CERN

INSTITUT DES SCIENCES DE L'HOMME ish

Ciapple

dkm²

Erasmus Mundus
master course in
**Data Mining and**
**Knowledge Management**
a european master

UPO
UNIVERSITÀ DEL PIEMONTE ORIENTALE

UNIVERSITÉ
LUMIÈRE
LYON 2
UNIVERSITÉ DE LYON

# MACHINE LEARNING IN PRACTICE

Hussein AL-NATSHEH, CIAPPLE

Jordan Open Source Association

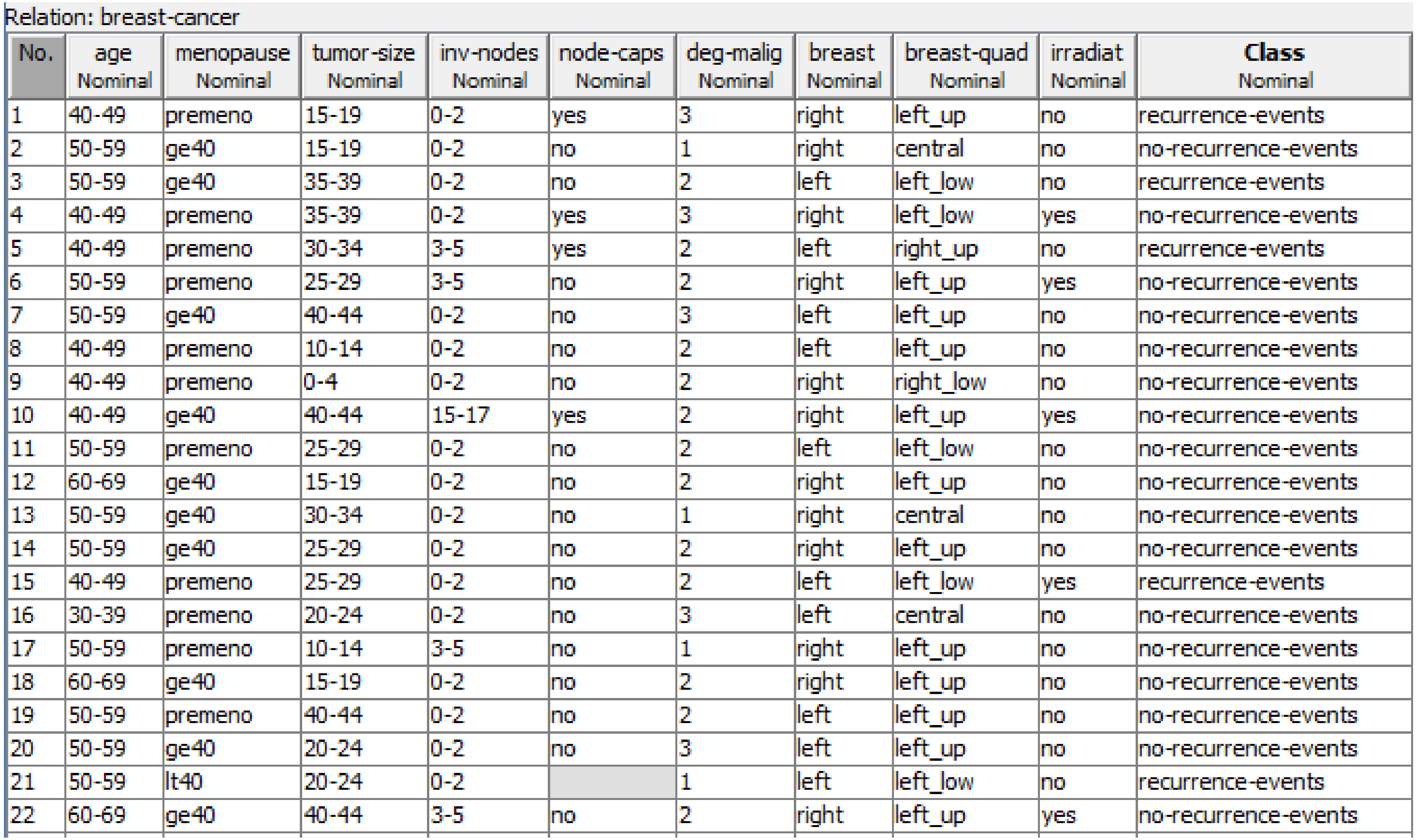
Amman, Aug 22, 2015

# AGENDA

- From supervised to unsupervised learning
- Identifying a ML problem?
- Recommended python libraries
- Data preparation
- Case study; Author name disambiguation of CERN digital library (use case)
- References, recommended readings and courses

# FROM SUPERVISED TO UNSUPERVISED LEARNING

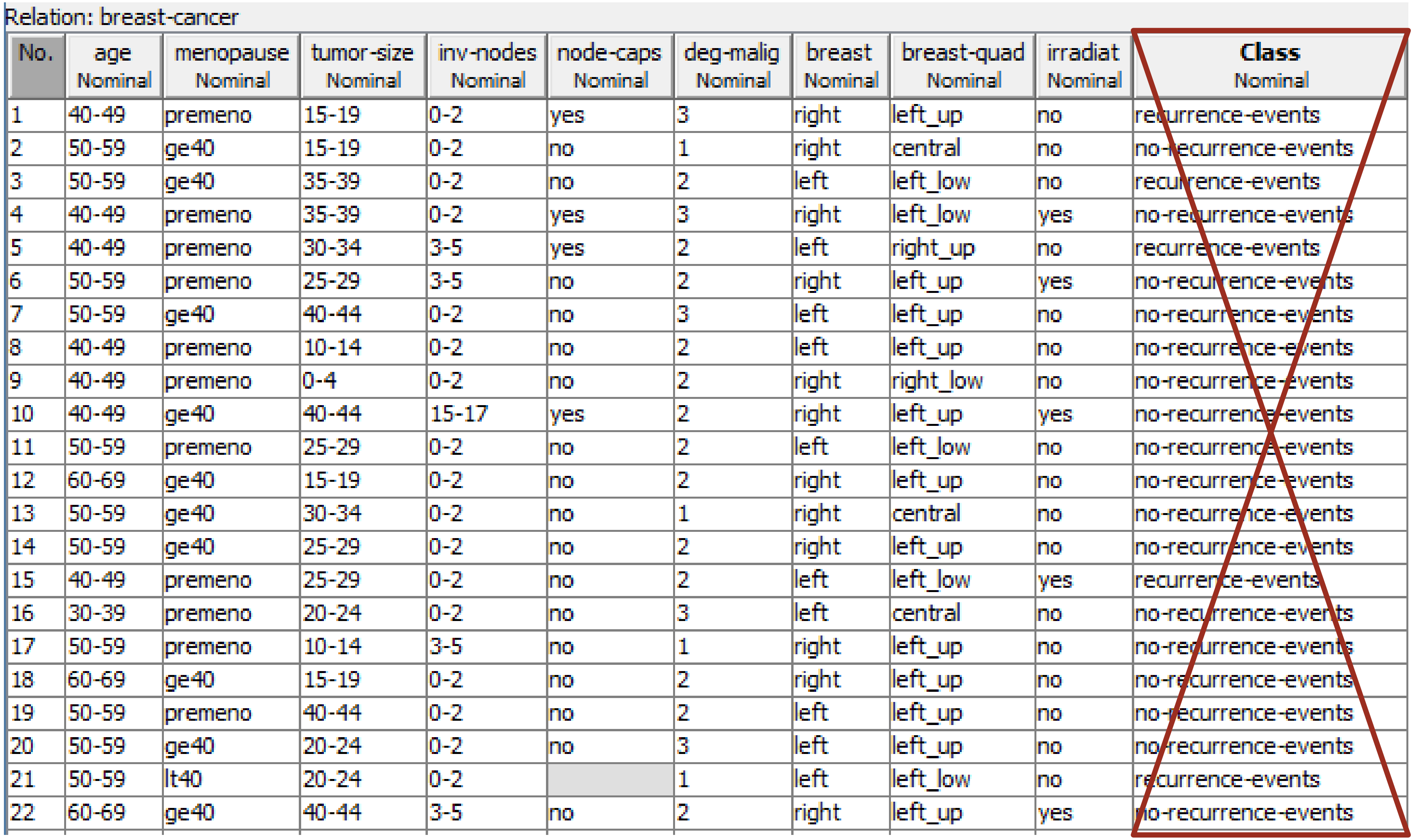
# RECALL SUPERVISED LEARNING



Relation: breast-cancer

| No. | age<br>Nominal | menopause<br>Nominal | tumor-size<br>Nominal | inv-nodes<br>Nominal | node-caps<br>Nominal | deg-malig<br>Nominal | breast<br>Nominal | breast-quad<br>Nominal | irradiat<br>Nominal | **Class**<br>Nominal |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 40-49 | premeno | 15-19 | 0-2 | yes | 3 | right | left_up | no | recurrence-events |
| 2 | 50-59 | ge40 | 15-19 | 0-2 | no | 1 | right | central | no | no-recurrence-events |
| 3 | 50-59 | ge40 | 35-39 | 0-2 | no | 2 | left | left_low | no | recurrence-events |
| 4 | 40-49 | premeno | 35-39 | 0-2 | yes | 3 | right | left_low | yes | no-recurrence-events |
| 5 | 40-49 | premeno | 30-34 | 3-5 | yes | 2 | left | right_up | no | recurrence-events |
| 6 | 50-59 | premeno | 25-29 | 3-5 | no | 2 | right | left_up | yes | no-recurrence-events |
| 7 | 50-59 | ge40 | 40-44 | 0-2 | no | 3 | left | left_up | no | no-recurrence-events |
| 8 | 40-49 | premeno | 10-14 | 0-2 | no | 2 | left | left_up | no | no-recurrence-events |
| 9 | 40-49 | premeno | 0-4 | 0-2 | no | 2 | right | right_low | no | no-recurrence-events |
| 10 | 40-49 | ge40 | 40-44 | 15-17 | yes | 2 | right | left_up | yes | no-recurrence-events |
| 11 | 50-59 | premeno | 25-29 | 0-2 | no | 2 | left | left_low | no | no-recurrence-events |
| 12 | 60-69 | ge40 | 15-19 | 0-2 | no | 2 | right | left_up | no | no-recurrence-events |
| 13 | 50-59 | ge40 | 30-34 | 0-2 | no | 1 | right | central | no | no-recurrence-events |
| 14 | 50-59 | ge40 | 25-29 | 0-2 | no | 2 | right | left_up | no | no-recurrence-events |
| 15 | 40-49 | premeno | 25-29 | 0-2 | no | 2 | left | left_low | yes | recurrence-events |
| 16 | 30-39 | premeno | 20-24 | 0-2 | no | 3 | left | central | no | no-recurrence-events |
| 17 | 50-59 | premeno | 10-14 | 3-5 | no | 1 | right | left_up | no | no-recurrence-events |
| 18 | 60-69 | ge40 | 15-19 | 0-2 | no | 2 | right | left_up | no | no-recurrence-events |
| 19 | 50-59 | premeno | 40-44 | 0-2 | no | 2 | left | left_up | no | no-recurrence-events |
| 20 | 50-59 | ge40 | 20-24 | 0-2 | no | 3 | left | left_up | no | no-recurrence-events |
| 21 | 50-59 | lt40 | 20-24 | 0-2 |  | 1 | left | left_low | no | recurrence-events |
| 22 | 60-69 | ge40 | 40-44 | 3-5 | no | 2 | right | left_up | yes | no-recurrence-events |

# UNSUPERVISED LEARNING



Relation: breast-cancer

| No. | age | menopause | tumor-size | inv-nodes | node-caps | deg-malig | breast | breast-quad | irradiat | Class |
|---|---|---|---|---|---|---|---|---|---|---|
| | Nominal | Nominal | Nominal | Nominal | Nominal | Nominal | Nominal | Nominal | Nominal | Nominal |
| 1 | 40-49 | premeno | 15-19 | 0-2 | yes | 3 | right | left_up | no | recurrence-events |
| 2 | 50-59 | ge40 | 15-19 | 0-2 | no | 1 | right | central | no | no-recurrence-events |
| 3 | 50-59 | ge40 | 35-39 | 0-2 | no | 2 | left | left_low | no | recurrence-events |
| 4 | 40-49 | premeno | 35-39 | 0-2 | yes | 3 | right | left_low | yes | no-recurrence-events |
| 5 | 40-49 | premeno | 30-34 | 3-5 | yes | 2 | left | right_up | no | recurrence-events |
| 6 | 50-59 | premeno | 25-29 | 3-5 | no | 2 | right | left_up | yes | no-recurrence-events |
| 7 | 50-59 | ge40 | 40-44 | 0-2 | no | 3 | left | left_up | no | no-recurrence-events |
| 8 | 40-49 | premeno | 10-14 | 0-2 | no | 2 | left | left_up | no | no-recurrence-events |
| 9 | 40-49 | premeno | 0-4 | 0-2 | no | 2 | right | right_low | no | no-recurrence-events |
| 10 | 40-49 | ge40 | 40-44 | 15-17 | yes | 2 | right | left_up | yes | no-recurrence-events |
| 11 | 50-59 | premeno | 25-29 | 0-2 | no | 2 | left | left_low | no | no-recurrence-events |
| 12 | 60-69 | ge40 | 15-19 | 0-2 | no | 2 | right | left_up | no | no-recurrence-events |
| 13 | 50-59 | ge40 | 30-34 | 0-2 | no | 1 | right | central | no | no-recurrence-events |
| 14 | 50-59 | ge40 | 25-29 | 0-2 | no | 2 | right | left_up | no | no-recurrence-events |
| 15 | 40-49 | premeno | 25-29 | 0-2 | no | 2 | left | left_low | yes | recurrence-events |
| 16 | 30-39 | premeno | 20-24 | 0-2 | no | 3 | left | central | no | no-recurrence-events |
| 17 | 50-59 | premeno | 10-14 | 3-5 | no | 1 | right | left_up | no | no-recurrence-events |
| 18 | 60-69 | ge40 | 15-19 | 0-2 | no | 2 | right | left_up | no | no-recurrence-events |
| 19 | 50-59 | premeno | 40-44 | 0-2 | no | 2 | left | left_up | no | no-recurrence-events |
| 20 | 50-59 | ge40 | 20-24 | 0-2 | no | 3 | left | left_up | no | no-recurrence-events |
| 21 | 50-59 | lt40 | 20-24 | 0-2 | | 1 | left | left_low | no | recurrence-events |
| 22 | 60-69 | ge40 | 40-44 | 3-5 | no | 2 | right | left_up | yes | no-recurrence-events |

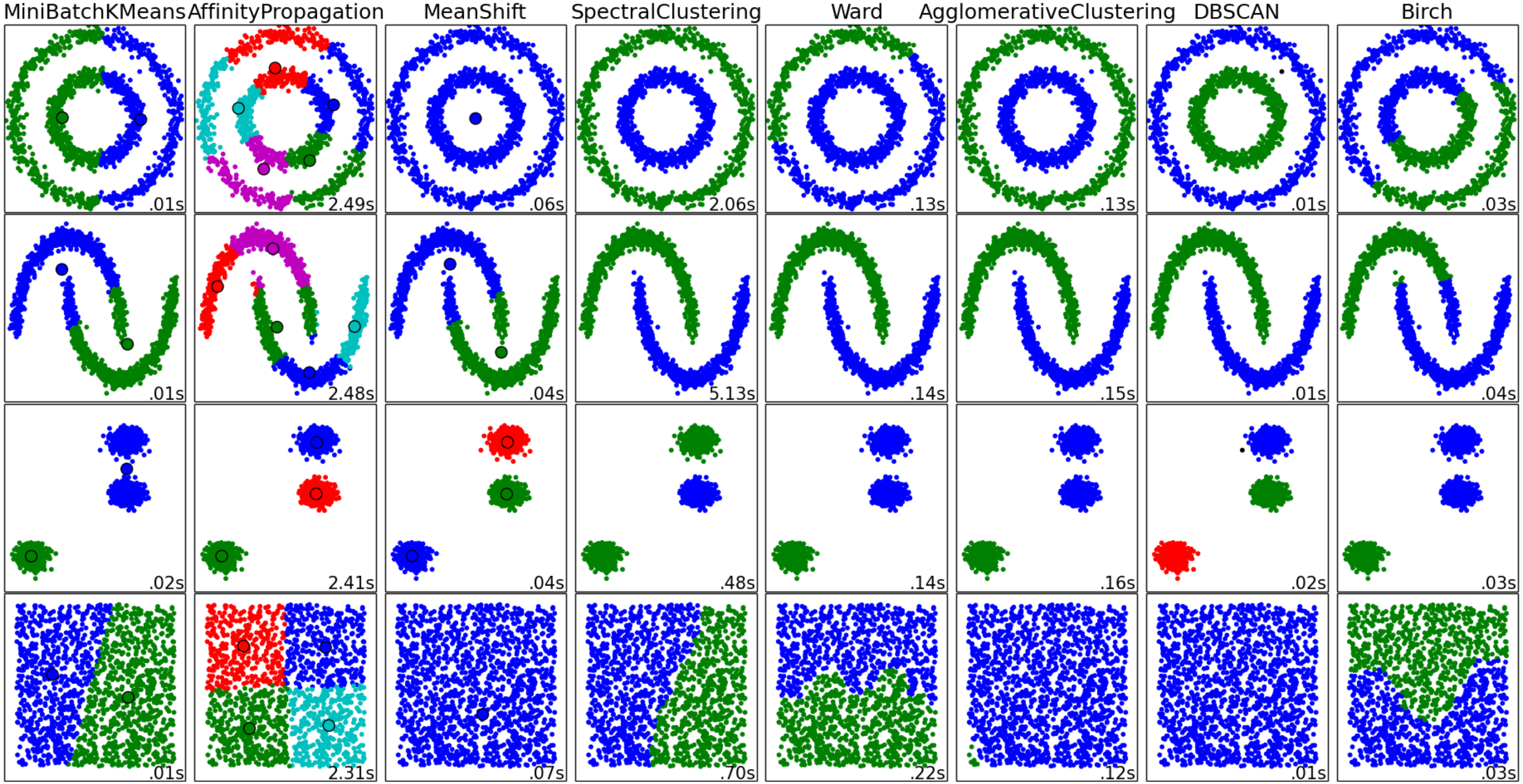
# CLUSTERING

- A flat partition (set of clusters or segments)
- A hierarchical tree or taxonomy (a set of nested partitions)
- Hard or soft (or fuzzy) memberships to clusters

# SOME CLUSTERING ALGORITHMS (1OF2)

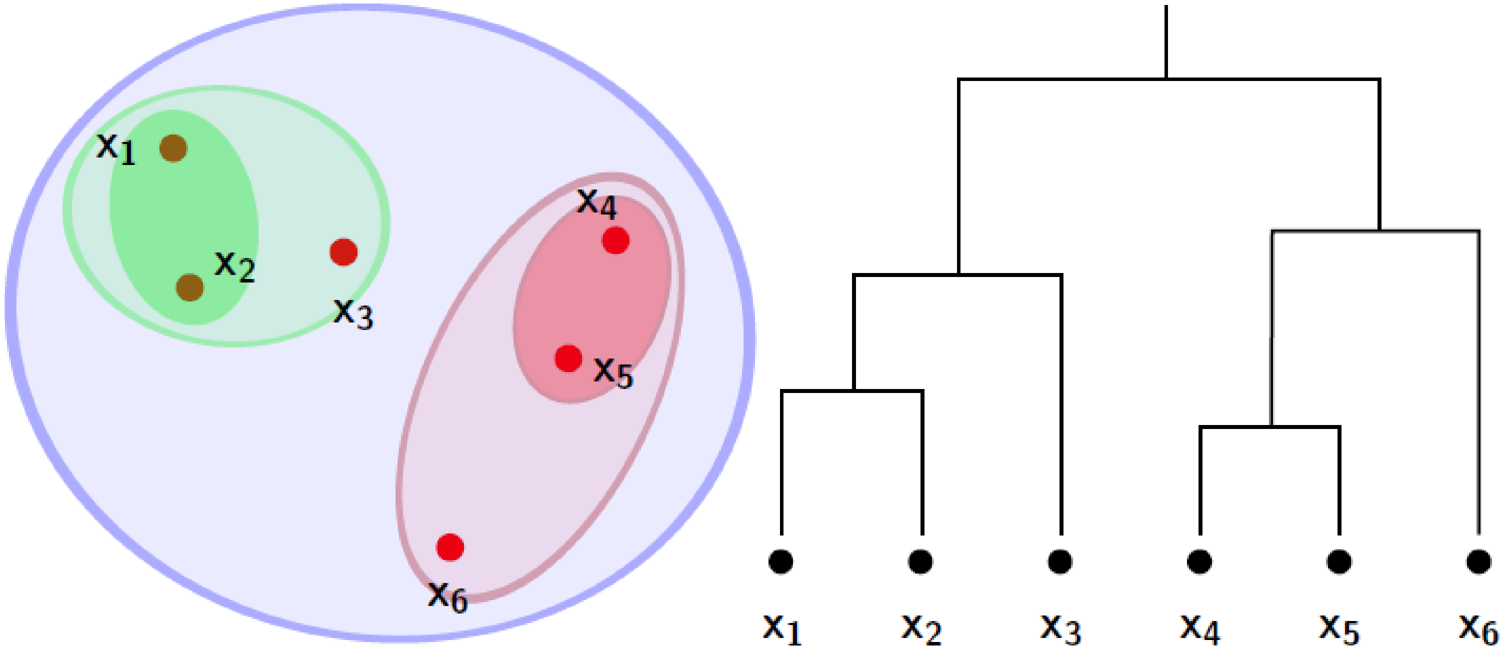
- K-Means, Expectation Maximization
- Hierarchal Agglomerative Clustering (HAC)
- Density-based spatial clustering of applications with noise (DBSCAN)
- Graph-based clustering (Modularity, Weighted communities, Clique percolation)
- Unsupervised Neural Networks (Adaptive Resonance Theory)

# SOME CLUSTERING ALGORITHMS (2OF2)



Source: http://scikit-learn.org/stable/modules/clustering.html#clustering

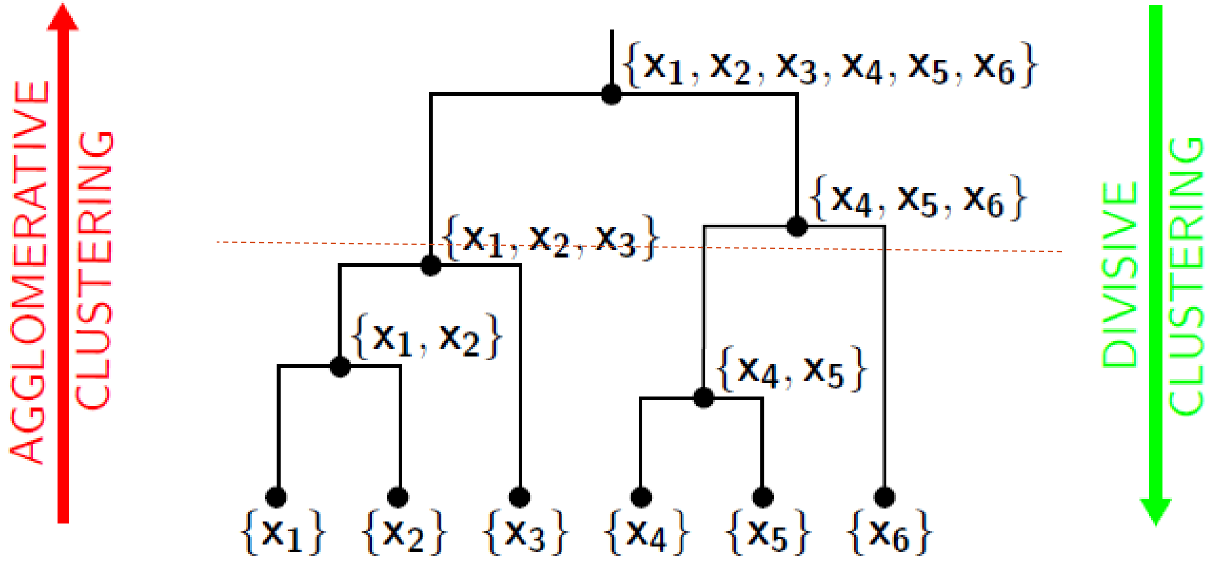
# HIERARCHAL AGGLOMERATIVE CLUSTERING (HAC) AND DENDROGRAM



Source:Data Clustering- Part 2, M2 DMKM Slides, Julien Ah-Pine, Universit´e Lyon 2, 2014

# CLUSTERING METHODS

- Graph methods
  - Single link
  - Complete link
  - Average
  - Weighted
- Geometric methods
  - Ward
  - Centroid
  - Median



Source:Data Clustering- Part 2, M2 DMKM Slides, Julien Ah-Pine, Universit´e Lyon 2, 2014

# DISSIMILARITY

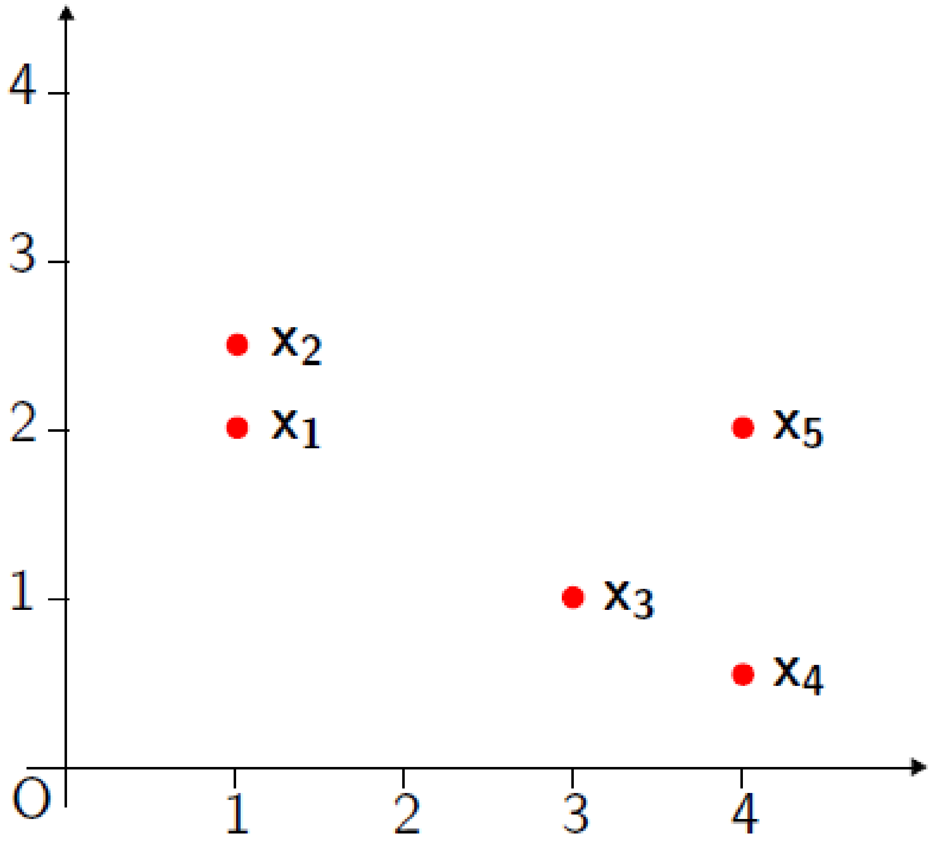
- Some distance functions
  - Euclidean distance
  - Jaccard Similarity
  - Cosine Similarity
  - Edit distance
- Distance estimator
  - Model an estimator to be used with a predict probability function
  - Better for merging feature importances
- Dissimilarity (affinity) matrix

# HIERARCHAL CLUSTERING EXAMPLE (1OF4)

We consider 5 data points in $\mathbb{R}^2$ :

- $x_1 = (1, 2)$
- $x_2 = (1, 2.5)$
- $x_3 = (3, 1)$
- $x_4 = (4, 0.5)$
- $x_5 = (4, 2)$

We consider the euclidean distance between data points.



Source:Data Clustering- Part 2, M2 DMKM Slides, Julien Ah-Pine, Universit´e Lyon 2, 2014

# HIERARCHAL CLUSTERING EXAMPLE (2OF4)

- Single Link method

$$D_{sl}(C_k, C_{k'}) = \min_{\mathbf{x}\in C_k, \mathbf{y}\in C_{k'}} \{D(\mathbf{x}, \mathbf{y})\}$$

```
import scipy.cluster.hierarchy as hac
def __init__(self, method="single", affinity="euclidean“)
self.linkage_ = hac.linkage(X,
                             method=self.method,
                             metric=self.affinity)
```

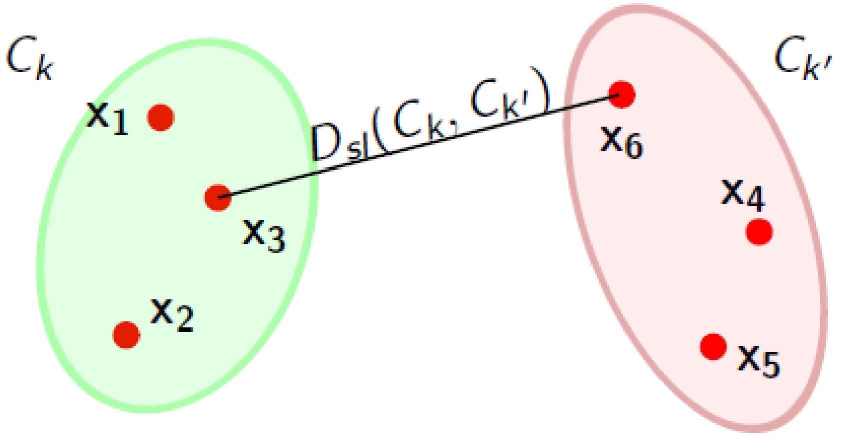


Image source: Data Clustering- Part 2, M2 DMKM Slides, Julien Ah-Pine, Universit´e Lyon 2, 2014

# HIERARCHAL CLUSTERING EXAMPLE (3OF4)

$$\mathbf{D} = \mathbf{D}_{\text{eucl}} = \mathbf{D}_{\text{sl}} = \begin{array}{c} \\ \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \end{array} \begin{array}{c} \begin{array}{ccccc} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 \end{array} \\ \begin{pmatrix} 0 & 0.5 & 2.24 & 3.35 & 3 \\ 0.5 & 0 & 2.5 & 3.61 & 3.04 \\ 2.24 & 2.5 & 0 & 1.12 & 1.41 \\ 3.35 & 3.61 & 1.12 & 0 & 1.5 \\ 3 & 3.04 & 1.41 & 1.5 & 0 \end{pmatrix} \end{array}$$

$$dend(h) = \left\{ \; \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} \quad \text{if } 0 \leq h \right.$$

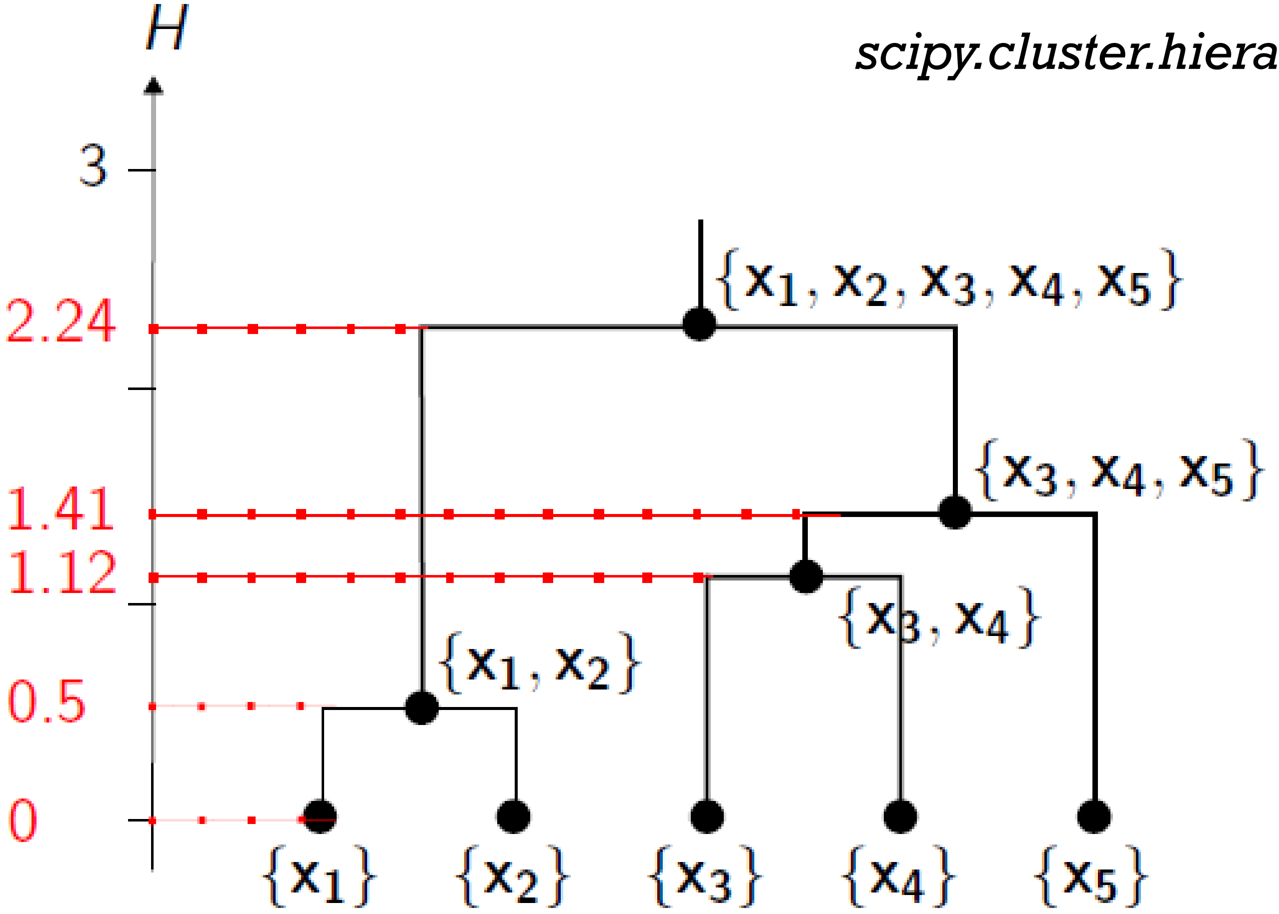
Merge $\mathbf{x}_1$ and $\mathbf{x}_2$

$$dend(h) = \begin{cases} \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0 \leq h < 0.5 \\ \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\} & \text{if } 0.5 \leq h \end{cases}$$

Source:Data Clustering- Part 2, M2 DMKM Slides, Julien Ah-Pine, Universit´e Lyon 2, 2014

# HIERARCHAL CLUSTERING EXAMPLE (4OF4)

*scipy.cluster.hierarchy.dendrogram()*



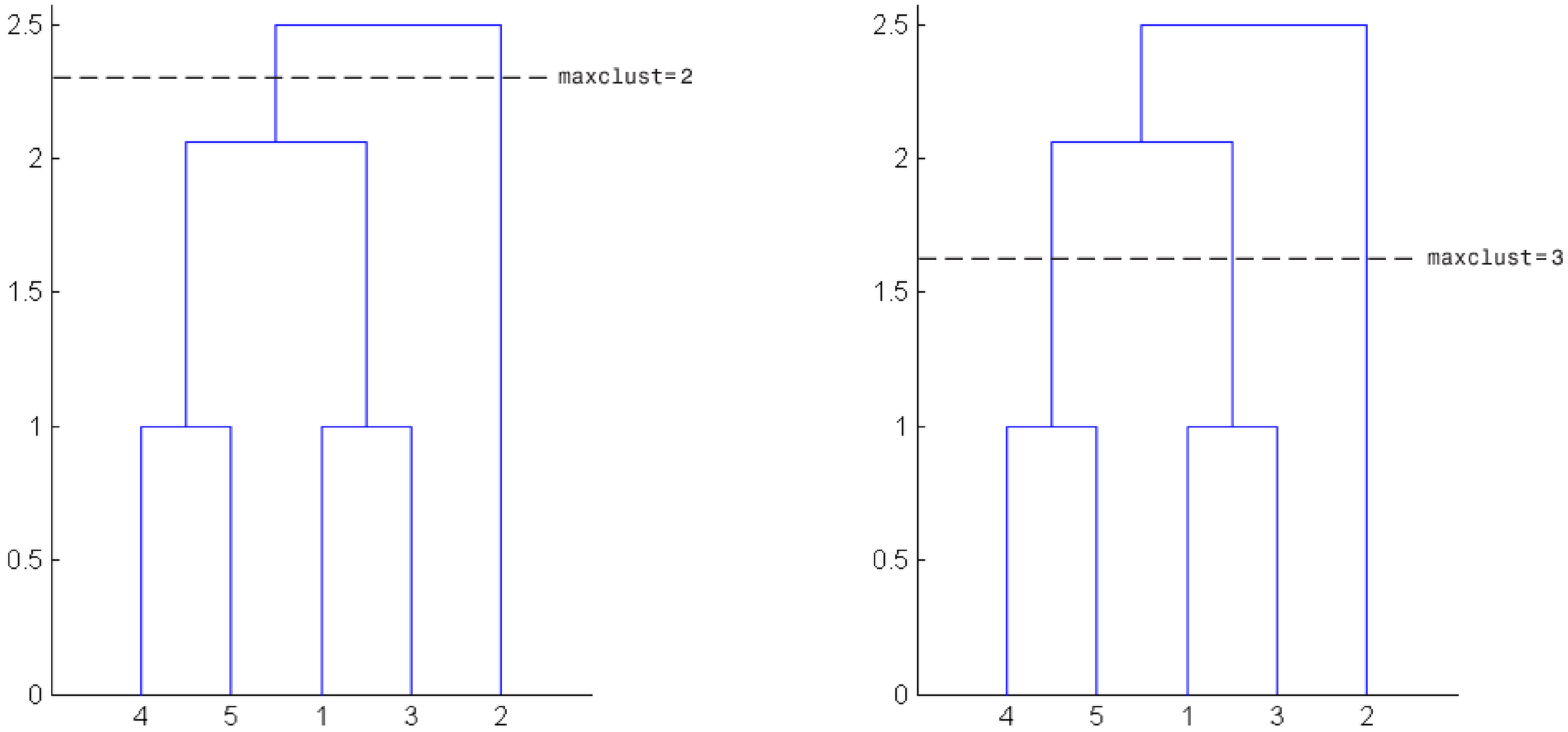
Source:Data Clustering- Part 2, M2 DMKM Slides, Julien Ah-Pine, Universit´e Lyon 2, 2014

# NUMBER OF CLUSTERS

- Flat cutting forming partitioned clustering from HAC
- Semi-supervised clustering
- DBSCAN

*labels = hac.fcluster(self.linkage_, threshold)*

# NUMBER OF CLUSTERS (SEMI-SUPERVISED)



1 and 5 must be in the same cluster → n_clust = 2

Image Source: Matlab

# IDENTIFYING A MACHINE LEARNING PROBLEM

# TASKS MAPPED TO ML (1 OF 2)

- Classification, recognition and completion
  - Decision trees
  - Supervised Neural Networks
  - Graph models: Hidden Markov Model, Bayesian Nets
- Optimization
  - Genetic Algorithms
  - Numerical Optimization and regression
- Smart Control
  - Fuzzy rules

# TASKS MAPPED TO ML (2 OF 2)

- Clustering, grouping and segmentation
  - Unsupervised Neural Networks
  - HAC, K-Means, DBSCAN, Graph-based clustering
- Simulation
  - Multi-agent systems
- Relations
  - Graph networks
  - Collaborative filtering
  - Association rules and frequent sets

# SOME ML APPLICATIONS (1 OF 2)

- Text mining: Topic and concept extraction → Unsupervised: Clustering, Graph and ontologies. Semi-supervised
- Pattern recognition: Character, speech and image recognition → Supervised: Classification
- Cross selling and personalized recommendation → Collaborative filtering, Association rules, frequent sets
- Churn management and customer retention → classification and clustering
- Games → search for optimal solution and multi-agent systems

# SOME ML APPLICATIONS (2 OF 2)

- Financials: risk management in insurance fraud detection → Graph, classification, clustering
- Security: Intrusion detection, spam filtering → Supervised: Classification
- Social networks: friends suggestion → Graph
- Bioinformatics (DNA)
- High energy physics (Higgs)
- Computer vision, Robotics …

# UNDERSTANDING THE DATA

- General statistics
- Correlations
- Plotting
- Understanding the source and the business behind the data
- Well-understanding of the requirements and the goal of the project

# DATA PREPARATION

# DATA CHALLENGES

- Feature extraction (What is a good feature?)
- Categories vs. continuous features vs. mixed
- Features Correlation
- Dimensionality reduction (the less the better for generalization)
- Missing values
- Normalization
- Outliers
- Feature selection
- Sampling
- Unbalanced dataset

# RECOMMENDED PYTHON LIBRARIES

# RECOMMENDED PYTHON LIBRARIES

- Scipy
- Numpy (vectorized data operations)
- Matplotlib (plotting)
- Scikit-Learn
- Pandas (data frames)
- NetworkX (Graphs analysis)
- Pylearn2 (deep learning and neural networks)
- BEARD (entity recognition) – *to be presented within the use case*

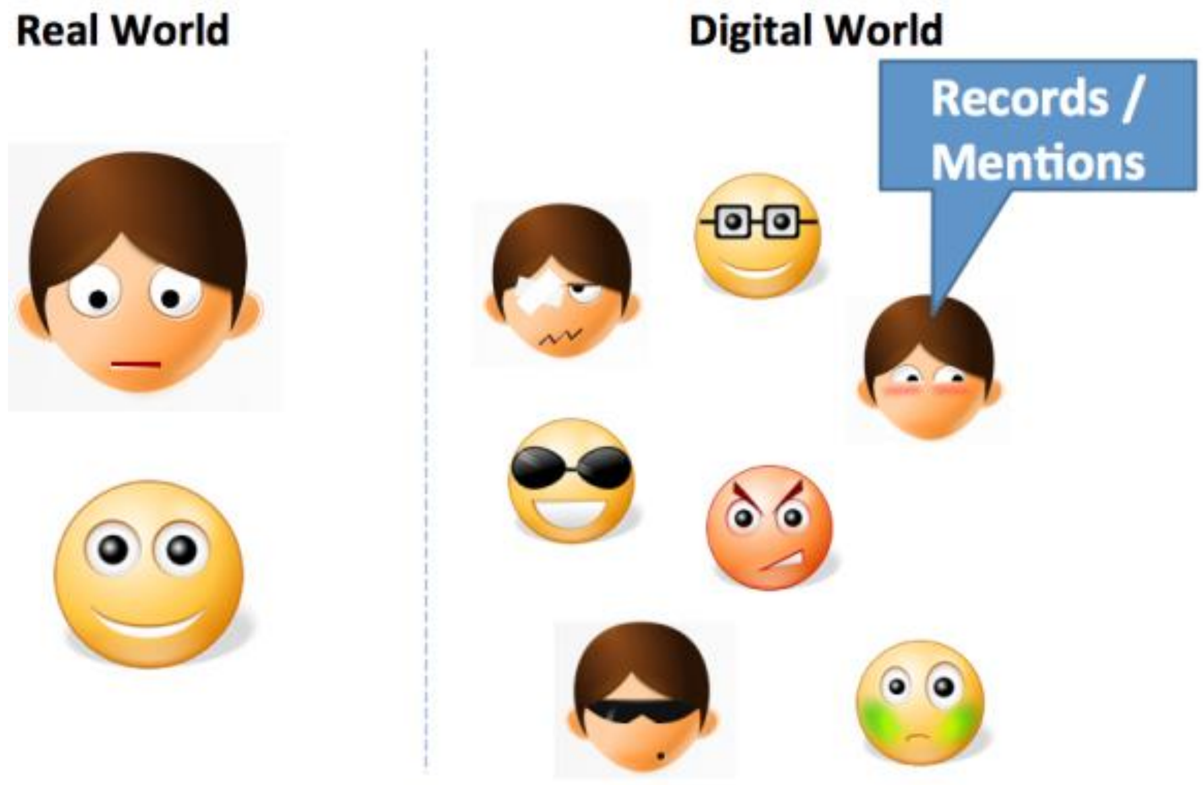
# SCIKIT-LEARN API

- fit
- *predict*
- *fit_transform*
- *Transformers*
- *Pipelines*
- *Features union*
- *Feature importances*
- *@property*
- Joblib and parallel computing

# CASE STUDY: AUTHOR NAME DISAMBIGUATION

CERN Digital Library

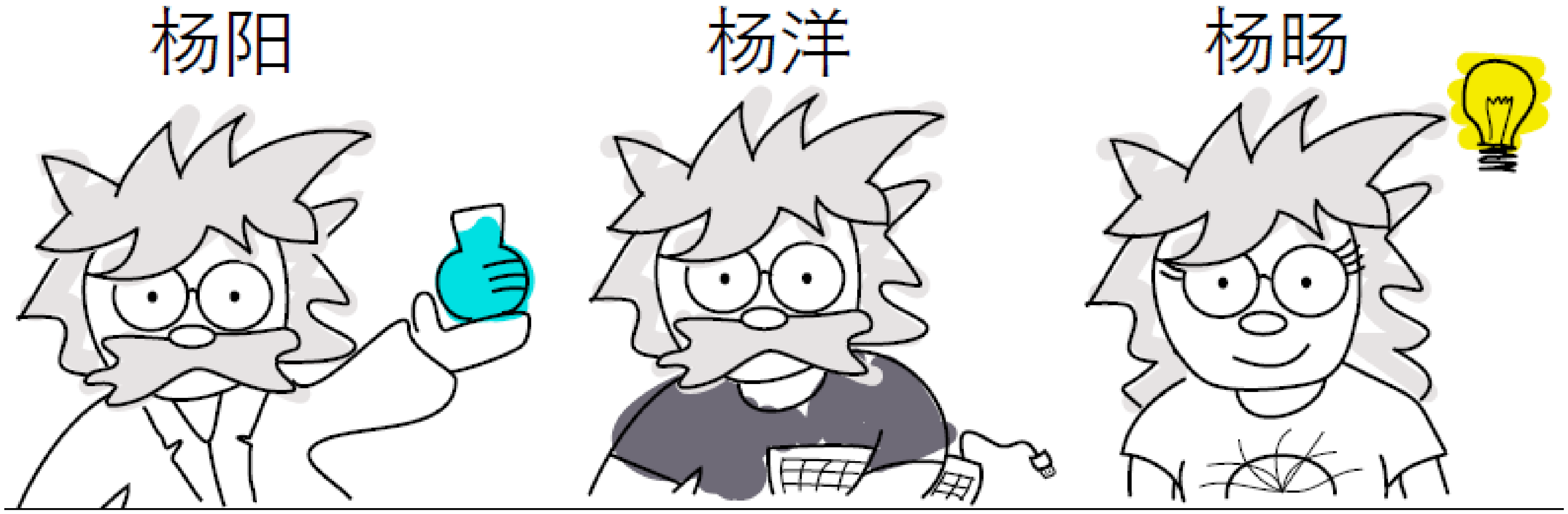
# ENTITY RESOLUTION, DUPLICATION



Source: http://www.datacommunitydc.org/blog/2013/08/entity-resolution-for-big-data

# WHAT IS THE PROBLEM?

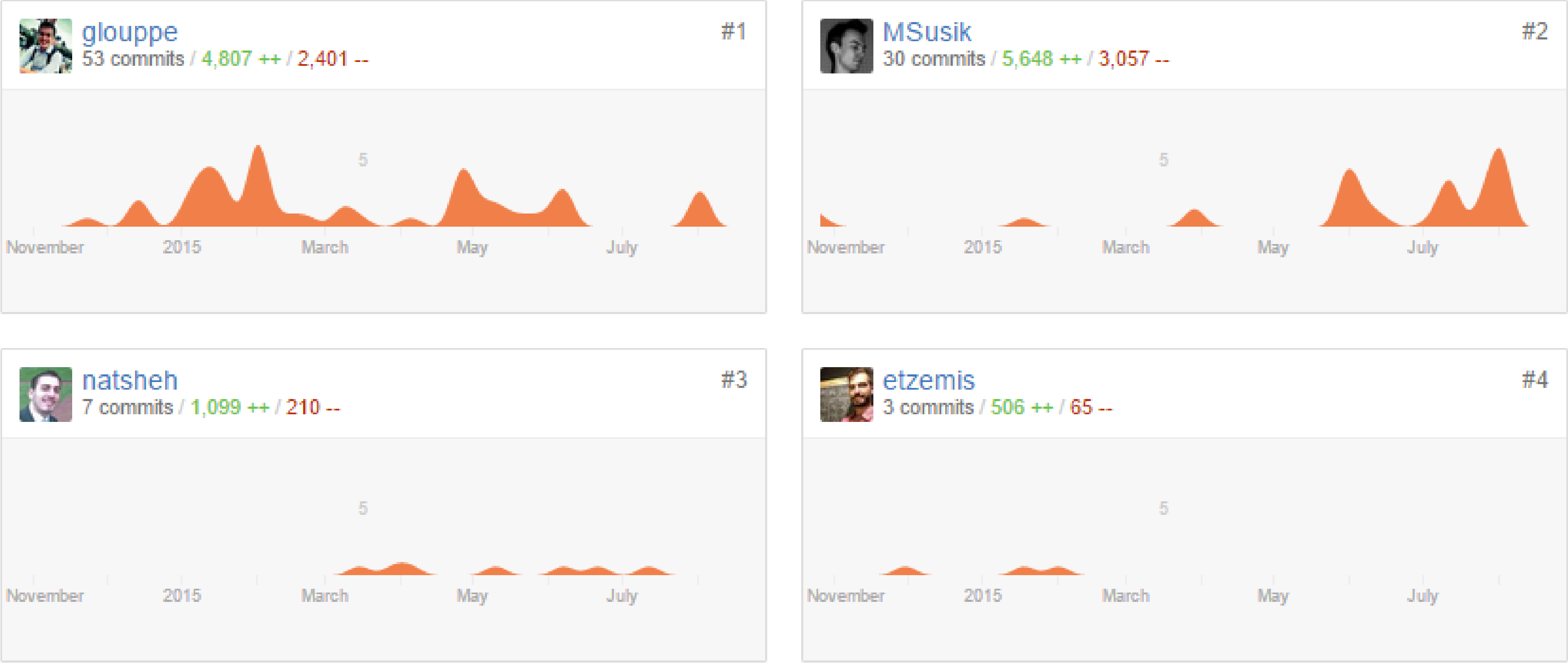
Please meet Yang Yang, Yang Yang, and Yang Yang!



Source: Laura Rueda Garcia, CERN

# OPEN SOURCE PROJECT @CERN: BEARD

Beard is a Python library of machine learning tools for Bibliographic Entity Automatic Recognition and Disambiguation.



glouppe #1
53 commits / 4,807 ++ / 2,401 --

MSusik #2
30 commits / 5,648 ++ / 3,057 --

natsheh #3
7 commits / 1,099 ++ / 210 --

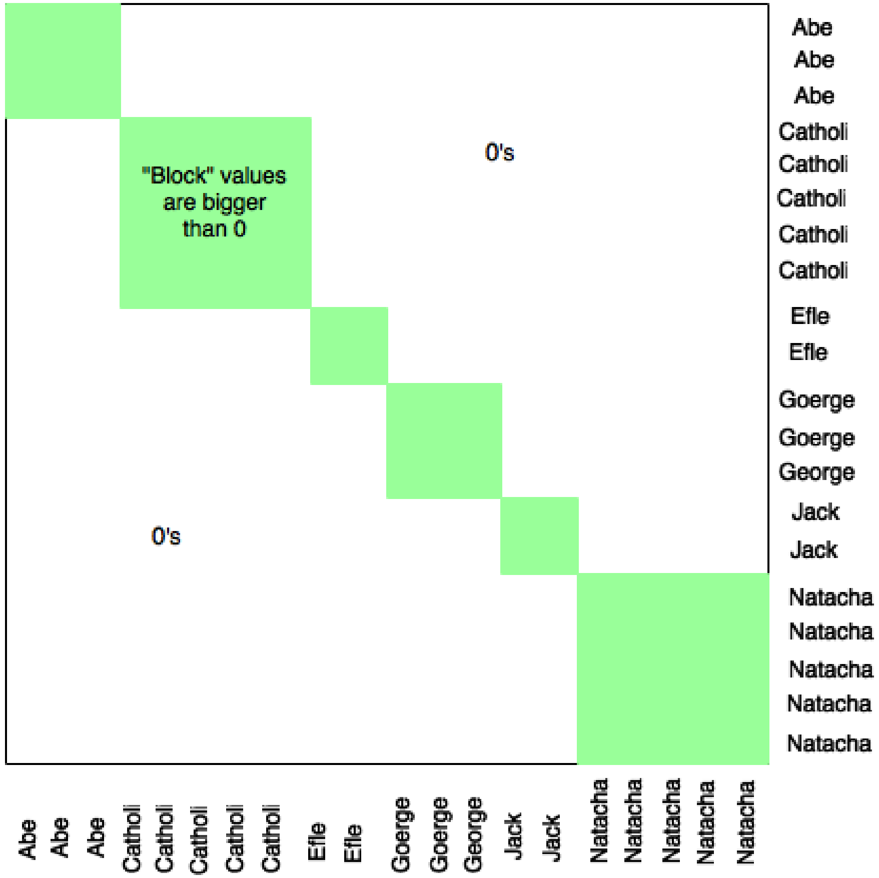
etzemis #4
3 commits / 506 ++ / 65 --

# DATA OVERVIEW

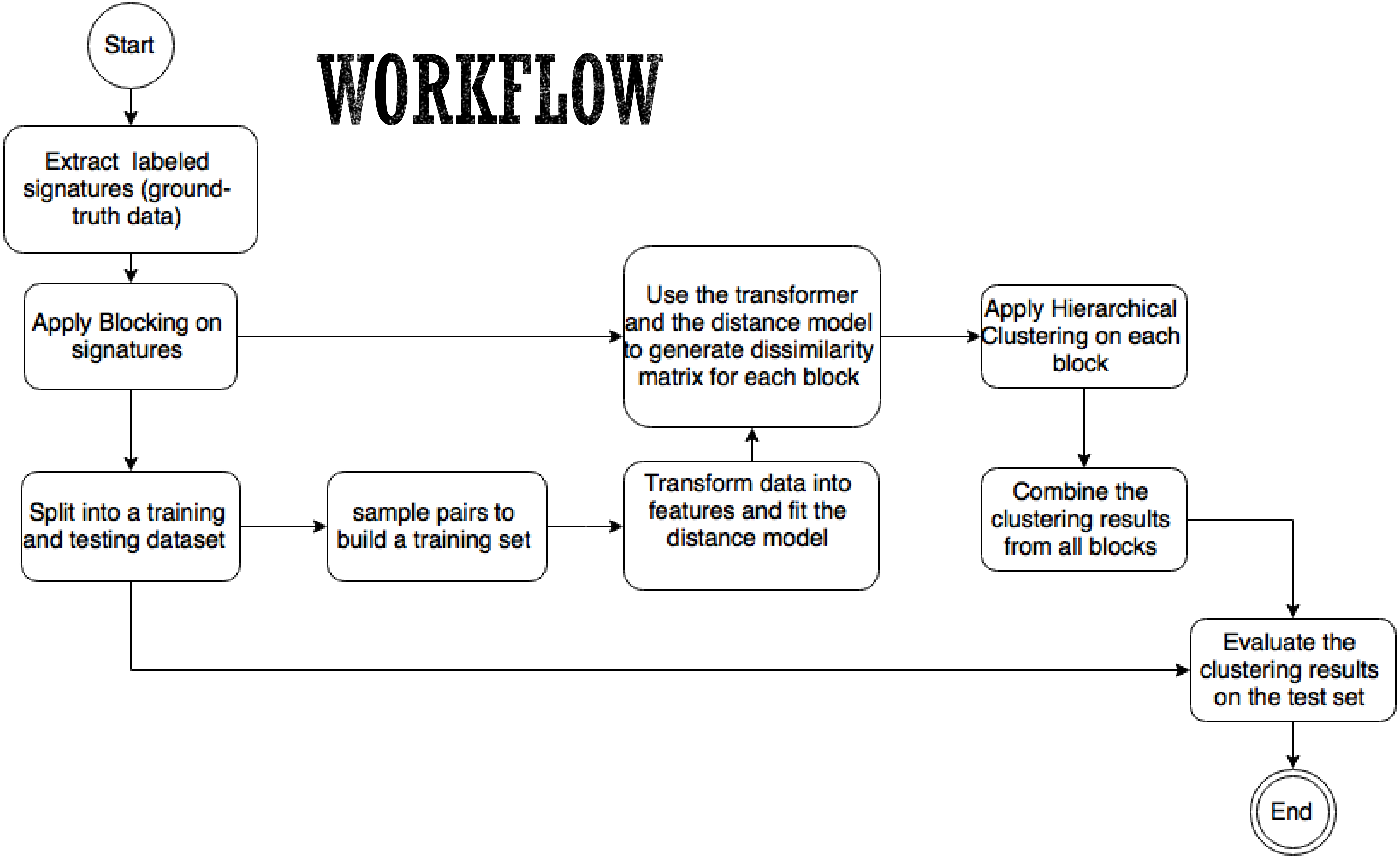
- Signature: Unique occurrence of author name and publication id

- We have over:
  - 1.07 million publications
  - 9.27 millions signatures
  - 1.2 million claimed signatures (Ground-truth)

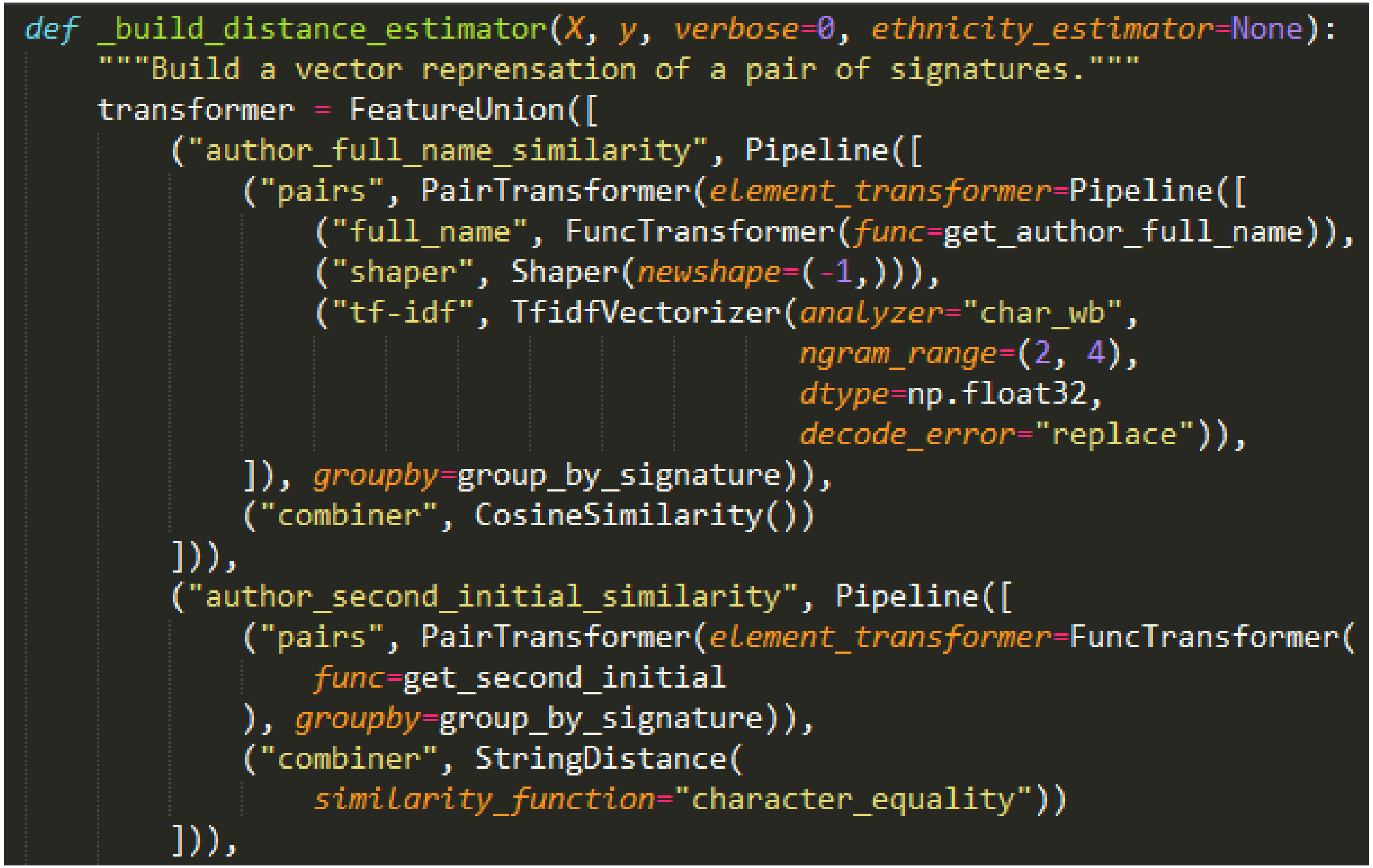
- Pair of signature (sig1, sig2, {0,1})
- Need for deviding into blocks



# WORKFLOW

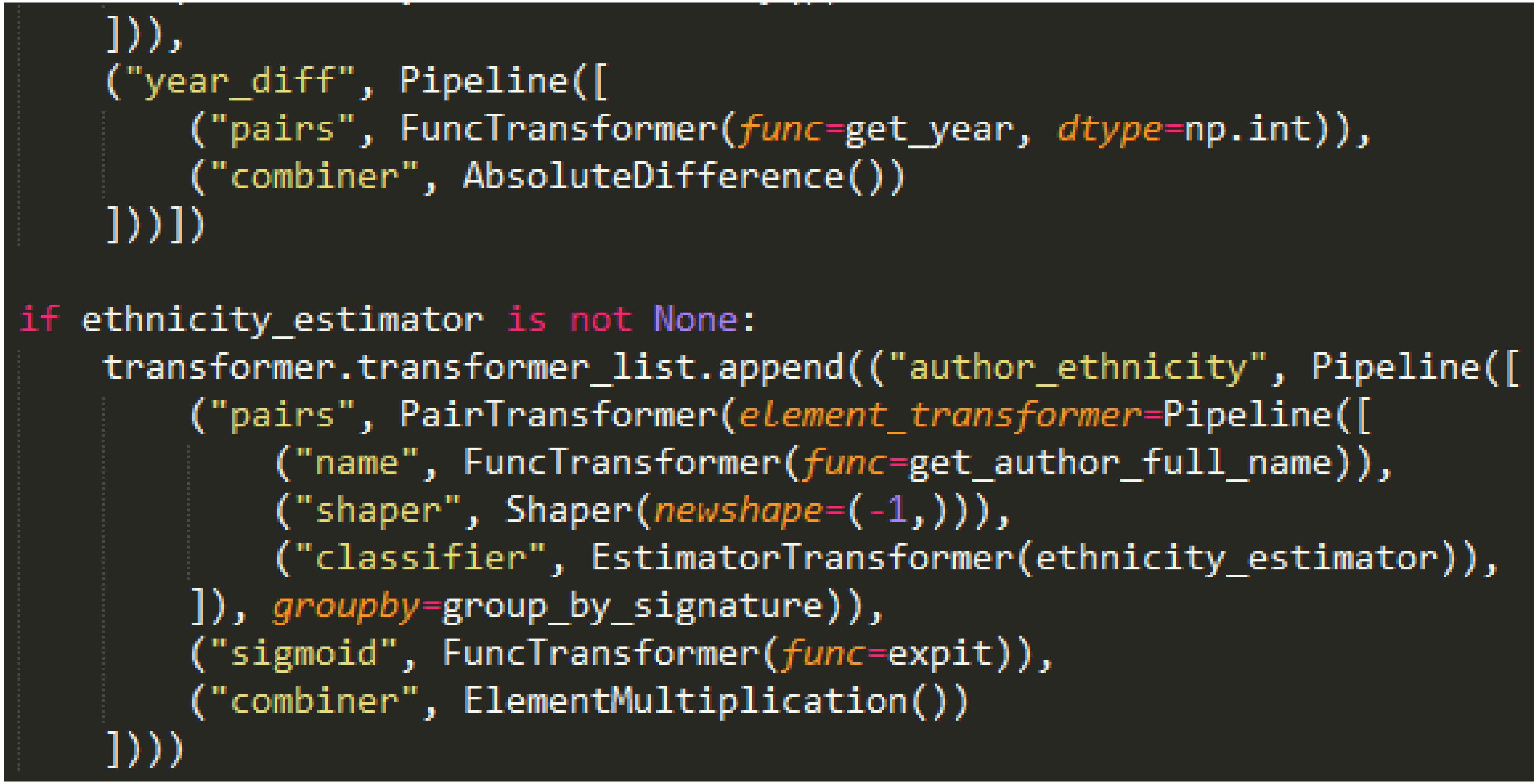

Start
Extract labeled signatures (ground-truth data)
Apply Blocking on signatures
Split into a training and testing dataset
sample pairs to build a training set
Transform data into features and fit the distance model
Use the transformer and the distance model to generate dissimilarity matrix for each block
Apply Hierarchical Clustering on each block
Combine the clustering results from all blocks
Evaluate the clustering results on the test set
End

# TRANSFORMER AND FEATURE UNION



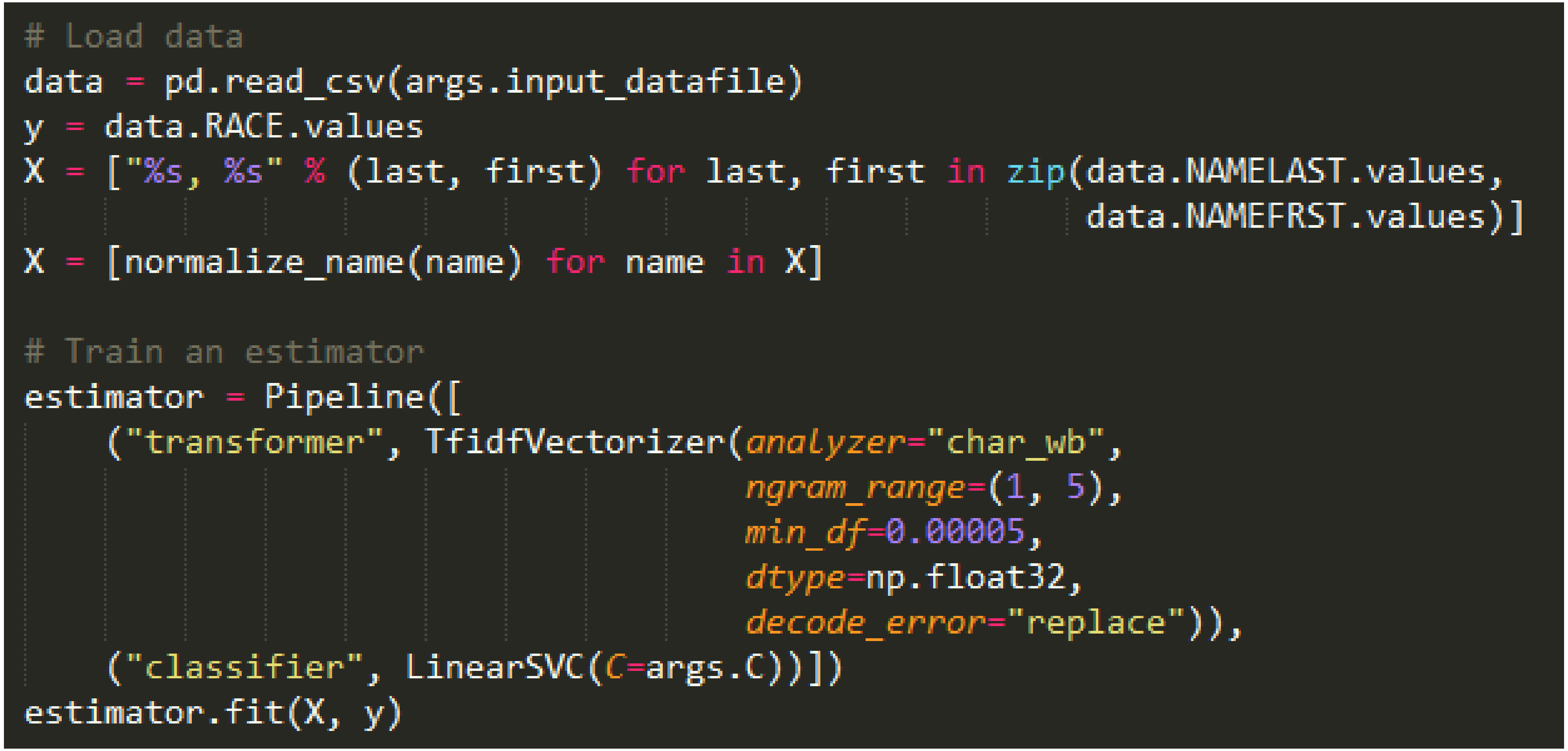
```
def _build_distance_estimator(X, y, verbose=0, ethnicity_estimator=None):
    """Build a vector reprensation of a pair of signatures."""
    transformer = FeatureUnion([
        ("author_full_name_similarity", Pipeline([
            ("pairs", PairTransformer(element_transformer=Pipeline([
                ("full_name", FuncTransformer(func=get_author_full_name)),
                ("shaper", Shaper(newshape=(-1,))),
                ("tf-idf", TfidfVectorizer(analyzer="char_wb",
                                          ngram_range=(2, 4),
                                          dtype=np.float32,
                                          decode_error="replace")),
            ]), groupby=group_by_signature)),
            ("combiner", CosineSimilarity())
        ])),
        ("author_second_initial_similarity", Pipeline([
            ("pairs", PairTransformer(element_transformer=FuncTransformer(
                func=get_second_initial
            ), groupby=group_by_signature)),
            ("combiner", StringDistance(
                similarity_function="character_equality"))
        ])),
```

# FEATURE COULD BE ESTIMATED



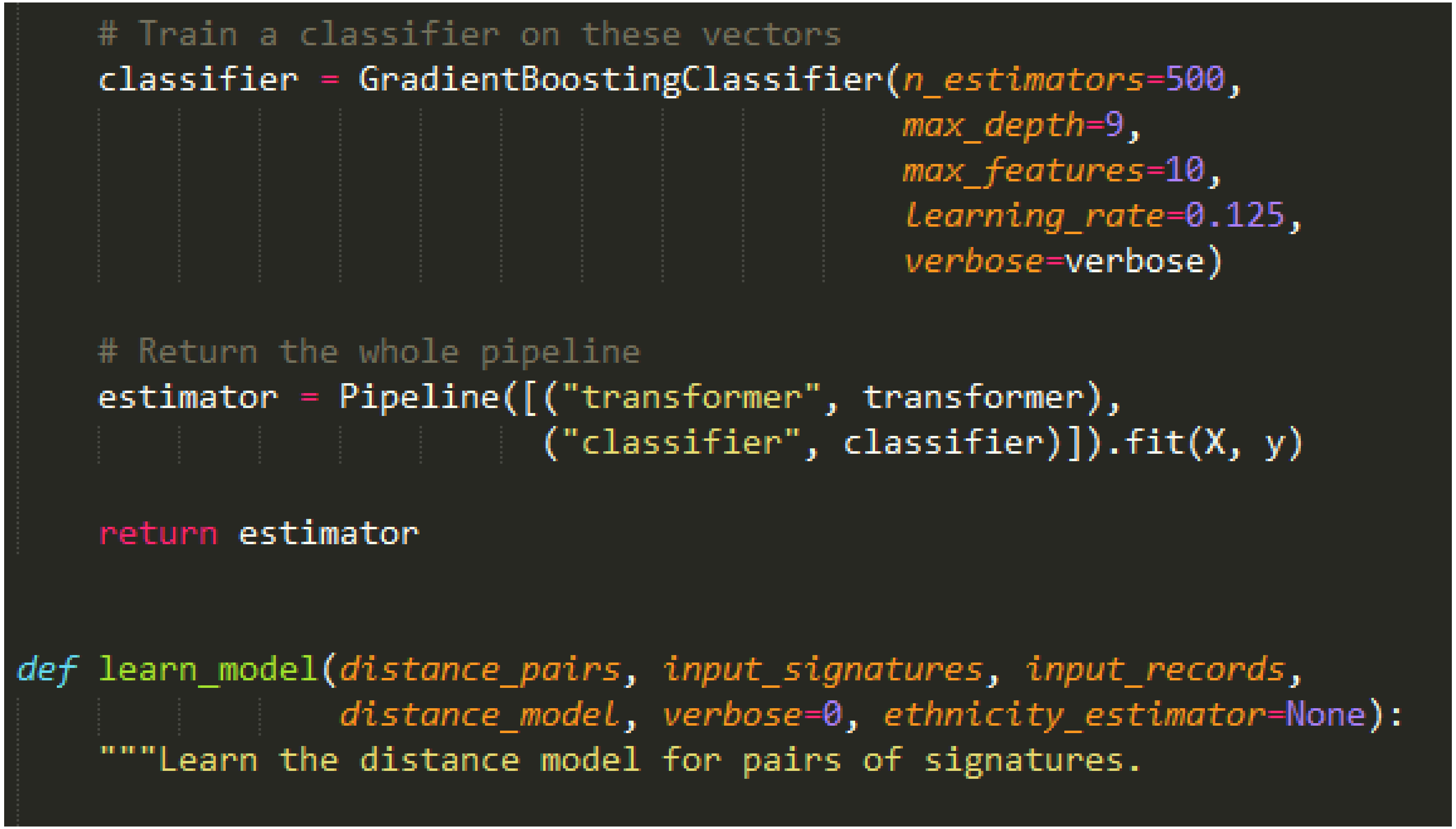
```
    ])),
    ("year_diff", Pipeline([
        ("pairs", FuncTransformer(func=get_year, dtype=np.int)),
        ("combiner", AbsoluteDifference())
    ]))])

if ethnicity_estimator is not None:
    transformer.transformer_list.append(("author_ethnicity", Pipeline([
        ("pairs", PairTransformer(element_transformer=Pipeline([
            ("name", FuncTransformer(func=get_author_full_name)),
            ("shaper", Shaper(newshape=(-1,))),
            ("classifier", EstimatorTransformer(ethnicity_estimator)),
        ]), groupby=group_by_signature)),
        ("sigmoid", FuncTransformer(func=expit)),
        ("combiner", ElementMultiplication())
    ])))
```

# ETHNICITY ESTIMATOR: EXTERNAL DATA



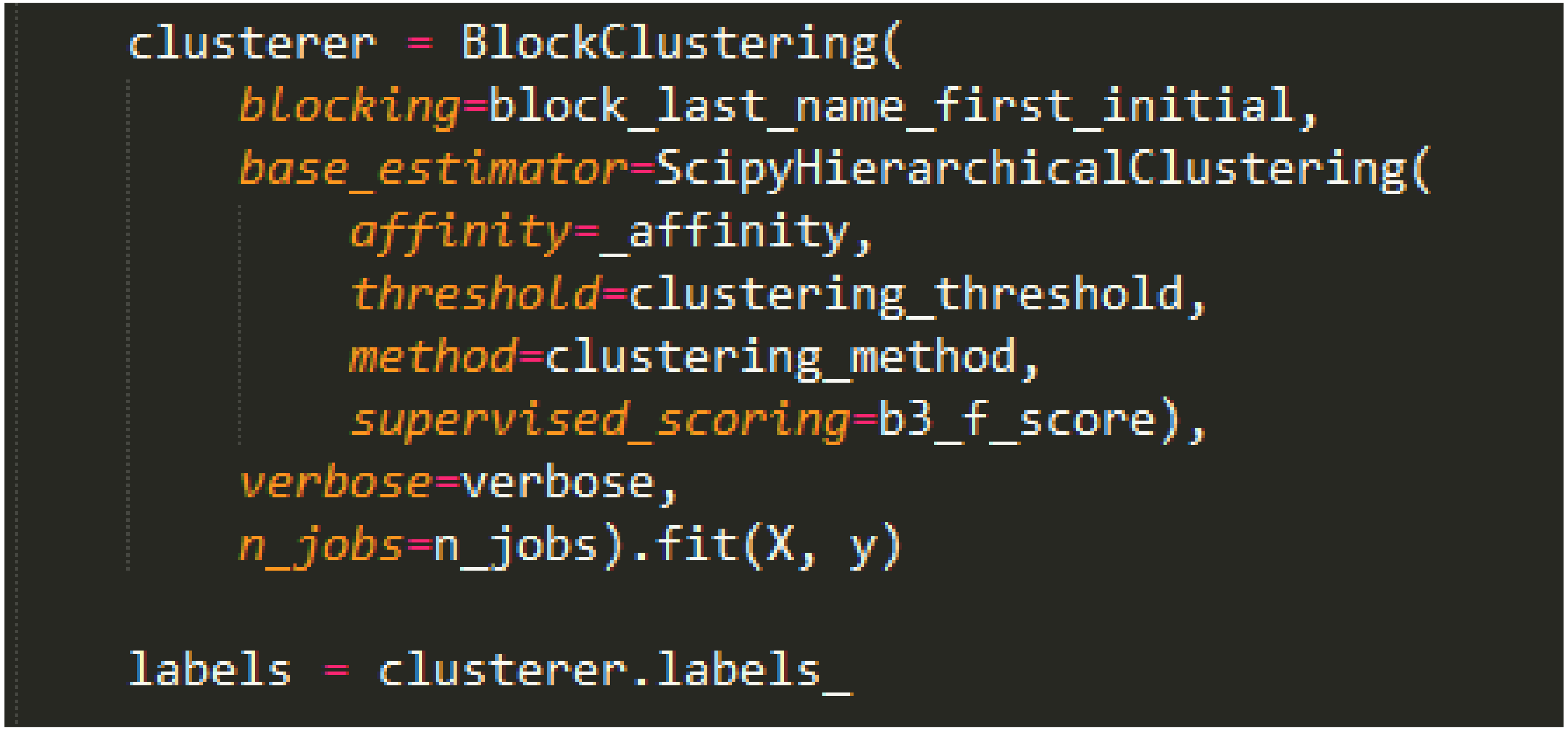
```python
# Load data
data = pd.read_csv(args.input_datafile)
y = data.RACE.values
X = ["%s, %s" % (last, first) for last, first in zip(data.NAMELAST.values,
                                                      data.NAMEFRST.values)]
X = [normalize_name(name) for name in X]

# Train an estimator
estimator = Pipeline([
    ("transformer", TfidfVectorizer(analyzer="char_wb",
                                    ngram_range=(1, 5),
                                    min_df=0.00005,
                                    dtype=np.float32,
                                    decode_error="replace")),
    ("classifier", LinearSVC(C=args.C))])
estimator.fit(X, y)
```

# PIPELINE OF TRANSFORMER AND CLASSIFIER



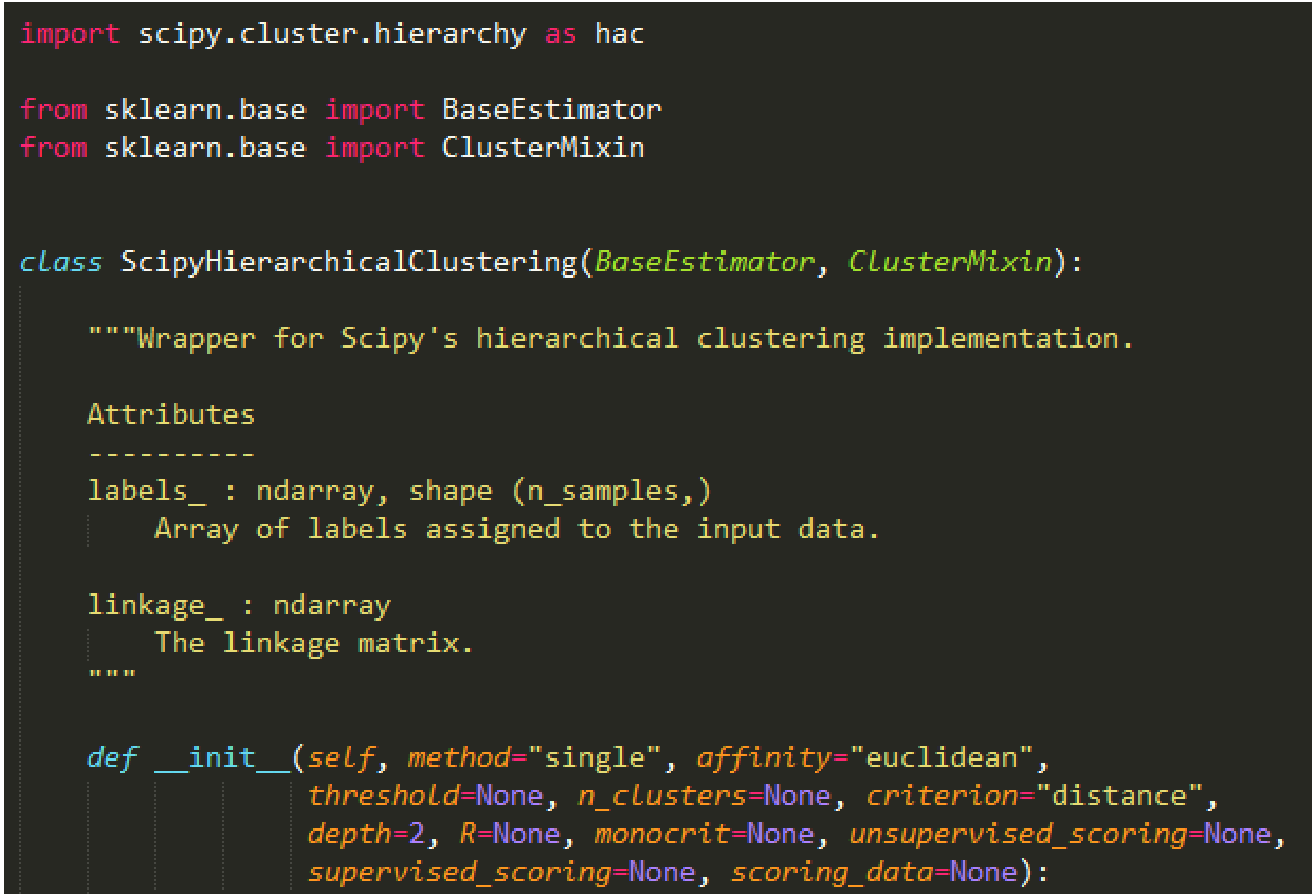
```python
    # Train a classifier on these vectors
    classifier = GradientBoostingClassifier(n_estimators=500,
                                            max_depth=9,
                                            max_features=10,
                                            learning_rate=0.125,
                                            verbose=verbose)

    # Return the whole pipeline
    estimator = Pipeline([("transformer", transformer),
                          ("classifier", classifier)]).fit(X, y)

    return estimator


def learn_model(distance_pairs, input_signatures, input_records,
                distance_model, verbose=0, ethnicity_estimator=None):
    """Learn the distance model for pairs of signatures.
```

# BLOCK CLUSTERING

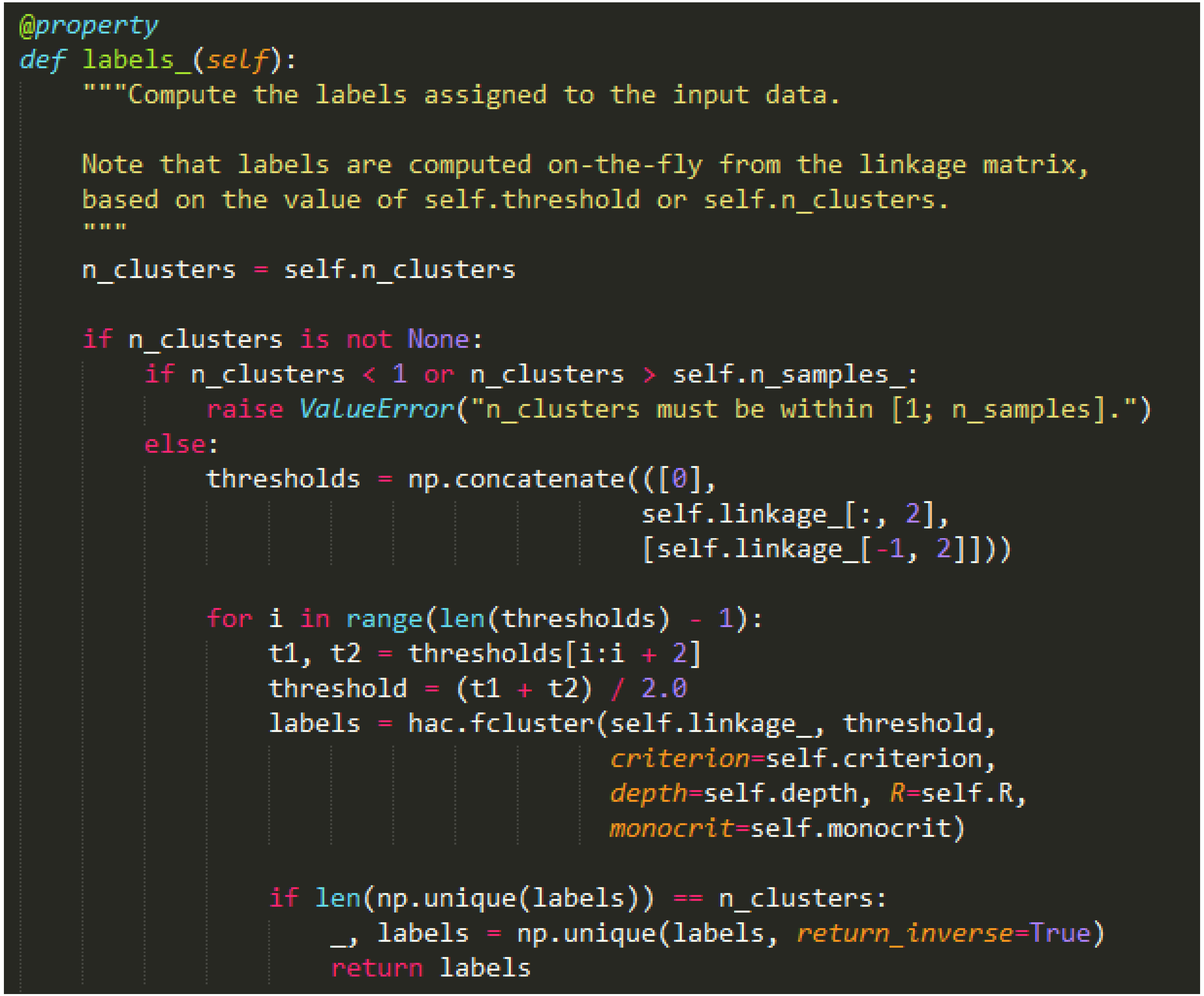


```
clusterer = BlockClustering(
    blocking=block_last_name_first_initial,
    base_estimator=ScipyHierarchicalClustering(
        affinity=_affinity,
        threshold=clustering_threshold,
        method=clustering_method,
        supervised_scoring=b3_f_score),
    verbose=verbose,
    n_jobs=n_jobs).fit(X, y)

labels = clusterer.labels_
```

# HAC CLUSTERING



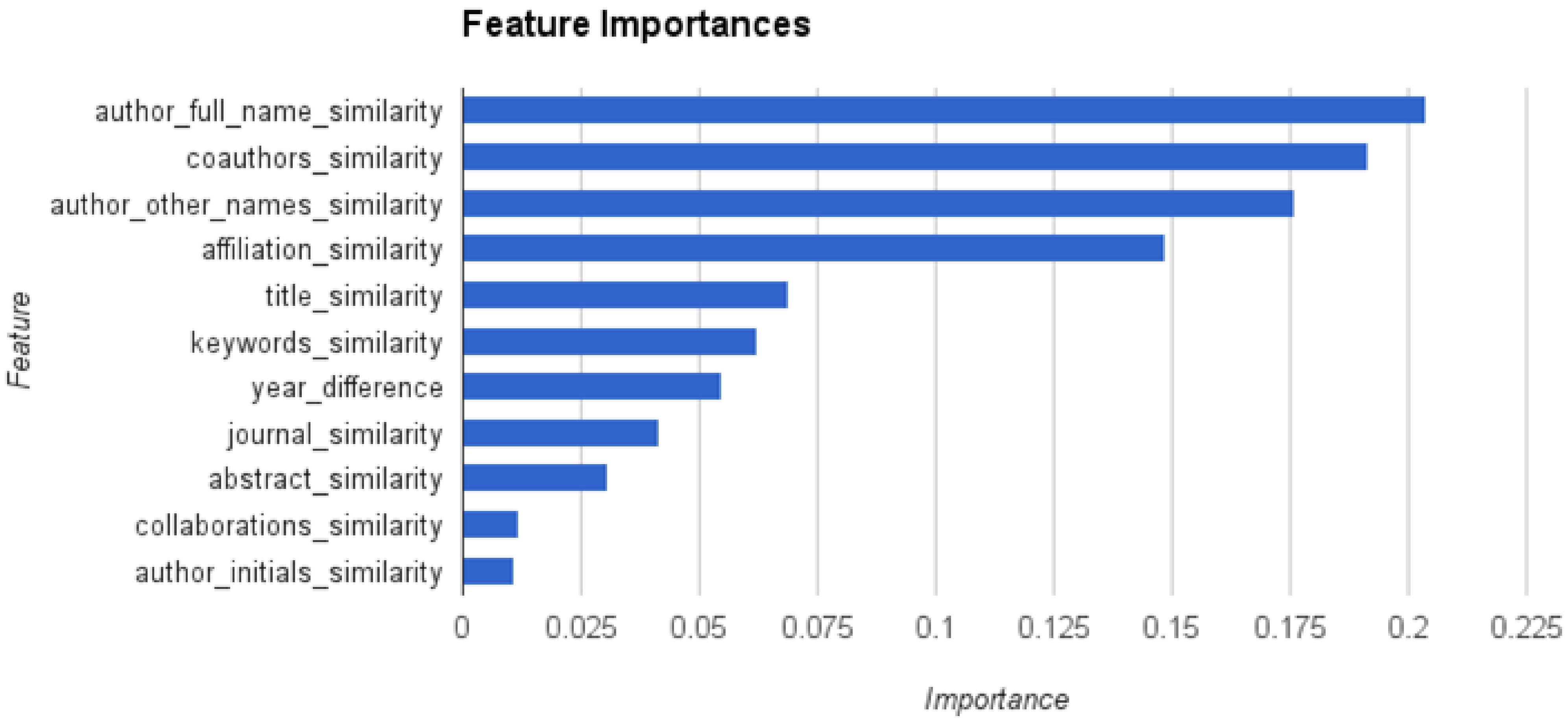
```python
import scipy.cluster.hierarchy as hac

from sklearn.base import BaseEstimator
from sklearn.base import ClusterMixin


class ScipyHierarchicalClustering(BaseEstimator, ClusterMixin):

    """Wrapper for Scipy's hierarchical clustering implementation.

    Attributes
    ----------
    labels_ : ndarray, shape (n_samples,)
        Array of labels assigned to the input data.

    linkage_ : ndarray
        The linkage matrix.
    """

    def __init__(self, method="single", affinity="euclidean",
                 threshold=None, n_clusters=None, criterion="distance",
                 depth=2, R=None, monocrit=None, unsupervised_scoring=None,
                 supervised_scoring=None, scoring_data=None):
```



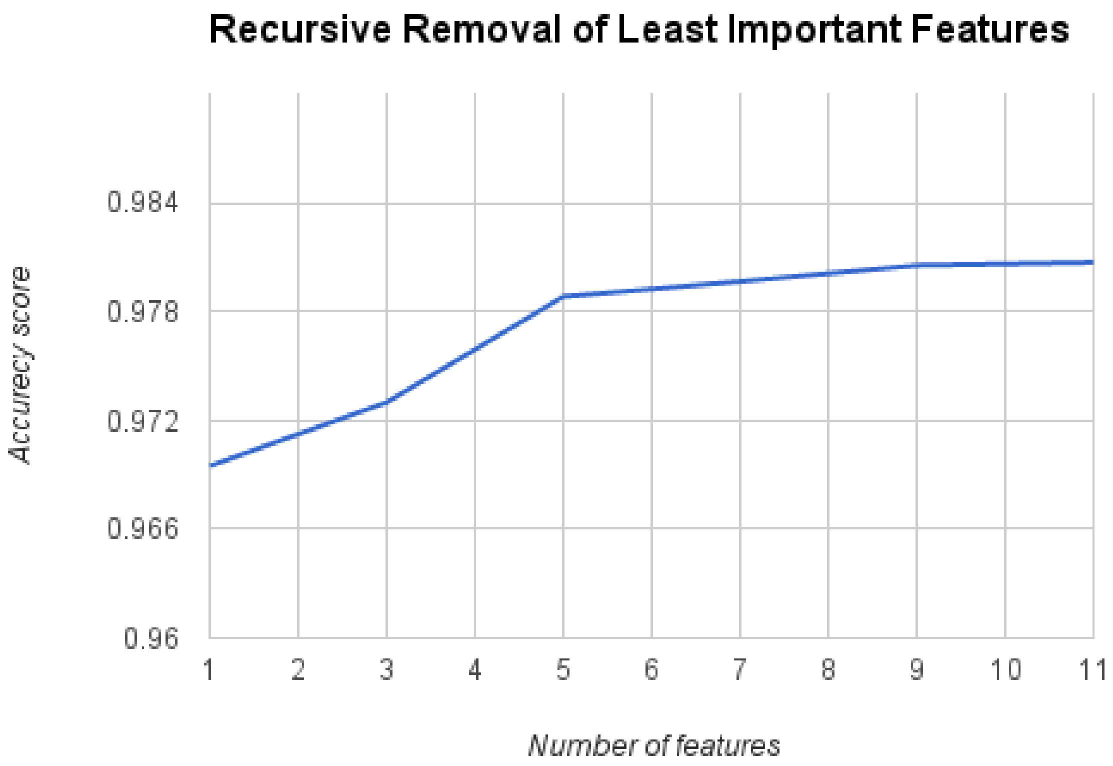
```
@property
def labels_(self):
    """Compute the labels assigned to the input data.

    Note that labels are computed on-the-fly from the linkage matrix,
    based on the value of self.threshold or self.n_clusters.
    """
    n_clusters = self.n_clusters

    if n_clusters is not None:
        if n_clusters < 1 or n_clusters > self.n_samples_:
            raise ValueError("n_clusters must be within [1; n_samples].")
        else:
            thresholds = np.concatenate(([0],
                                        self.linkage_[:, 2],
                                        [self.linkage_[-1, 2]]))

            for i in range(len(thresholds) - 1):
                t1, t2 = thresholds[i:i + 2]
                threshold = (t1 + t2) / 2.0
                labels = hac.fcluster(self.linkage_, threshold,
                                      criterion=self.criterion,
                                      depth=self.depth, R=self.R,
                                      monocrit=self.monocrit)

                if len(np.unique(labels)) == n_clusters:
                    _, labels = np.unique(labels, return_inverse=True)
                    return labels
```

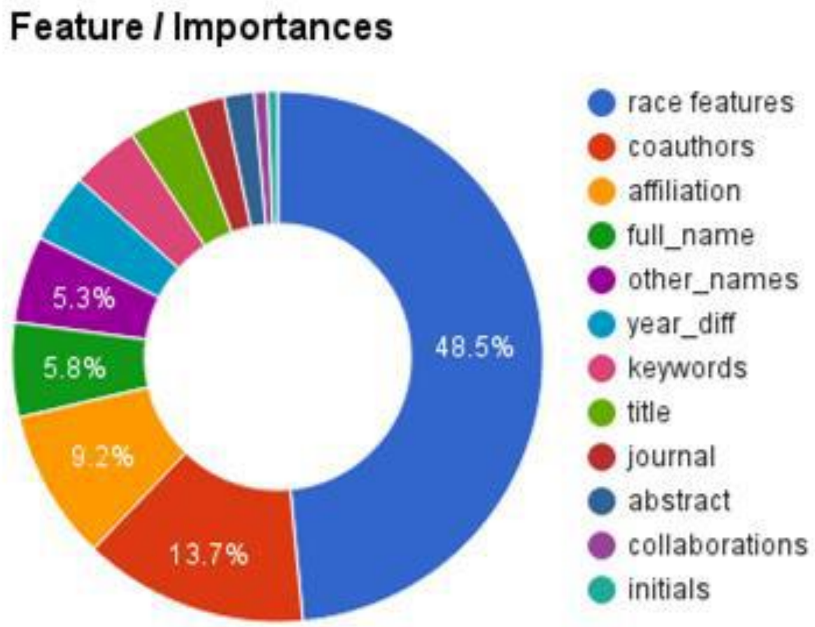
# PROPERTY

# FEATURE IMPORTANCES

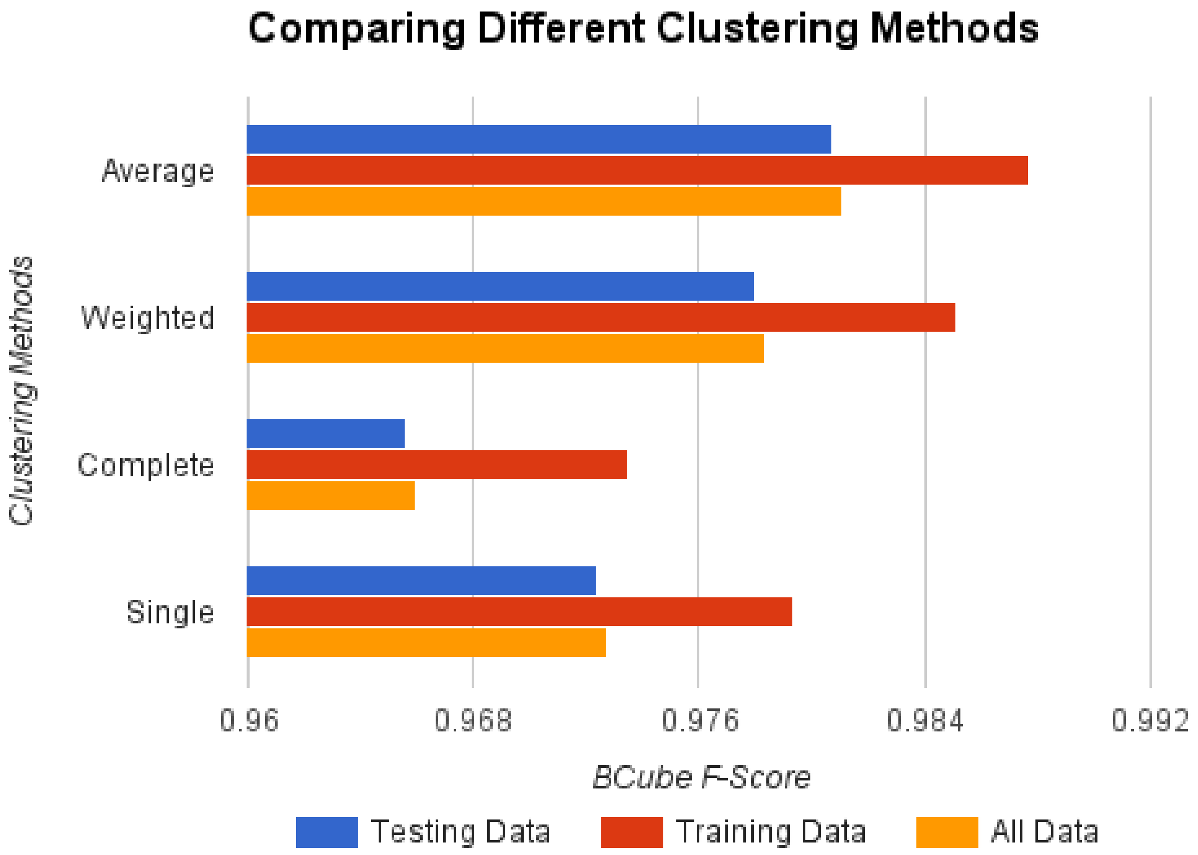


# RECURSIVE REMOVAL





# CLUSTERING METHODS



# REFERENCES AND RECOMMENDED READINGS

# REFERENCES

- Data Clustering- Part 2, M2 DMKM Slides, Julien Ah-Pine, Universit´e Lyon 2, 2014
- BEARD: https://github.com/inveniosoftware/beard
- Scikit-learn: http://scikit-learn.org
- Hussein, AL-NATSHEH. "Bibliographic Entity Automatic Recognition and Disambiguation.“, CERN, 2015

  https://preprints.cern.ch/record/2036112/files/CERN-THESIS-2015-098.pdf
- Buitinck, Lars, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae et al. "API design for machine learning software: experiences from the scikit-learn project." *arXiv preprint arXiv:1309.0238* (2013).

  http://arxiv.org/pdf/1309.0238.pdf

# RECOMMENDED ML COURSES

Stanford University
Machine Learning

University of Washington
Introduction to Data Science

Practice on:

kaggle

# HUSSEIN AL-NATSHEH

Thanks for attending!

## CONTACT DETAILS:

- https://ch.linkedin.com/in/natsheh
- https://twitter.com/hnatsheh
- h.natsheh@ciapple.com

Ciapple

Better Knowledge... For Better Decisions