0
21	defect_free	28	defect_free\28.j...	902117	1984	1488	0.0706239	0.03409
22	defect_free	29	defect_free\29.j...	1251948	1488	1984	0.0776182	0
23	defect_free	3	defect_free\3.jpg	638133	1984	1488	0	0.0042975
24	defect_free	30	defect_free\30.j...	1136502	1984	1488	0.228346	0
25	defect_free	31	defect_free\31.j...	1213784	1984	1488	0.13187	0.00901064
26	defect_free	32	defect_free\32.j...	733058	1984	1488	0.319206	0.0446304
27	defect_free	33	defect_free\33.j...	769720	1488	1984	0.474398	0
28	defect_free	34	defect_free\34.j...	1070479	1488	1984	0.103713	0.00309253
29	defect_free	35	defect_free\35.j...	1197234	1488	1984	0.289641	0.0161295

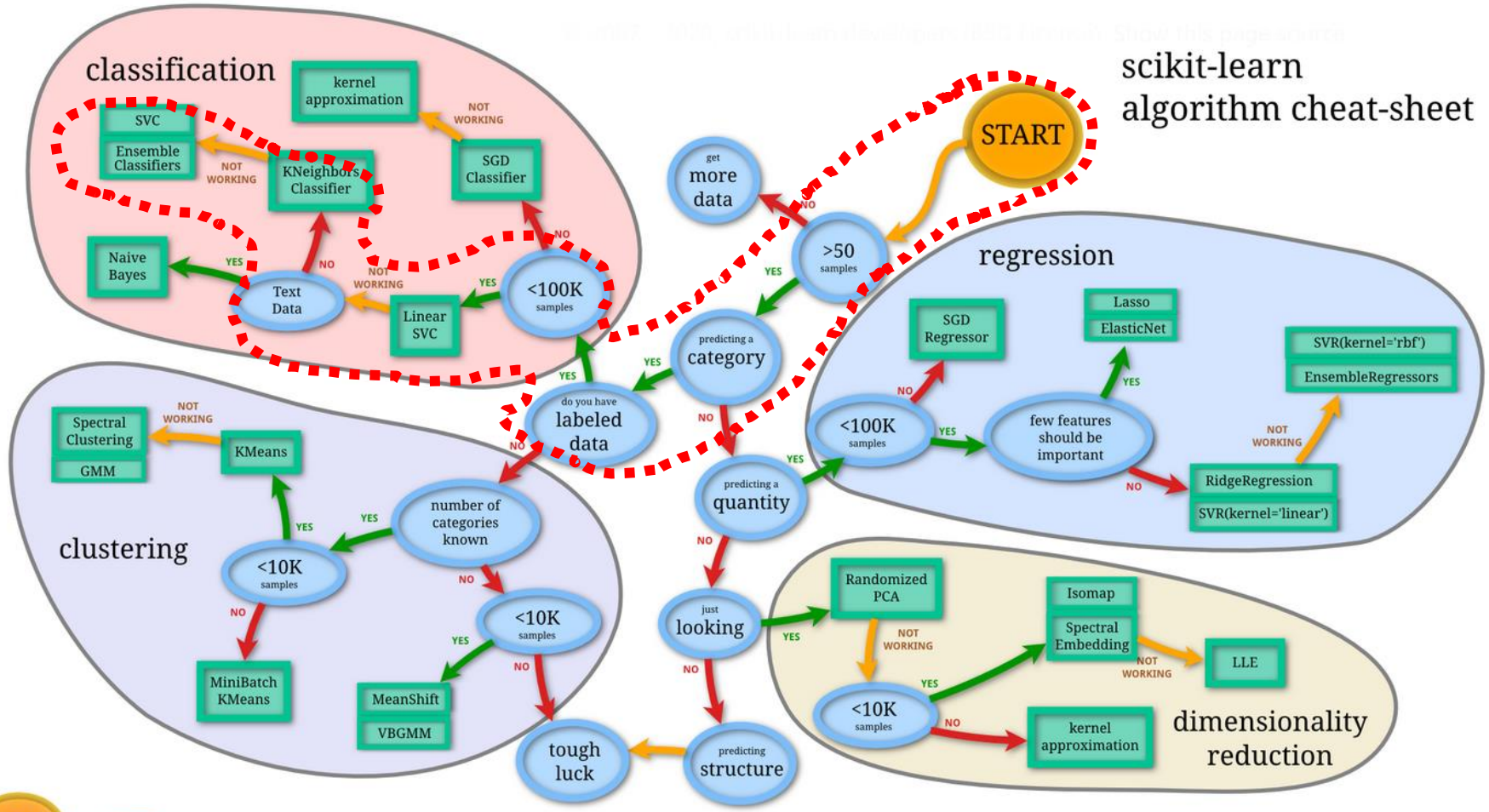
- Including new Data Table View show the new information that was added to the data in a result of image transformation to vectors generated after features selection in Image Embeddings

Step 7

- The dataset of images for fabric stains is already prepared for the classification task
- We need to select classification model from the Model panel
- We will compare several ML models:
  - SVM (linear)
  - kNN
  - Random forest
  - (Decision) Tree
  - Naïve Bayes



- According to the recommendation for ML method selection our hypothesis is that the best performance will be achieved by SVM method (see the decision highlighted in red), because:
  - We have more than 50 examples in the dataset (we have 466 images)
  - The problem that we try to solve is to identify the category of the image from the optical scanner of fabric
  - The training dataset contains labeled data with two categories
  - The dataset contains 466 <100K examples



Back

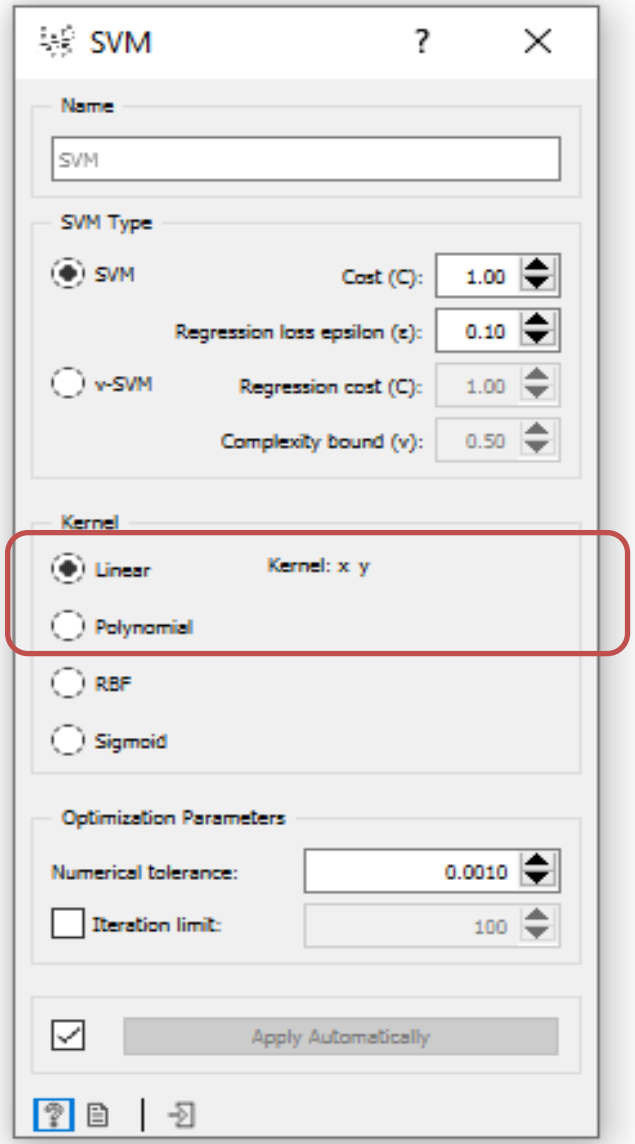
Copyright image – source: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

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

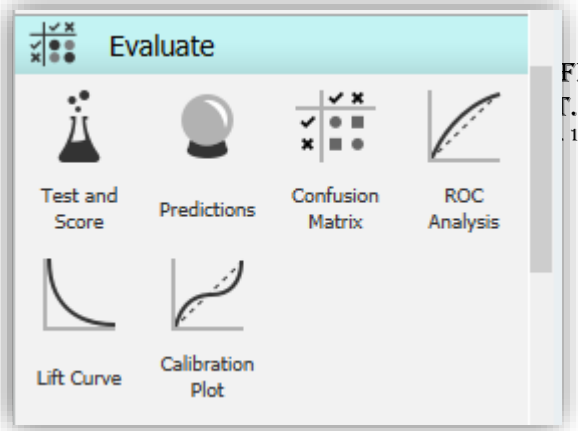
Step 8

- The first ML model that is selected is SVM - Support Vector Machine – Linear kernel

SVM



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



FIA UNIVERSITY
T. KLIMENT OHRIDSKI
1888
ICT-TEX

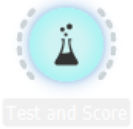
Step 9

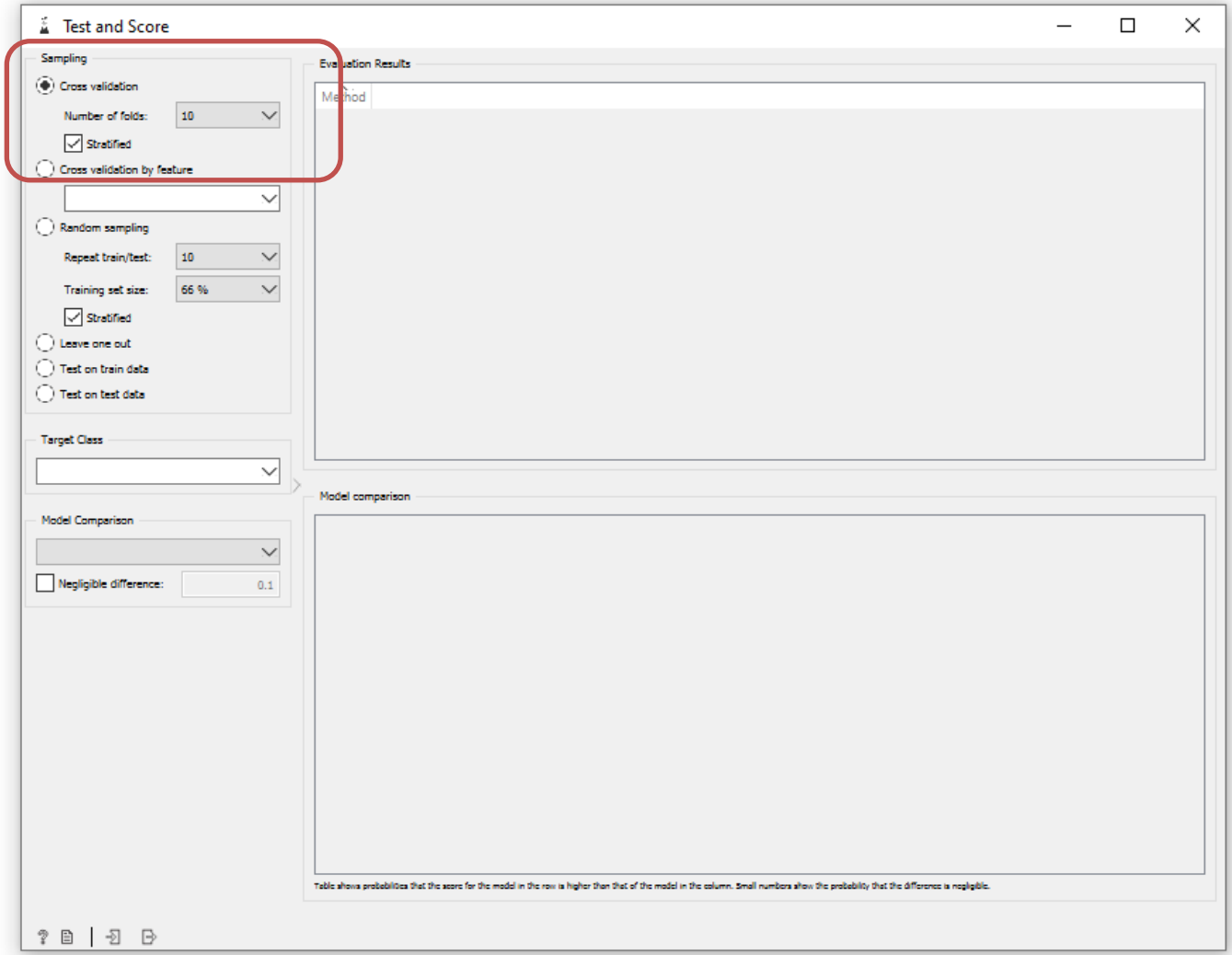
- For evaluation is used Test and Score widget
- We need to select the training and test set.
- There are the following options:
  - Using the current dataset as training set for the ML model and additional dataset to be used as test set. In our case this is not the option, because we do not have additional data
  - Using the same dataset as training set for the ML model and to test with them. This is not considered as a good approach, because the trained model is bias of the data and there are not used new “unseen” data for testing.

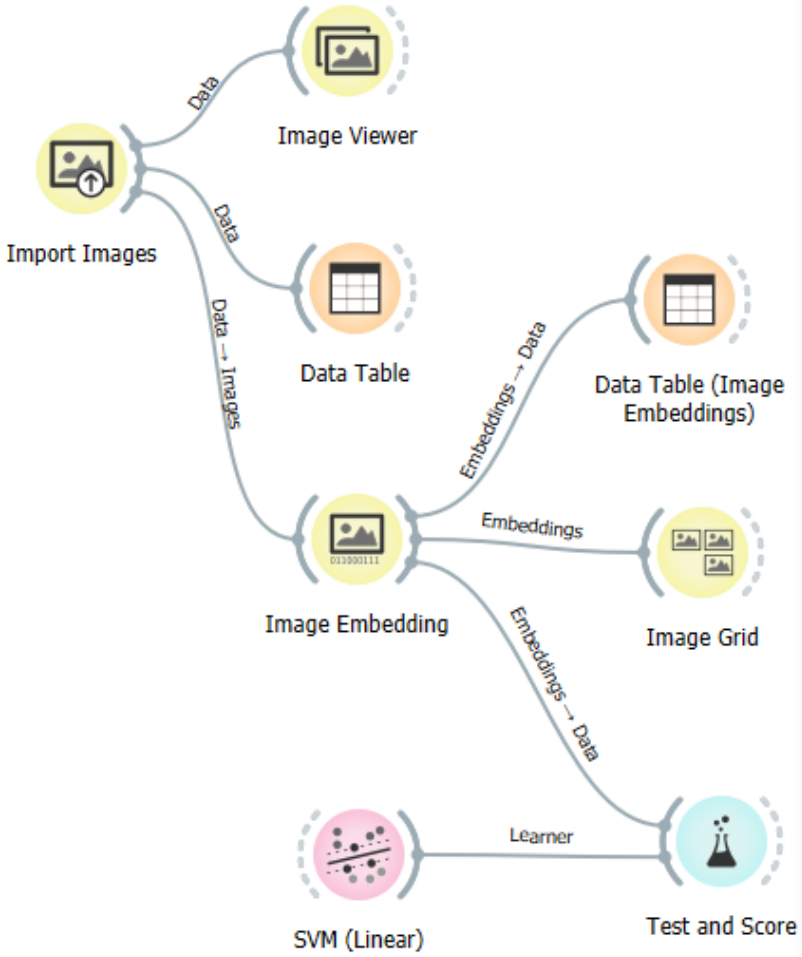
- There are options to split the current dataset on two subsets and to use one part of them as training data and the remaining part as test data, called data sampling:
  - One approach is to use random sampling by specifying the ratio between train:test examples. The most common ratio is 2:1, i.e 66% of the data are used as training and the remaining part as test. In this approach some. Although using different train and test set, the random sample has a huge impact on the result obtained, which may differ significantly from the actual classifications.
  - One of the best methods is to use K-fold cross validation. Selecting in advance some integer K .The dataset is splatted in K parts. The training/testing of the ML model is repeated K times, and for each iteration one of the K subsets is chosen as test, and the remaining K-1 subsets are used as training. Common values for K are 3, 5, 7, 10. For smaller datasets is better to choose higher values of K, because only small part of data will not be used for training and there are lower chances to miss significant information in the ML model training.

3-fold cross validation

iteration 1	1 test	2 train	3 train
iteration 2	1 train	2 test	3 train
iteration 3	1 train	2 train	3 test


Test and Score


Test and Score
Sampling
Cross validation
Number of folds: 10
Stratified
Cross validation by feature
Random sampling
Repeat train/test: 10
Training set size: 66 %
Stratified
Leave one out
Test on train data
Test on test data
Target Class
Model Comparison
Negligible difference: 0.1
Evaluation Results
Method
Model comparison



Test and Score

Sampling
- Cross validation
  - Number of folds: 10
  - Stratified
- Cross validation by feature
- Random sampling
  - Repeat train/test: 10
  - Training set size: 66 %
  - Stratified
- Leave one out
- Test on train data
- Test on test data

Target Class
(Average over classes)

Model Comparison
Area under ROC curve
Negligible difference: 0.1

Evaluation Results

Model	AUC	CA	F1	Precision	Recall
SVM	0.897	0.867	0.865	0.864	0.867

Model Comparison by AUC

	SVM
SVM	

Table shows probabilities that the score for the model in the row is higher than that of the model in the column. Small numbers show the probability that the difference is negligible.

466 466

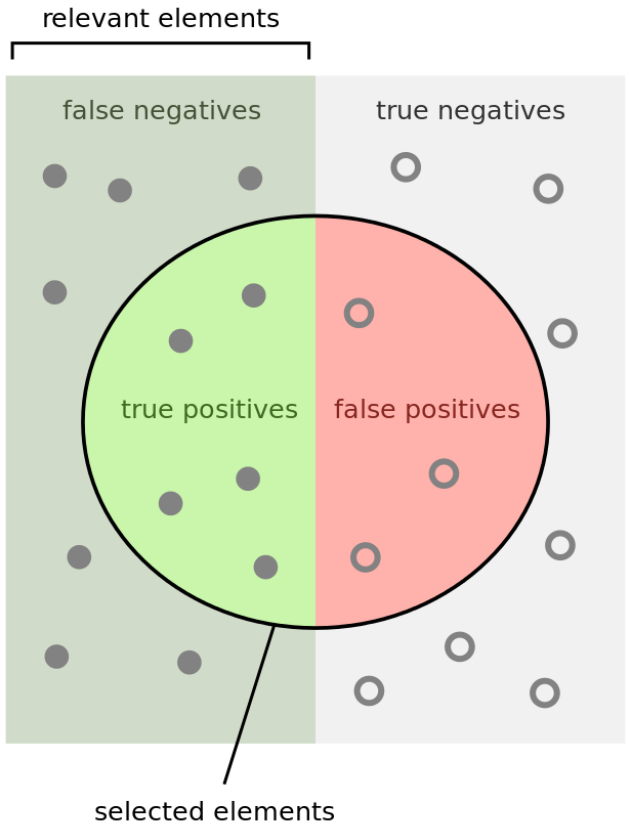
Co-funded by the
Erasmus+ Programme
of the European Union

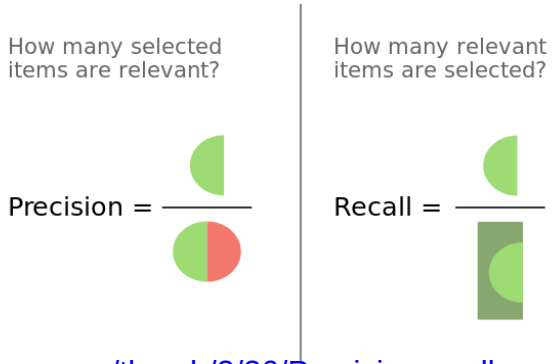
Evaluation Metrics

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{Precission \cdot Recall}{Precission + Recall}$$





true positive (TP)

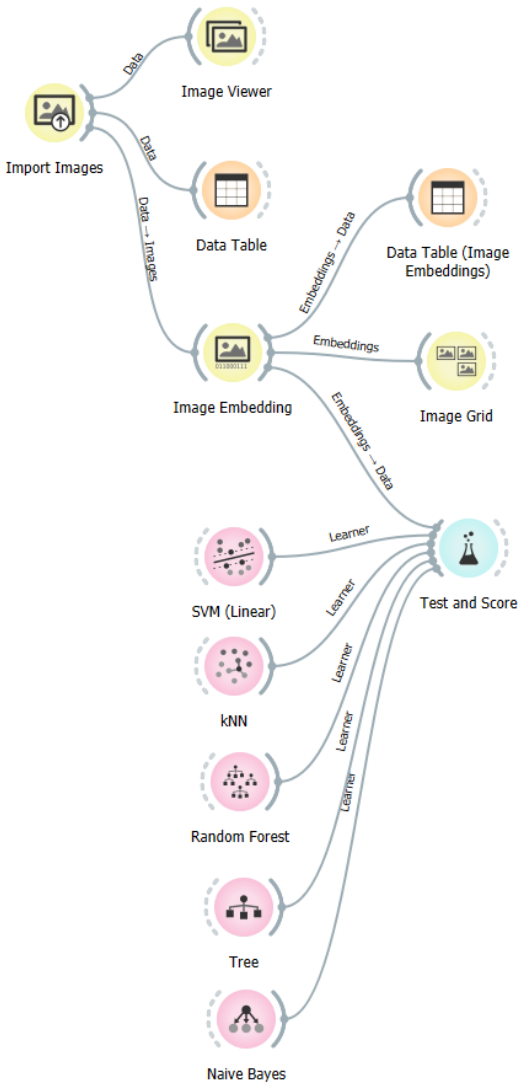
true negative (TN)

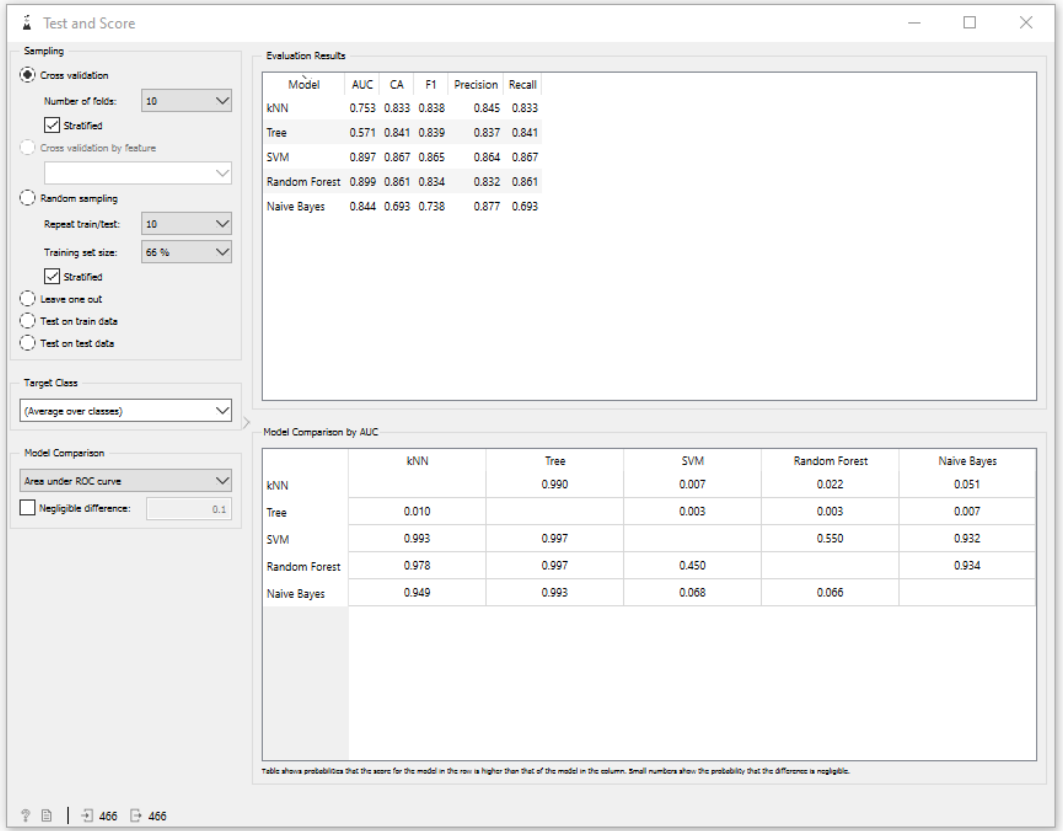
false positive (FP)

false negative (FN)

Step 10

For comparison with other ML models, we add kNN, Random forest, (Decision) Tree and Naïve Bayes widgets from Model panel, using their default parameters


Import Images
Image Viewer
Data Table
Data Table (Image Embeddings)
Image Embedding
Image Grid
Test and Score
SVM (Linear)
kNN
Random Forest
Tree
Naive Bayes
Data
Data
Data → Images
Embeddings → Data
Embeddings
Embeddings → Data
Learner
Learner
Learner
Learner
Learner


Test and Score
Sampling
Cross validation
Number of folds: 10
Stratified
Cross validation by feature
Random sampling
Repeat train/test: 10
Training set size: 66 %
Stratified
Leave one out
Test on train data
Test on test data
Target Class
(Average over classes)
Model Comparison
Area under ROC curve
Negligible difference: 0.1
Evaluation Results
Model AUC CA F1 Precision Recall
kNN 0.753 0.833 0.838 0.845 0.833
Tree 0.571 0.841 0.839 0.837 0.841
SVM 0.897 0.867 0.865 0.864 0.867
Random Forest 0.899 0.861 0.834 0.832 0.861
Naive Bayes 0.844 0.693 0.738 0.877 0.693
Model Comparison by AUC
kNN Tree SVM Random Forest Naive Bayes
kNN 0.990 0.007 0.022 0.051
Tree 0.010 0.003 0.003 0.007
SVM 0.993 0.997 0.550 0.932
Random Forest 0.978 0.997 0.450 0.934
Naive Bayes 0.949 0.993 0.068 0.066
466 466

Discussion

- The results of ML classification models training show that the best performance have:
  - SVM (Linear) with highest F1 score 0.865
  - Naïve Bayes has highest Precision 0.877, but lacks Recall.
  - SVM (Linear) has highest Recall 0.867
- The huge variety of stains features, like shapes, size, number of stains, does not allow to achieve better performance of ML models due to relatively small training dataset, that does not allow to be investigated the majority of possible features combinations.
- 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/

CONTACTS

Coordinator:
Technical University of Sofia

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

Web-site: ICT-TEX.eu

Author:
Assoc. Professor Svetla Boytcheva
Institute of Information and Communication Technologies,
Bulgarian Academy of Sciences

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

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