SOSP AI System Workshop
Shanghai
Oct. 28, 2017

EasyML: Ease the Process of Machine Learning with Data Flow

Jun Xu

Institute of Computing Technology, Chinese Academy of Sciences

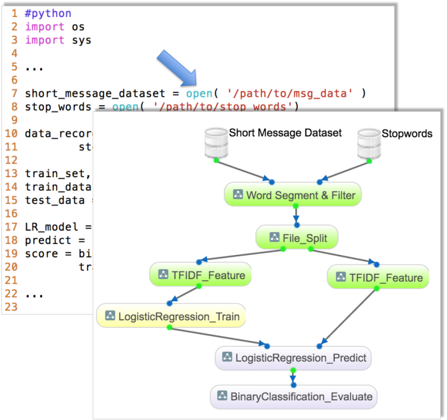
Oct. 28, 2017

Applying Machine Learning is not Easy

Collaborating and sharing

Share the data, algorithms, and experience

UI



```
#python
import os
import sys

...

short_message_dataset = open( '/path/to/msg_data' )
stop_words = open( '/path/to/stop_words')

data_recor
         st

train_set,
train_data
test_data

LR_model =
predict =
score = bi
         tr

...
```

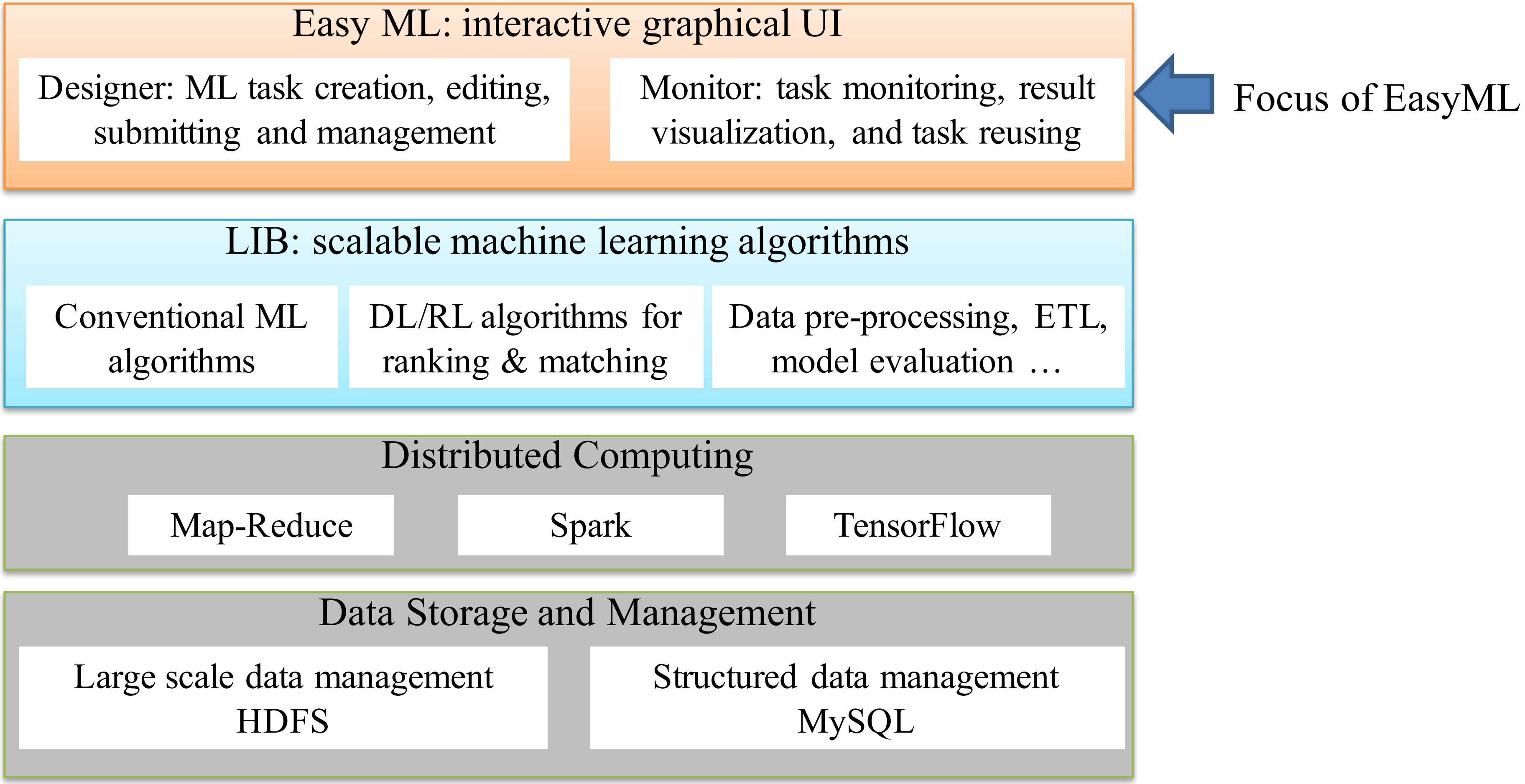
Simple UI is helpful

Mobility

Can access the service everywhere

The barrier comes not only from the advanced algorithms, but also from the complex process of using the algorithms!

Large Scale Machine Learning System@ICT



Easy ML: interactive graphical UI

Designer: ML task creation, editing, submitting and management

Monitor: task monitoring, result visualization, and task reusing

Focus of EasyML

LIB: scalable machine learning algorithms

Conventional ML algorithms

DL/RL algorithms for ranking & matching

Data pre-processing, ETL, model evaluation …

Distributed Computing

Map-Reduce

Spark

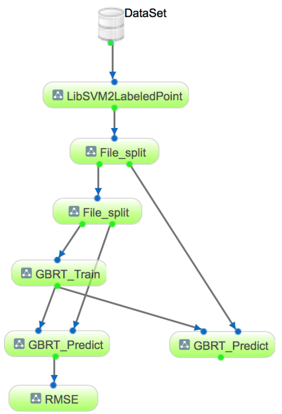
TensorFlow

Data Storage and Management

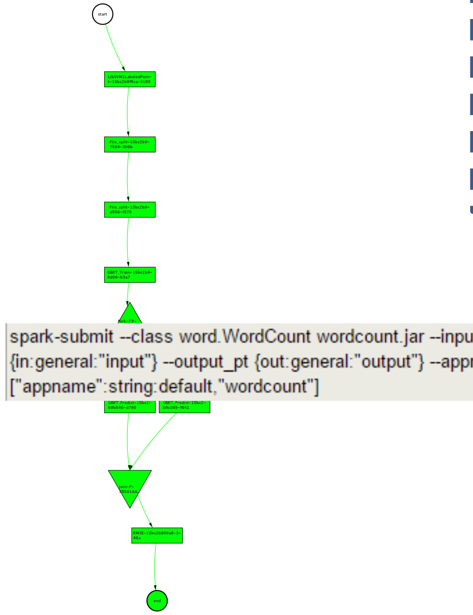
Large scale data management HDFS

Structured data management MySQL

Design of Easy Machine Learning

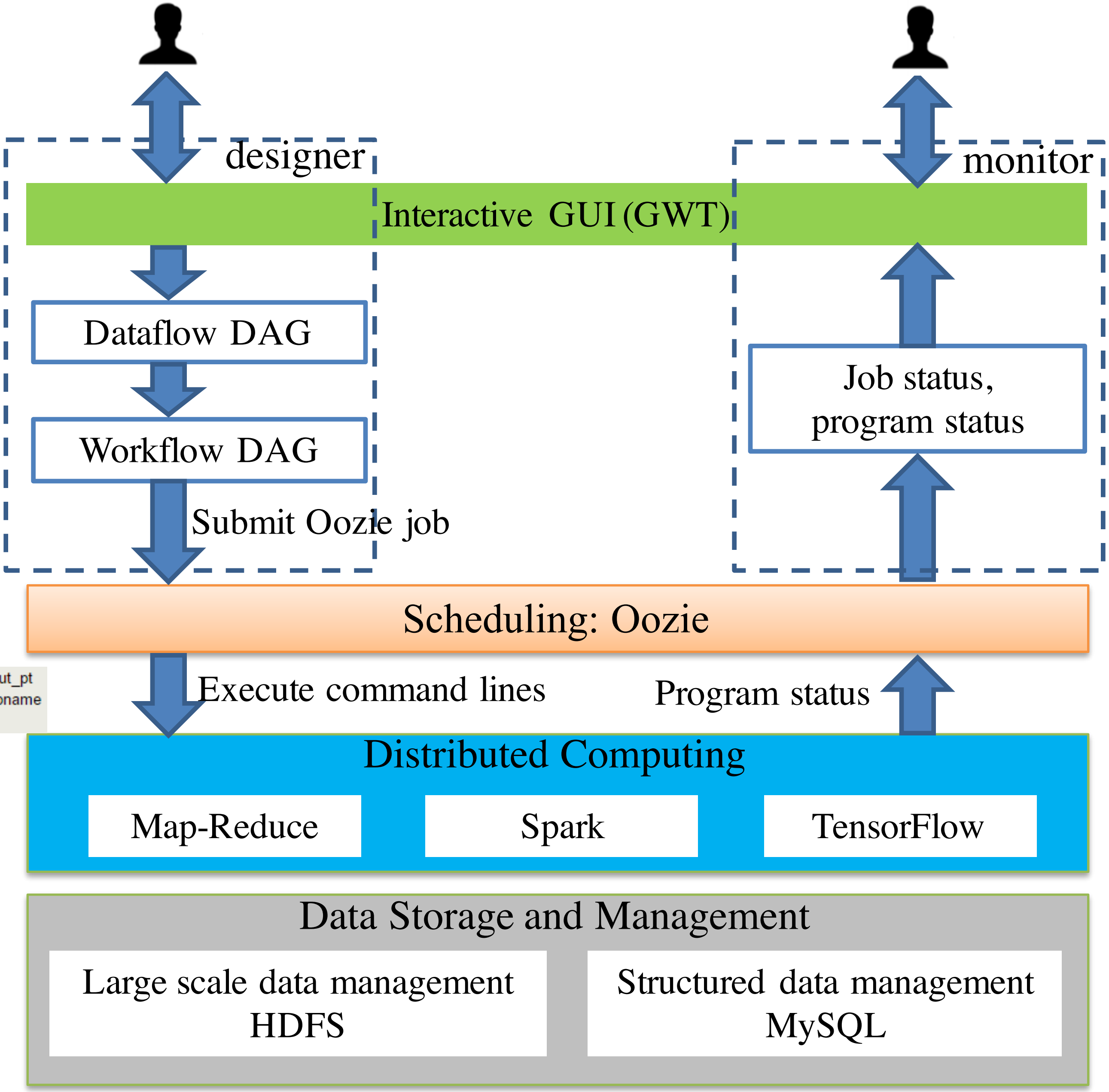


Node: program / data
Edge: dataflow



```
spark-submit --class word.WordCount wordcount.jar --input_pt
{in:general:"input"} --output_pt {out:general:"output"} --appname
["appname":string:default,"wordcount"]
```

Node: program / start / end / fork / join
Edge: dependency



designer

monitor

Interactive GUI (GWT)

Dataflow DAG

Workflow DAG

Submit Oozie job

Job status, program status

Scheduling: Oozie

Execute command lines

Program status

Distributed Computing

Map-Reduce

Spark

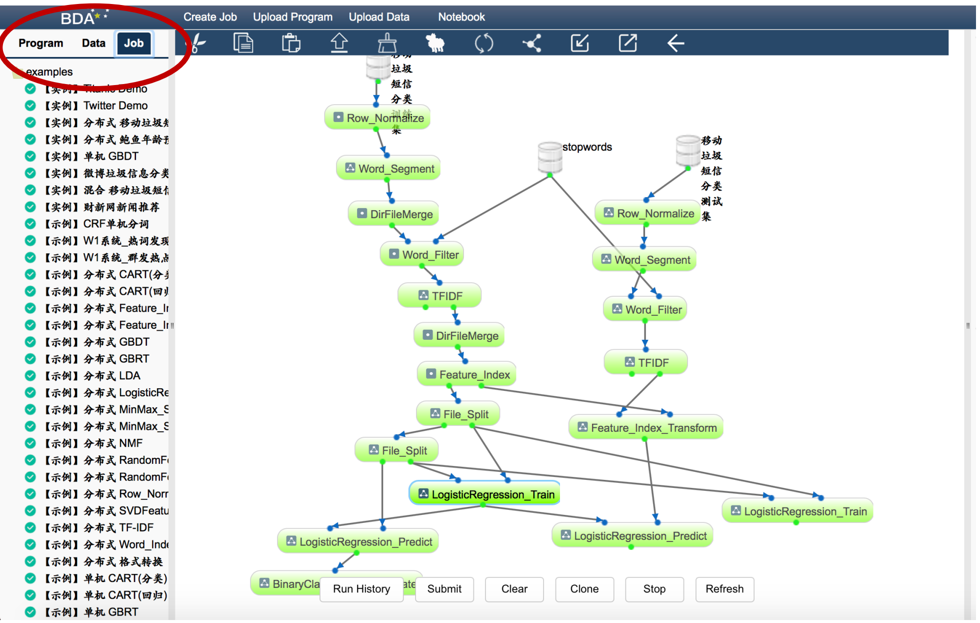
TensorFlow

Data Storage and Management

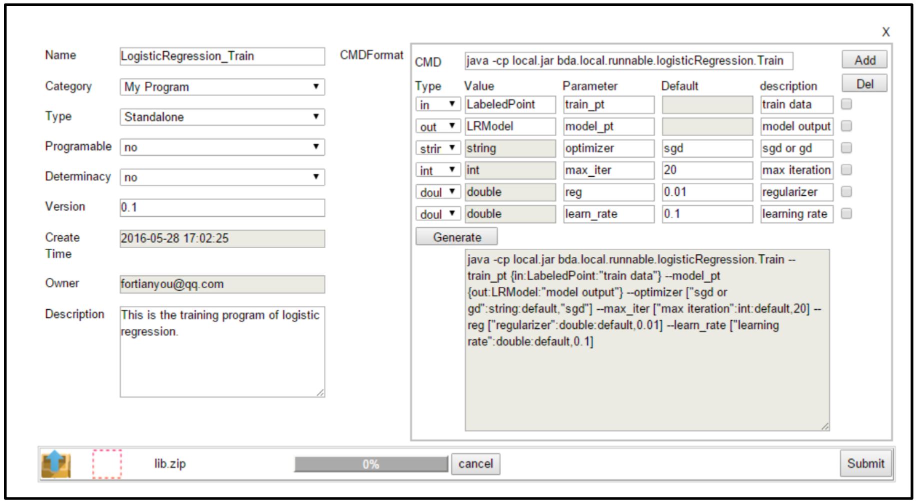
Large scale data management HDFS

Structured data management MySQL

Key Features — Resource Management



examples
- 【实例】Titanic Demo
- 【实例】Twitter Demo
- 【实例】分布式 移动垃圾
- 【实例】分布式 鲍鱼年龄
- 【实例】单机 GBDT
- 【实例】微博垃圾信息分类
- 【实例】混合 移动垃圾短
- 【实例】财新网新闻推荐
- 【示例】CRF单机分词
- 【示例】W1系统_热词发
- 【示例】W1系统_群发热点
- 【示例】分布式 CART(分
- 【示例】分布式 CART(回
- 【示例】分布式 Feature_I
- 【示例】分布式 Feature_I
- 【示例】分布式 GBDT
- 【示例】分布式 GBRT
- 【示例】分布式 LDA
- 【示例】分布式 LogisticRe
- 【示例】分布式 MinMax_S
- 【示例】分布式 MinMax_S
- 【示例】分布式 NMF
- 【示例】分布式 RandomF
- 【示例】分布式 RandomF
- 【示例】分布式 Row_Norr
- 【示例】分布式 SVDFeatu
- 【示例】分布式 TF-IDF
- 【示例】分布式 Word_Inde
- 【示例】分布式 格式转换
- 【示例】单机 CART(分类)
- 【示例】单机 CART(回归)
- 【示例】单机 GBRT



Name	LogisticRegression_Train
Category	My Program
Type	Standalone
Programable	no
Determinacy	no
Version	0.1
Create Time	2016-05-28 17:02:25
Owner	fortianyou@qq.com
Description	This is the training program of logistic regression.

CMDFormat

CMD: java -cp local.jar bda.local.runnable.logisticRegression.Train

Type	Value	Parameter	Default	description	
in	LabeledPoint	train_pt		train data	□
out	LRModel	model_pt		model output	□
strir	string	optimizer	sgd	sgd or gd	□
int	int	max_iter	20	max iteration	□
doul	double	reg	0.01	regularizer	□
doul	double	learn_rate	0.1	learning rate	□

```
java -cp local.jar bda.local.runnable.logisticRegression.Train --
train_pt {in:LabeledPoint:"train data"} --model_pt
{out:LRModel:"model output"} --optimizer ["sgd or
gd":string:default,"sgd"] --max_iter ["max iteration":int:default,20] --
reg ["regularizer":double:default,0.01] --learn_rate ["learning
rate":double:default,0.1]
```

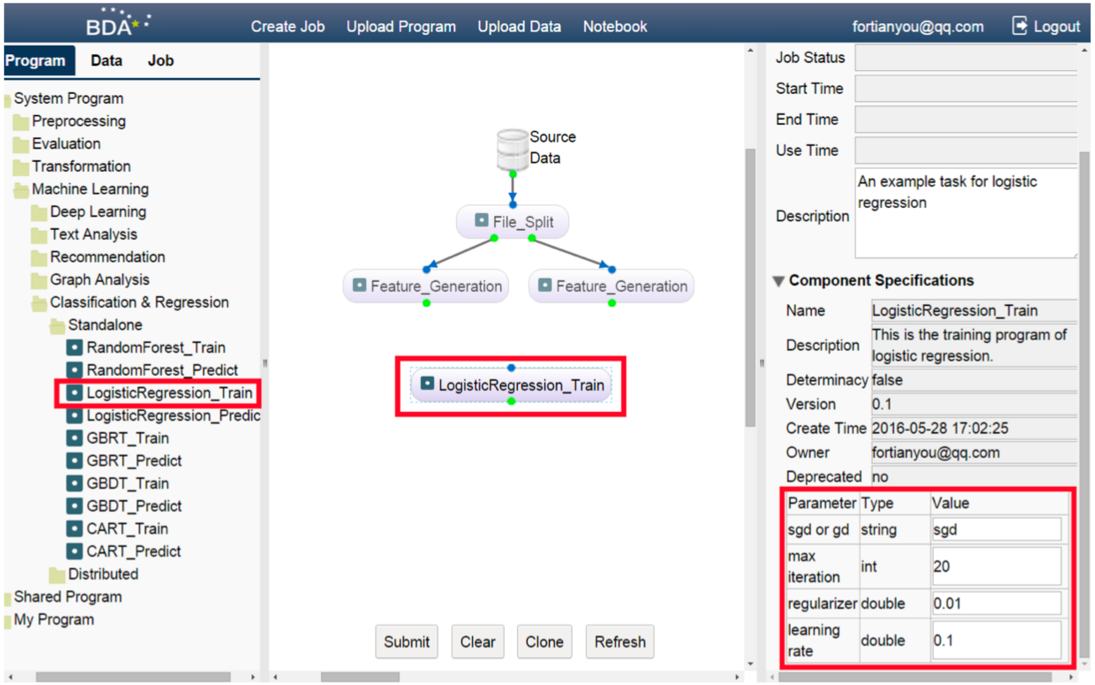
lib.zip

Managing programs, data, and tasks

Uploading new algorithms

- Managing the algorithms, data, and tasks
- Uploading algorithms and data

Key Features — Task Design



Job Status

Start Time

End Time

Use Time

Description: An example task for logistic regression

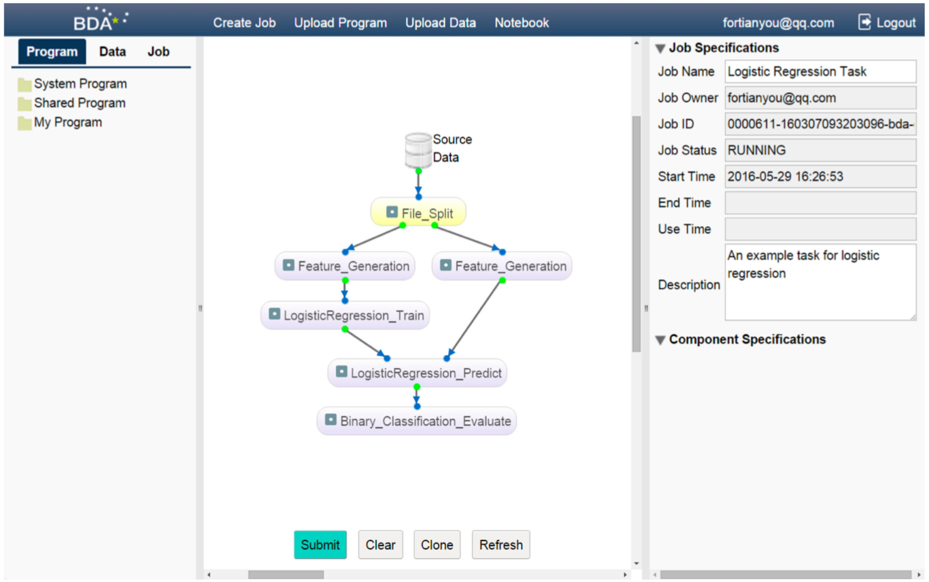
▼ Component Specifications

Name	LogisticRegression_Train
Description	This is the training program of logistic regression.
Determinacy	false
Version	0.1
Create Time	2016-05-28 17:02:25
Owner	fortianyou@qq.com
Deprecated	no

Parameter	Type	Value
sgd or gd	string	sgd
max iteration	int	20
regularizer	double	0.01
learning rate	double	0.1

- Creating the task DAG (usually by cloning and editing an existing task) with drag-and-drop manner
- Setting the parameters for each node
- Submitting for execution

Key Features — Task Monitoring

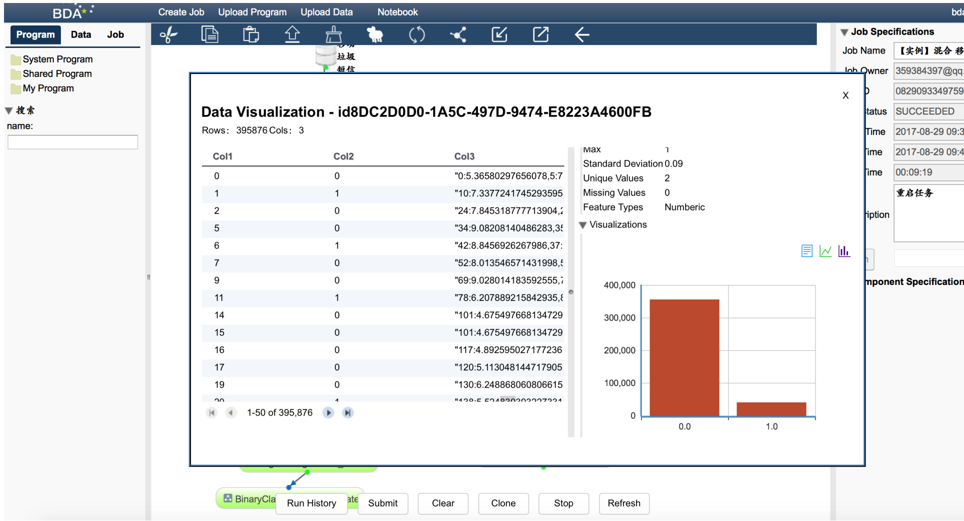


▼ Job Specifications

Job Name: Logistic Regression Task
Job Owner: fortianyou@qq.com
Job ID: 0000611-160307093203096-bda-
Job Status: RUNNING
Start Time: 2016-05-29 16:26:53
End Time:
Use Time:
Description: An example task for logistic regression

▼ Component Specifications

Task/node status monitoring



Data Visualization - id8DC2D0D0-1A5C-497D-9474-E8223A4600FB

Rows： 395876 Cols： 3

Col1	Col2	Col3
0	0	"0:5.36580297656078,5:7
1	1	"10:7.3377241745293595
2	0	"24:7.845318777713904,2
5	0	"34:9.08208140486283,35
6	1	"42:8.8456926267986,37:
7	0	"52:8.013546571431998,5
9	0	"69:9.028014183592555,7
11	1	"78:6.207889215842935,8
14	0	"101:4.675497668134729
15	0	"101:4.675497668134729
16	0	"117:4.892595027177236
17	0	"120:5.113048144717905
19	0	"130:6.248868060806615
20	1	"138:5.524[illegible]

1-50 of 395,876

Max	1
Standard Deviation	0.09
Unique Values	2
Missing Values	0
Feature Types	Numberic

▼ Visualizations

▼ Job Specifications

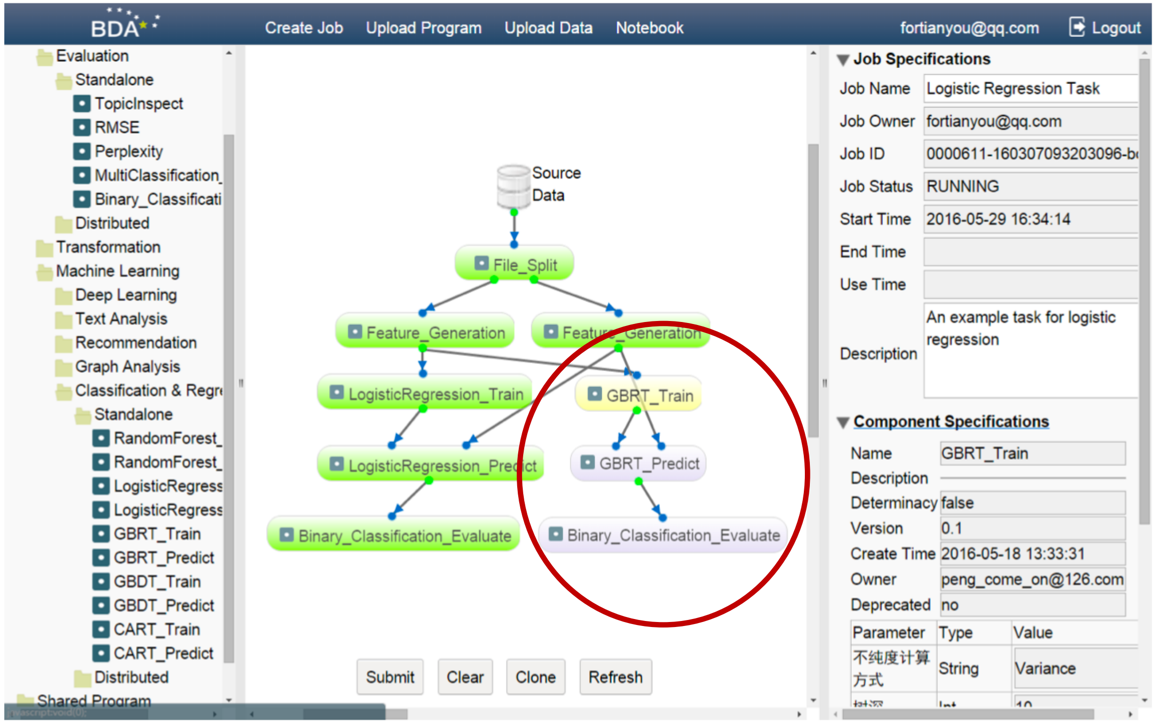
Job Name: 【实例】混合 移
Job Owner: 359384397@qq
ID: 0829093349759
Status: SUCCEEDED
Time: 2017-08-29 09:3
Time: 2017-08-29 09:4
Time: 00:09:19
Description: 重启任务

Component Specification

Data/results visualization

- Monitoring status of tasks and nodes
- Checking / downloading the outputs
- Visualizing the data / models

Key Features — Task Reusing



Job Specifications

Job Name	Logistic Regression Task
Job Owner	fortianyou@qq.com
Job ID	0000611-160307093203096-b
Job Status	RUNNING
Start Time	2016-05-29 16:34:14
End Time	
Use Time	
Description	An example task for logistic regression

Component Specifications

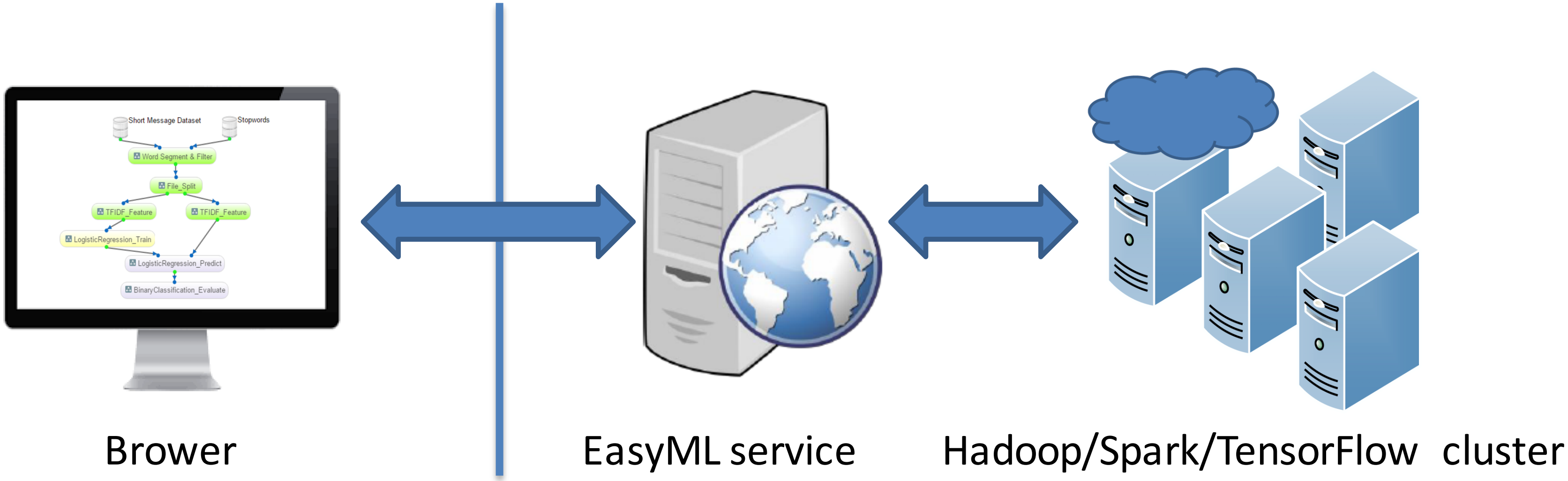
Name	GBRT_Train
Description	
Determinacy	false
Version	0.1
Create Time	2016-05-18 13:33:31
Owner	peng_come_on@126.com
Deprecated	no

Parameter	Type	Value
不纯度计算方式	String	Variance
树深	Int	10

- Editing (appending nodes, deleting nodes, and changing parameters) and re-submiting

Deploy as Web Service

http://159.226.40.104:18080/dev

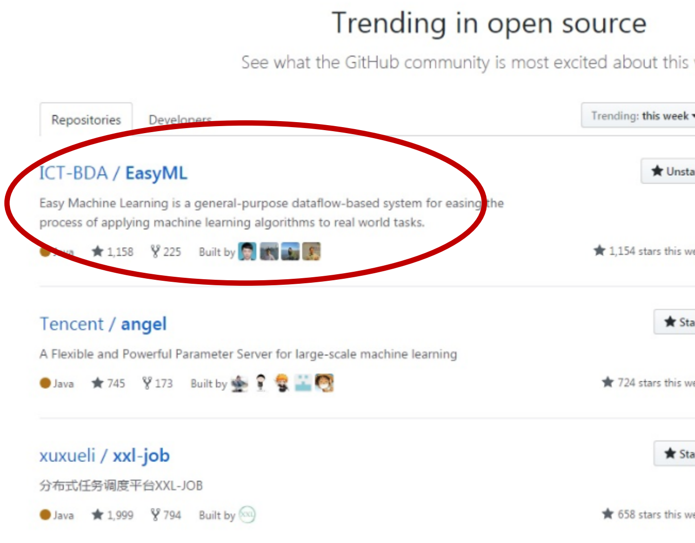


- Advantages
  - **Sharing**: share data/programs/tasks among users
  - **Collaborating**: working together for one task
  - **Mobility**: accessing with web browsers anywhere
  - **Open**: ETL for data import/export; can run third-party algorithms

Source Shared at Github

https://github.com/ICT-BDA/EasyML

- Top 1 Java project at Github trending for one week
- 1400 + stars and ~300 forks
- CIKM 2016 best demo candidate
  [Guo et al., CIKM ’16]



Trending in open source

See what the GitHub community is most excited about this

ICT-BDA / EasyML

Easy Machine Learning is a general-purpose dataflow-based system for easing the process of applying machine learning algorithms to real world tasks.

Java ★ 1,158 225 Built by ★ 1,154 stars this we

Tencent / angel

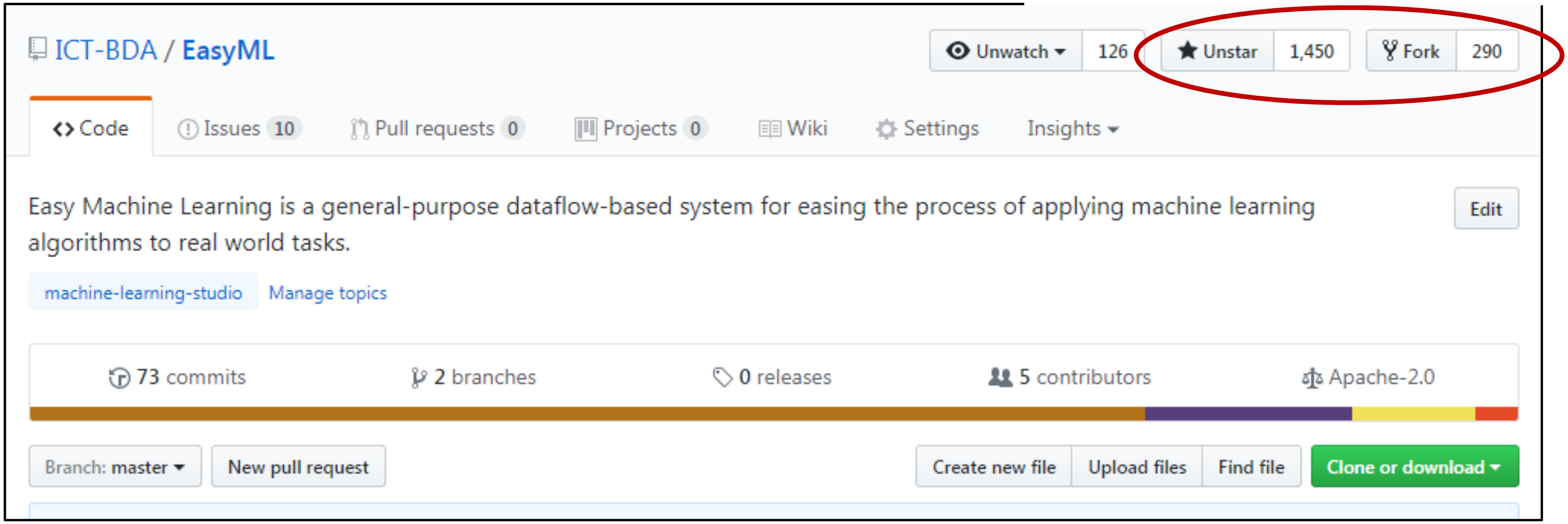
A Flexible and Powerful Parameter Server for large-scale machine learning

Java ★ 745 173 Built by ★ 724 stars this we

xuxueli / xxl-job

分布式任务调度平台XXL-JOB

Java ★ 1,999 794 Built by ★ 658 stars this we



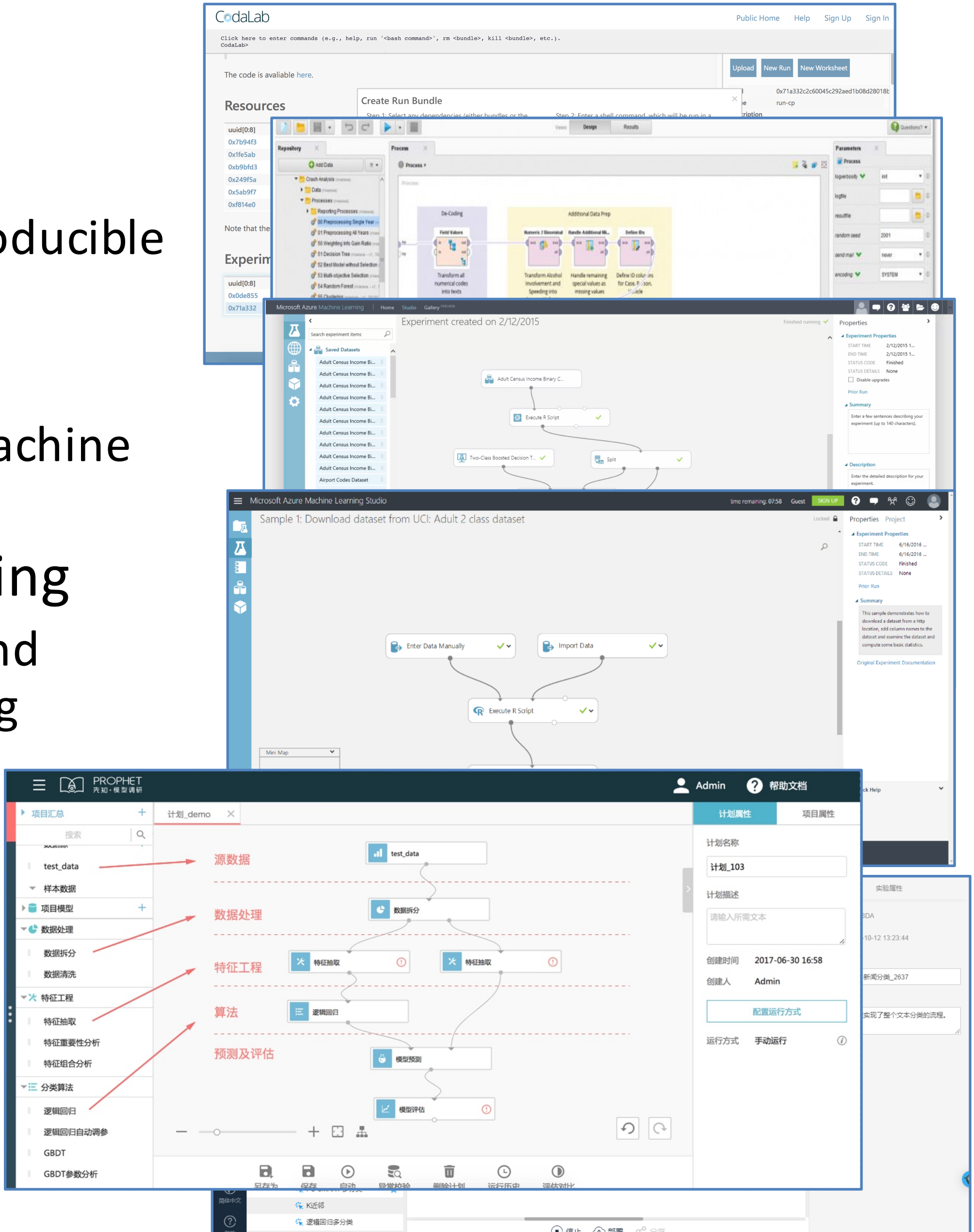
ICT-BDA / EasyML

Easy Machine Learning is a general-purpose dataflow-based system for easing the process of applying machine learning algorithms to real world tasks.

73 commits | 2 branches | 0 releases | 5 contributors | Apache-2.0

Related Systems

- Stanford CodaLab
  - A collaborative platform for reproducible research
- Rapid Miner Studio
  - Interactive GUI and integrated machine learning algorithms
- Microsoft Azure machine learning
  - GUI-based IDE for constructing and operationalizing machine learning workflow on Azure
- Alibaba 御膳房 / DT PAI
- The Fourth Paradigm Prophet



Summary

- Ease the process of using machine learning
  - Machine learning process as dataflow DAG
  - Interactive GUI for designing and managing ML tasks
  - Deployed as a web service
    - Resource management
    - Task design
    - Task monitoring
    - Result reusing
  - Source code @Github
    https://github.com/ICT-BDA/EasyML

EasyML Team

- Xinjie Chen, ICT CAS
- Zhaohui Li, ICT CAS
- Tianyou Guo, Sogou Inc.
- Jianpeng Hou, Google China
- Ping Li, Tencent Wechat
- Jiashuo Cao, CUIT
- Dong Huang, UCAS
- Xiaohui Yan
- Xueqi Cheng, ICT CAS

Thanks!

junxu@ict.ac.cn

http://www.bigdatalab.ac.cn/~junxu