"INTRODUÇÃO AO APRENDIZADO DE MÁQUINA AUTOMATIZADO (AUTOML)"

Rafael Gomes Mantovani

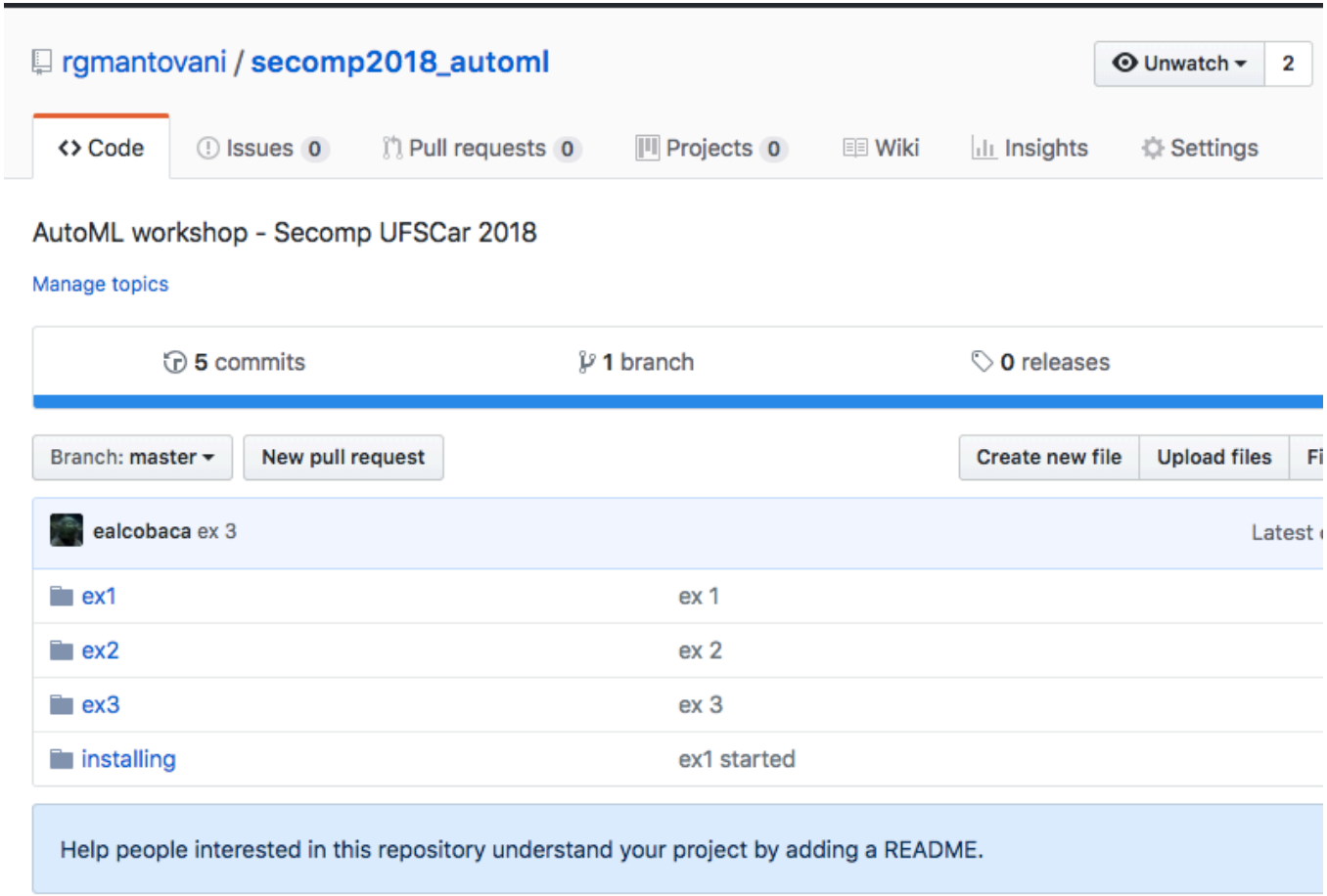
Universidade de São Paulo (USP), Brazil

São Carlos - SP
Setembro, 2018

Universidade de São Paulo (USP), São Carlos, Brasil
Institute de Ciências Matemáticas e de Computação - ICMC

Antes de começarmos ...

- github.com/rgmantovani/secomp2018_automl



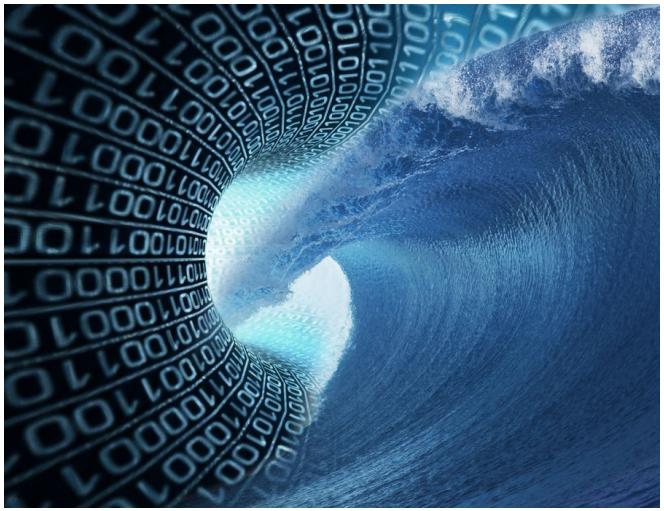
Roteiro

1. Breve Introdução a *Machine Learning* (ML)
2. OpenML / mlr / mlrMBO / RStudio
3. Exemplos / Benchmark
4. Automated Machine Learning (AutoML)
5. Literatura em AutoML
6. Exemplos / AutoML / CASH
7. Considerações finais

Roteiro

1. **Breve Introdução a *Machine Learning* (ML)**
2. **OpenML / mlr / mlrMBO / RStudio**
3. **Exemplos / Benchmark**
4. **Automated Machine Learning (AutoML)**
5. **Literatura em AutoML**
6. **Exemplos / AutoML / CASH**
7. **Considerações finais**

Contextualização

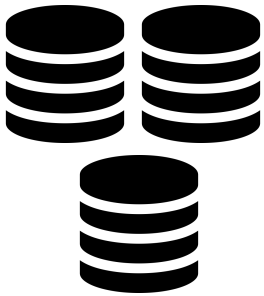


poucos dados

imensa quantidade de dados (big data)

Contextualização

Pessoas



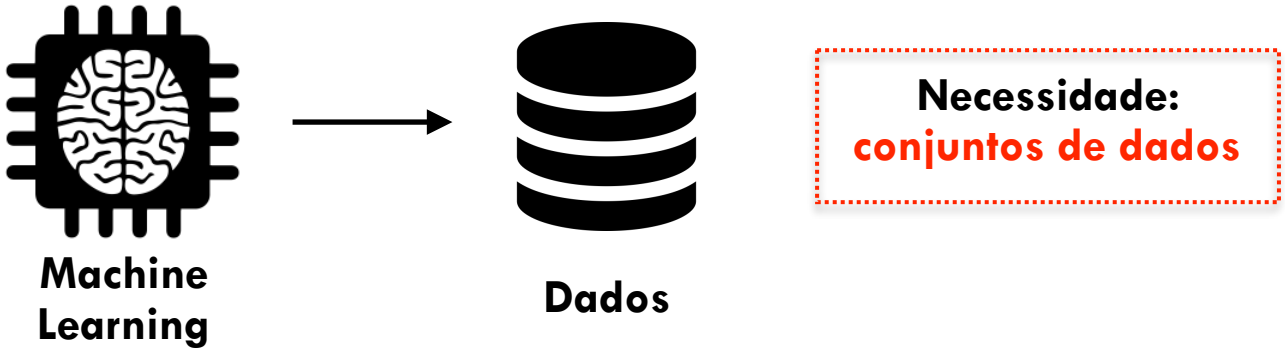
Dados

Dispositivos

- Dados são **continuamente**:
  - gerados, coletados, processados e transmitidos

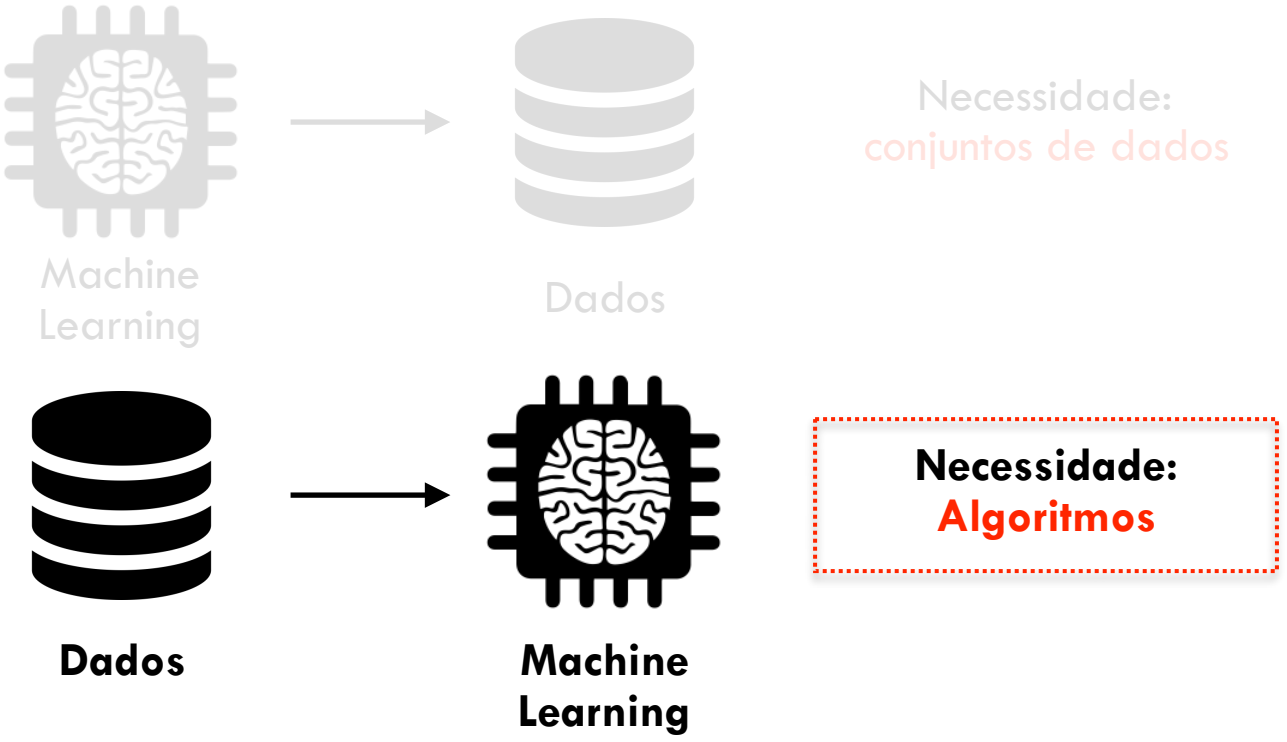
Contextualização

- Mudança de realidade

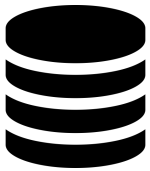


Contextualização

- Mudança de realidade



Aprendizado de Máquina



Dados

Algoritmo AM

Conhecimento

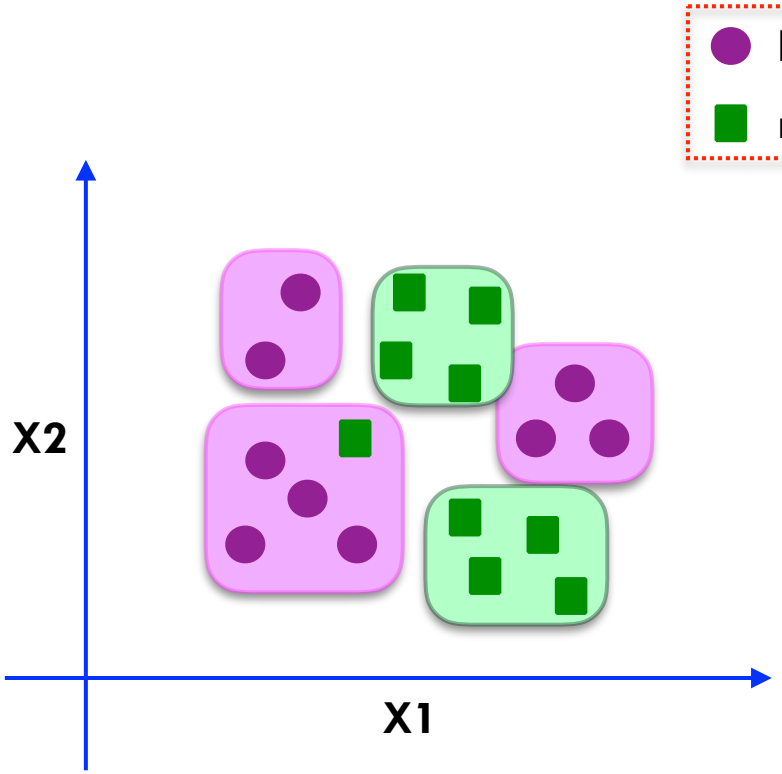
Solução

- Inteligência Artificial
- Automatiza a construção de modelos para solucionar problemas!

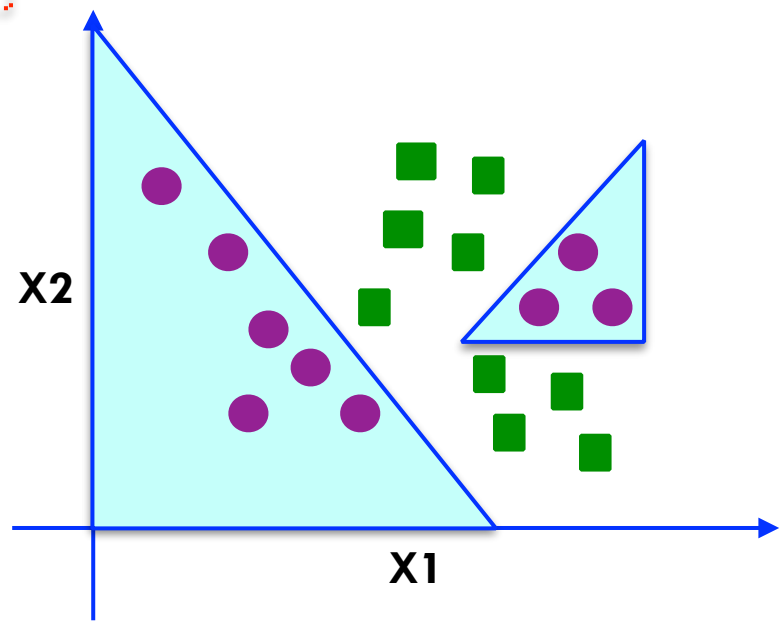
Aprendizado de Máquina

- Tarefas:
  - Reconhecimento de imagem/som
  - Recomendação de produtos (filmes, itens, …)
  - Detectar fraude (banco, acesso)
  - Detecção de cancer, e diversas doenças
  - Navegação autônoma
  - etc …

Aprendizado de Máquina



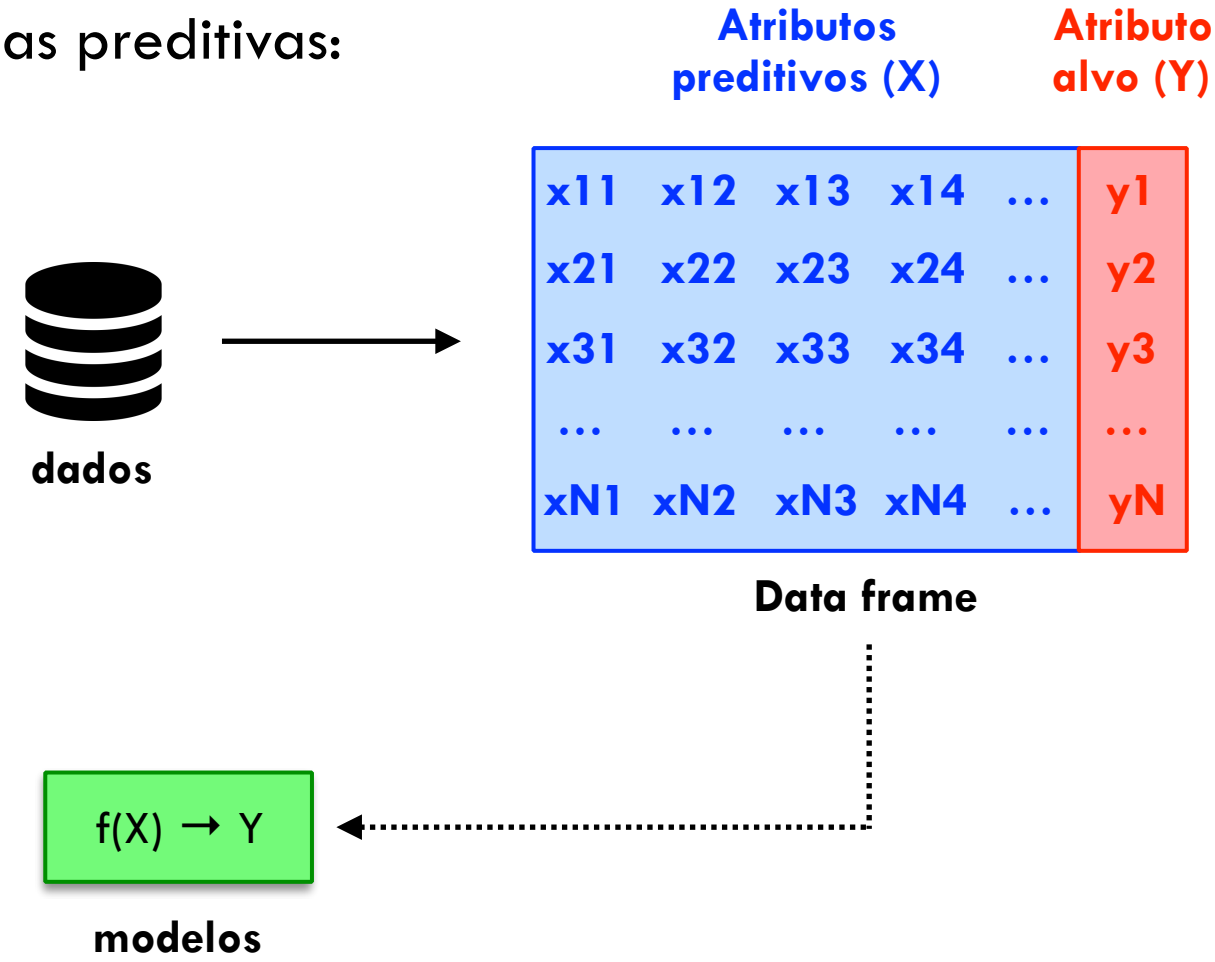
Tarefa descritiva
(Clustering)



Tarefa preditiva
(Classificação/ Regressão)

Dataset

- Tarefas preditivas:



Dataset

- Tarefas preditivas

Iris

Atributos preditivos (X)

Atributo alvo (Y)

	sepallength	sepalwidth	petallength	petalwidth	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-setosa
10	5.4	3.7	1.5	0.2	Iris-setosa
11	4.8	3.4	1.6	0.2	Iris-setosa
12	4.8	3.0	1.4	0.1	Iris-setosa
13	4.3	3.0	1.1	0.1	Iris-setosa
14	5.8	4.0	1.2	0.2	Iris-setosa
15	5.7	4.4	1.5	0.4	Iris-setosa

Showing 1 to 16 of 150 entries

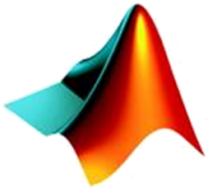
Aprendizado de Máquina

- Quantos algoritmos de ML existem?

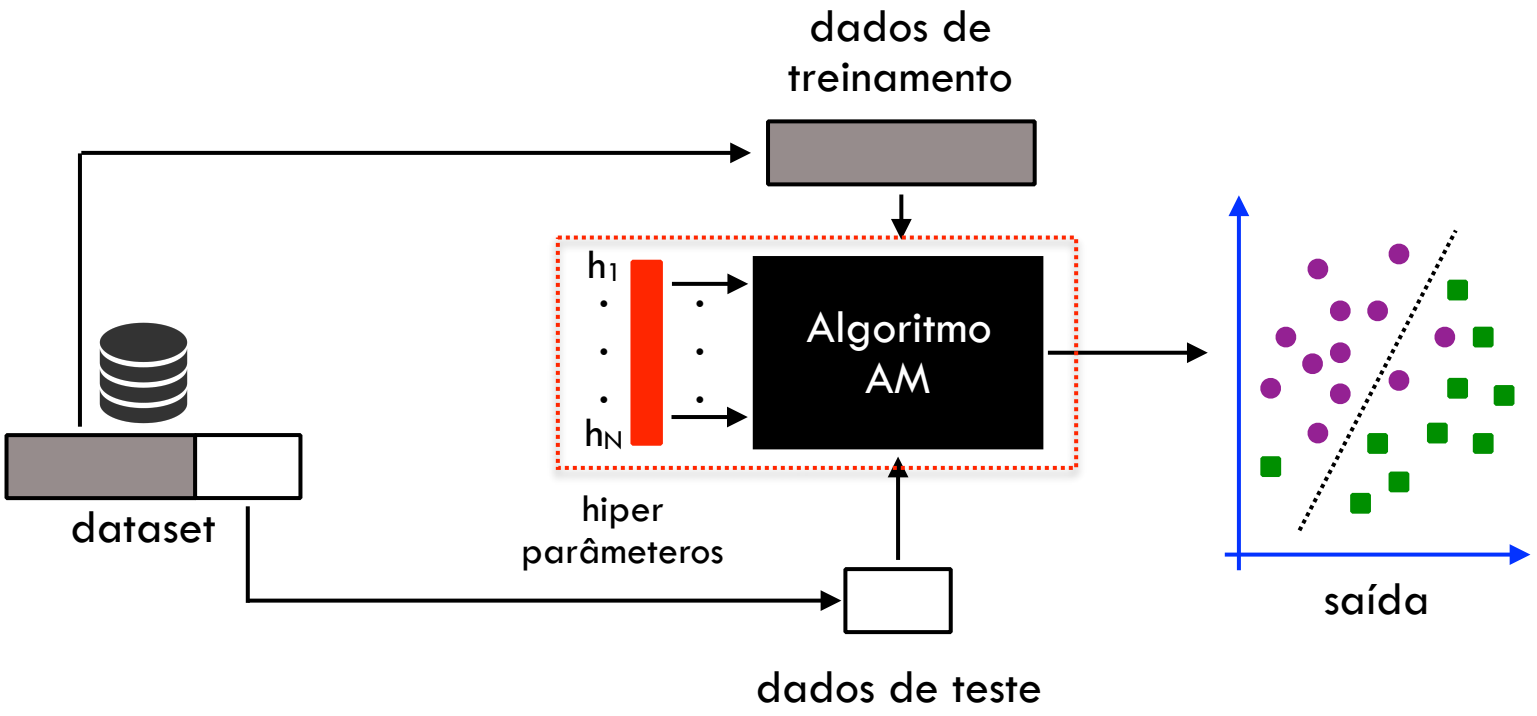
Aprendizado de Máquina

- Quantos algoritmos de ML existem?

...



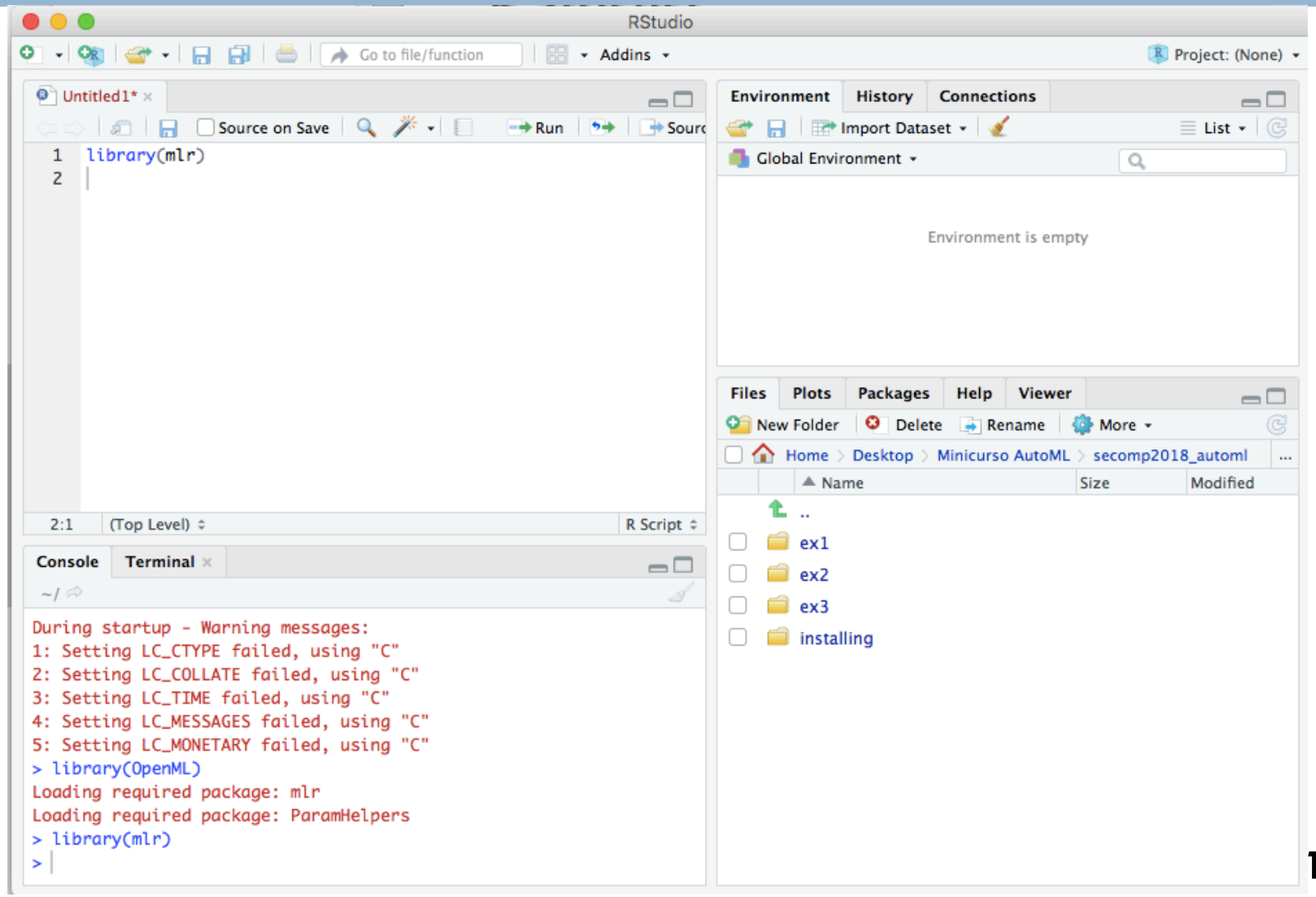
Aprendizado de Máquina



Roteiro

1. Breve Introdução a *Machine Learning* (ML)
2. OpenML / mlr / mlrMBO / RStudio
3. Exemplos / Benchmark
4. Automated Machine Learning (AutoML)
5. Literatura em AutoML
6. Exemplos / AutoML / CASH
7. Considerações finais

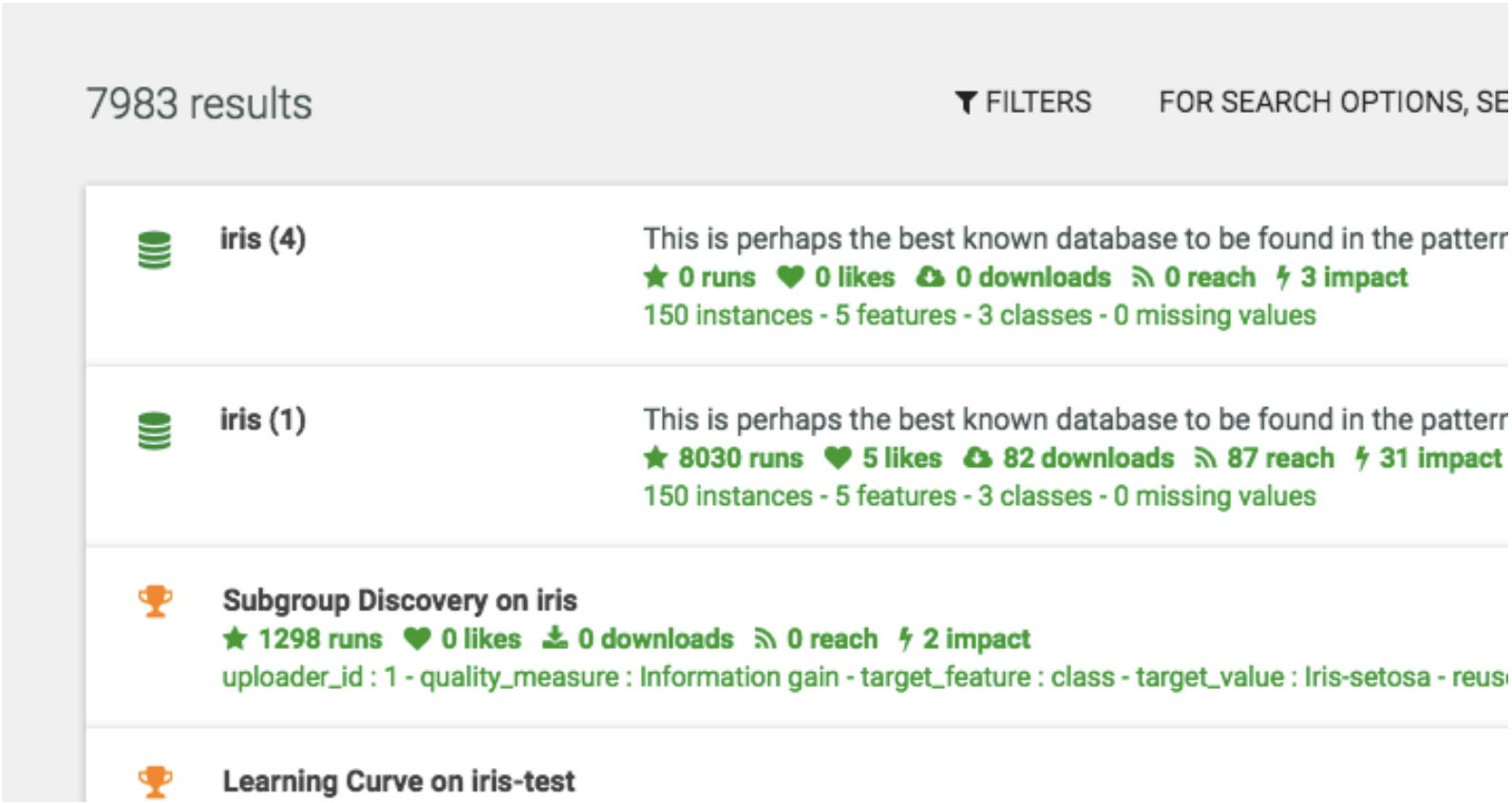
RStudio


RStudio
Go to file/function
Addins
Project: (None)
Untitled1*
Source on Save
Run
1 library(mlr)
2
2:1
(Top Level)
R Script
Console
Terminal
~/
During startup - Warning messages:
1: Setting LC_CTYPE failed, using "C"
2: Setting LC_COLLATE failed, using "C"
3: Setting LC_TIME failed, using "C"
4: Setting LC_MESSAGES failed, using "C"
5: Setting LC_MONETARY failed, using "C"
> library(OpenML)
Loading required package: mlr
Loading required package: ParamHelpers
> library(mlr)
>
Environment
History
Connections
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home
Desktop
Minicurso AutoML
secomp2018_automl
Name
Size
Modified
..
ex1
ex2
ex3
installing

OpenML


Search
OpenML
beta
Machine learning, better, together
20072
data sets
Find or add **data** to analyse
67888
tasks
Download or create scientific **tasks**
6092
flows
Find or add data analysis **flows**
9012316
runs
Upload and explore all **results** online.

OpenML


7983 results
FILTERS
FOR SEARCH OPTIONS, SE
iris (4)
This is perhaps the best known database to be found in the patterr
0 runs 0 likes 0 downloads 0 reach 3 impact
150 instances - 5 features - 3 classes - 0 missing values
iris (1)
This is perhaps the best known database to be found in the patterr
8030 runs 5 likes 82 downloads 87 reach 31 impact
150 instances - 5 features - 3 classes - 0 missing values
Subgroup Discovery on iris
1298 runs 0 likes 0 downloads 0 reach 2 impact
uploader_id : 1 - quality_measure : Information gain - target_feature : class - target_value : Iris-setosa - reus
Learning Curve on iris-test

iris

ARFF CSV JSON XML RDF V. 1

active ARFF Publicly available Visibility: public Uploaded 06-04-2014 by Jan van Rijn

5 likes downloaded by 82 people , 104 total downloads 0 issues 0 downvotes

study_1 study_25 study_4 study_41 study_50 study_52 study_7 study_86 study_88 study_89 uci + Add tag

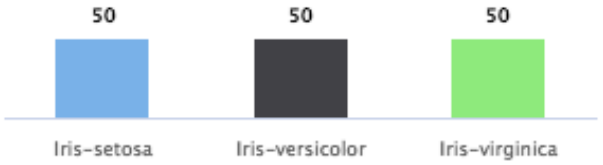
Help us complete this description → Edit

Author: R.A. Fisher
Source: UCI - 1936 - Donated by Michael Marshall
Please cite:

Iris Plants Database
This is perhaps the best known database to be found in the pattern recognition literature. Fisher's paper is a classic in the field and is referenced frequently to this day. (See Duda & Hart, for example.) The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly

▾ Show all

5 features

class **(target)**	nominal	3 unique values 0 missing	
sepallength	numeric	35 unique values 0 missing	

mlr - Machine Learning in R

Machine Learning in R

- CRAN release site
- Detailed Tutorial: Online as HTML
- mlr cheatsheet
- Install the development version

```
devtools::install_github("mlr-org/mlr")
```

- Further installation instructions
- Ask a question about mlr on Stackoverflow
- We are on Slack (Request invitation: code{at}jakob-r.de)
- We have a blog on mlr
- A list of possible enhancements to mlr is available on the wiki - contributors welcome!
- We are in the top 20 of the most starred R packages on Github, as reported by metacran.

mlr - Machine Learning in R

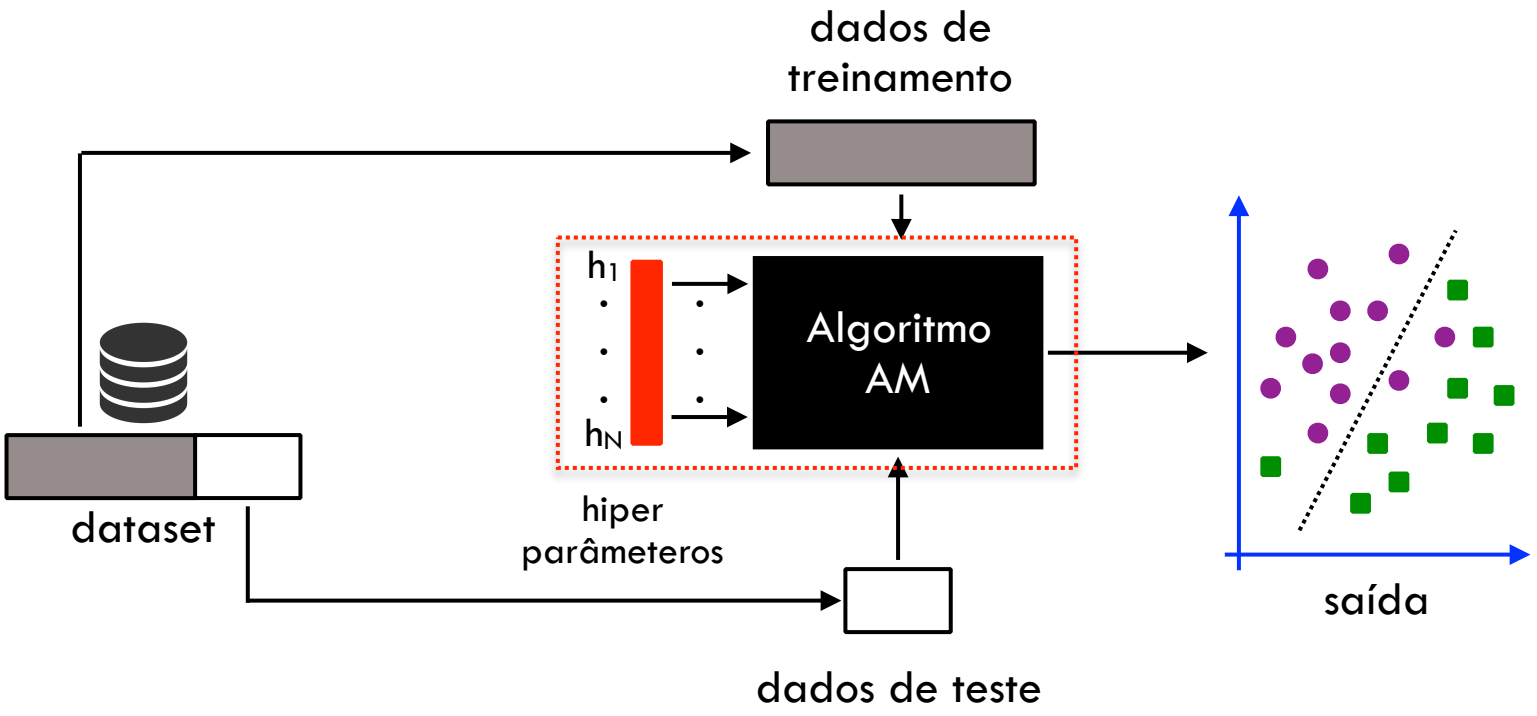
- Página principal:
  - https://github.com/mlr-org/mlr

- Tutoriais:
  - https://mlr-org.github.io/mlr/
  - https://mlr-org.github.io/mlr/articles/wrapper.html
  - https://mlr-org.github.io/mlr/articles/integrated_learners.html
  - https://mlr-org.github.io/mlr/articles/measures.html
  - https://mlr-org.github.io/mlr/articles/advanced_tune.html

Roteiro

1. Breve Introdução a *Machine Learning* (ML)
2. OpenML / mlr / mlrMBO / RStudio
3. **Exemplos / Benchmark**
4. Automated Machine Learning (AutoML)
5. Literatura em AutoML
6. Exemplos / AutoML / CASH
7. Considerações finais

Exemplo 01


dados de
treinamento
h1
hN
Algoritmo
AM
dataset
hiper
parâmeteros
saída
dados de teste

Exemplo 01

1. Obter um dataset (visualizar)
2. Criar uma tarefa de classificação (**target**)
3. Escolher um algoritmo (**learner**)
4. Definir metodologia (**rdesc**),
5. Definir medida de avaliação (**measure**)
6. Treinar um modelo
7. Predizer as classes dos dados de teste
8. Avaliar os resultados

Exemplo 01

1. OpenML::getOMLDataset()
2. mlr::makeClassifTask()
3. mlr::makeLearner()
4. mlr::makeResampleDesc()
5. mlr::resample()
6. print()

Exemplo 01

```r
library(mlr)
library(OpenML)

data.obj = OpenML::getOMLDataSet(data.id=61)
dataset  = data.obj$data

task = mlr::makeClassifTask(data = dataset, target = "class")

rf = mlr::makeLearner("classif.randomForest")

rdesc = mlr::makeResampleDesc("Holdout", split = 2/3)

res = mlr::resample(learner=rf, task=task, resampling=rdesc,
              measures=list(acc, bac), show.info = TRUE)

pred = mlr::getRRPredictions(r)

print(res)
print(pred)
```

Exercício 01

1. Repetir o exemplo anterior usando:
    A. outro algoritmo de classificação

       https://mlr-org.github.io/mlr/articles/integrated_learners.html

    B. adicionando novas medidas de desempenho
    C. usando validação cruzada como resampling ("CV")
2. Comparar com os resultados obtidos anteriormente. O que houve?

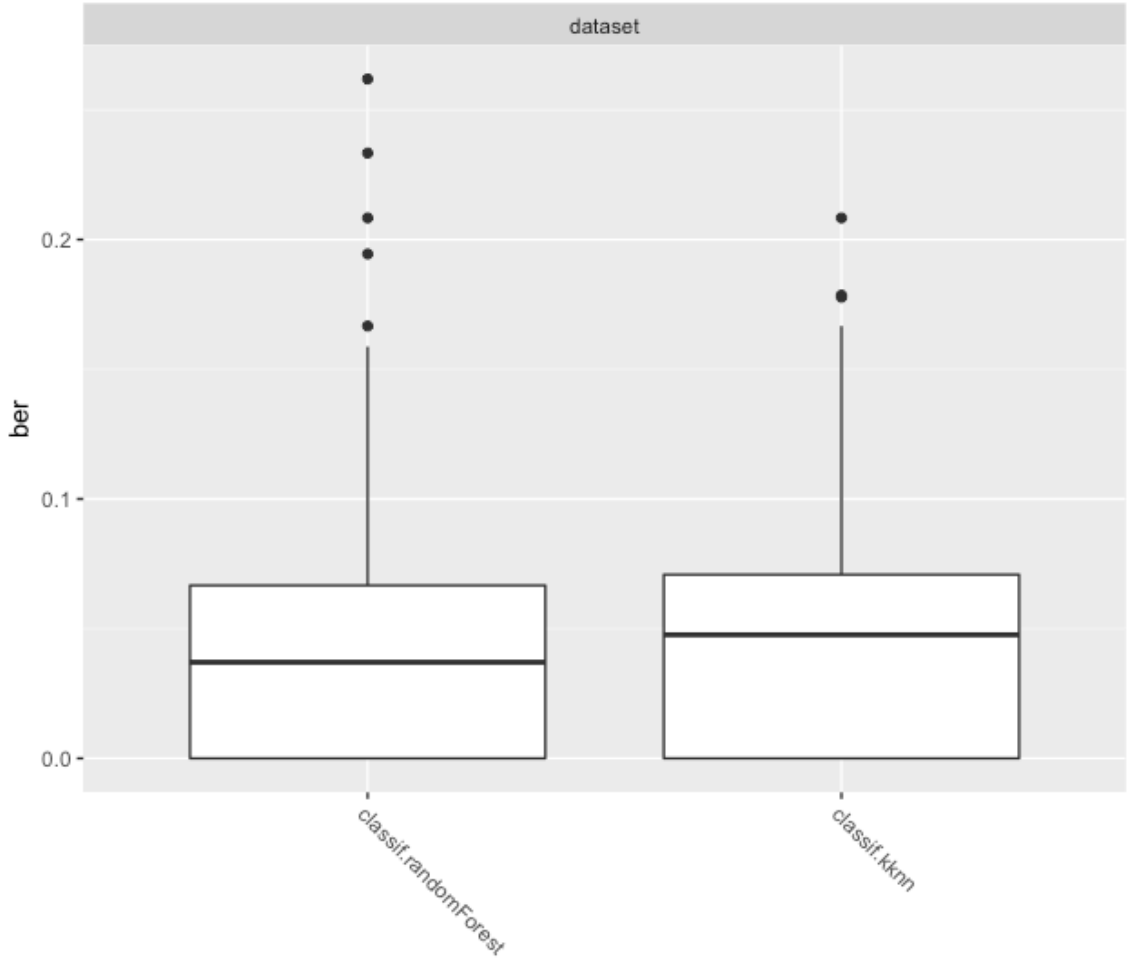
Exemplo 02

1. Avaliar mais de um algoritmo em um mesmo dataset
   - mlr::makeLearner()
   - mlr::benchmark()

Exemplo 02

```r
library(mlr)
library(OpenML)

data.obj = OpenML::getOMLDataSet(data.id=61)
dataset  = data.obj$data

task = mlr::makeClassifTask(data = dataset, target = "class")

rf = mlr::makeLearner("classif.randomForest")
nn = mlr::makeLearner("classif.kknn")

rdesc = mlr::makeResampleDesc("RepCV", reps = 10, folds = 10)

res = mlr::benchmark(learners=list(rf, nn), tasks=task, resampling=rdesc,
                     measures=list(acc, ber), show.info = TRUE)
print(res)
plotBMRBoxplots(bmr = res, measure = ber)
```

Exemplo 02



Exercício 02

- Avaliar vários algoritmos distinto no seguinte setup:
  1. Dataset: balance-scale (data.id = 11)
  2. Algoritmos: naiveBayes, SVM, J48, RF, k-NN
  3. Medida: Balanced error rate (ber)
  4. Resampling: 5 x 5-CV

- Qual o melhor algoritmo?

Exercício 03

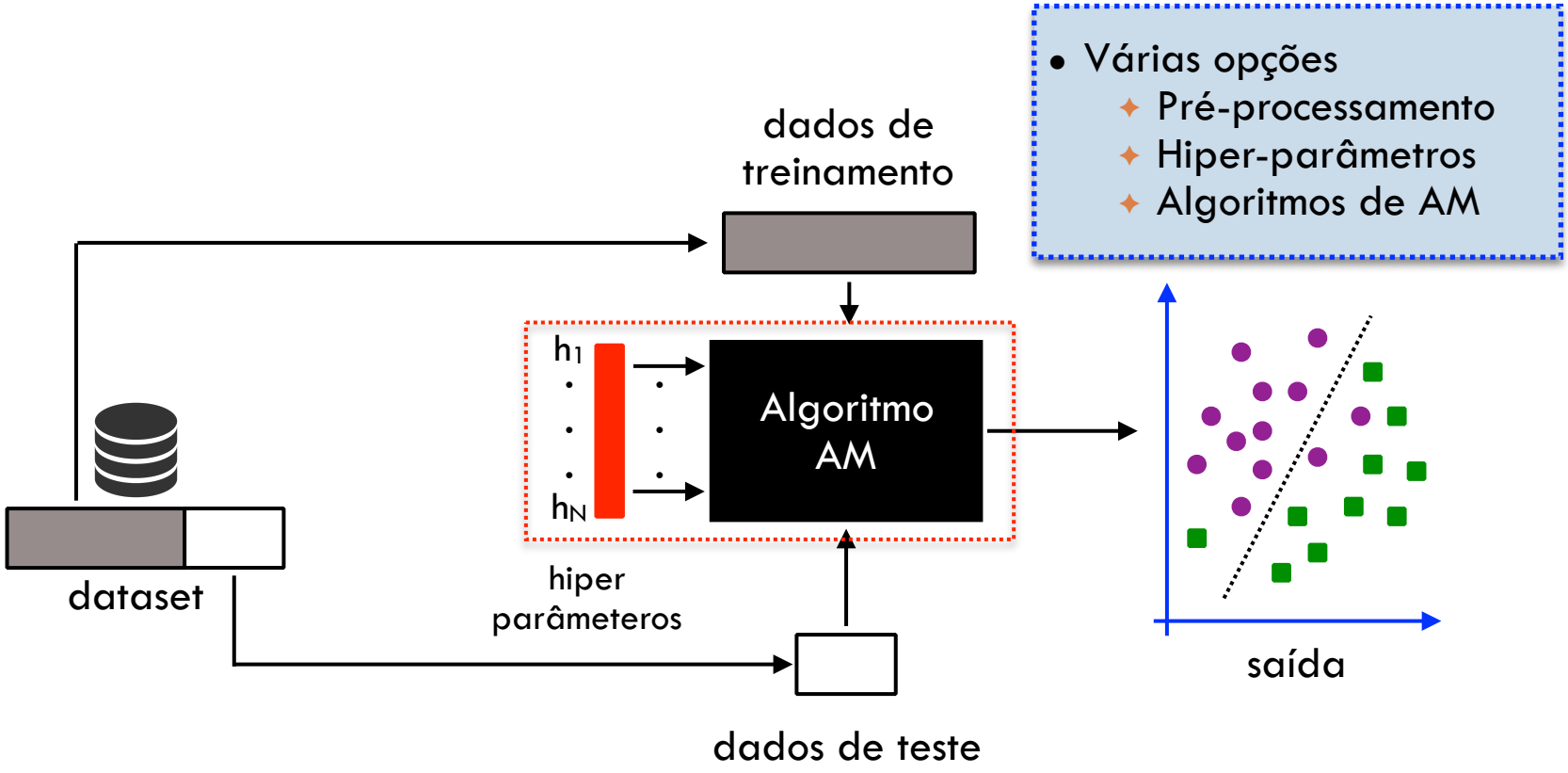
- Avaliar vários algoritmos em vários datasets (*benchmarking*)
  - plotar resultados, plotBMRBoxplots()
  - verificar resultados por meio de teste estatísticos
  - g mlr::generateCritDifferencesData(res, p.value = 0.05, test = “nemenyi”)
  - plotCritDifferences(g)

- Qual o melhor algoritmo?

Roteiro

1. Breve Introdução a *Machine Learning* (ML)
2. OpenML / mlr / mlrMBO / RStudio
3. Exemplos / Benchmark
4. **Automated Machine Learning (AutoML)**
5. Literatura em AutoML
6. Exemplos / AutoML / CASH
7. Considerações finais

AutoML

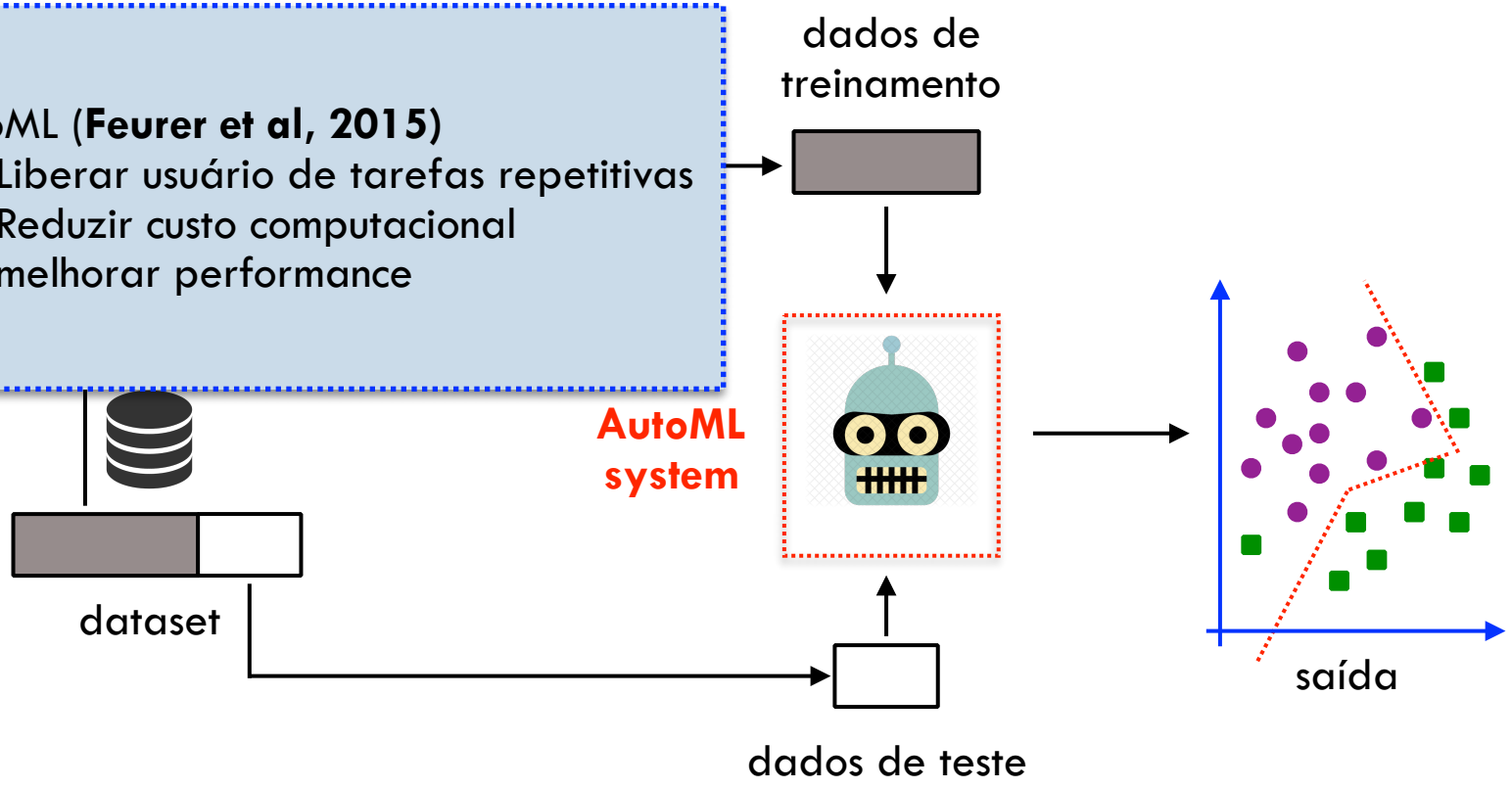

• Várias opções
✦ Pré-processamento
✦ Hiper-parâmetros
✦ Algoritmos de AM
dados de treinamento
h1
hN
Algoritmo AM
dataset
hiper parâmeteros
saída
dados de teste

AutoML

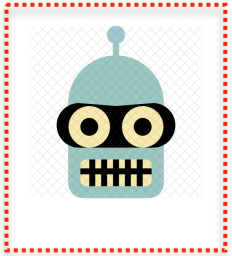
- Mundo real:
  - existem **centenas** de algoritmos de AM
    - diferentes viéses
    - forma de resolver um problema, forma de aprender
  - testar diversos algoritmos
  - hiper-parâmetros
  - pré-processamento
  - ensembles
- muito tempo gasto :/
- nem todos podem usar
- **QUESTÃO: Qual melhor configuração usar?**

AutoML

- AutoML (**Feurer et al, 2015**)
  - Liberar usuário de tarefas repetitivas
  - Reduzir custo computacional
  - melhorar performance



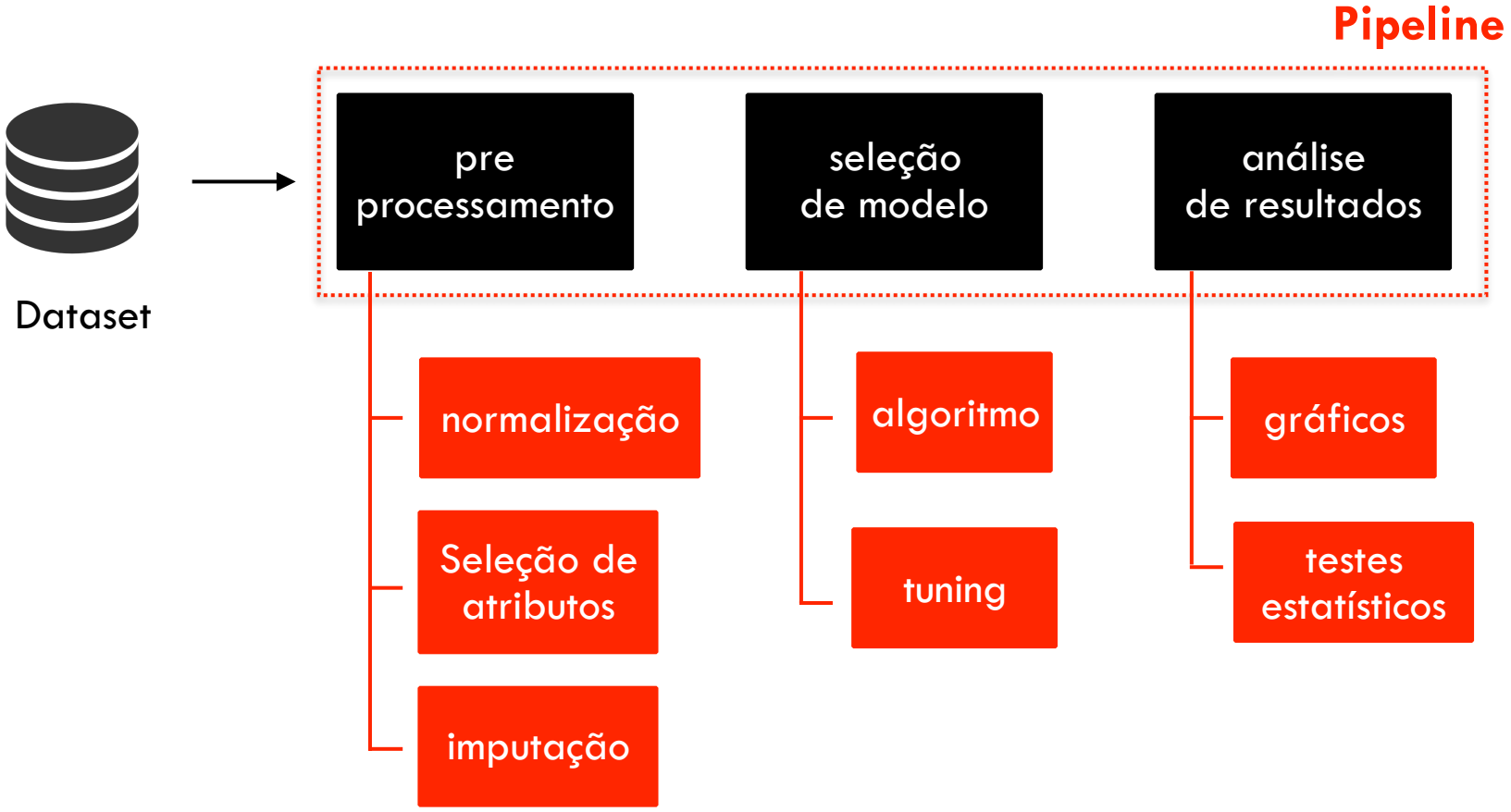
AutoML



AutoML systems

- AutoML (**Feurer et al, 2015**)
  - Meta-aprendizado
  - Otimização
  - Aprendizado por transferência
  - Processamento paralelo
  - …

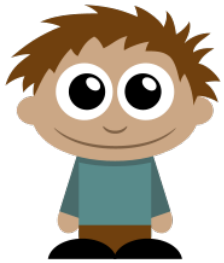
AutoML


Pipeline
Dataset
pre processamento
seleção de modelo
análise de resultados
normalização
Seleção de atributos
imputação
algoritmo
tuning
gráficos
testes estatísticos

AutoML

Especialista

- Podem explorar mais opções de pipelines
- Demandar mais tempo em outras tarefas prioritárias



Novos usuários

- Solução automatizada
- Definir tempo e quantidade de memória

Roteiro

1. Breve Introdução a *Machine Learning* (ML)
2. OpenML / mlr / mlrMBO / RStudio
3. Exemplos / Benchmark
4. Automated Machine Learning (AutoML)
5. **Literatura em AutoML**
6. Exemplos / AutoML / CASH
7. Considerações finais

Auto-WEKA

Auto-WEKA

Home | Papers | Software | Licensing

Many different machine learning algorithms exist that can easily be used off the shelf, many of these methods are implemented in th
can drastically change their performance, and there are a staggeringly large number of possible alternatives overall. Auto-WEKA co
going beyond previous methods that address these issues in isolation. Auto-WEKA does this using a fully automated approach, leve
users to more effectively identify machine learning algorithms and hyperparameter settings appropriate to their applications, and he

Latest news

29 November, 2016	Our paper on Auto-WEKA 2.0 was accepted for publication at the Journal of Machine Learning Research, op
10 November, 2016	We've released a new version with lots of new features and stability fixes.
06 July, 2016	We've updated the WEKA version, support returning more than one configuration and fixed a few bugs.
May, 2016	**Auto-WEKA is back with a vengeance! We have just released version 2.0 with a new interface that int**
12 November, 2013	Bug fixes, updated documentation, more usability tweaks
9 August, 2013	New version of Auto-WEKA code, emphasis on usability
5 June, 2013	Added datasets, updated Auto-WEKA code
2 March, 2013	Initial release of Auto-WEKA

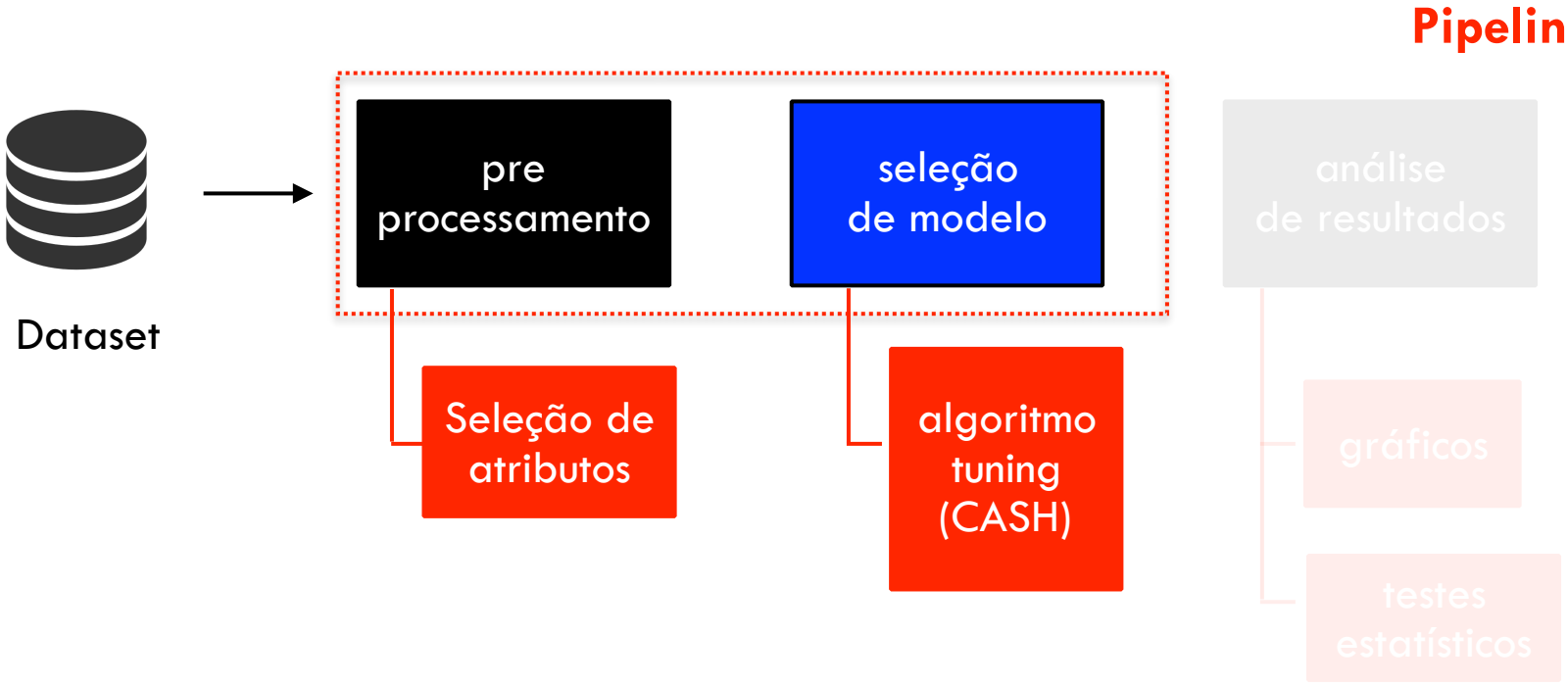
People

- Lars Kotthoff, Assistant Professor (University of Wymonig)
- Chris Thornton, M.Sc. Student (UBC)
- Frank Hutter, Assistant Professor (Freiburg University)
- Holger Hoos, Professor (UBC)
- Kevin Leyton-Brown, Professor (UBC)

Auto-WEKA

- Proposto pelo pessoal da University of British Columbia
  - v1 (2013), v2 (2016)
- Seleciona algoritmos + hiper-parâmetros simultaneamente
  - *CASH - Combined Algorithm Selection and Hyperparameter Optimization*
  - Otimização Bayesiana (SMAC)
- WEKA (JAVA)

Auto-WEKA


Pipeline
Dataset
pre processamento
seleção de modelo
análise de resultados
Seleção de atributos
algoritmo tuning (CASH)
gráficos
testes estatísticos

Auto-skLearn

auto-sklearn

auto-sklearn is an automated machine learning toolkit and a drop-in replacement for a scikit-learn estimator:

```
>>> import autosklearn.classification
>>> cls = autosklearn.classification.AutoSklearnClassifier()
>>> cls.fit(X_train, y_train)
>>> predictions = cls.predict(X_test)
```

auto-sklearn frees a machine learning user from algorithm selection and hyperparameter tuning. It leverages r
Bayesian optimization, *meta-learning* and *ensemble construction*. Learn more about the technology behind *a*
paper published at NIPS 2015 .

Example

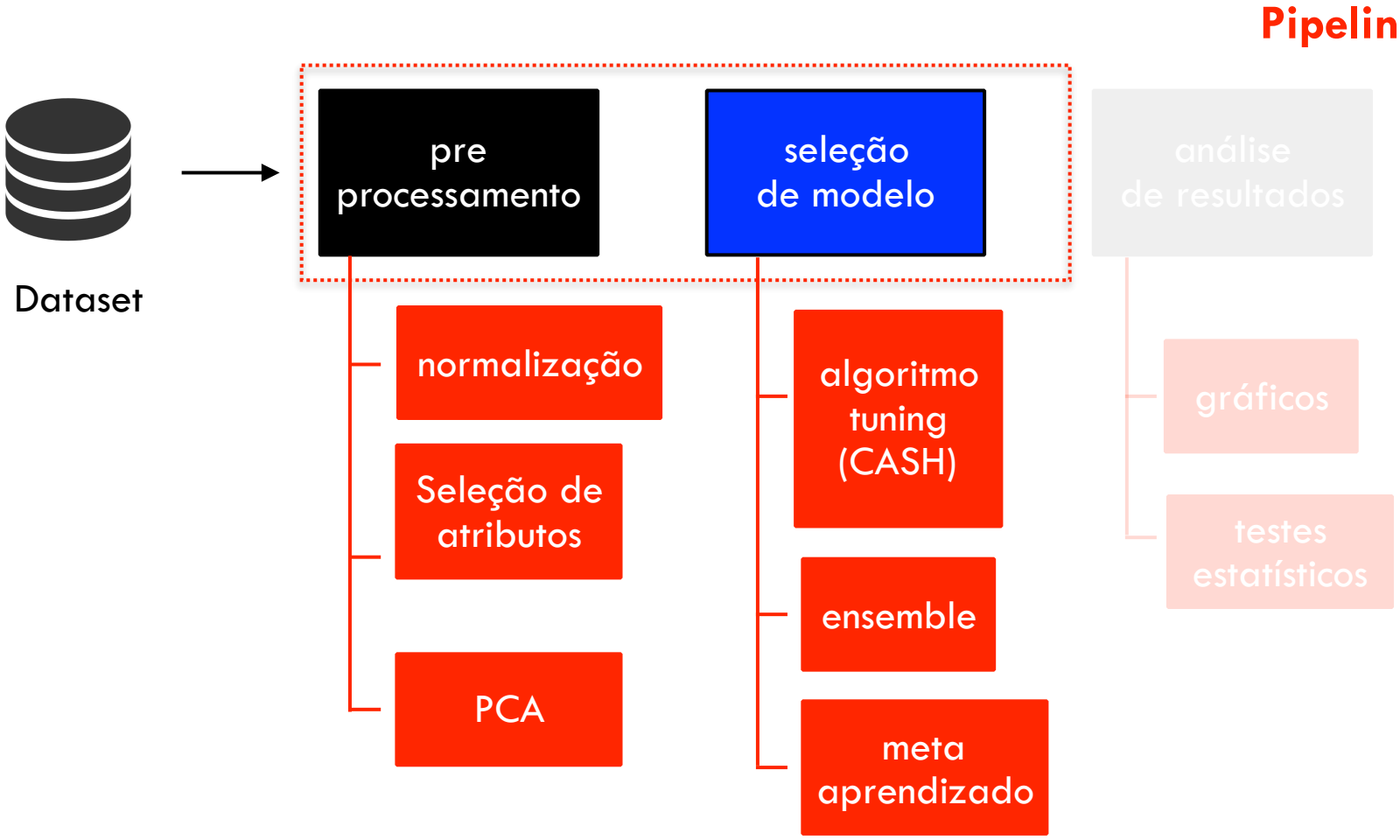
```
>>> import autosklearn.classification
>>> import sklearn.model_selection
>>> import sklearn.datasets
>>> import sklearn.metrics
>>> X, y = sklearn.datasets.load_digits(return_X_y=True)
>>> X_train, X_test, y_train, y_test = \
        sklearn.model_selection.train_test_split(X, y, random_state=1)
>>> automl = autosklearn.classification.AutoSklearnClassifier()
>>> automl.fit(X_train, y_train)
>>> y_hat = automl.predict(X_test)
>>> print("Accuracy score", sklearn.metrics.accuracy_score(y_test, y_hat))
```

This will run for one hour and should result in an accuracy above 0.98.

Auto-skLearn

- Proposto pelo pessoal da Universidade de Freiburg
  - v1 (2015)
- Reimplementação do Auto-WEKA
  - Meta-learning
  - Ensembles
- scikit-Learn (Python)

Auto-skLearn



TPOT

Master status: build passing | build passing | health 95% | coverage 96%

Development status: build passing | build passing | health 95% | coverage 96%

Package information: python 2.7 | python 3.6 | license LGPL v3 | pypi package 0.9.5

Consider TPOT your **Data Science Assistant**. TPOT is a Python Automated Machine Learning tool that optimizes machine learning pipelines using genetic programming.

```
tpot.export('tpot_mnist_pipeline.py')
!cat tpot_mnist_pipeline.py
```

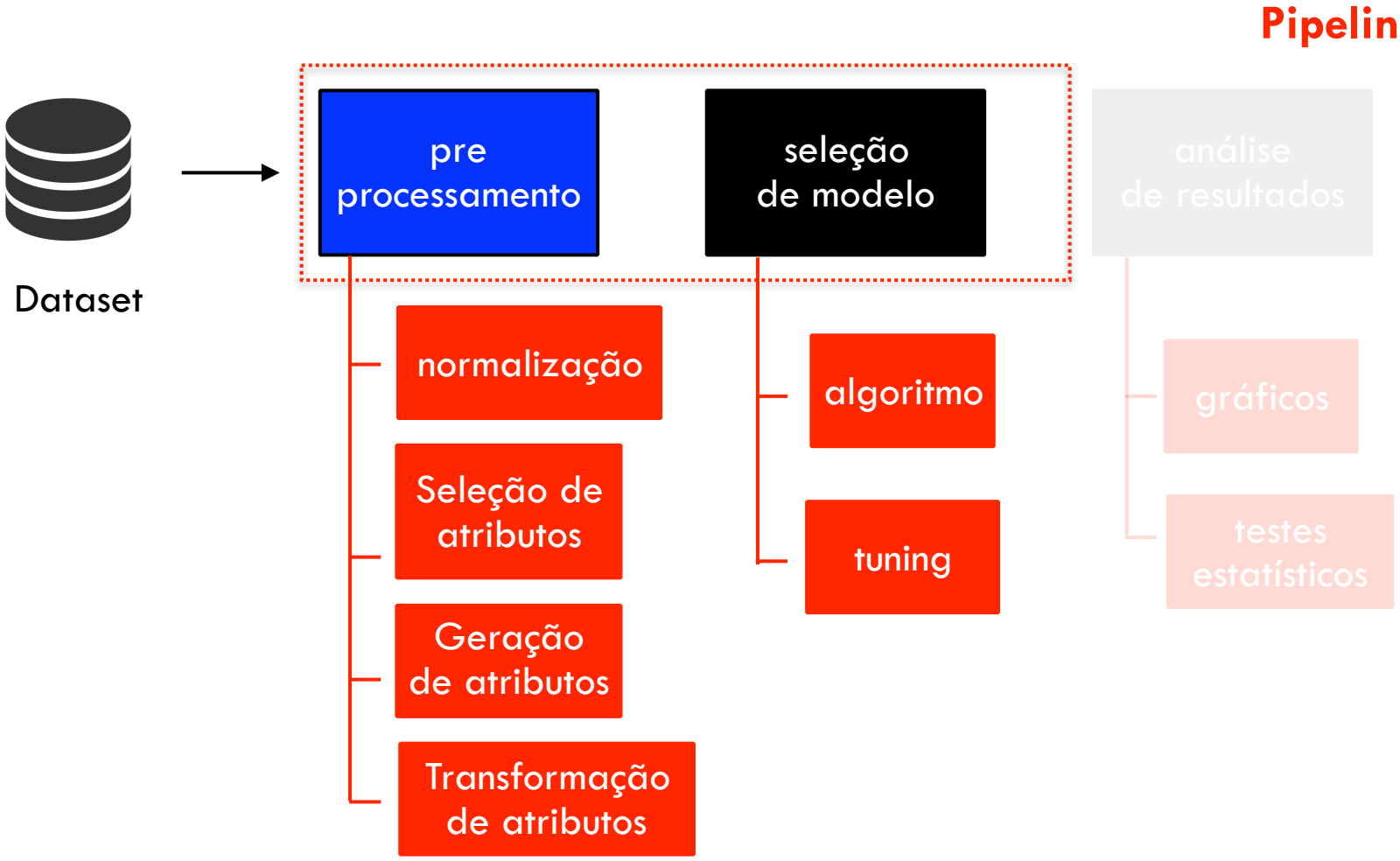
```
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

# NOTE: Make sure that the class is labeled 'class' in the data file
tpot_data = np.recfromcsv('PATH/TO/DATA/FILE', delimiter='COLUMN_SEPARATOR', dtype=np.float64)
features = np.delete(tpot_data.view(np.float64).reshape(tpot_data.size, -1), tpot_data.dtype.names.index('class'), ax
is=1)
```

TPOT

- *Tree-based Pipeline Optimization*
- Proposto pelo pessoal da Universidade da Pensilvania
  - v1 (2016)

- Programação Genética para evoluir Pipelines
  - Pipelines - Grafos, Árvores
  - scikit-Learn (Python)

TPOT


Pipeline
Dataset
pre processamento
seleção de modelo
análise de resultados
normalização
Seleção de atributos
Geração de atributos
Transformação de atributos
algoritmo
tuning
gráficos
testes estatísticos

Roteiro

1. Breve Introdução a *Machine Learning* (ML)
2. OpenML / mlr / mlrMBO / RStudio
3. Exemplos / Benchmark
4. Automated Machine Learning (AutoML)
5. Literatura em AutoML
6. **Exemplos / AutoML / CASH**
7. Considerações finais

Exemplo 04

- Ajustar hiper-parâmetros (*tuning*):
  - dataset: Vehicle (id = 54)
  - algoritmo: Random Forest
  - hiper-parâmetros:
    - nTree, mtry, maxnodes, nodesize
  - Random Search
    - 50 iterações

Exemplo 04

- Ajustar hiper-parâmetros (*tuning*):
  - makeParamSet()
  - makeTuneControlRandom()
  - tuneParams()
  - generateHyperParsEffectData()
  - plotHyperParsEffect()

Exemplo 04

```r
# Create a search space
ps = makeParamSet(
  makeIntegerParam("ntree", lower = 10, upper = 600),
  makeIntegerParam("mtry", lower = 1, upper = 10)
)

### Random search with 100 iterations
ctrl  = makeTuneControlRandom(maxit = 50L)
rdesc = makeResampleDesc("CV", iters = 10L, stratify=TRUE)
me = ber

# Performing the tuning
res = tuneParams("classif.randomForest", task = task, resampling = rdesc,
                 par.set = ps, measure=me, control = ctrl)
print(res)

res_data = generateHyperParsEffectData(res, partial.dep=TRUE)
res_plt = plotHyperParsEffect(res_data, x = "iteration", y = "bac.test.mean",
                              plot.type = "line",
                              partial.dep.learn='regr.randomForest')
print(res_plt)
```

Exercício 04

- Ajustar hiper-parâmetros (*tuning*):
  - algoritmo: SVM
  - hiper-parâmetros:
    - cost [2^(-15), 2^(+15)]
    - gamma [2^(-15), 2^(+15)]
  - Random Search

Exemplo 05

- Selecionar atributos mais relevantes
  - dataset: AP_Prostate_Uterus (tipo de câncer)
  - Information Gain
    - número de features
    - % de features
    - threshold
  - Algoritmos:
    - Rpart, kNN

Exemplo 05

- Selecionar atributos mais relevantes
  - generateFilterValuesData()
  - filterFeatures()
  - benchmark()

Exemplo 05

```
13  cat("All features ....\n")
14  all.task = makeClassifTask(data = data, target = "Tissue")
15  print(all.task)
16  cat("-----------------------------------------------------\n\n")
17
18  # feature values
19  cat("Filtering ....\n")
20  fv = generateFilterValuesData(all.task, method = "information.gain")
21  print(fv)
22  cat("-----------------------------------------------------\n\n")
23
24  cat("4 best ....\n")
25  # Keep the 2 most important features
26  fil.abs.task = filterFeatures(all.task, fval=fv, abs=4)
27  print(fil.abs.task)
28  cat("-----------------------------------------------------\n\n")
29
```

Exercício 05

- Proposta:
  - outro dataset: + de 100 features
  - terceiro algoritmo
  - plotar resultados

Exemplo 06

- CASH
  - Algoritmos: SVM, RF, KNN, NB, J48, Rpart
  - Hiper-parâmetros: espaço misto
  - Random Search
  - Medidas:
    - Balanced Error Rate
    - Runtime

- Gráficos de análises

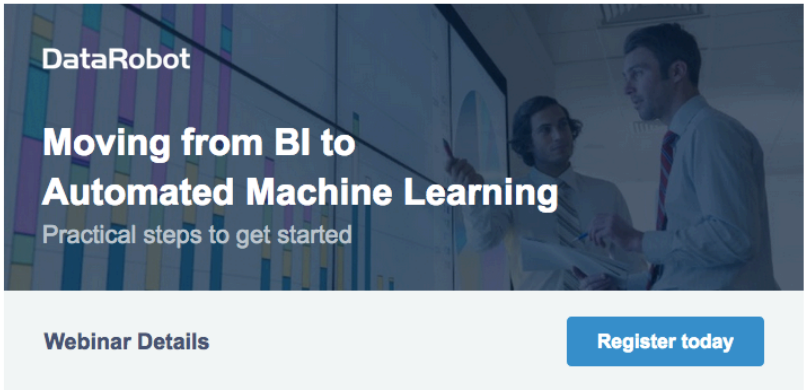
Exemplo 06

- Funções
  - makeModelMultiplexer()
  - makeModelMultiplexerParamSet()
  - makeResampleDesc()
  - makeTuneMultiCritControlRandom()
  - tuneParamsMultiCrit()
  - plotTuneMultiCritResult()

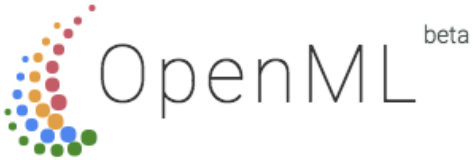
Roteiro

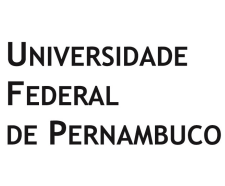
1. Breve Introdução a *Machine Learning* (ML)
2. OpenML / mlr / mlrMBO / RStudio
3. Exemplos / Benchmark
4. Automated Machine Learning (AutoML)
5. Literatura em AutoML
6. Exemplos / AutoML / CASH
7. Considerações finais

Eventos de AutoML



Grupos de AutoML





Questionamentos

- E os cientistas de dados?
- Pessoas ficarão desempregadas?
- Bolhas sociais?
- Privacidade?
- Quando a SkyNet chegará?

Questionamentos

Agradecimentos =)

- Edésio - ICMC/USP
- Prof André - ICMC/USP
- Prof Ricardo Cerri - UFSCar
- Organização Secomp

Obrigado

Rafael Gomes Mantovani

rgmantovani@usp.br