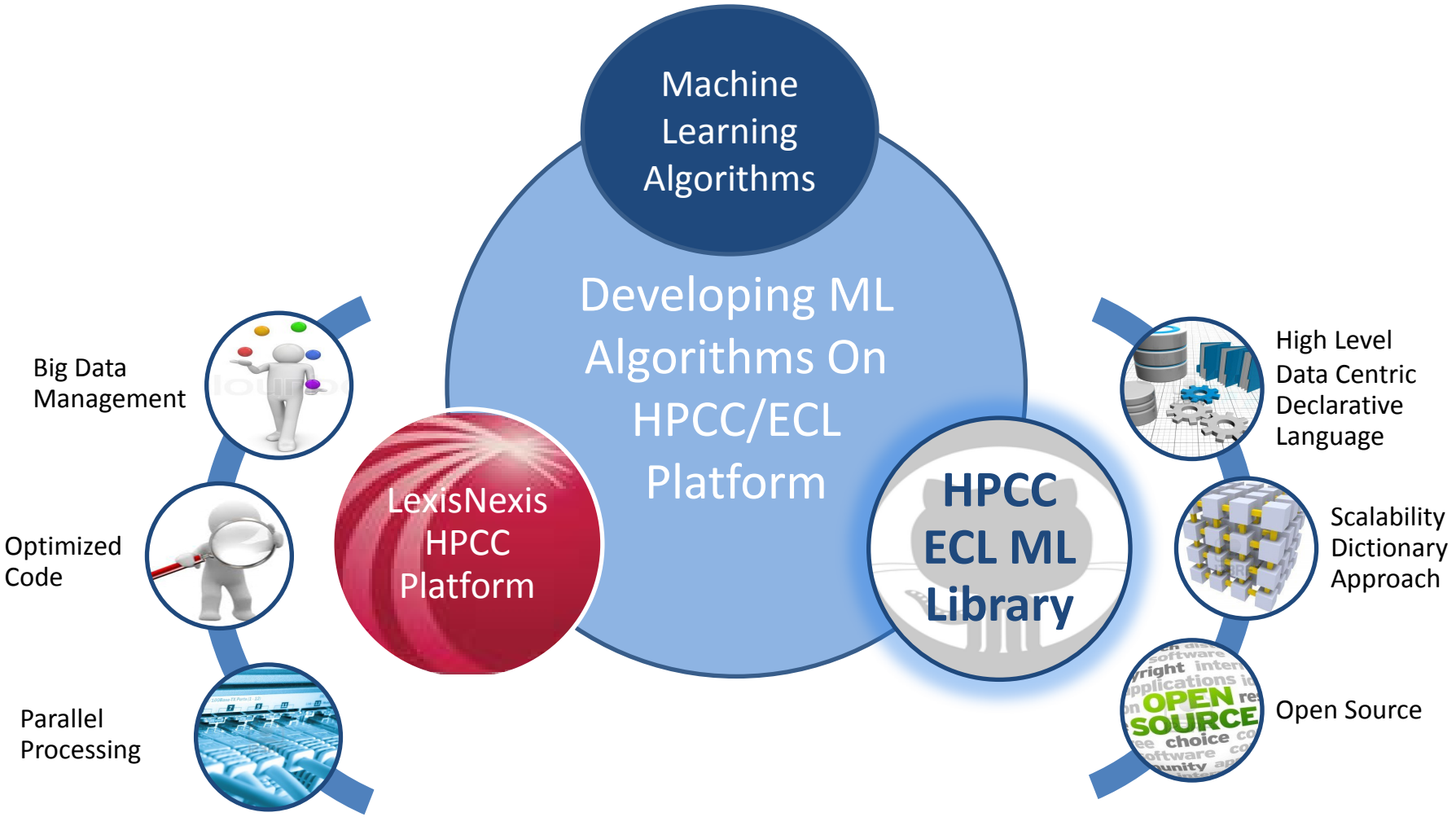
Developing Machine Learning Algorithms on HPCC/ECL Platform

Industry/University Cooperative Research

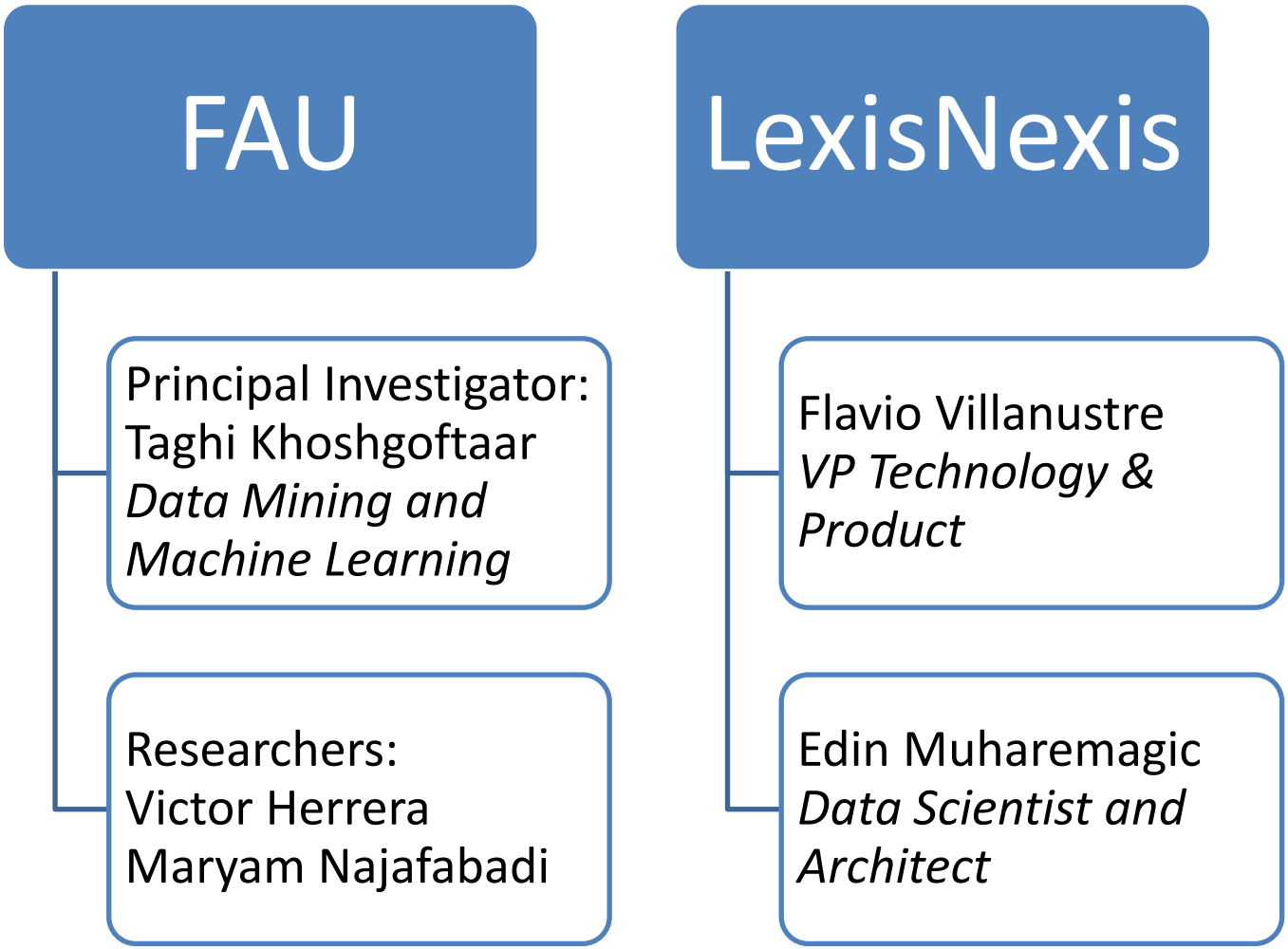
LexisNexis/Florida Atlantic University

Victor Herrera – FAU – Fall 2014

LexisNexis/Florida Atlantic University Cooperative Research



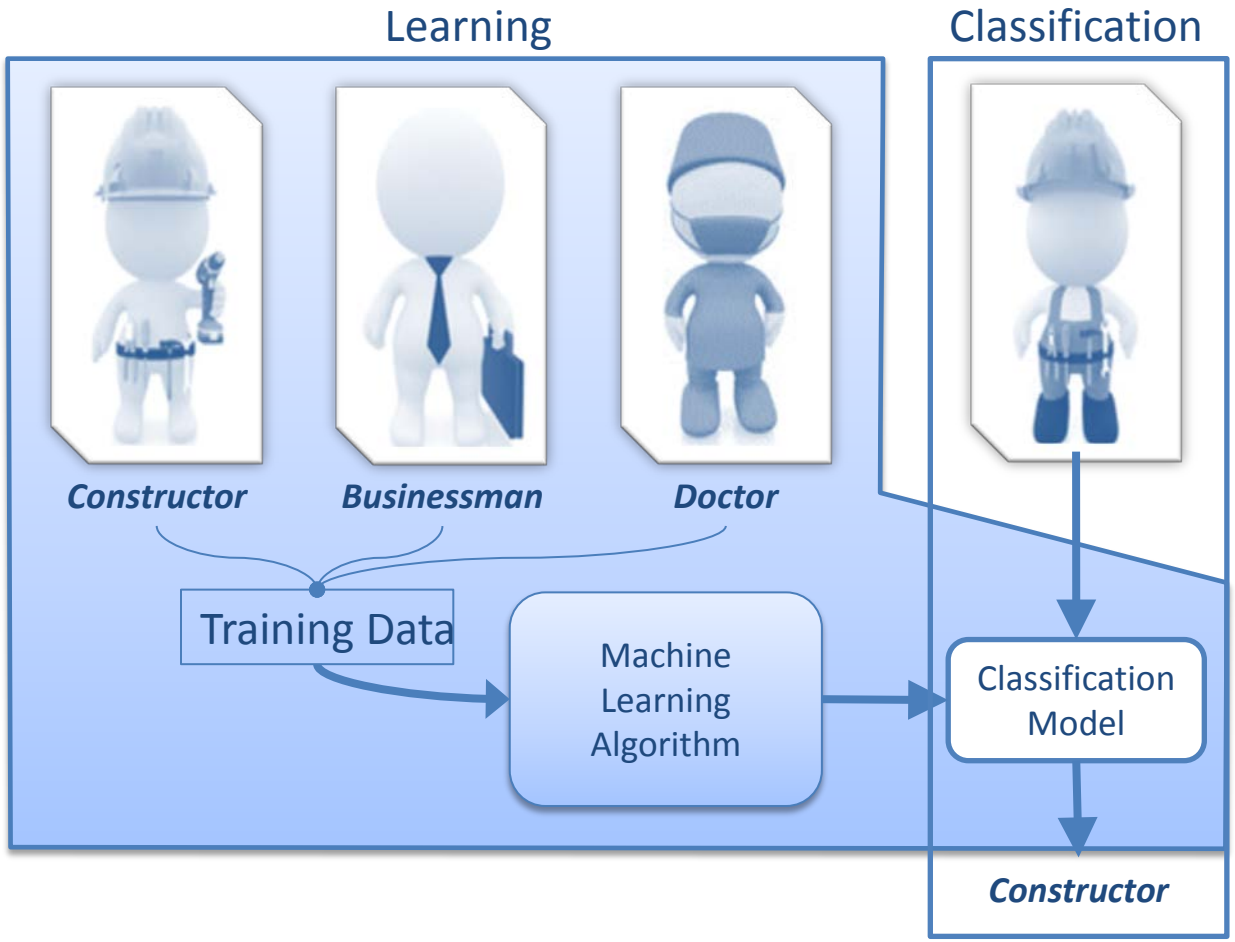
Project Team



Project Scope

Implement Machine Learning Algorithms in HPCC ECL-ML library, focusing on:

- Supervised Learning for Classification
- Big Data
- Parallel Computing



ECL-ML Learning Library

Project Starting Point
April 2012

ECL-ML Open Source Blueprint

Machine Learning Documentation

HPCC-ECL Training

HPCC-ECL Documentation Forums

ML-ECL Library Contributor

- Commit
- Pull Request

Review Code

- Comment
- Merge

ECL – ML Library

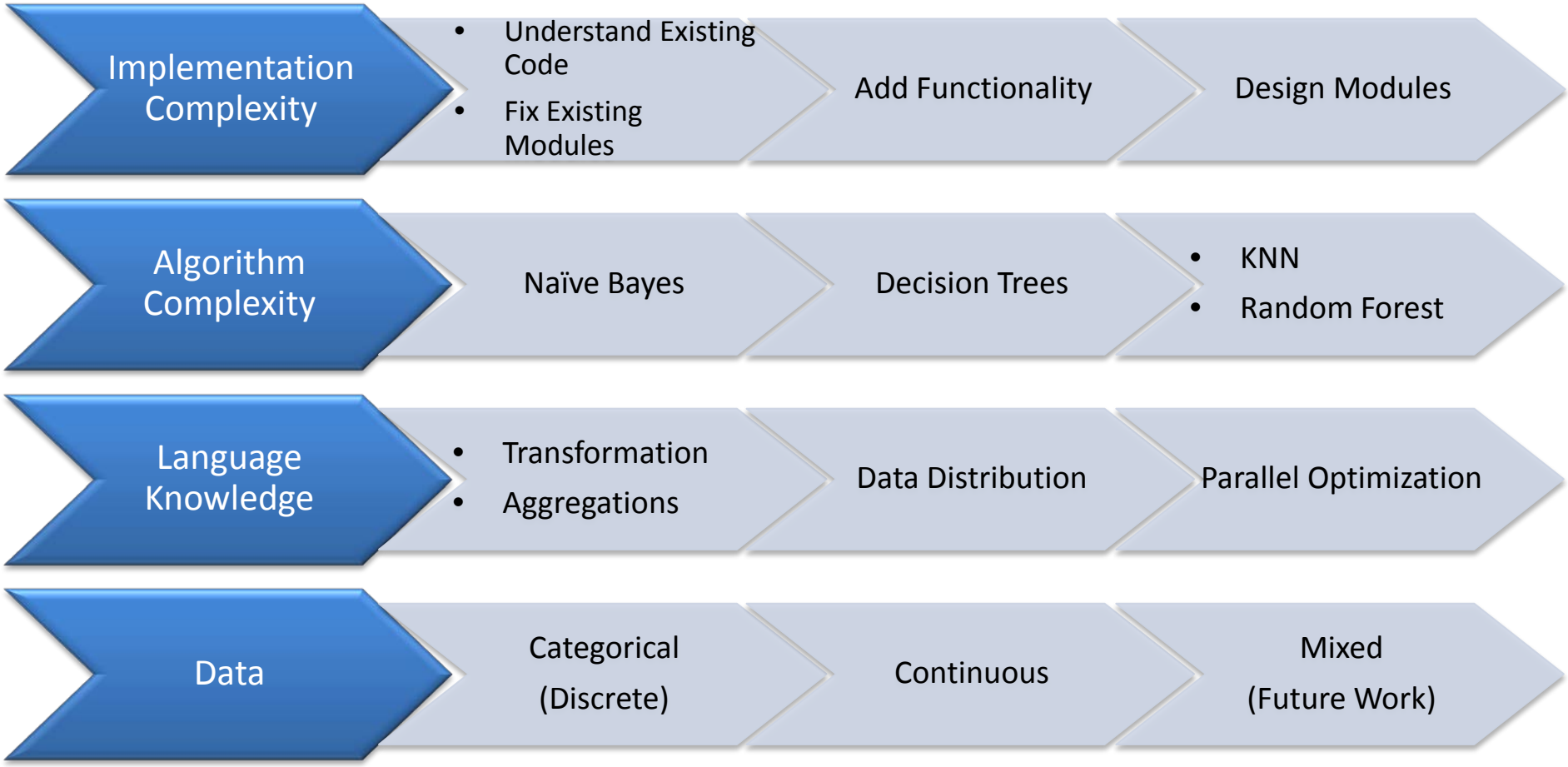
Process comments

ML-ECL Library Administrator

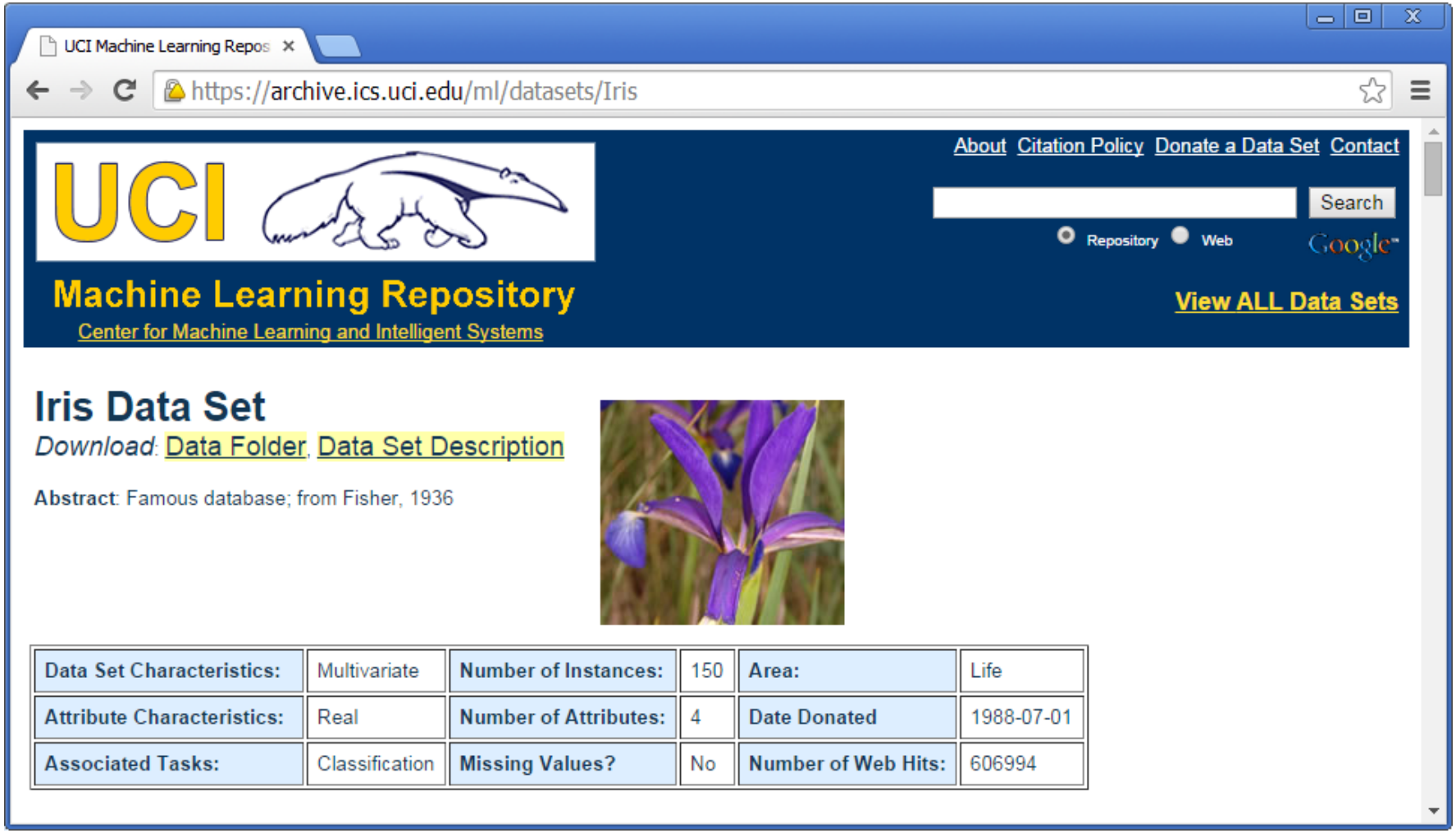
- Use
- Contribute

Community

Experience Progress



Example: Classifying Iris-Setosa
Decision Tree vs. Random Forest

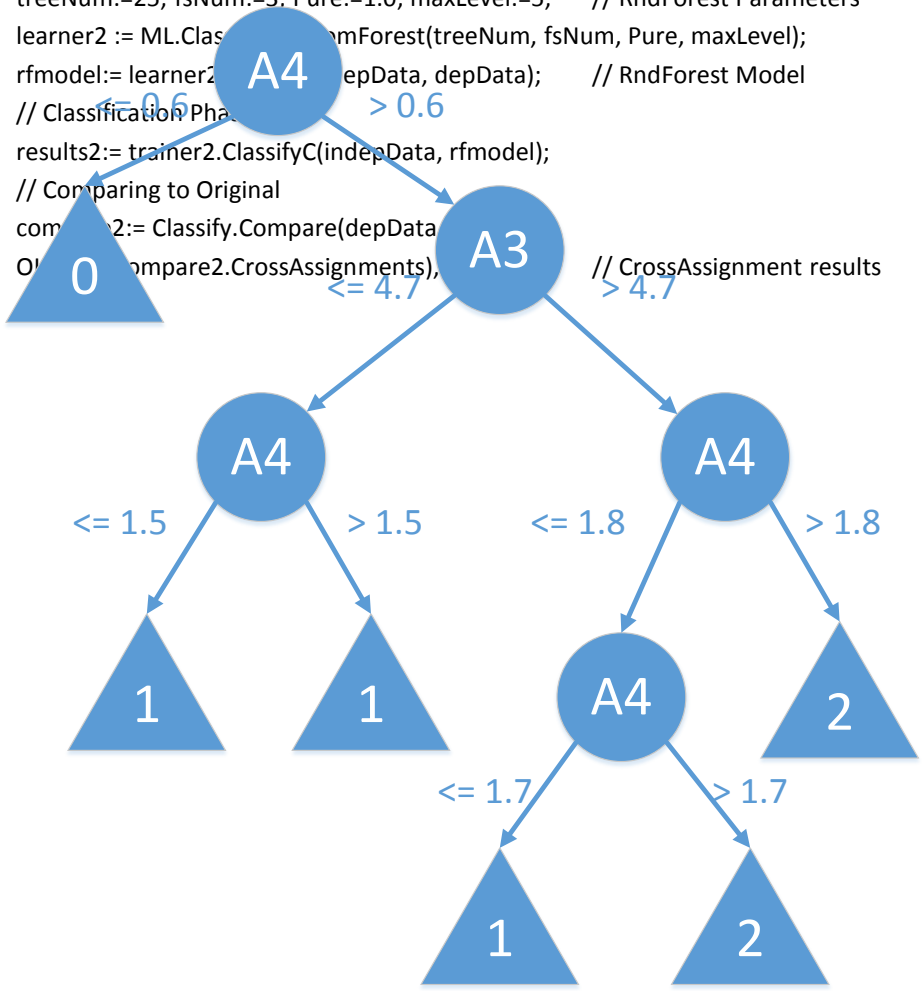


Example: Classifying Iris-Setosa
Decision Tree vs. Random Forest

```
// Learning Phase
minNumObj:= 2;    maxLevel := 5;                    // DecTree Parameters
learner1:= ML.Classify.DecisionTree.C45Binary(minNumObj, maxLevel);
tmodel:= learner1.LearnC(indepData, depData);       // DecTree Model
// Classification Phase
results1:= learner1.ClassifyC(indepData, tmodel);
// Comparing to Original
compare1:= Classify.Compare(depData, results1);
OUTPUT(compare1.CrossAssignments), ALL);            // CrossAssignment results
```

```
// Learning Phase
treeNum:=25; fsNum:=3; Pure:=1.0; maxLevel:=5;      // RndForest Parameters
learner2 := ML.Clas                                 mForest(treeNum, fsNum, Pure, maxLevel);
rfmodel:= learner2                                  epData, depData);       // RndForest Model
// Classification Pha
results2:= trainer2.ClassifyC(indepData, rfmodel);
// Comparing to Original
com                                                 2:= Classify.Compare(depData
O                                                   mpare2.CrossAssignments),   // CrossAssignment results
```

##	node_id	level	number	value	high_fork	new_node_id
1	1	1	4	0.6	0	2
2	1	1	4	0.6	1	3
3	2	2	0	0.0	0	0
4	3	2	3	4.7	0	4
5	3	2	3	4.7	1	5
6	4	3	4	1.5	0	6
7	4	3	4	1.5	1	7
8	5	3	4	1.8	0	8
9	5	3	4	1.8	1	9
10	6	4	0	1.0	0	0
11	7	4	0	1.0	1	0
12	8	4	4	1.7	0	10
13	8	4	4	1.7	1	11
14	9	4	0	2.0	1	0
15	10	5	0	1.0	0	0
16	11	5	0	2.0	0	0



Example: Classifying Iris-Setosa
Decision Tree vs. Random Forest

```
// Learning Phase
minNumObj:= 2;    maxLevel := 5;                    // DecTree Parameters
learner1:= ML.Classify.DecisionTree.C45Binary(minNumObj, maxLevel);
tmodel:= learner1.LearnC(indepData, depData);       // DecTree Model
// Classification Phase
results1:= learner1.ClassifyC(indepData, tmodel);
// Comparing to Original
compare1:= Classify.Compare(depData, results1);
OUTPUT(compare1.CrossAssignments), ALL);            // CrossAssignment results
```

```
// Learning Phase
treeNum:=25; fsNum:=3; Pure:=1.0; maxLevel:=5;      // RndForest Parameters
learner2 := ML.Classify.RandomForest(treeNum, fsNum, Pure, maxLevel);
rfmodel:= learner2.LearnC(indepData, depData);      // RndForest Model
// Classification Phase
results2:= trainer2.ClassifyC(indepData, rfmodel);
// Comparing to Original
compare2:= Classify.Compare(depData, results2);
OUTPUT(compare2.CrossAssignments), ALL);            // CrossAssignment results
```

##	classifier	c_actual	c_modeled	cnt
1	1	0	0	50
2	1	0	1	0
3	1	0	2	0
4	1	1	0	0
5	1	1	1	49
6	1	1	2	1
7	1	2	0	0
8	1	2	1	2
9	1	2	2	48

##	classifier	c_actual	c_modeled	cnt
1	1	0	0	50
2	1	0	1	0
3	1	0	2	0
4	1	1	0	0
5	1	1	1	50
6	1	1	2	0
7	1	2	0	0
8	1	2	1	0
9	1	2	2	50

Project Contribution

Supervised Learning

Naïve Bayes	Decision Tree	Case Based	Ensemble	Classification
Maximum Likelihood	Top-Down Induction	KD-Tree Partitioning	Data Sampling Tree Bagging	Class Probability
Probabilistic Model	Recursive Data Partitioning	Nearest Neighbor Search	F. Sampling Rdn Subspace	Missing Values Classification
	Decision Tree Model	No Model – Lazy KNN Algorithm	Ensemble Tree Model	Area Under ROC

Overall Summary

- Development of Machine Learning Algorithms in ECL Language, focused on the implementation of Classification Algorithms, Big Data and Parallel Processing.
- Added functionality to Naïve Bayes Classifier and implemented Decision Trees, KNN and Random Forest Classifiers into ECL-ML library.
- Currently working on Big Data Learning & Evaluation :
  - Experiment design and implementation of a Big Data CASE Study
  - Analysis of Results.
  - Classifiers enhancements based upon analysis of results.

Thank you.