A Knowledge Flow as a Software Product Line

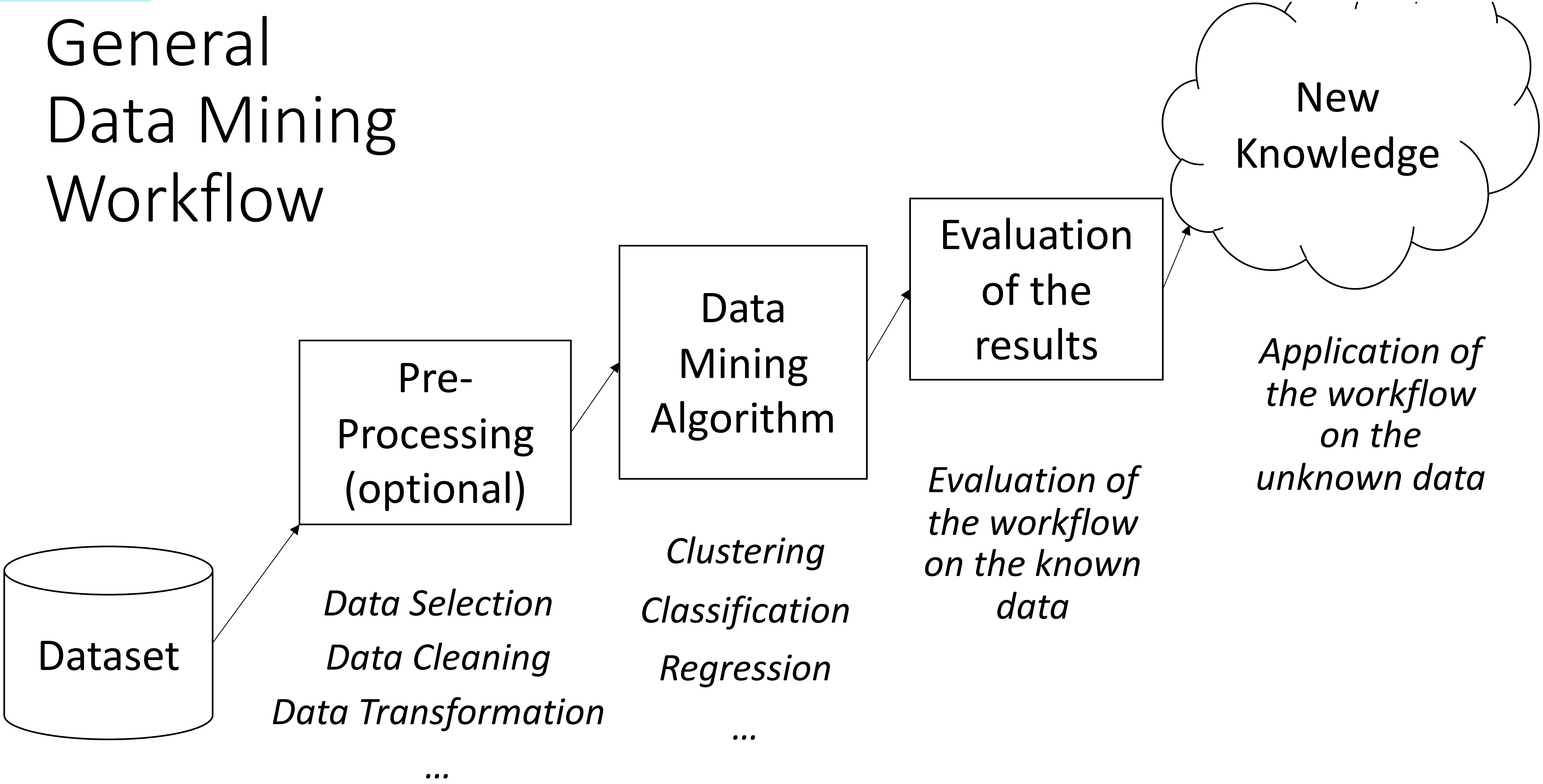
Presented by: Luca Parisi

Supervisors:

Prof. Mireille Blay-Fornarino (I3S)

Prof. Frederic Precioso (I3S)

General Data Mining Workflow



Data Selection

Data Cleaning

Data Transformation

…

Clustering

Classification

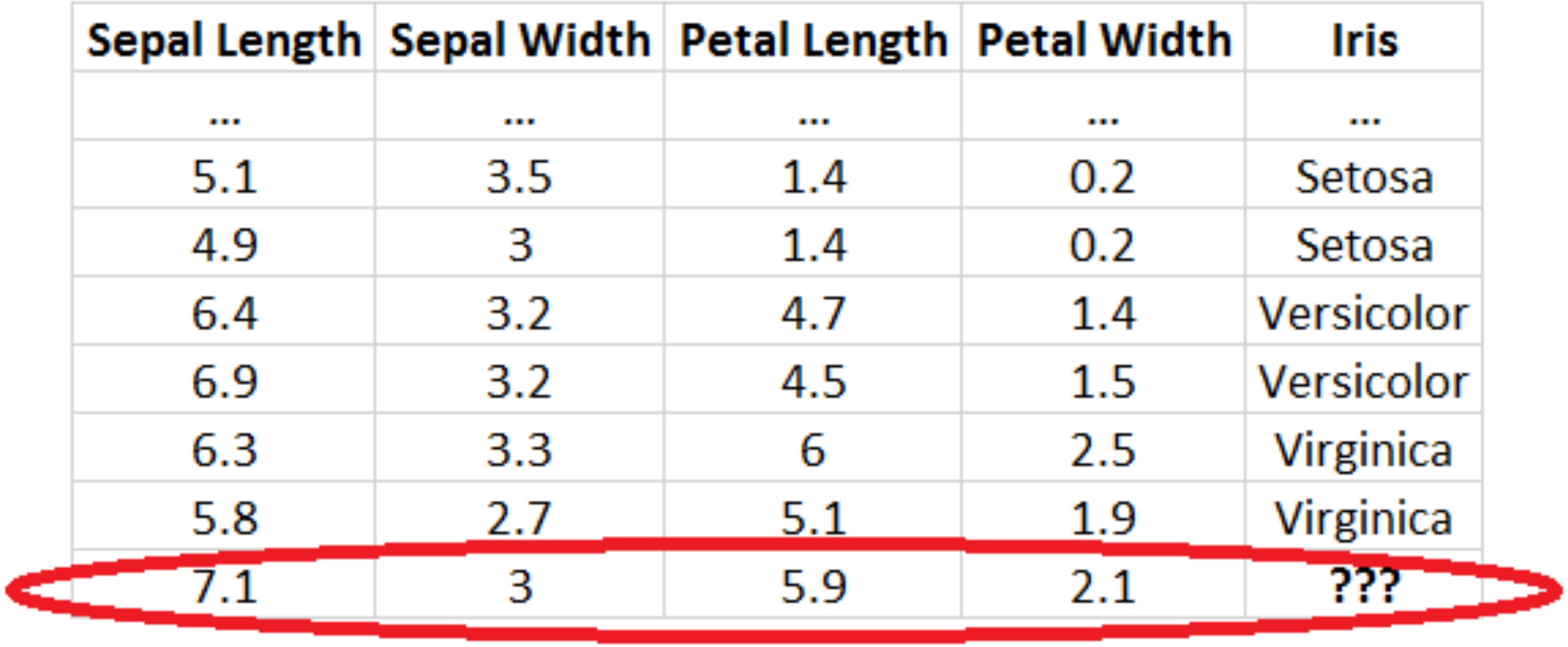
Regression

…

Evaluation of the workflow on the known data

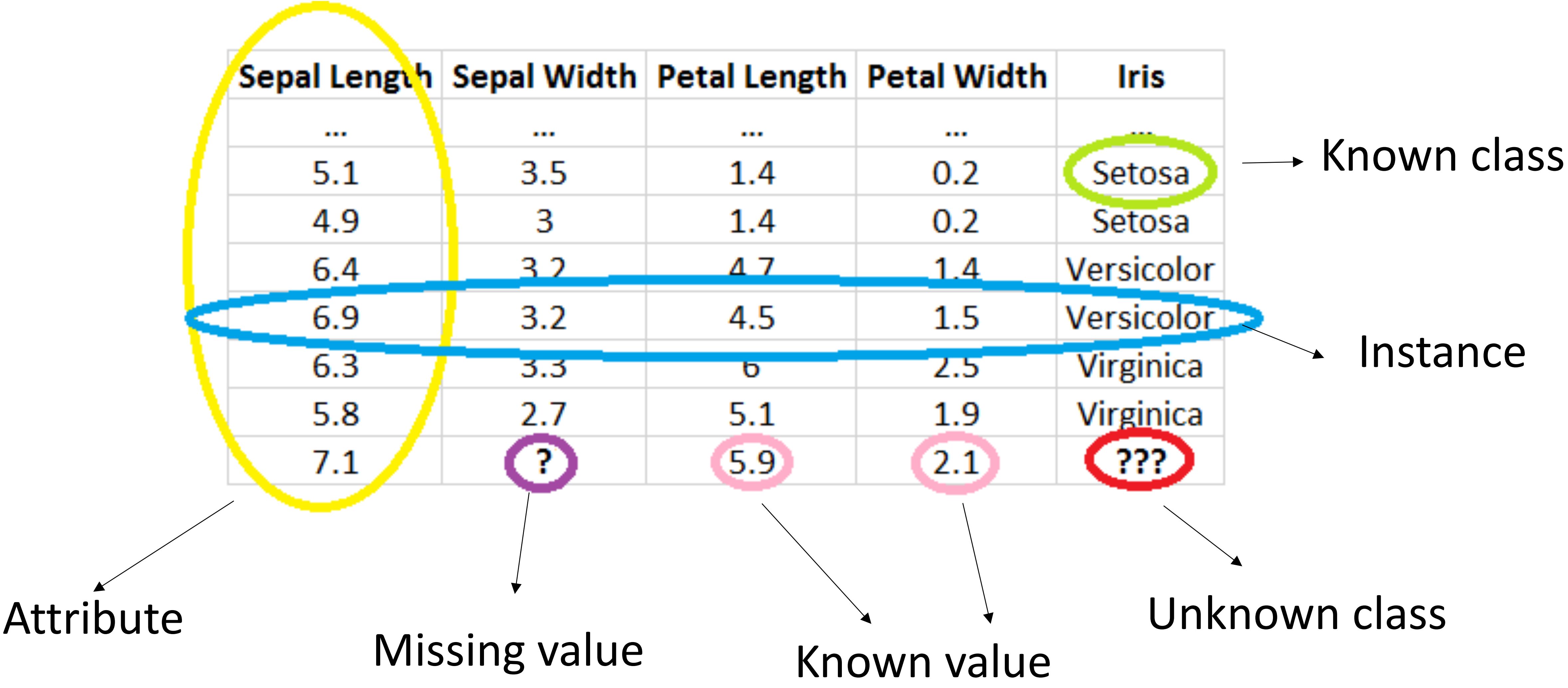
Application of the workflow on the unknown data

Supervised Classification: Example Iris Dataset



Sepal Length	Sepal Width	Petal Length	Petal Width	Iris
...	...	...	...	...
5.1	3.5	1.4	0.2	Setosa
4.9	3	1.4	0.2	Setosa
6.4	3.2	4.7	1.4	Versicolor
6.9	3.2	4.5	1.5	Versicolor
6.3	3.3	6	2.5	Virginica
5.8	2.7	5.1	1.9	Virginica
7.1	3	5.9	2.1	**???**

Supervised Classification: Example Iris Dataset



Sepal Length	Sepal Width	Petal Length	Petal Width	Iris
…	…	…	…	…
5.1	3.5	1.4	0.2	Setosa
4.9	3	1.4	0.2	Setosa
6.4	3.2	4.7	1.4	Versicolor
6.9	3.2	4.5	1.5	Versicolor
6.3	3.3	6	2.5	Virginica
5.8	2.7	5.1	1.9	Virginica
7.1	?	5.9	2.1	???

Known class

Instance

Attribute

Missing value

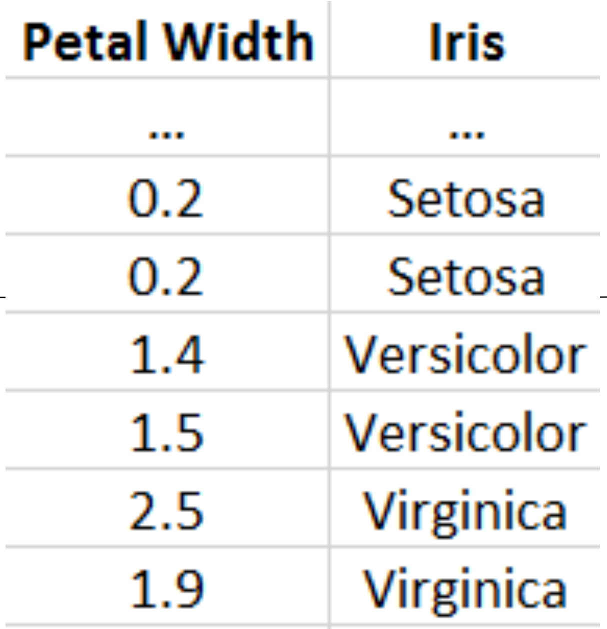
Known value

Unknown class

Numeric and Nominal Attributes

Numeric

Attribute



Petal Width	Iris
...	...
0.2	Setosa
0.2	Setosa
1.4	Versicolor
1.5	Versicolor
2.5	Virginica
1.9	Virginica

Nominal

Attribute

Evaluation: Training and Test set

- **Training set:** Subset of instances with known classes. Used to train the classifier (automatic learning)

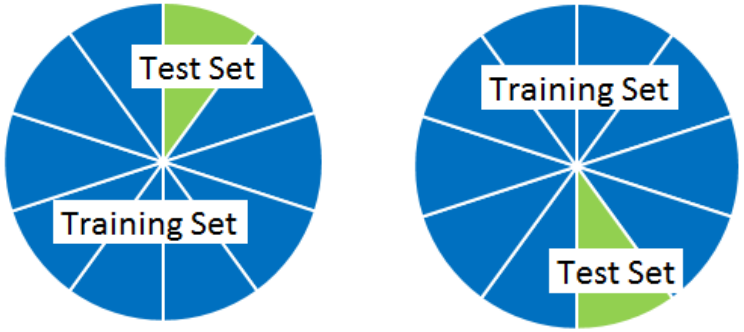
- **Test set:** Remaining instances with known classes. Used to check if the predictions of the classifier the correct class or not

- **Accuracy:** Indicator for evaluation. It is the % of correct predicted instances of the test set.

Evaluation Statistically Sound: Cross Validation

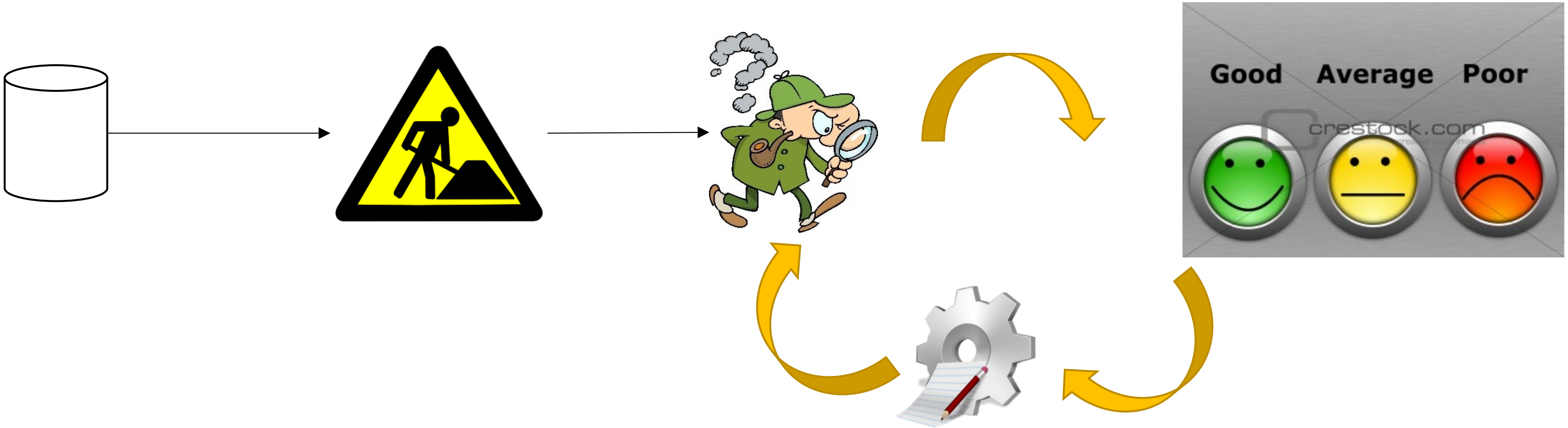
- **Cross Validation:** technique to evaluate several times a classifier on different training sets and test sets.

- **N-Folds:** Divide data into N partitions, then for N times:
  - N-1/N of data is the training set
  - 1/N is the test set.
  - Evaluate and change folds



- **Accuracy:** Is the average of the N accuracies found on each test.

Classification: General Evaluation Workflow



Dataset with known classes

Pre-Processing (optional)

Classification Algorithm

Evaluation: Cross Validation

Algorithm parameters tuning

Motivations

- **Motivation #1**: We want to skip the evaluation phase. We want to be able to know in advance what the best workflow will be, according to the input dataset.

- **Motivation #2**: Implement a system that helps the user in choosing the best workflow according to the input dataset

Challenge: Amount of algorithms

- **Challenge #1**: In the data mining literature there exist a lot of algorithms of classification.

- **(old) Research question:** Can we identify an algorithm who is always better than the others?

- **No-Free-Lunch Theorem**: The best classifier will not be the same for each dataset, it depends on the input dataset [1]

Challenge: Pre-processing Dependency

- **Challenge #2:** The application of a pre-processing technique on the user's input data changes the performances of the algorithms of classification.

- **Research question:** What is the impact of the pre-processing on the algorithms of classification?

Challenge: A lot of variability

- **Challenge #3**: The combination of choice of pre-processing techniques, choice of classification algorithms and parameters tuning creates a lot of variability

- **Hot topic**: Impossible to try all the combinations of workflows in order to find the best workflow (Limit of time).

- **Research question**: Without doing the evaluation, can we know in advance a workflow that has always the best accuracy?

Challenge: User Constraints

- **Challenge #4**: Beyond the accuracy, the users may have some constraint of:
  - Total time of execution of the workflow
  - RAM required by the workflow during the execution

- **Research question**: Without doing the evaluation, can we know in advance what is the behavior of classifiers w.r.t. the total time and RAM required by a workflow?

Related Work

- **M.F.Delgado et al. (2014)** « *Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?* » Journal of Machine Learning Research 15 (2014) 3133-3181

- **Experiments:** 179 classifiers implemented in Weka [2], R [3], C and MatLab on 121 datasets coming from the UCI repository [4],

- **Results:** They reported a ranking of classifiers in order to see which classifier works best in general, basing on the average accuracy.

Delgado’s Experiment Features

- **Pre-processing technique:** Each attribute has been converted into a standardized numeric attribute (zero mean and standard deviation one).

- **Evaluation**: 4-fold cross validation, where the class labels have been equally separated between training and test set

Critics and Questions

- **Critics:**
  - They have compared the performances of the classifiers by considering only 1 pre-processing technique
  - For cross validation, they have prepared the folds ad-hoc for each dataset.

- **Question:** Does the ranking change if a classifier works best with a different kind of pre-processing?

- **Question:** Does the ranking change if we use a 10-Fold cross validation without guaranteeing an equal separation of classes between training and test set?

Contribution: Paper Comparison

- **Deep study:** Of the algorithms of classification in order to know how they work and which kind of data they can manage (numeric, nominal, missing values, …) [5] [6] [7]
- **Experiment comparison:** Replicate in part the experiments of the previous paper but considering in plus:
  - 10 different pre-processing techniques to apply to each dataset and the original dataset (without pre-processing)
  - For cross validation, use both the 4-Folds and the 10-Folds without guaranteeing an equal separation of classes between training and test set

Our Experiments

- **Environment :** Based on the java Weka's APIs:
    - 65 / 179 algorithms of classifications implemented in Weka
    - 102 / 121 datasets coming from the UCI repository

- **Parameters tuning:** Each algorithm has the same parameters of the ones reported into the paper.

- **Weka behavior:** It applies automatically a hidden pre-processing on the dataset if the algorithm is not compatible with it (<u>not taken into account in the paper</u>)

Our Experiments

- **Algorithm properties :** We have tested an algorithm on a dataset only if it is compatible with it.

  - Manage **multi class** problems (#classes > 2)
  - Require **nominal attributes**
  - Require **numeric attributes**
  - Manage **missing values**

Our Experiments

- **Pre-processing:** We have chosen a combination of 10 pre-processing techniques that change the dataset properties.

  - **Replace missing values** with mean (numeric) / mode (nominal)
  - **Discretization**: Convert numeric attributes in nominal attributes
  - **Binarization**: Convert nominal attributes in numeric attributes (0 - 1)
  - **Attribute selection**: Selection of a subset of attributes, trying to remove the high-correlated ones.

Comparison of Results

- **Comparable results:** 102 / 121 datasets, we need to recompute the results reported on the paper on the 102 datasets.

- **Ranking comparison:** 2 rankings of 2 different average accuracies:

  1. The accuracy comes from the pre-processed dataset used into the paper
  2. The accuracy comes from the best pre-processed dataset

Results: Pre-Processing Impact

Top 5 classifiers, 4-Folds cross validation:

Classifier	**% Avg Accuracy Paper Pre-Proc.**	**Classifier**	**% Avg Accuracy Best Pre-Proc.**
Nearest Neighbour (1)	94,32	RotationForest RandomTree	97,75
Nearest Neighbour (K)	94,17	Nearest Neighbour (1)	97,60
Random Forest	93,69	Random Forest	97,44
RotationForest RandomTree	93,50	Nearest Neighbour (K)	97,36
RandomComittee RandomTree	93,36	MultiboostAB, RandomTree (from 6th)	97,32

Results: Pre-Processing Impact

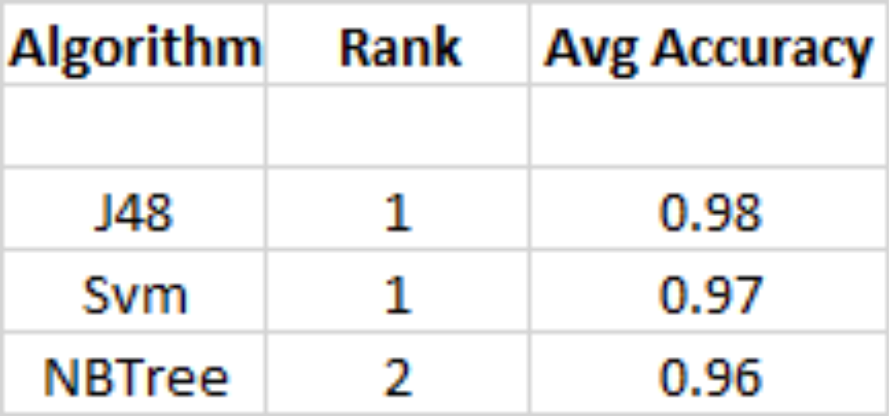
Top 5 classifiers 10-Folds cross validation:

Classifier	% Avg Accuracy Paper Pre-Proc.	Classifier	% Avg Accuracy Best Pre-Proc.
RotationForest J48	76,21	RotationForest J48	79,63
Random Forest	75,76	RotationForest RandomTree	79,47
Logistic Model Tree	75,69	Logistic Model Tree	79,17
RotationForest RandomTree	75,55	Random Forest	79,05
Bagging PART	75,15	MultiboostAB NBTree (from 8th)	78,74

Contribution: Groups of Classifiers

- **Significant difference:** From the results of the experiments, we want to distinguish the classifiers that have a significant difference w.r.t. each performace (accuracy, time, RAM)

- **Rank:** value that groups together the classifiers which do not have a significant difference.

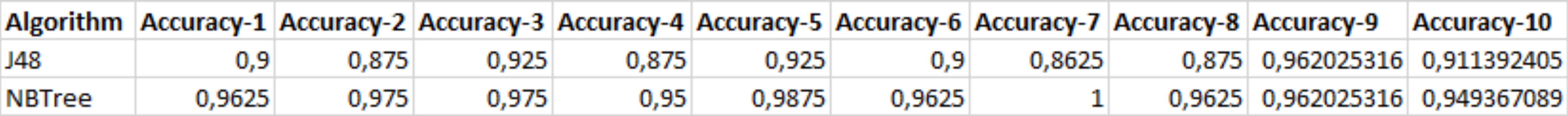


Algorithm	Rank	Avg Accuracy
J48	1	0.98
Svm	1	0.97
NBTree	2	0.96

Significant Difference: Statistical Tests

- **Arrays of values:** The statistical tests compare 2 arrays of values instead of 2 simple numbers (ex. 2 accuracies)

- **Idea:** For each performance (accuracy, time, RAM), we take the values obtained by the 4-Fold and 10-Fold cross validation

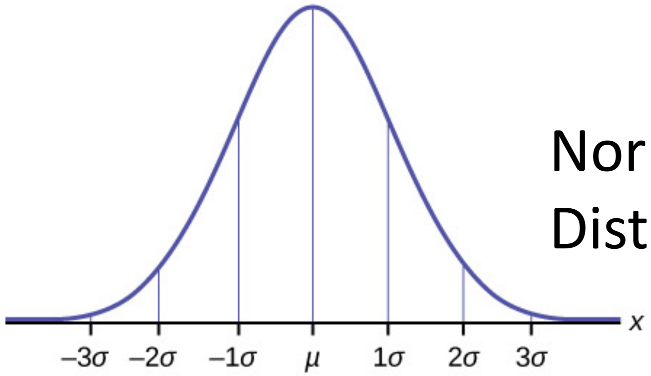


Algorithm	Accuracy-1	Accuracy-2	Accuracy-3	Accuracy-4	Accuracy-5	Accuracy-6	Accuracy-7	Accuracy-8	Accuracy-9	Accuracy-10
J48	0,9	0,875	0,925	0,875	0,925	0,9	0,8625	0,875	0,962025316	0,911392405
NBTree	0,9625	0,975	0,975	0,95	0,9875	0,9625	1	0,9625	0,962025316	0,949367089

- **Null hypothesis:** The difference of the means of the arrays is zero.

Statistical Tests: Method

- **Kolmogorov-Smirnov test:** Do the 2 arrays follow a normal distribution?



Normal Distribution

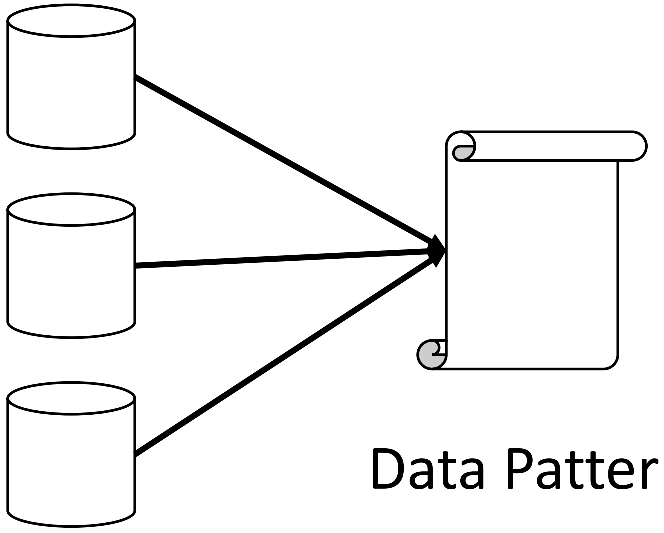
- **Student’s paired T-test :** Precise test, requires the condition of normality of data

- **Wilcoxon signed-rank test :** Less precise test, it does not require the condition of normality

Current Work: Data Pattern

- **Problem:** For new datasets, can we find the same best workflow (rank = 1) on some tested datasets without doing the evaluation?

- **Idea:**



Data Pattern

Input datasets

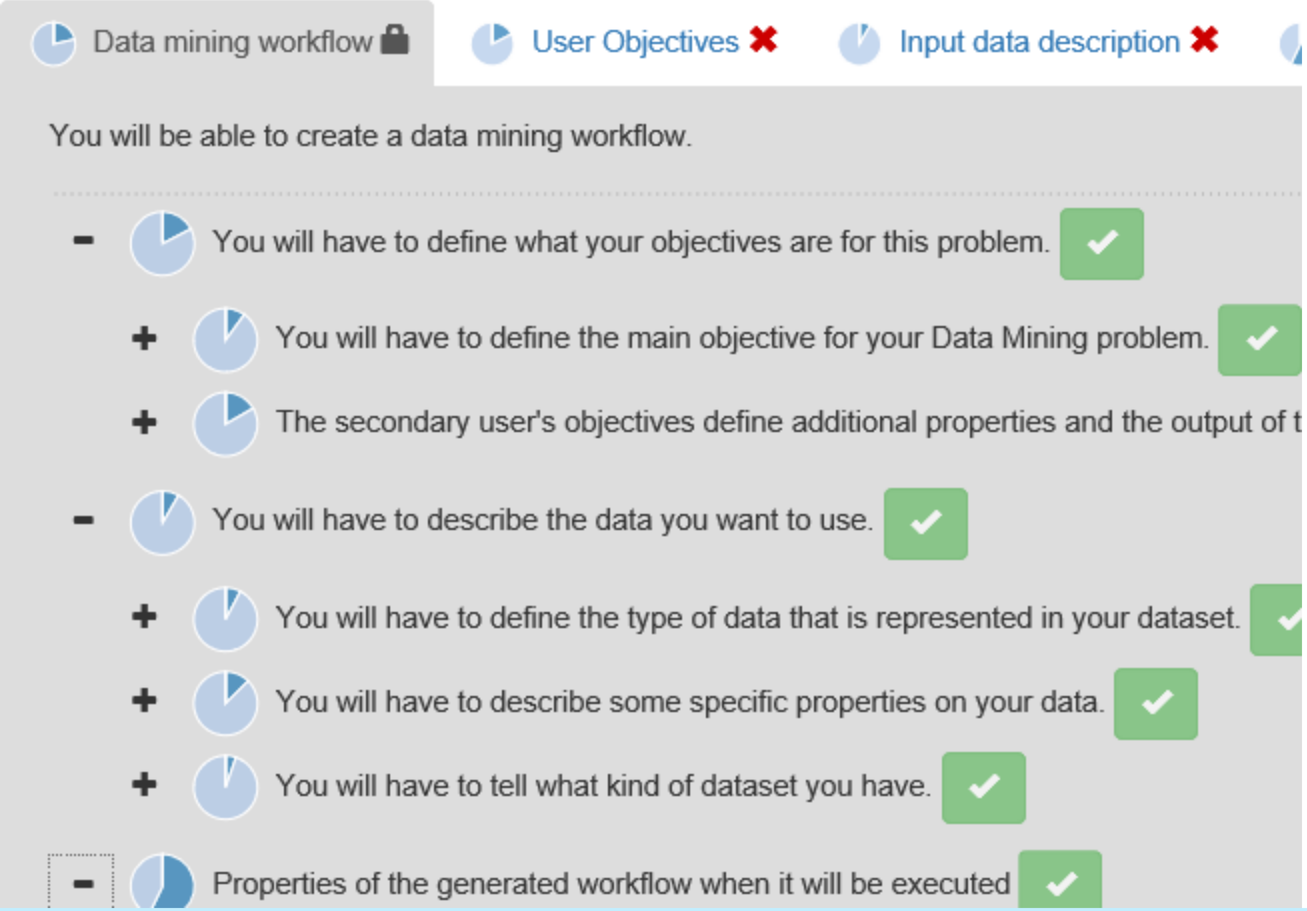
If the best workflow depends on the data pattern…

Final Conclusions

- **Pre-processing impact:** We have proved that the impact of the pre-processing cannot be ignored when comparing the classifiers
- **Work in progress:** we are currently working in finding data patterns where a workflow has always rank = 1
- **RockFLOWS:** This is the theoretical work that is behind the generator of workflows that is being implemented by my colleagues.

RockFLOWS: Workflows as Software Product Line

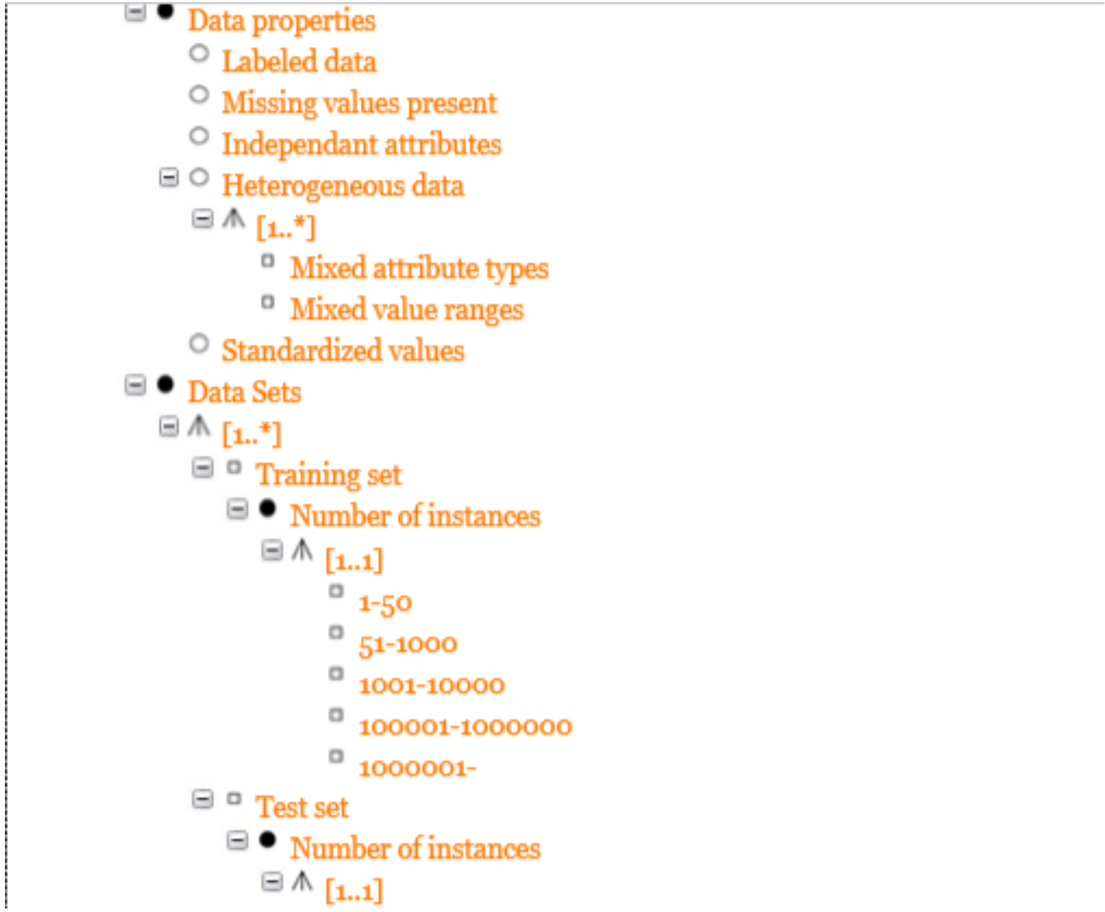
User Interface



You will be able to create a data mining workflow.

- You will have to define what your objectives are for this problem. ✓
  - + You will have to define the main objective for your Data Mining problem. ✓
  - + The secondary user's objectives define additional properties and the output of t
- You will have to describe the data you want to use. ✓
  - + You will have to define the type of data that is represented in your dataset. ✓
  - + You will have to describe some specific properties on your data. ✓
  - + You will have to tell what kind of dataset you have. ✓
- Properties of the generated workflow when it will be executed ✓

Feature Model



```
● Data properties
    ○ Labeled data
    ○ Missing values present
    ○ Independant attributes
    ○ Heterogeneous data
        [1..*]
            Mixed attribute types
            Mixed value ranges
    ○ Standardized values
● Data Sets
    [1..*]
        Training set
            ● Number of instances
                [1..1]
                    1-50
                    51-1000
                    1001-10000
                    100001-1000000
                    1000001-
        Test set
            ● Number of instances
                [1..1]
```

Future Work

- **Data pattern: are we in the right way?** If we find at least one data pattern where the same classifier is always the best (avg rank = 1) => Yes

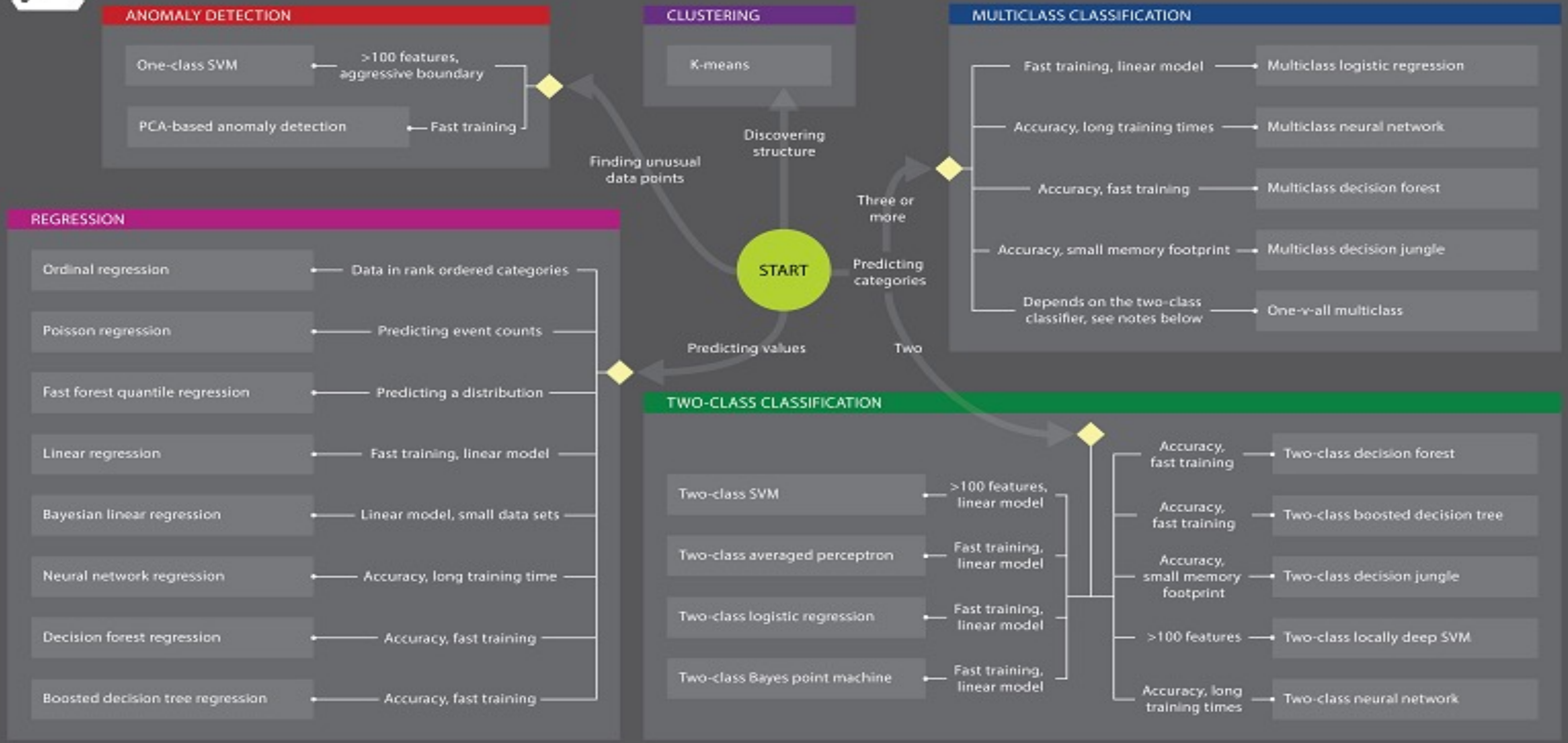
- **Other ways:** If the data patterns do not lead to any success, we need to find another strategy to foresee the behavior of the workflow without doing the evaluation.

References

- [1]: Wolpert, David (1996), “*The Lack of A Priori Distinctions between Learning Algorithms*”, Neural Computation, pp. 1341-1390.
- [2]: Weka: « *http://www.cs.waikato.ac.nz/ml/weka/* »
- [3]: R: « *https://www.r-project.org/* »
- [4]: UCI: « *https://archive.ics.uci.edu/ml/datasets.html* »
- [5]: Ian Witten, Eibe Frank, Mark Hall: « *Data Mining: Practical Machine Learning Tools and Techniques* » ISBN: 978-0-12-374856-0
- [6]: Guojun Gan, Chaoqun Ma, Jianhong Wu: « *Data Clustering: Theory, Algorithms, and Applications* » ISBN-13: 978-0898716238
- [7]: Data Mining course: « *http://bias.csr.unibo.it/golfarelli/DataMining/* »

Microsoft Azure Machine Learning: Algorithm Cheat Sheet

This cheat sheet helps you choose the best Azure Machine Learning Studio algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the question you're trying to answer.



START

Finding unusual data points → ANOMALY DETECTION

- >100 features, aggressive boundary → One-class SVM
- Fast training → PCA-based anomaly detection

Discovering structure → CLUSTERING

- K-means

Predicting categories — Three or more → MULTICLASS CLASSIFICATION

- Fast training, linear model → Multiclass logistic regression
- Accuracy, long training times → Multiclass neural network
- Accuracy, fast training → Multiclass decision forest
- Accuracy, small memory footprint → Multiclass decision jungle
- Depends on the two-class classifier, see notes below → One-v-all multiclass

Predicting categories — Two → TWO-CLASS CLASSIFICATION

- >100 features, linear model → Two-class SVM
- Fast training, linear model → Two-class averaged perceptron
- Fast training, linear model → Two-class logistic regression
- Fast training, linear model → Two-class Bayes point machine
- Accuracy, fast training → Two-class decision forest
- Accuracy, fast training → Two-class boosted decision tree
- Accuracy, small memory footprint → Two-class decision jungle
- >100 features → Two-class locally deep SVM
- Accuracy, long training times → Two-class neural network

Predicting values → REGRESSION

- Data in rank ordered categories → Ordinal regression
- Predicting event counts → Poisson regression
- Predicting a distribution → Fast forest quantile regression
- Fast training, linear model → Linear regression
- Linear model, small data sets → Bayesian linear regression
- Accuracy, long training time → Neural network regression
- Accuracy, fast training → Decision forest regression
- Accuracy, fast training → Boosted decision tree regression

Created by the Azure Machine Learning Team   Email: AzurePoster@microsoft.com   Download this poster: http://aka.ms/MLCheatSheet   Microsoft

Motivations

- **Motivation #1**: Create a generator and executor of data mining workflow that helps the user in selecting the best workflow w.r.t. its input data

- **Valid data mining workflows**: the constraints of the feature model allow the generation of only valid workflows.

- **Example:** If an algorithm works only on numeric data, we can’t apply it on nominal data.

Motivations

- **Motivation #2**: The generator is intended for both data mining user experts and not expert ones.

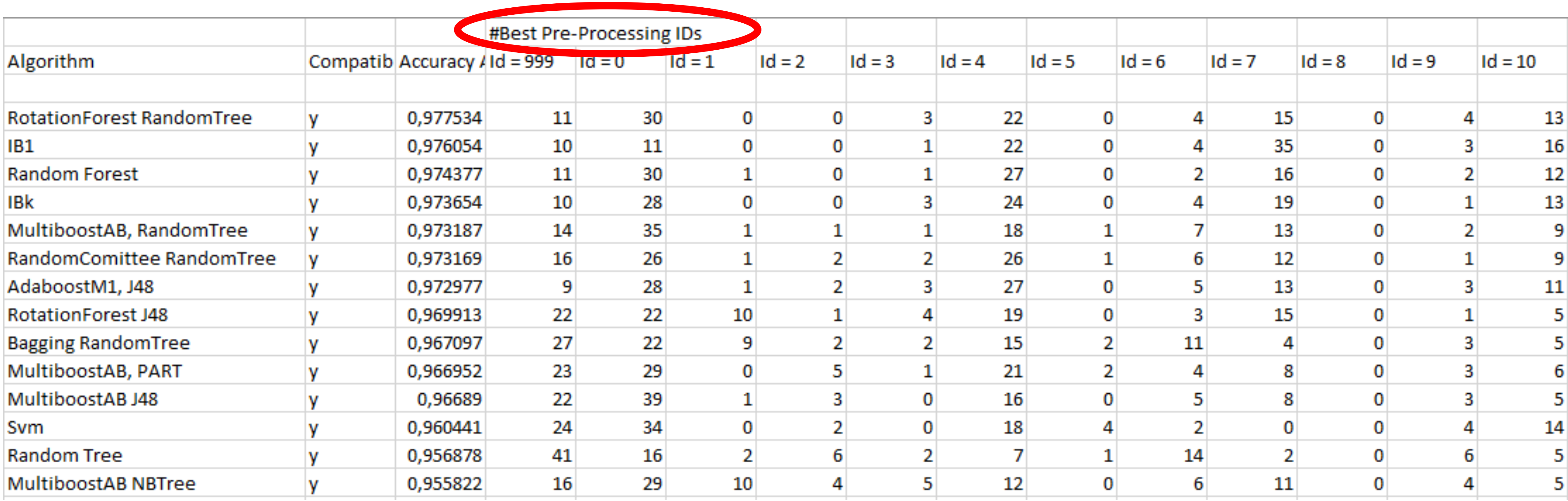
- **Generic vs Specific:** If the user knows which components to use, he/she can select them manually, otherwise we want to define a strategy to select the best ones automatically.

- **Example:** Do you know which data mining algorithm to use?
  - Yes => we build the workflow with the specified algorithm (specific)
  - No => we can compare all the suitable algorithms (generic)

Conclusions: Pre-processing impact

- **% of paper pre-processing = best pre-processing:**
  - 4-Folds: 10.94%,
  - 10-Folds: 26,04%,

- **Distribution of best pre-processing:** It is equally distributed among the techniques of pre-processing, there is not in general a clear winner



			#Best Pre-Processing IDs											
Algorithm	Compatib	Accuracy	Id = 999	Id = 0	Id = 1	Id = 2	Id = 3	Id = 4	Id = 5	Id = 6	Id = 7	Id = 8	Id = 9	Id = 10
RotationForest RandomTree	y	0,977534	11	30	0	0	3	22	0	4	15	0	4	13
IB1	y	0,976054	10	11	0	0	1	22	0	4	35	0	3	16
Random Forest	y	0,974377	11	30	1	0	1	27	0	2	16	0	2	12
IBk	y	0,973654	10	28	0	0	3	24	0	4	19	0	1	13
MultiboostAB, RandomTree	y	0,973187	14	35	1	1	1	18	1	7	13	0	2	9
RandomComittee RandomTree	y	0,973169	16	26	1	2	2	26	1	6	12	0	1	9
AdaboostM1, J48	y	0,972977	9	28	1	2	3	27	0	5	13	0	3	11
RotationForest J48	y	0,969913	22	22	10	1	4	19	0	3	15	0	1	5
Bagging RandomTree	y	0,967097	27	22	9	2	2	15	2	11	4	0	3	5
MultiboostAB, PART	y	0,966952	23	29	0	5	1	21	2	4	8	0	3	6
MultiboostAB J48	y	0,96689	22	39	1	3	0	16	0	5	8	0	3	5
Svm	y	0,960441	24	34	0	2	0	18	4	2	0	0	4	14
Random Tree	y	0,956878	41	16	2	6	2	7	1	14	2	0	6	5
MultiboostAB NBTree	y	0,955822	16	29	10	4	5	12	0	6	11	0	4	5

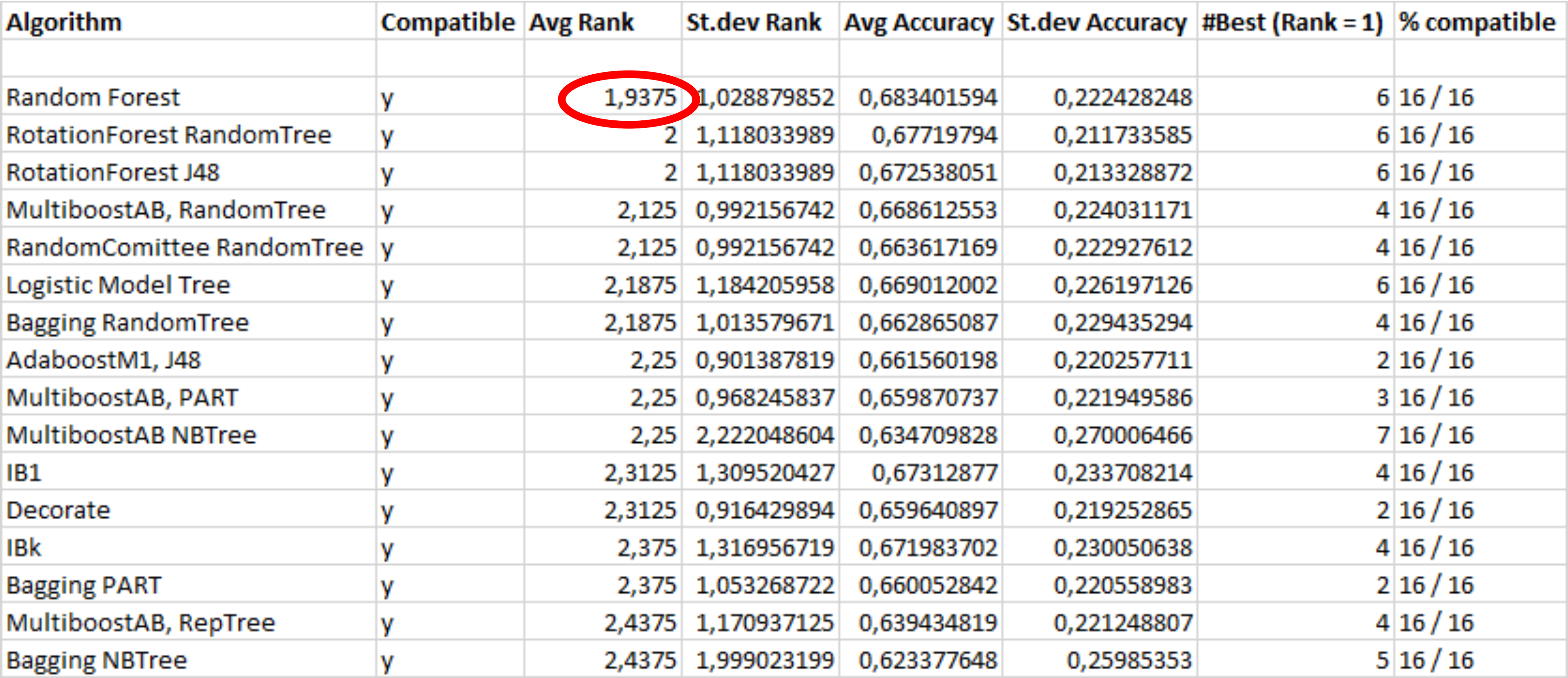
Contribution: Data Pattern

- **How to define a pattern?:** This is the hardest part. The dataset complexity itself is hard to define
  (Tin Kam Ho and Mitra Basu. “*Complexity measures of supervised classification problems”.* IEEE Trans. on Pattern Analysis and Machine Intelligence, 24(3):289–300, 2002)

Data Patterns: Failed Experiment

- **Example:** We take the results from all the datasets that have the following properties:
  - No missing values
  - Attributes type: only numeric
  - 0 < #attributes < 10
  - 20 < #training instances < 1000
  - #classes > 2
  - Attributes selection: done

Data Patterns: Failed Experiments



Algorithm	Compatible	Avg Rank	St.dev Rank	Avg Accuracy	St.dev Accuracy	#Best (Rank = 1)	% compatible
Random Forest	y	1,9375	1,028879852	0,683401594	0,222428248	6	16 / 16
RotationForest RandomTree	y	2	1,118033989	0,67719794	0,211733585	6	16 / 16
RotationForest J48	y	2	1,118033989	0,672538051	0,213328872	6	16 / 16
MultiboostAB, RandomTree	y	2,125	0,992156742	0,668612553	0,224031171	4	16 / 16
RandomComittee RandomTree	y	2,125	0,992156742	0,663617169	0,222927612	4	16 / 16
Logistic Model Tree	y	2,1875	1,184205958	0,669012002	0,226197126	6	16 / 16
Bagging RandomTree	y	2,1875	1,013579671	0,662865087	0,229435294	4	16 / 16
AdaboostM1, J48	y	2,25	0,901387819	0,661560198	0,220257711	2	16 / 16
MultiboostAB, PART	y	2,25	0,968245837	0,659870737	0,221949586	3	16 / 16
MultiboostAB NBTree	y	2,25	2,222048604	0,634709828	0,270006466	7	16 / 16
IB1	y	2,3125	1,309520427	0,67312877	0,233708214	4	16 / 16
Decorate	y	2,3125	0,916429894	0,659640897	0,219252865	2	16 / 16
IBk	y	2,375	1,316956719	0,671983702	0,230050638	4	16 / 16
Bagging PART	y	2,375	1,053268722	0,660052842	0,220558983	2	16 / 16
MultiboostAB, RepTree	y	2,4375	1,170937125	0,639434819	0,221248807	4	16 / 16
Bagging NBTree	y	2,4375	1,999023199	0,623377648	0,25985353	5	16 / 16

Computation of Rank

- **Rank:** value that indicate how much the classifier is good w.r.t. the specific performance (accuracy, time, RAM)

- **Algorithm:**
  1. Sort the classifiers from the best performance value to the worst one
  2. Check the statistical difference among the consecutive classifiers and set the same rank to the ones who aren’t different, rank + 1 to the different ones
  3. Upper cycle: until convergence, take the classifiers with the same rank. Then, check difference between the 1st and the 3rd, the 1st and the 4th, … If some is different, they have a different rank
  4. Lower cycle: same as upper cycle, but by starting from the worst one.