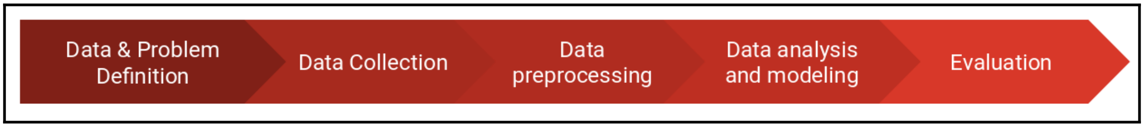
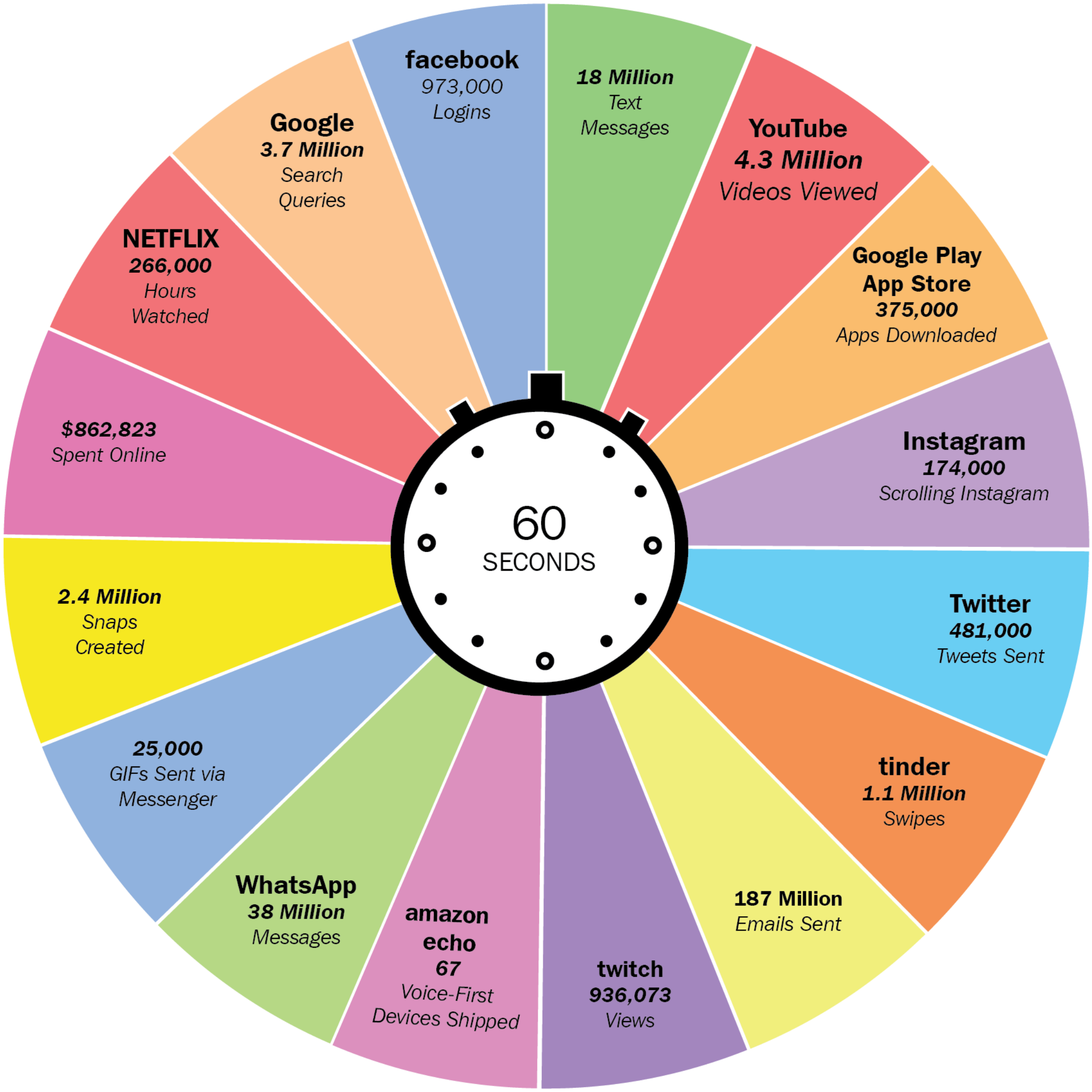
Chapter 1: Applied Machine Learning Quick Start



2018 This Is What Happens In An Internet Minute



60 SECONDS

18 Million
Text Messages

YouTube
4.3 Million
Videos Viewed

Google Play App Store
375,000
Apps Downloaded

Instagram
174,000
Scrolling Instagram

Twitter
481,000
Tweets Sent

tinder
1.1 Million
Swipes

187 Million
Emails Sent

twitch
936,073
Views

amazon echo
67
Voice-First Devices Shipped

WhatsApp
38 Million
Messages

25,000
GIFs Sent via Messenger

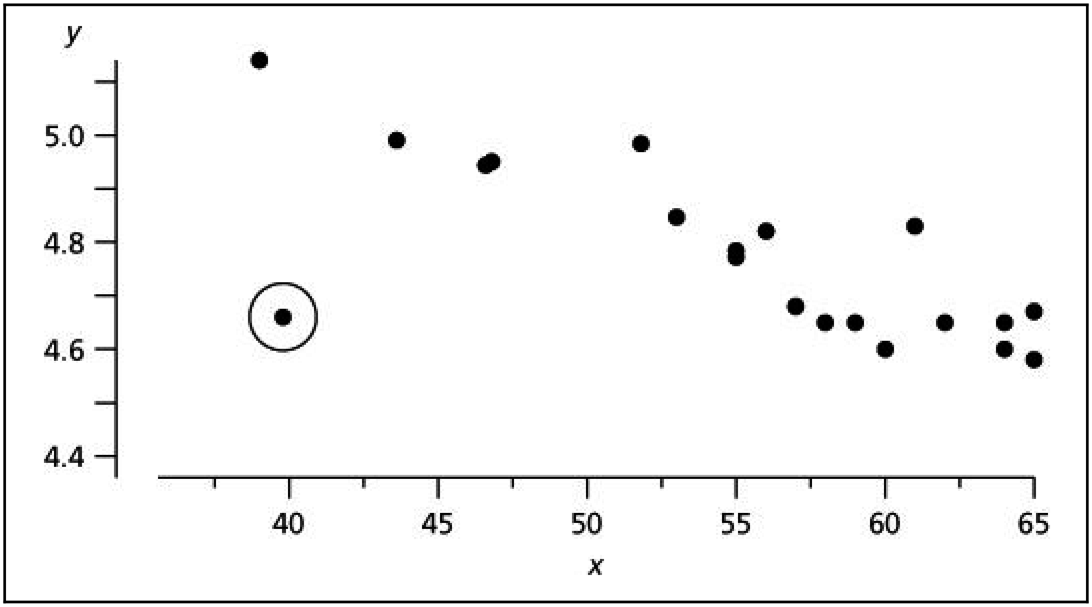
2.4 Million
Snaps Created

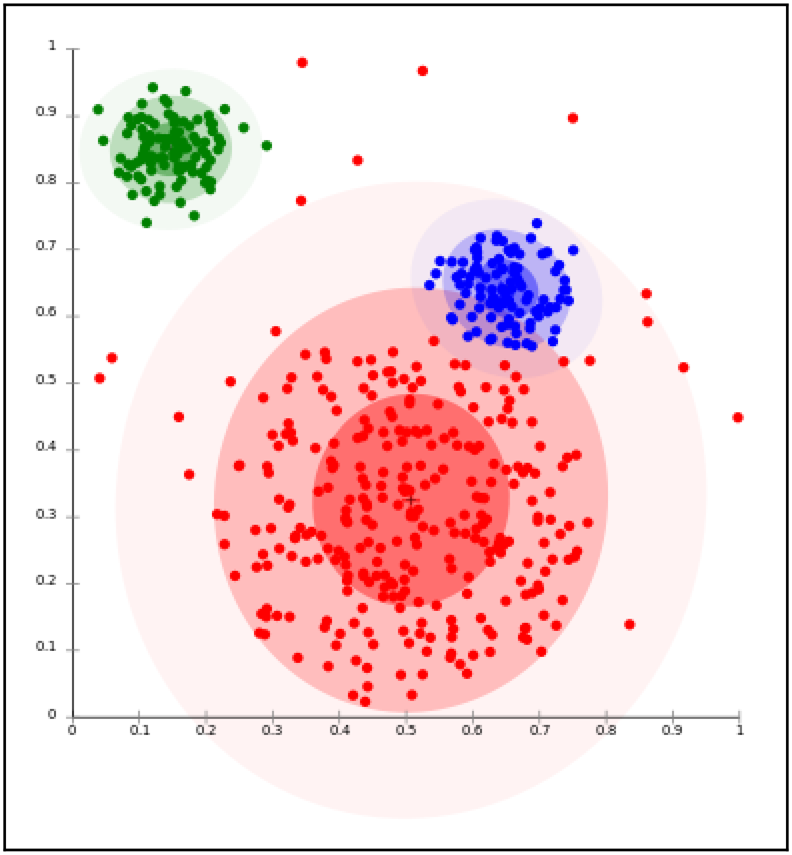
$862,823
Spent Online

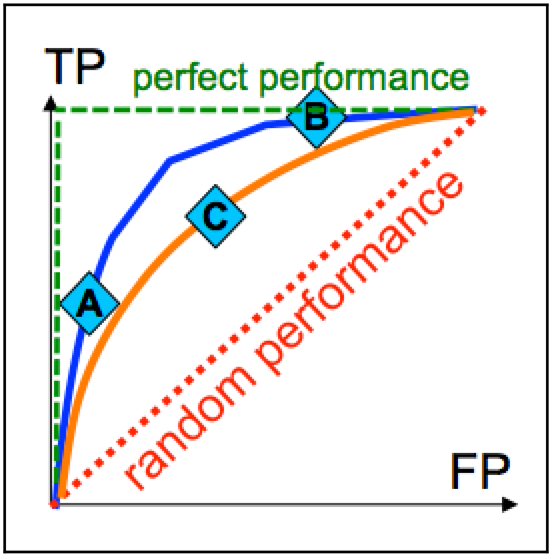
NETFLIX
266,000
Hours Watched

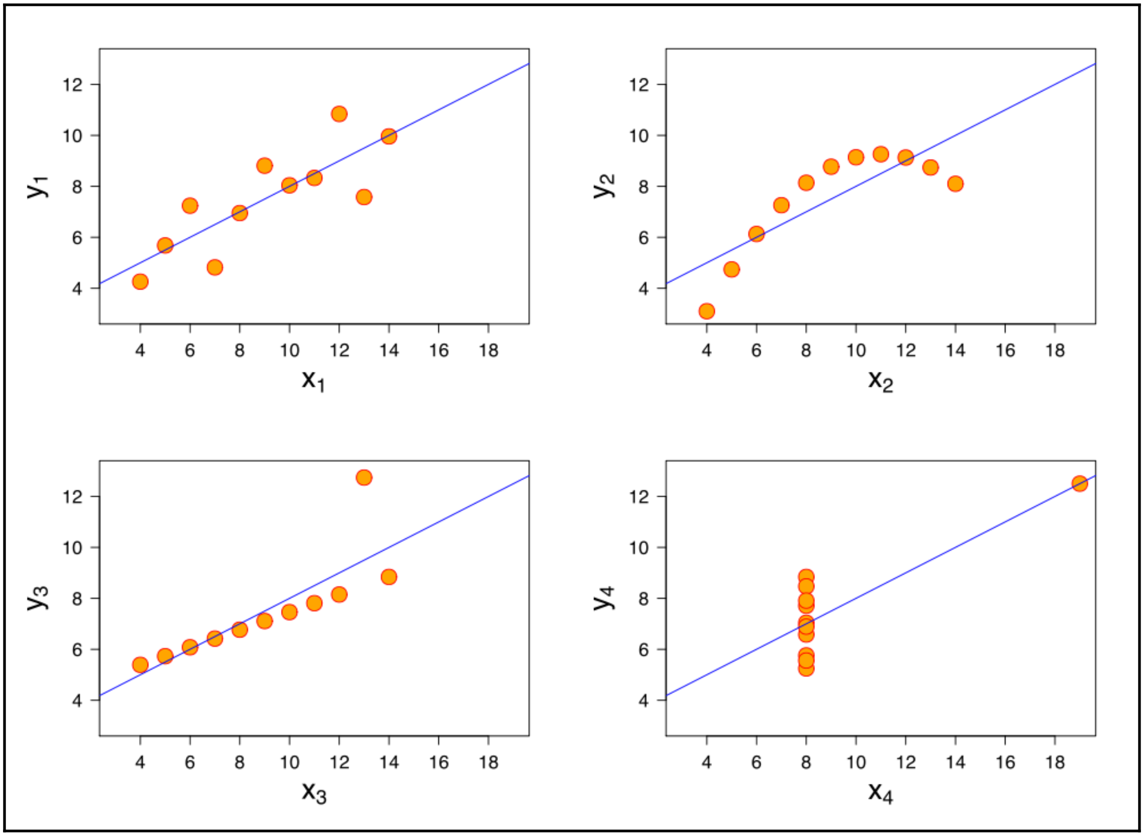
Google
3.7 Million
Search Queries

facebook
973,000
Logins

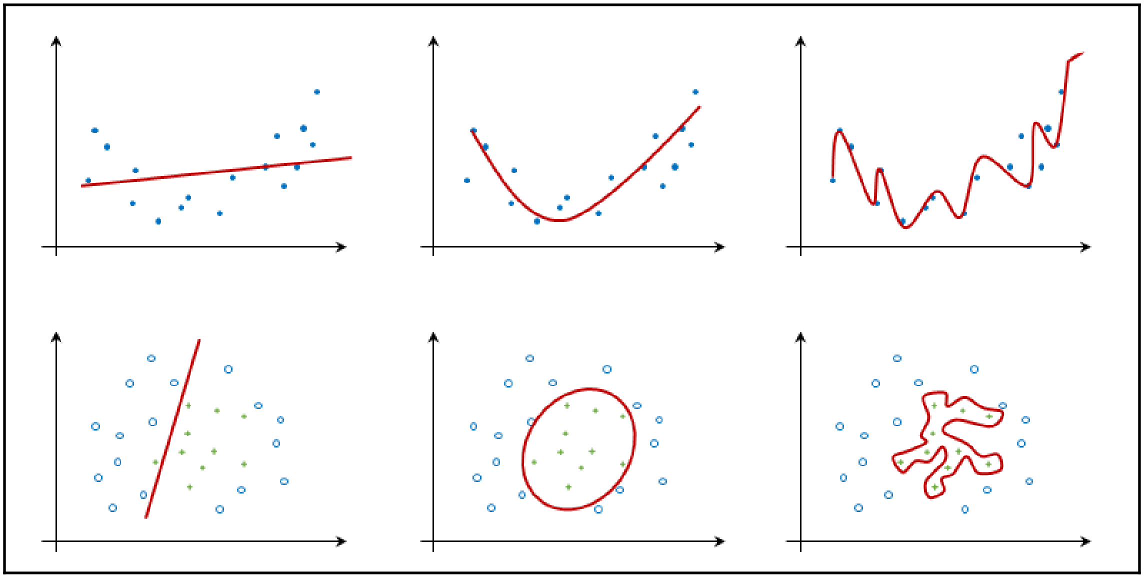

y
5.0
4.8
4.6
4.4
40
45
50
55
60
65
x

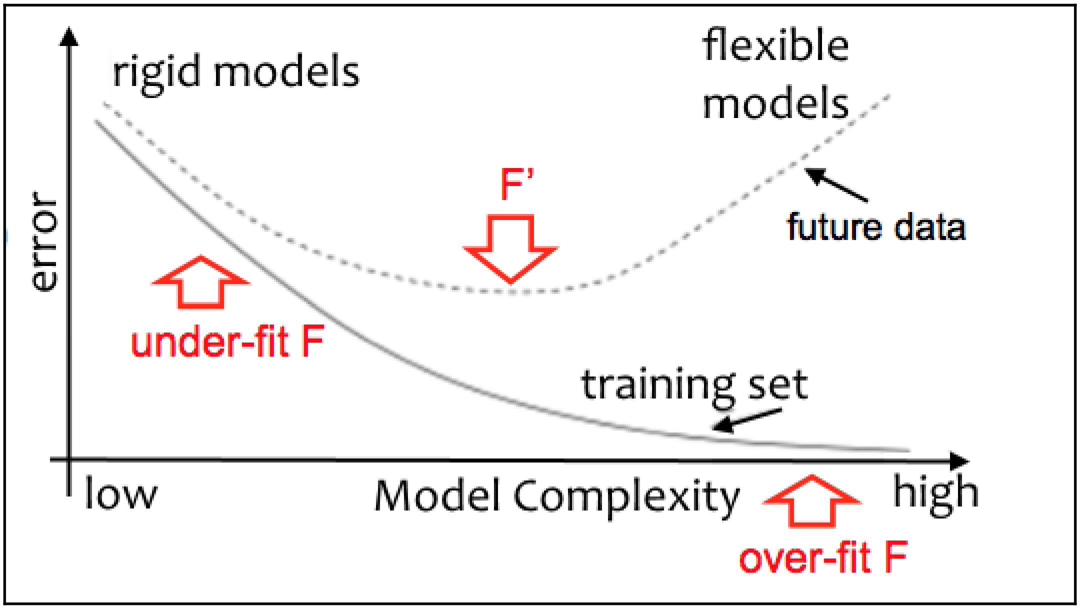

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

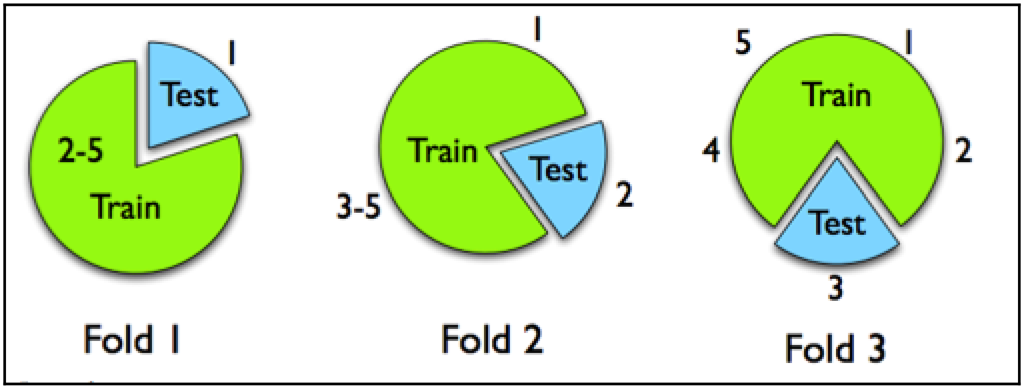

TP
perfect performance
B
C
A
random performance
FP



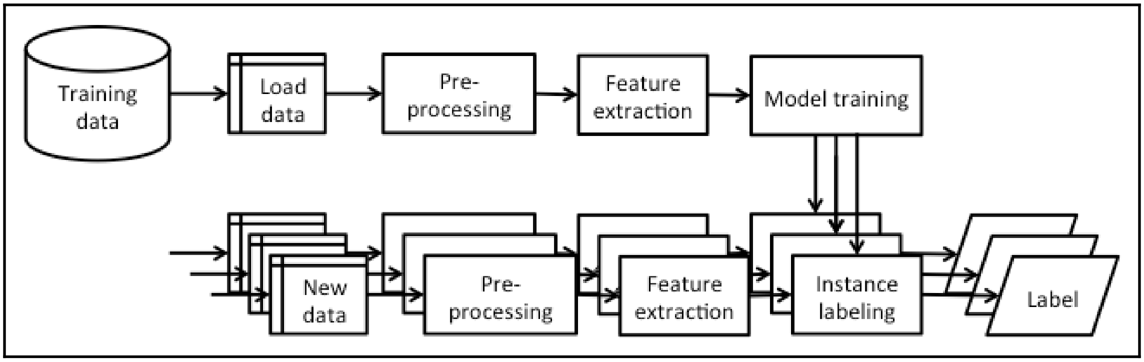
y$_1$
12
10
8
6
4
4 6 8 10 12 14 16 18
x$_1$
y$_2$
12
10
8
6
4
4 6 8 10 12 14 16 18
x$_2$
y$_3$
12
10
8
6
4
4 6 8 10 12 14 16 18
x$_3$
y$_4$
12
10
8
6
4
4 6 8 10 12 14 16 18
x$_4$



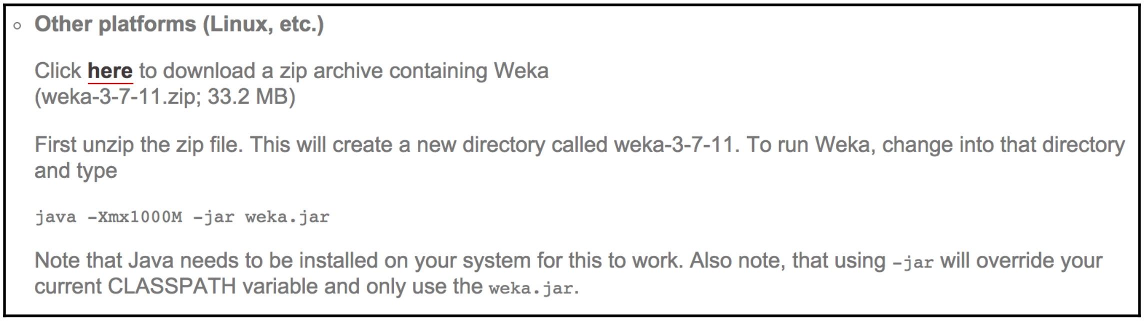

rigid models
flexible models
F'
future data
error
under-fit F
training set
low
Model Complexity
high
over-fit F


Test
1
2-5
Train
Fold 1
1
Train
Test
2
3-5
Fold 2
5
1
Train
4
2
Test
3
Fold 3

Chapter 2: Java Libraries and Platforms for Machine Learning



Chapter 3: Basic Algorithms – Classification, Regression, and Clustering



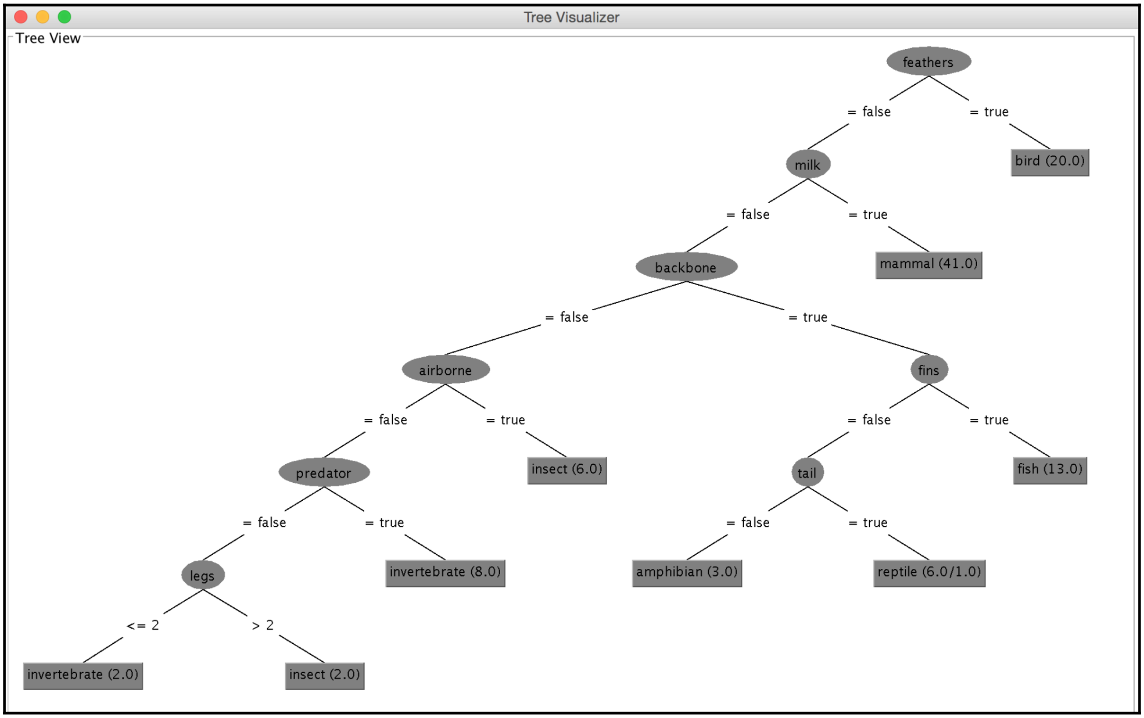
- **Other platforms (Linux, etc.)**

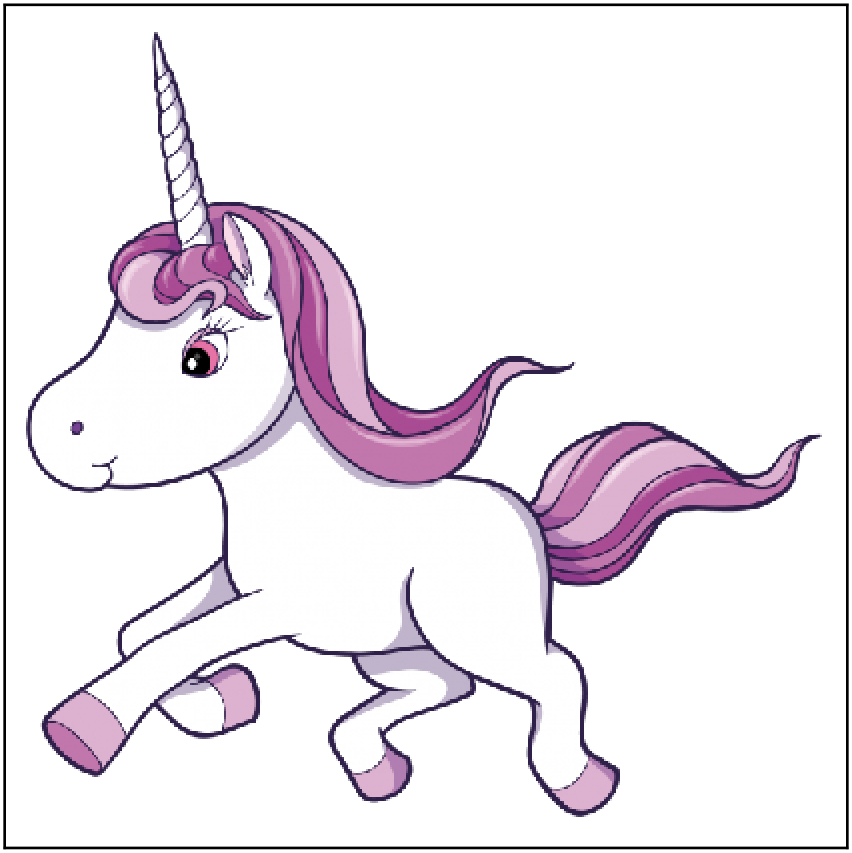
  Click **here** to download a zip archive containing Weka (weka-3-7-11.zip; 33.2 MB)

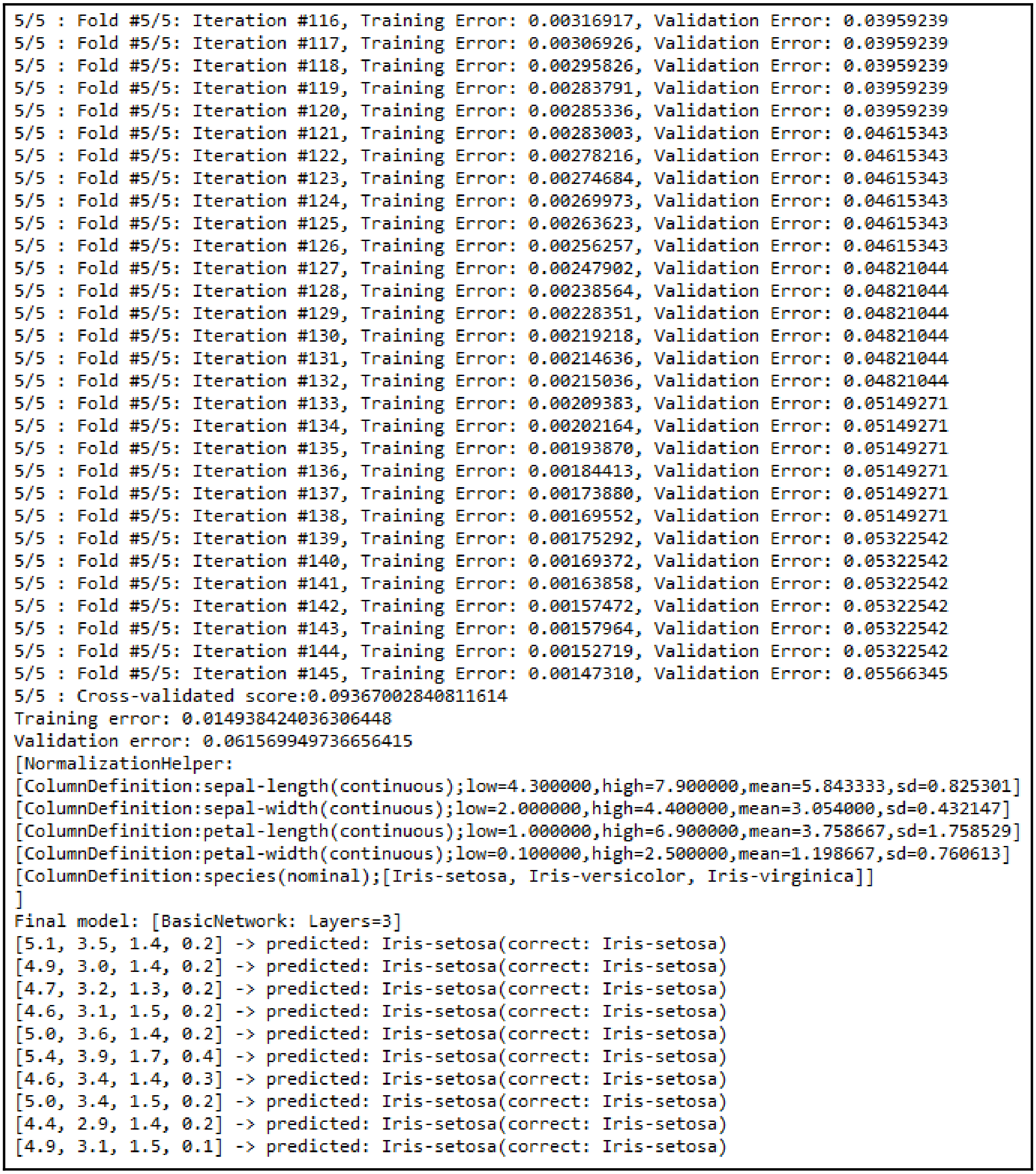
  First unzip the zip file. This will create a new directory called weka-3-7-11. To run Weka, change into that directory and type

  ```
  java -Xmx1000M -jar weka.jar
  ```

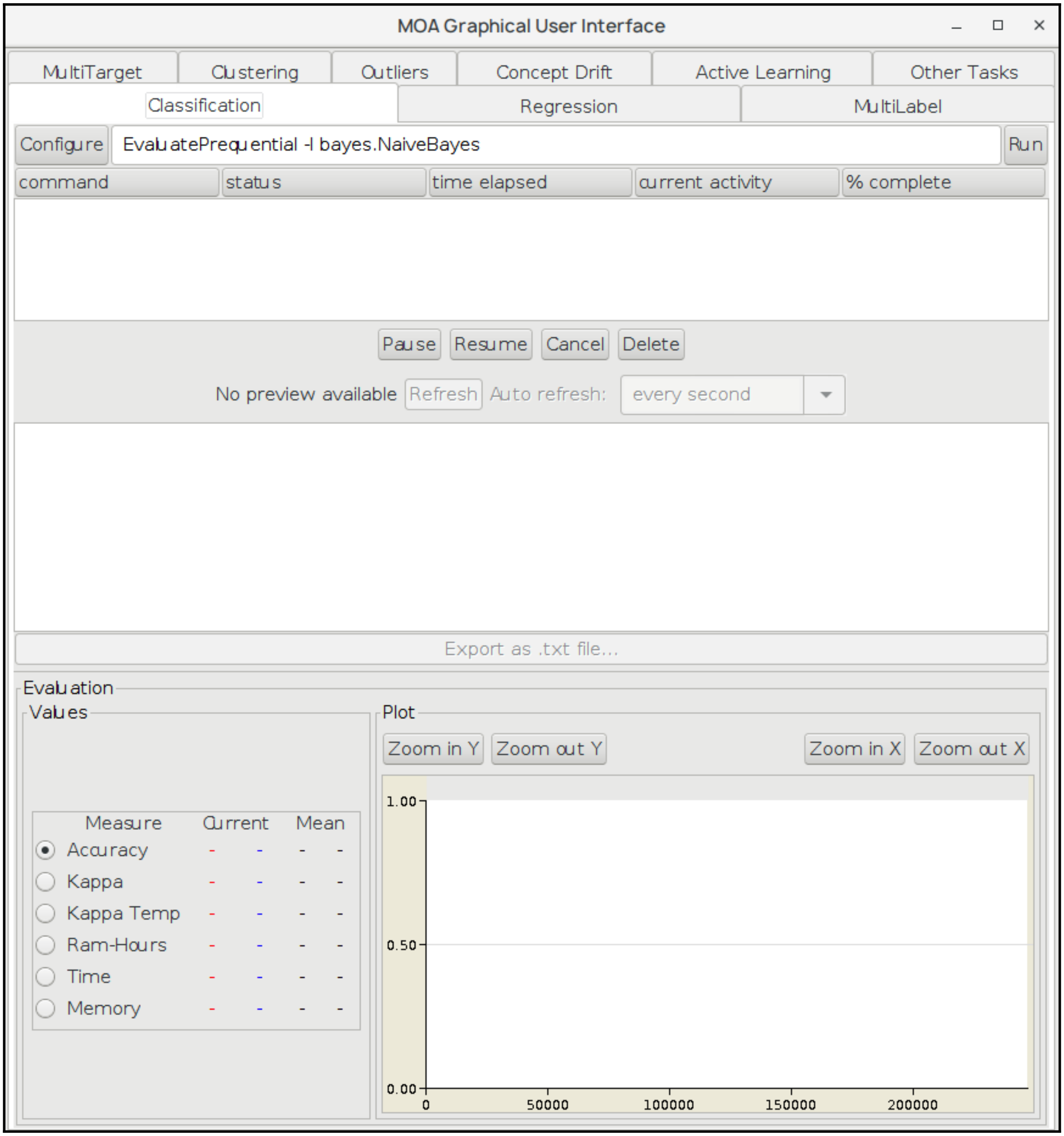
  Note that Java needs to be installed on your system for this to work. Also note, that using `-jar` will override your current CLASSPATH variable and only use the `weka.jar`.





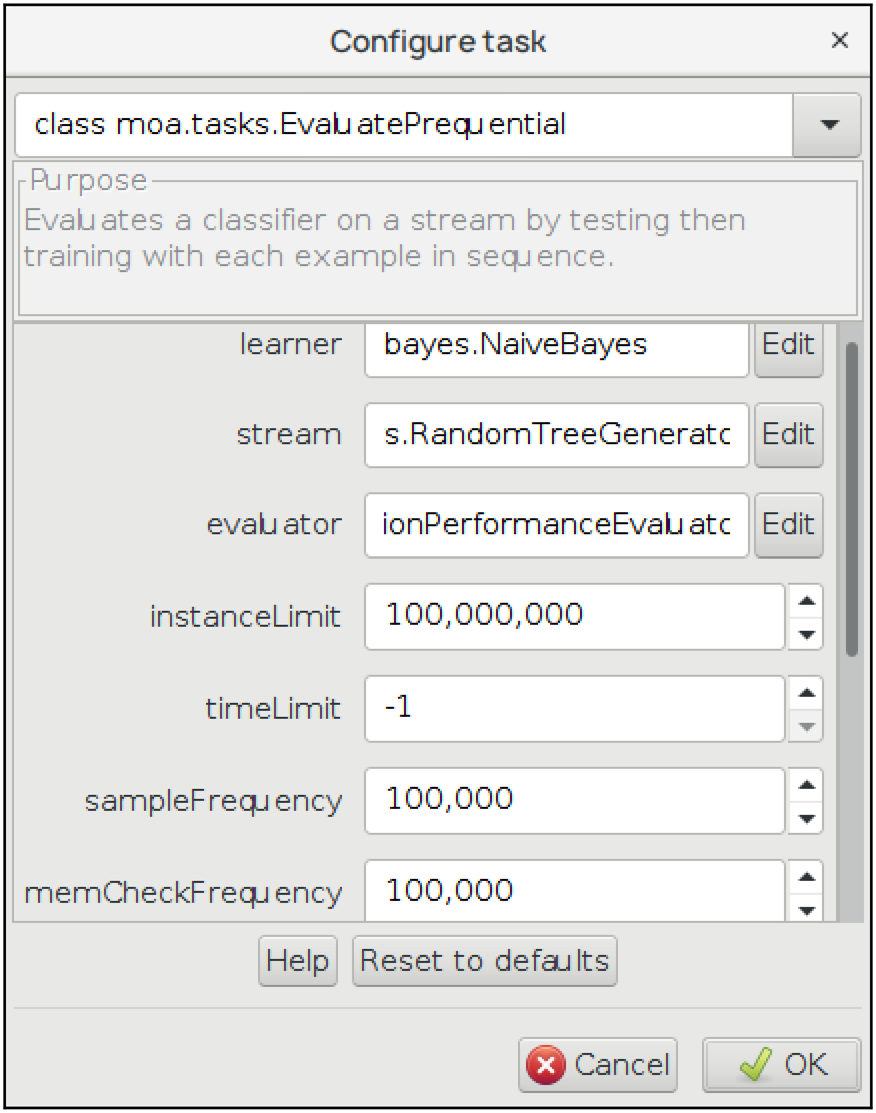


```
5/5 : Fold #5/5: Iteration #116, Training Error: 0.00316917, Validation Error: 0.03959239
5/5 : Fold #5/5: Iteration #117, Training Error: 0.00306926, Validation Error: 0.03959239
5/5 : Fold #5/5: Iteration #118, Training Error: 0.00295826, Validation Error: 0.03959239
5/5 : Fold #5/5: Iteration #119, Training Error: 0.00283791, Validation Error: 0.03959239
5/5 : Fold #5/5: Iteration #120, Training Error: 0.00285336, Validation Error: 0.03959239
5/5 : Fold #5/5: Iteration #121, Training Error: 0.00283003, Validation Error: 0.04615343
5/5 : Fold #5/5: Iteration #122, Training Error: 0.00278216, Validation Error: 0.04615343
5/5 : Fold #5/5: Iteration #123, Training Error: 0.00274684, Validation Error: 0.04615343
5/5 : Fold #5/5: Iteration #124, Training Error: 0.00269973, Validation Error: 0.04615343
5/5 : Fold #5/5: Iteration #125, Training Error: 0.00263623, Validation Error: 0.04615343
5/5 : Fold #5/5: Iteration #126, Training Error: 0.00256257, Validation Error: 0.04615343
5/5 : Fold #5/5: Iteration #127, Training Error: 0.00247902, Validation Error: 0.04821044
5/5 : Fold #5/5: Iteration #128, Training Error: 0.00238564, Validation Error: 0.04821044
5/5 : Fold #5/5: Iteration #129, Training Error: 0.00228351, Validation Error: 0.04821044
5/5 : Fold #5/5: Iteration #130, Training Error: 0.00219218, Validation Error: 0.04821044
5/5 : Fold #5/5: Iteration #131, Training Error: 0.00214636, Validation Error: 0.04821044
5/5 : Fold #5/5: Iteration #132, Training Error: 0.00215036, Validation Error: 0.04821044
5/5 : Fold #5/5: Iteration #133, Training Error: 0.00209383, Validation Error: 0.05149271
5/5 : Fold #5/5: Iteration #134, Training Error: 0.00202164, Validation Error: 0.05149271
5/5 : Fold #5/5: Iteration #135, Training Error: 0.00193870, Validation Error: 0.05149271
5/5 : Fold #5/5: Iteration #136, Training Error: 0.00184413, Validation Error: 0.05149271
5/5 : Fold #5/5: Iteration #137, Training Error: 0.00173880, Validation Error: 0.05149271
5/5 : Fold #5/5: Iteration #138, Training Error: 0.00169552, Validation Error: 0.05149271
5/5 : Fold #5/5: Iteration #139, Training Error: 0.00175292, Validation Error: 0.05322542
5/5 : Fold #5/5: Iteration #140, Training Error: 0.00169372, Validation Error: 0.05322542
5/5 : Fold #5/5: Iteration #141, Training Error: 0.00163858, Validation Error: 0.05322542
5/5 : Fold #5/5: Iteration #142, Training Error: 0.00157472, Validation Error: 0.05322542
5/5 : Fold #5/5: Iteration #143, Training Error: 0.00157964, Validation Error: 0.05322542
5/5 : Fold #5/5: Iteration #144, Training Error: 0.00152719, Validation Error: 0.05322542
5/5 : Fold #5/5: Iteration #145, Training Error: 0.00147310, Validation Error: 0.05566345
5/5 : Cross-validated score:0.09367002840811614
Training error: 0.014938424036306448
Validation error: 0.061569949736656415
[NormalizationHelper:
[ColumnDefinition:sepal-length(continuous);low=4.300000,high=7.900000,mean=5.843333,sd=0.825301]
[ColumnDefinition:sepal-width(continuous);low=2.000000,high=4.400000,mean=3.054000,sd=0.432147]
[ColumnDefinition:petal-length(continuous);low=1.000000,high=6.900000,mean=3.758667,sd=1.758529]
[ColumnDefinition:petal-width(continuous);low=0.100000,high=2.500000,mean=1.198667,sd=0.760613]
[ColumnDefinition:species(nominal);[Iris-setosa, Iris-versicolor, Iris-virginica]]
]
Final model: [BasicNetwork: Layers=3]
[5.1, 3.5, 1.4, 0.2] -> predicted: Iris-setosa(correct: Iris-setosa)
[4.9, 3.0, 1.4, 0.2] -> predicted: Iris-setosa(correct: Iris-setosa)
[4.7, 3.2, 1.3, 0.2] -> predicted: Iris-setosa(correct: Iris-setosa)
[4.6, 3.1, 1.5, 0.2] -> predicted: Iris-setosa(correct: Iris-setosa)
[5.0, 3.6, 1.4, 0.2] -> predicted: Iris-setosa(correct: Iris-setosa)
[5.4, 3.9, 1.7, 0.4] -> predicted: Iris-setosa(correct: Iris-setosa)
[4.6, 3.4, 1.4, 0.3] -> predicted: Iris-setosa(correct: Iris-setosa)
[5.0, 3.4, 1.5, 0.2] -> predicted: Iris-setosa(correct: Iris-setosa)
[4.4, 2.9, 1.4, 0.2] -> predicted: Iris-setosa(correct: Iris-setosa)
[4.9, 3.1, 1.5, 0.1] -> predicted: Iris-setosa(correct: Iris-setosa)
```



```
EvaluatePrequential -l bayes.NaiveBayes
```

Measure	Current		Mean	
● Accuracy	-	-	-	-
○ Kappa	-	-	-	-
○ Kappa Temp	-	-	-	-
○ Ram-Hours	-	-	-	-
○ Time	-	-	-	-
○ Memory	-	-	-	-



class moa.tasks.EvaluatePrequential

Purpose

Evaluates a classifier on a stream by testing then training with each example in sequence.

learner bayes.NaiveBayes

stream s.RandomTreeGeneratc

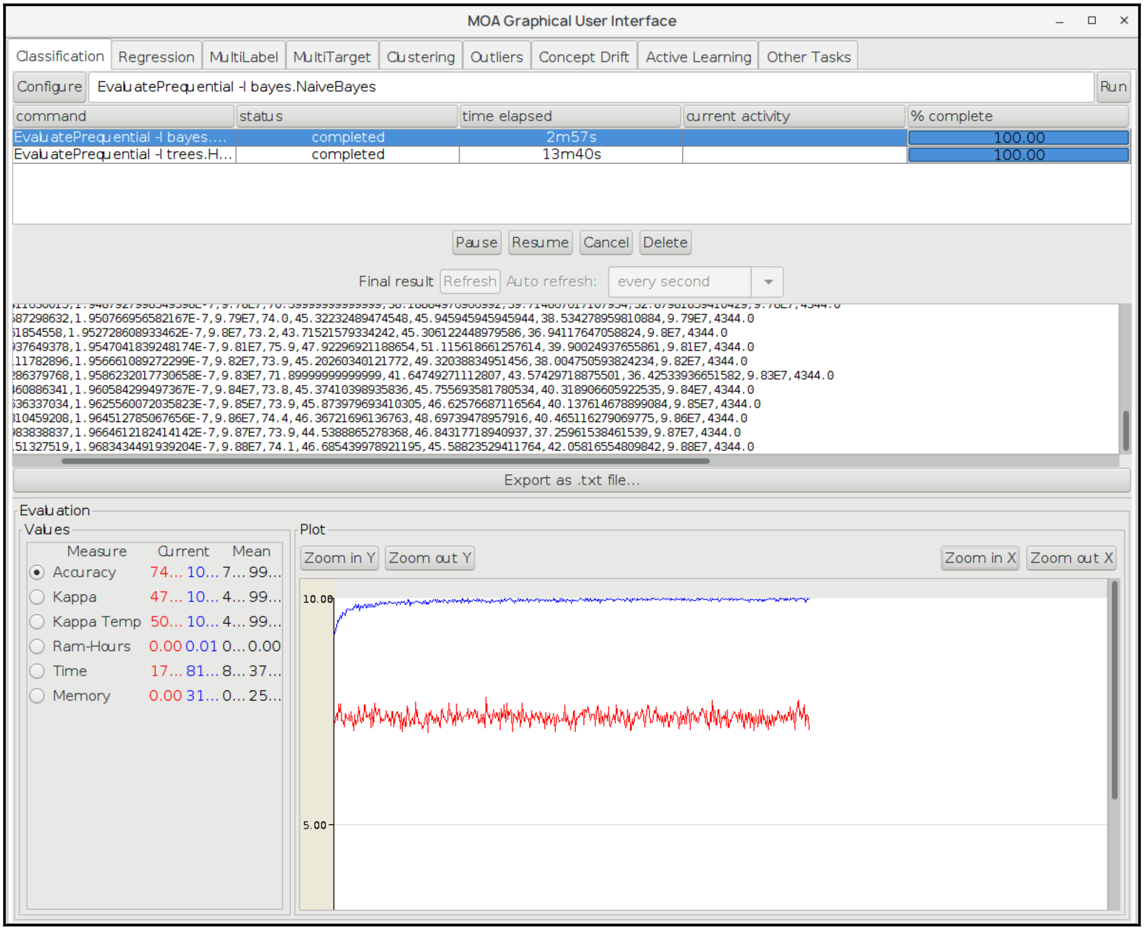
evaluator ionPerformanceEvaluatc

instanceLimit 100,000,000

timeLimit -1

sampleFrequency 100,000

memCheckFrequency 100,000



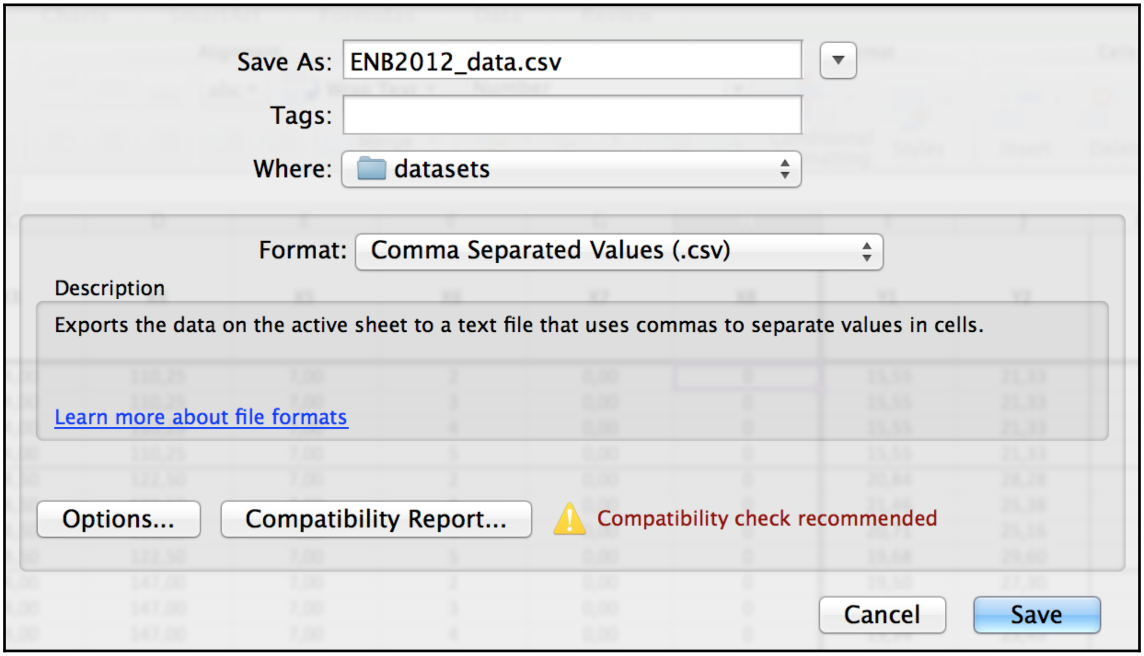
EvaluatePrequential -l bayes.NaiveBayes

command	status	time elapsed	current activity	% complete
EvaluatePrequential -l bayes....	completed	2m57s		100.00
EvaluatePrequential -l trees.H...	completed	13m40s		100.00

Final result Auto refresh: every second

```
587298632,1.950766956582167E-7,9.79E7,74.0,45.32232489474548,45.945945945945944,38.534278959810884,9.79E7,4344.0
51854558,1.952728608933462E-7,9.8E7,73.2,43.71521579334242,45.306122448979586,36.94117647058824,9.8E7,4344.0
937649378,1.9547041839248174E-7,9.81E7,75.9,47.92296921188654,51.115618661257614,39.90024937655861,9.81E7,4344.0
.11782896,1.956661089272299E-7,9.82E7,73.9,45.20260340121772,49.32038834951456,38.004750593824234,9.82E7,4344.0
286379768,1.9586232017730658E-7,9.83E7,71.89999999999999,41.64749271112807,43.57429718875501,36.42533936651582,9.83E7,4344.0
460886341,1.960584299497367E-7,9.84E7,73.8,45.37410398935836,45.755693581780534,40.318906605922535,9.84E7,4344.0
536337034,1.9625560072035823E-7,9.85E7,73.9,45.873979693410305,46.62576687116564,40.137614678899084,9.85E7,4344.0
310459208,1.964512785067656E-7,9.86E7,74.4,46.36721696136763,48.69739478957916,40.465116279069775,9.86E7,4344.0
983838837,1.9664612182414142E-7,9.87E7,73.9,44.5388865278368,46.84317718940937,37.25961538461539,9.87E7,4344.0
.51327519,1.9683434491939204E-7,9.88E7,74.1,46.685439978921195,45.58823529411764,42.05816554809842,9.88E7,4344.0
```

	Measure	Current		Mean
●	Accuracy	74...	10...	7... 99...
○	Kappa	47...	10...	4... 99...
○	Kappa Temp	50...	10...	4... 99...
○	Ram-Hours	0.00	0.01	0... 0.00
○	Time	17...	81...	8... 37...
○	Memory	0.00	31...	0... 25...



Save As: ENB2012_data.csv

Tags:

Where: datasets

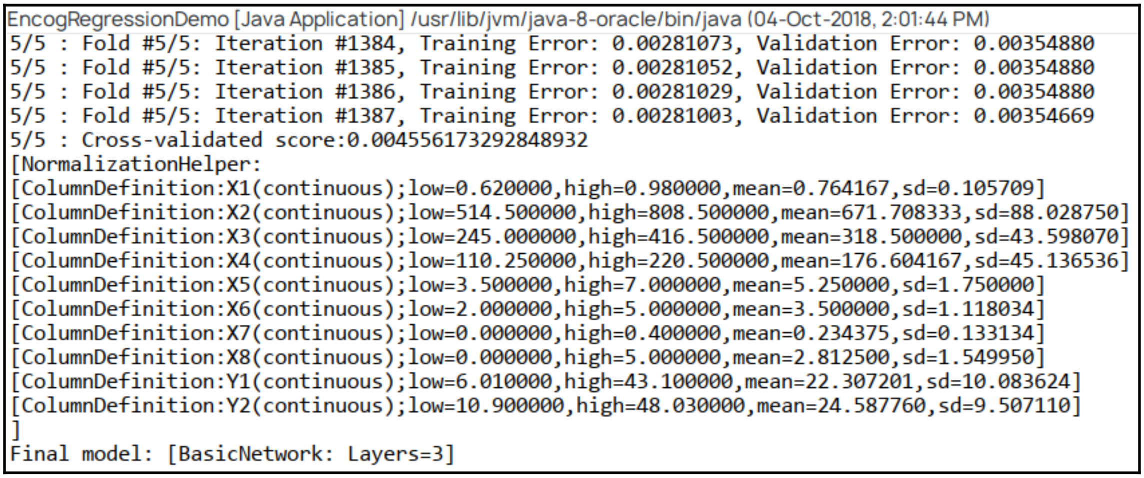
Format: Comma Separated Values (.csv)

Description

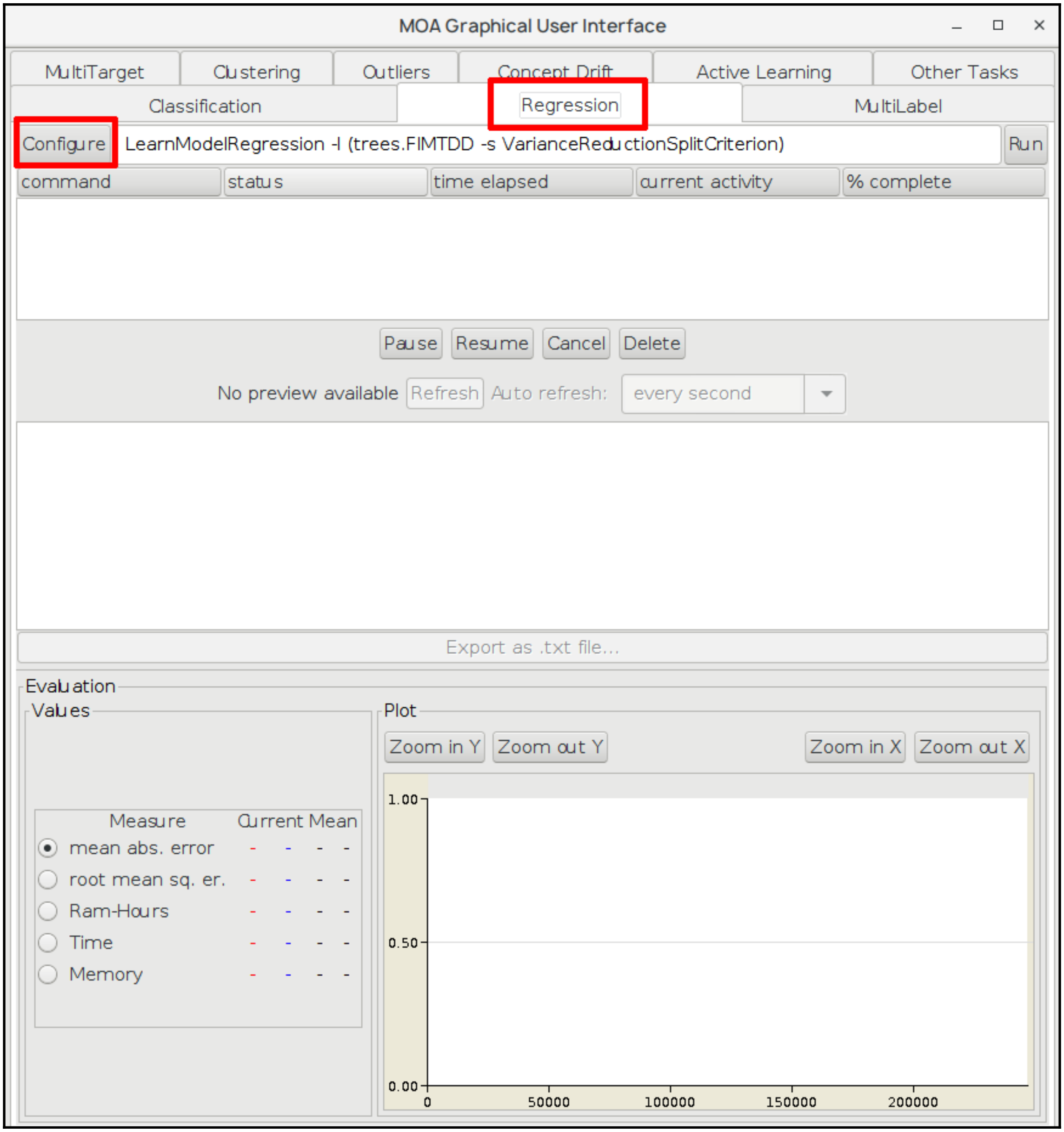
Exports the data on the active sheet to a text file that uses commas to separate values in cells.

Learn more about file formats

Compatibility check recommended

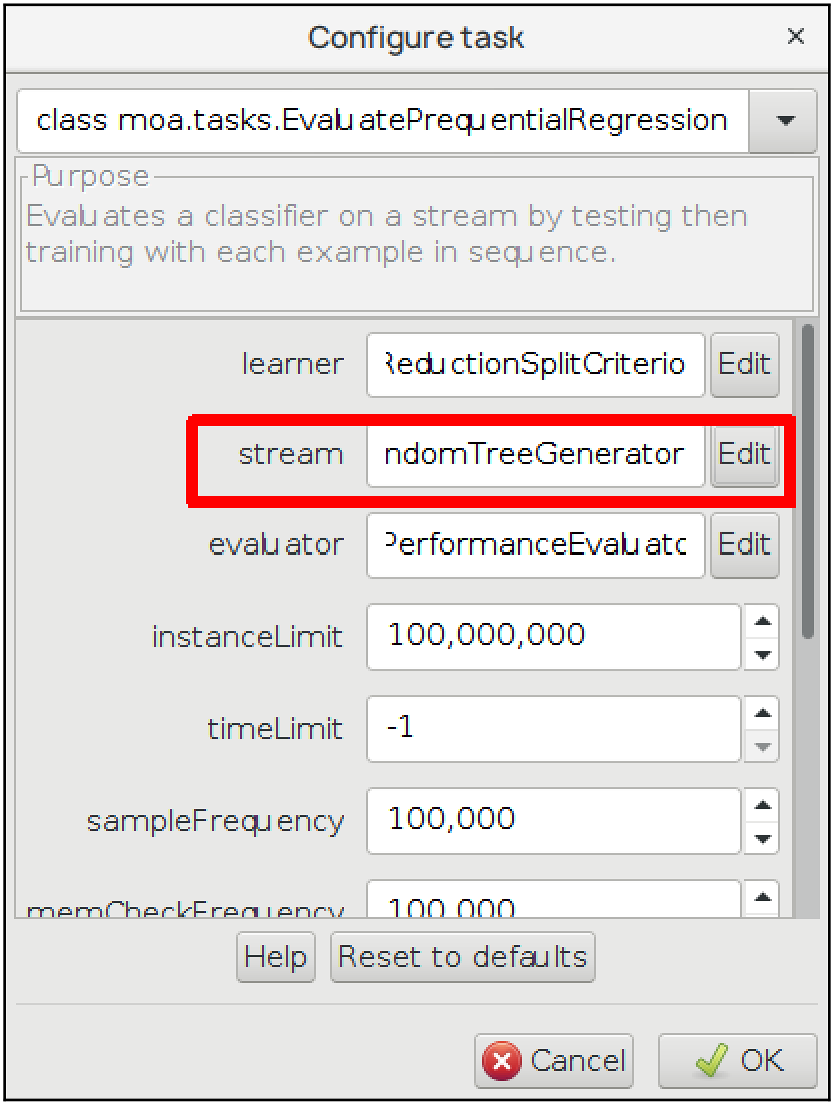


```
EncogRegressionDemo [Java Application] /usr/lib/jvm/java-8-oracle/bin/java (04-Oct-2018, 2:01:44 PM)
5/5 : Fold #5/5: Iteration #1384, Training Error: 0.00281073, Validation Error: 0.00354880
5/5 : Fold #5/5: Iteration #1385, Training Error: 0.00281052, Validation Error: 0.00354880
5/5 : Fold #5/5: Iteration #1386, Training Error: 0.00281029, Validation Error: 0.00354880
5/5 : Fold #5/5: Iteration #1387, Training Error: 0.00281003, Validation Error: 0.00354669
5/5 : Cross-validated score:0.004556173292848932
[NormalizationHelper:
[ColumnDefinition:X1(continuous);low=0.620000,high=0.980000,mean=0.764167,sd=0.105709]
[ColumnDefinition:X2(continuous);low=514.500000,high=808.500000,mean=671.708333,sd=88.028750]
[ColumnDefinition:X3(continuous);low=245.000000,high=416.500000,mean=318.500000,sd=43.598070]
[ColumnDefinition:X4(continuous);low=110.250000,high=220.500000,mean=176.604167,sd=45.136536]
[ColumnDefinition:X5(continuous);low=3.500000,high=7.000000,mean=5.250000,sd=1.750000]
[ColumnDefinition:X6(continuous);low=2.000000,high=5.000000,mean=3.500000,sd=1.118034]
[ColumnDefinition:X7(continuous);low=0.000000,high=0.400000,mean=0.234375,sd=0.133134]
[ColumnDefinition:X8(continuous);low=0.000000,high=5.000000,mean=2.812500,sd=1.549950]
[ColumnDefinition:Y1(continuous);low=6.010000,high=43.100000,mean=22.307201,sd=10.083624]
[ColumnDefinition:Y2(continuous);low=10.900000,high=48.030000,mean=24.587760,sd=9.507110]
]
Final model: [BasicNetwork: Layers=3]
```



LearnModelRegression -l (trees.FIMTDD -s VarianceReductionSplitCriterion)

Measure	Current	Mean		
● mean abs. error	-	-	-	-
○ root mean sq. er.	-	-	-	-
○ Ram-Hours	-	-	-	-
○ Time	-	-	-	-
○ Memory	-	-	-	-



Configure task

class moa.tasks.EvaluatePrequentialRegression

Purpose

Evaluates a classifier on a stream by testing then training with each example in sequence.

learner: ReductionSplitCriterio

stream: ndomTreeGenerator

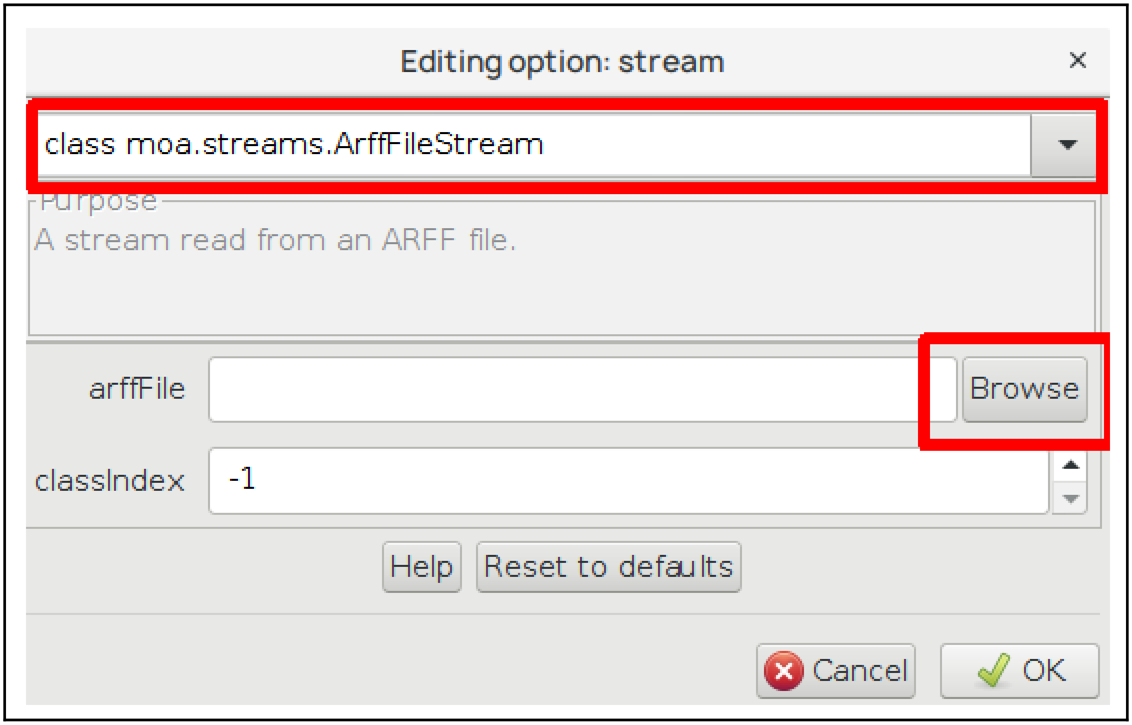
evaluator: PerformanceEvaluatc

instanceLimit: 100,000,000

timeLimit: -1

sampleFrequency: 100,000

memCheckFrequency: 100,000



Editing option: stream

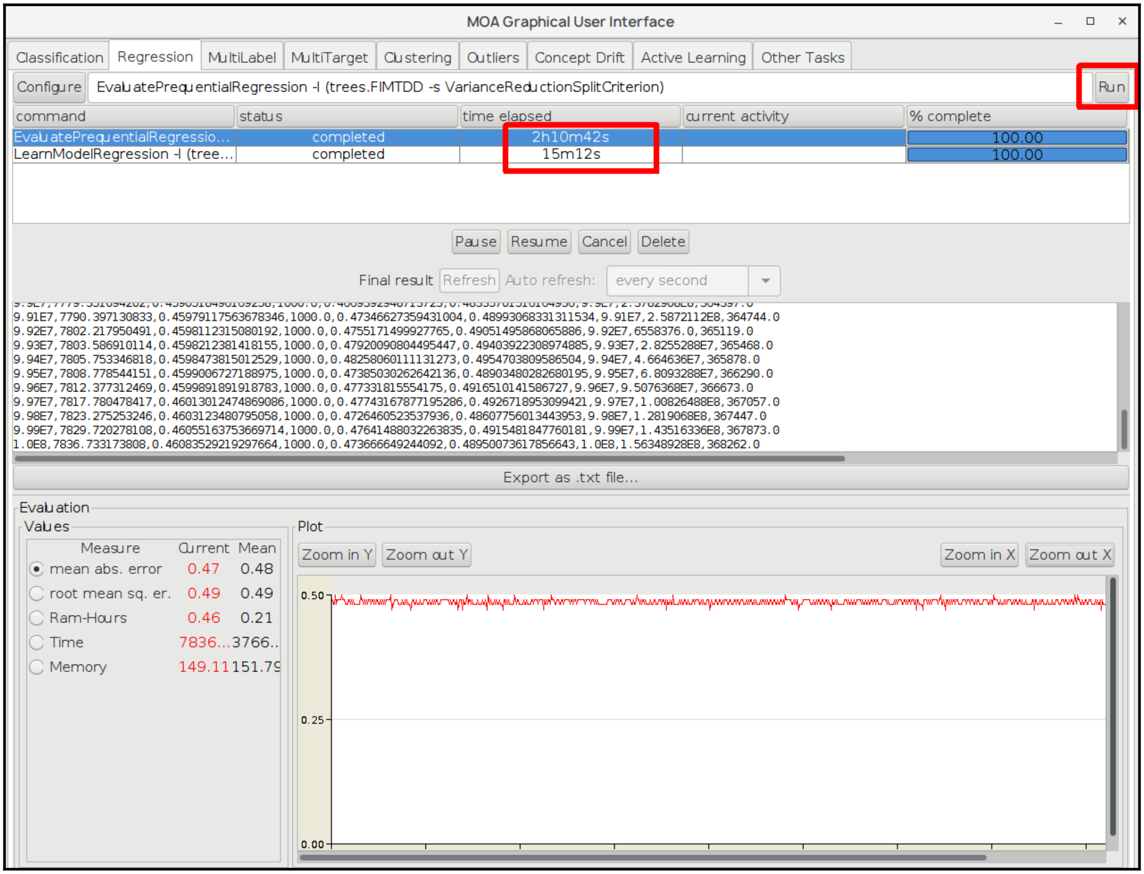
class moa.streams.ArffFileStream

Purpose

A stream read from an ARFF file.

arffFile

classIndex -1

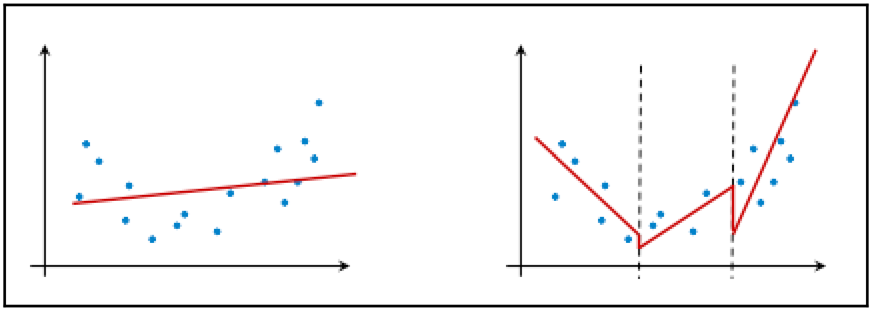


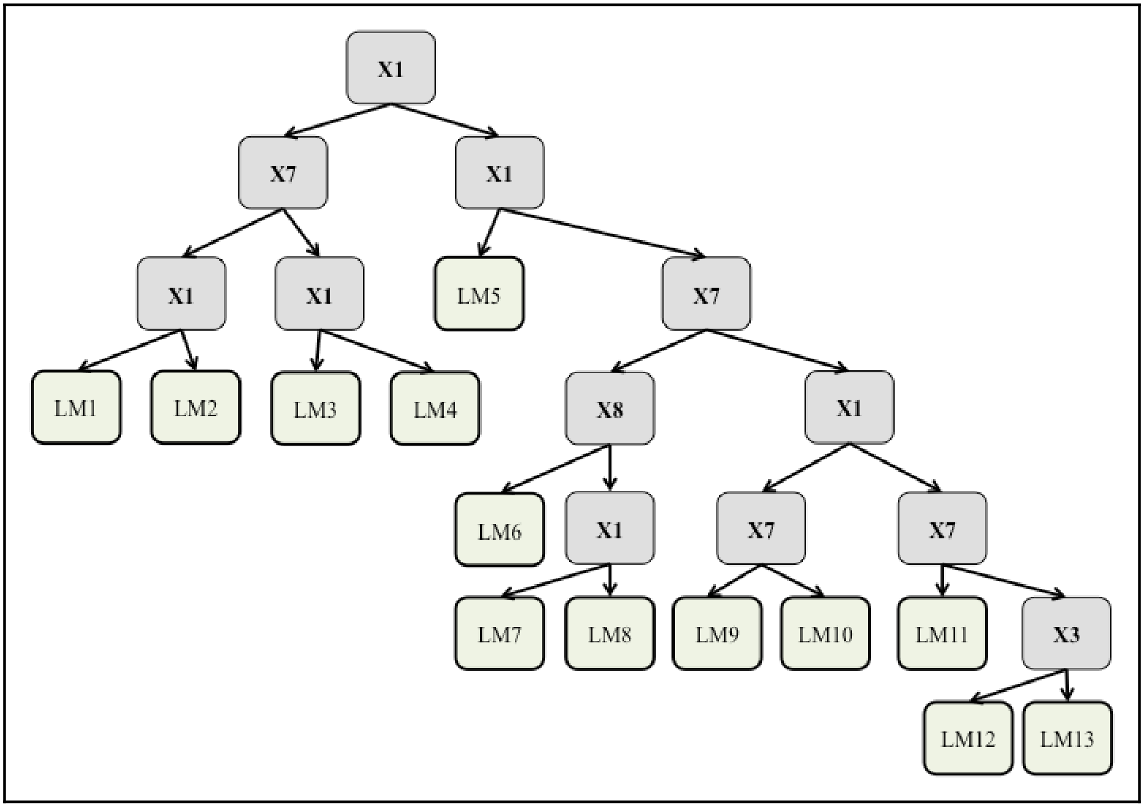
EvaluatePrequentialRegression -l (trees.FIMTDD -s VarianceReductionSplitCriterion)

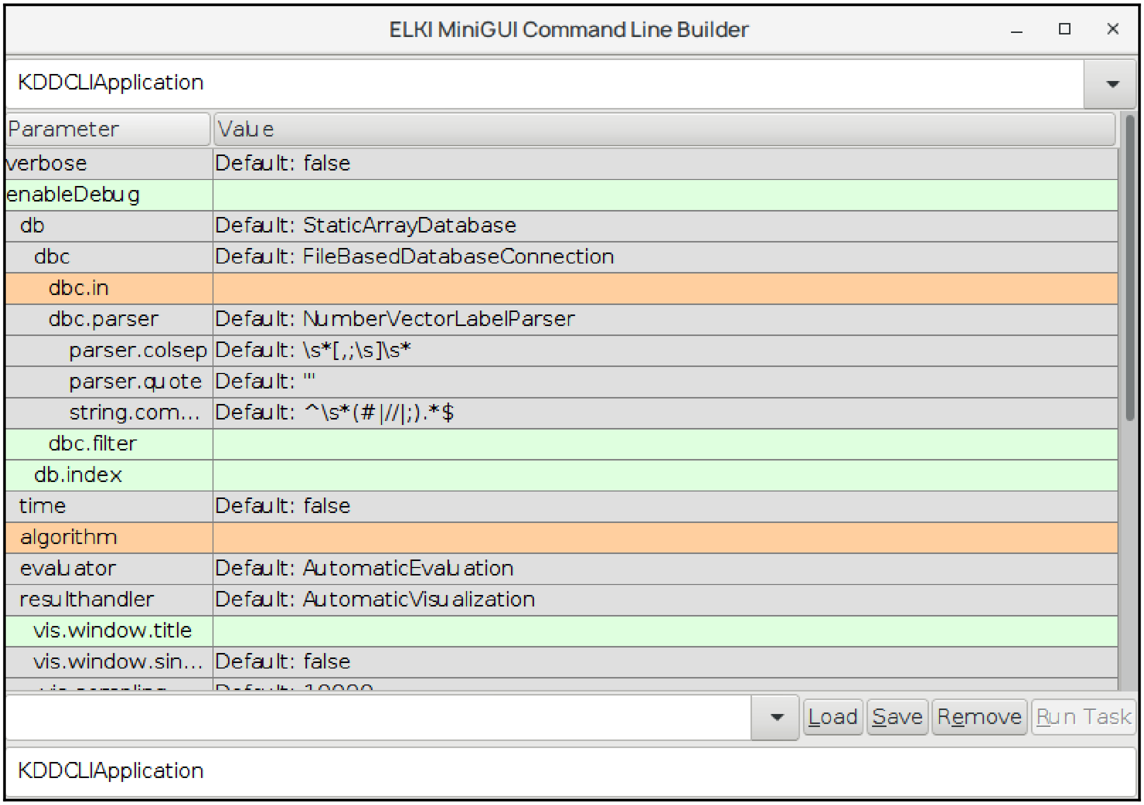
command	status	time elapsed	current activity	% complete
EvaluatePrequentialRegressio...	completed	2h10m42s		100.00
LearnModelRegression -l (tree...	completed	15m12s		100.00

```
9.9E7,7779.331094202,0.4590318490169258,1000.0,0.4669392948713723,0.4833370131010493 0,9.9E7,2.3782908E8,364397.0
9.91E7,7790.397130833,0.45979117563678346,1000.0,0.47346627359431004,0.48993068331311534,9.91E7,2.5872112E8,364744.0
9.92E7,7802.217950491,0.4598112315080192,1000.0,0.4755171499927765,0.49051495868065886,9.92E7,6558376.0,365119.0
9.93E7,7803.586910114,0.4598212381418155,1000.0,0.47920090804495447,0.49403922308974885,9.93E7,2.8255288E7,365468.0
9.94E7,7805.753346818,0.4598473815012529,1000.0,0.48258060111131273,0.4954703809586504,9.94E7,4.664636E7,365878.0
9.95E7,7808.778544151,0.4599006727188975,1000.0,0.47385030262642136,0.48903480282680195,9.95E7,6.8093288E7,366290.0
9.96E7,7812.377312469,0.4599891891918783,1000.0,0.47733181555417 5,0.4916510141586727,9.96E7,9.5076368E7,366673.0
9.97E7,7817.780478417,0.46013012474869086,1000.0,0.47743167877195286,0.4926718953099421,9.97E7,1.00826488E8,367057.0
9.98E7,7823.275253246,0.4603123480795058,1000.0,0.4726460523537936,0.48607756013443953,9.98E7,1.2819068E8,367447.0
9.99E7,7829.720278108,0.46055163753669714,1000.0,0.47641488032263835,0.4915481847760181,9.99E7,1.43516336E8,367873.0
1.0E8,7836.733173808,0.46083529219297664,1000.0,0.473666649244092,0.48950073617856643,1.0E8,1.56348928E8,368262.0
```

	Measure	Current	Mean
●	mean abs. error	0.47	0.48
○	root mean sq. er.	0.49	0.49
○	Ram-Hours	0.46	0.21
○	Time	7836...	3766..
○	Memory	149.11	151.79



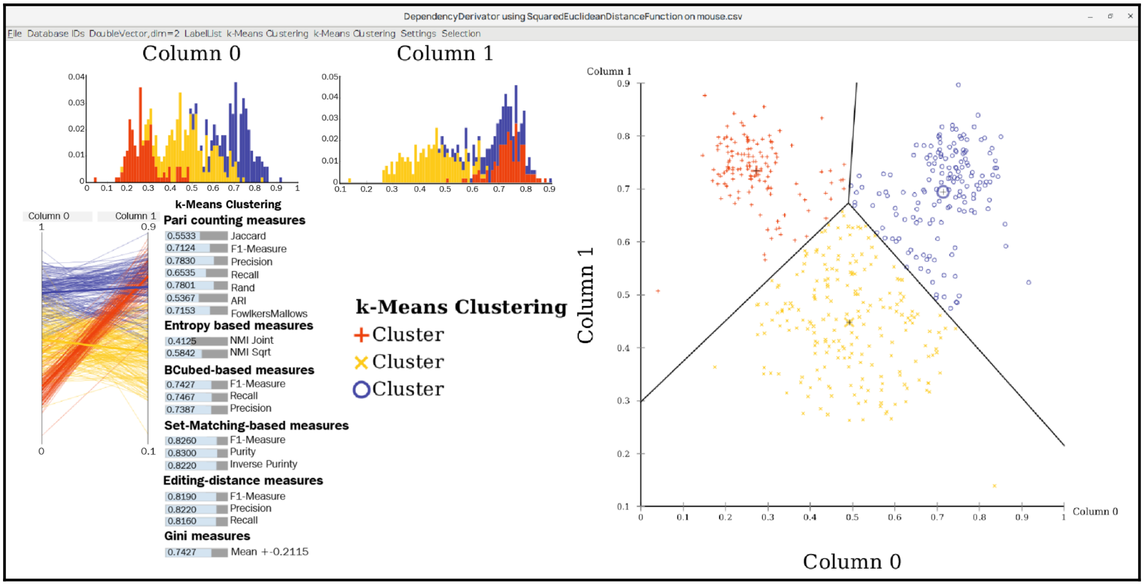

X1
X7
X1
X1
X1
LM5
X7
LM1
LM2
LM3
LM4
X8
X1
LM6
X1
X7
X7
LM7
LM8
LM9
LM10
LM11
X3
LM12
LM13



KDDCLIApplication

Parameter	Value
verbose	Default: false
enableDebug	
db	Default: StaticArrayDatabase
dbc	Default: FileBasedDatabaseConnection
dbc.in	
dbc.parser	Default: NumberVectorLabelParser
parser.colsep	Default: \s*[,;\s]\s*
parser.quote	Default: "'
string.com...	Default: ^\s*(#|//|;).*$
dbc.filter	
db.index	
time	Default: false
algorithm	
evaluator	Default: AutomaticEvaluation
resulthandler	Default: AutomaticVisualization
vis.window.title	
vis.window.sin...	Default: false
vis.sampling	Default: 10000

KDDCLIApplication



k-Means Clustering

Pari counting measures

0.5533	Jaccard
0.7124	F1-Measure
0.7830	Precision
0.6535	Recall
0.7801	Rand
0.5367	ARI
0.7153	FowlkersMallows

Entropy based measures

0.4125	NMI Joint
0.5842	NMI Sqrt

BCubed-based measures

0.7427	F1-Measure
0.7467	Recall
0.7387	Precision

Set-Matching-based measures

0.8260	F1-Measure
0.8300	Purity
0.8220	Inverse Purinty

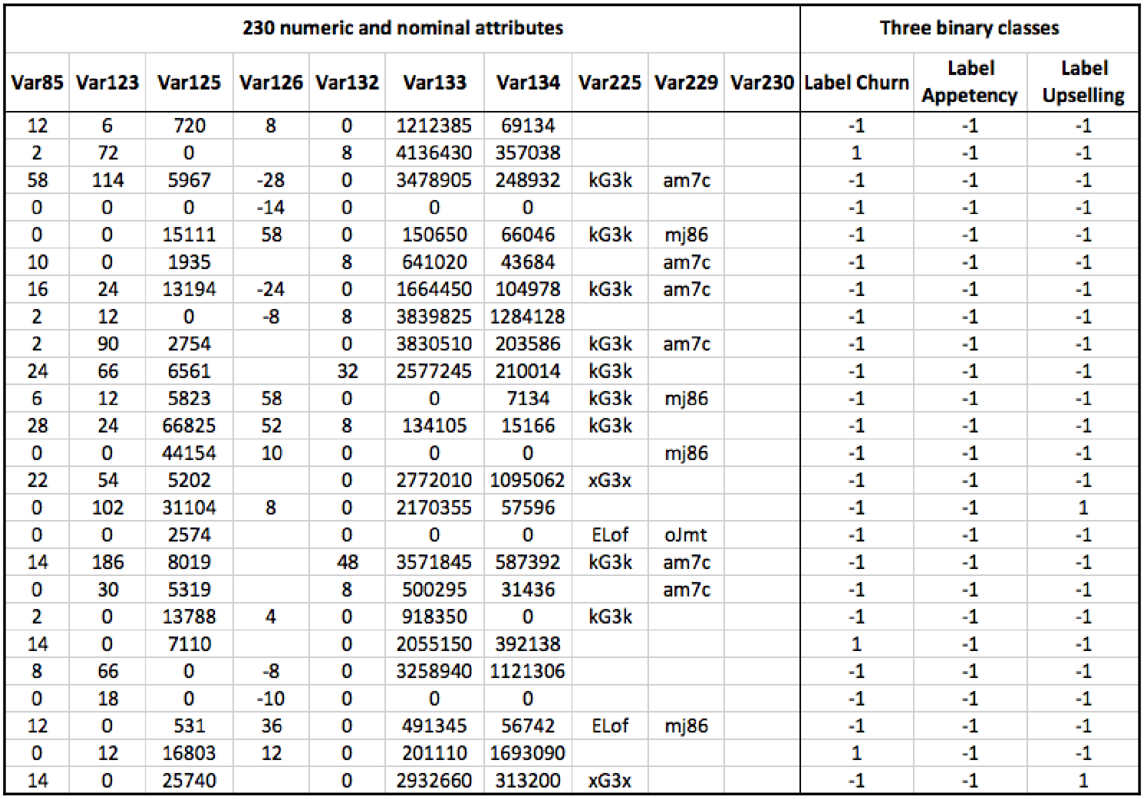
Editing-distance measures

0.8190	F1-Measure
0.8220	Precision
0.8160	Recall

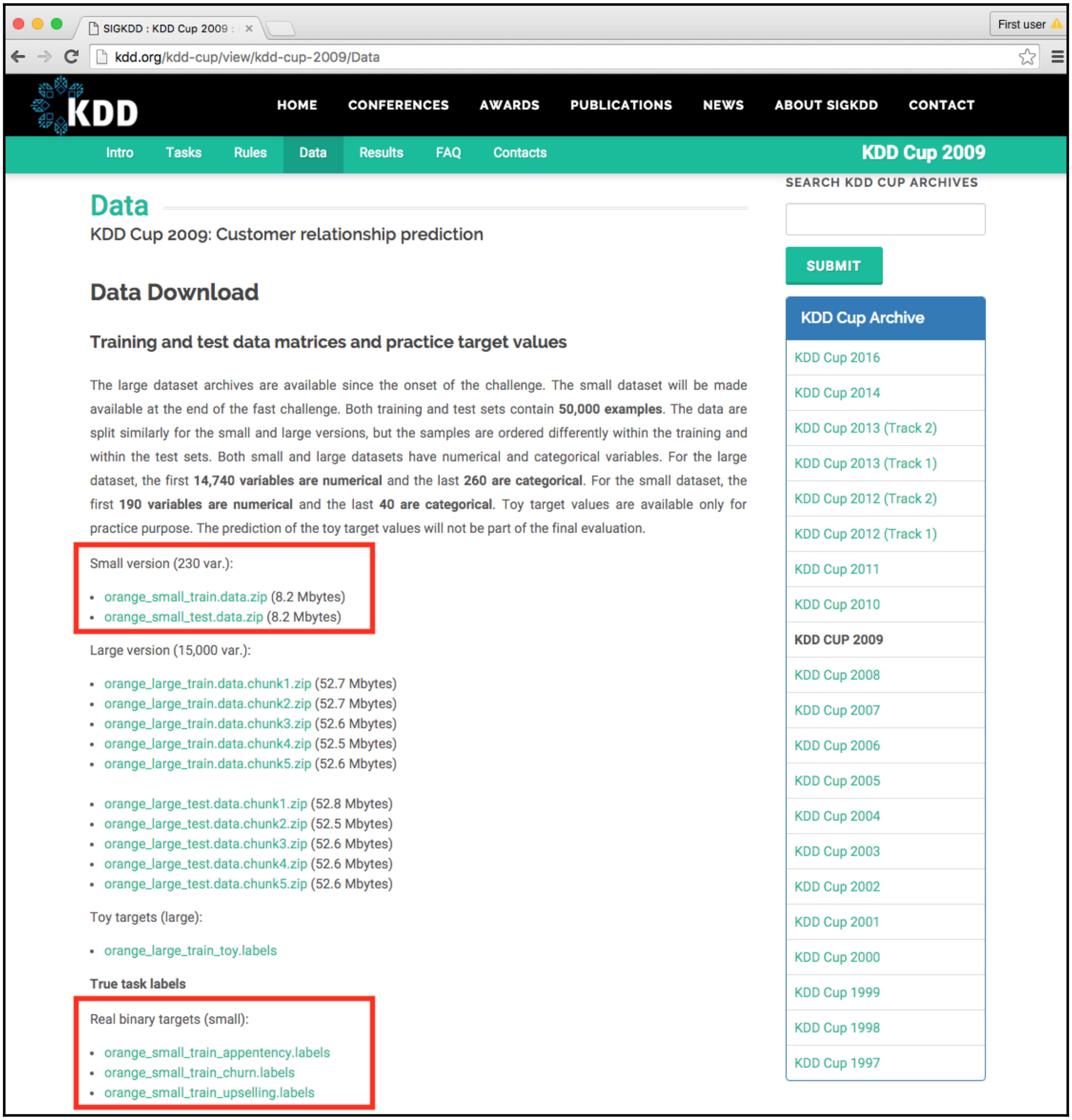
Gini measures

0.7427	Mean +-0.2115

Chapter 4: Customer Relationship Prediction with Ensembles



230 numeric and nominal attributes										Three binary classes		
Var85	Var123	Var125	Var126	Var132	Var133	Var134	Var225	Var229	Var230	Label Churn	Label Appetency	Label Upselling
12	6	720	8	0	1212385	69134				-1	-1	-1
2	72	0		8	4136430	357038				1	-1	-1
58	114	5967	-28	0	3478905	248932	kG3k	am7c		-1	-1	-1
0	0	0	-14	0	0	0				-1	-1	-1
0	0	15111	58	0	150650	66046	kG3k	mj86		-1	-1	-1
10	0	1935		8	641020	43684		am7c		-1	-1	-1
16	24	13194	-24	0	1664450	104978	kG3k	am7c		-1	-1	-1
2	12	0	-8	8	3839825	1284128				-1	-1	-1
2	90	2754		0	3830510	203586	kG3k	am7c		-1	-1	-1
24	66	6561		32	2577245	210014	kG3k			-1	-1	-1
6	12	5823	58	0	0	7134	kG3k	mj86		-1	-1	-1
28	24	66825	52	8	134105	15166	kG3k			-1	-1	-1
0	0	44154	10	0	0	0		mj86		-1	-1	-1
22	54	5202		0	2772010	1095062	xG3x			-1	-1	-1
0	102	31104	8	0	2170355	57596				-1	-1	1
0	0	2574		0	0	0	ELof	oJmt		-1	-1	-1
14	186	8019		48	3571845	587392	kG3k	am7c		-1	-1	-1
0	30	5319		8	500295	31436		am7c		-1	-1	-1
2	0	13788	4	0	918350	0	kG3k			-1	-1	-1
14	0	7110		0	2055150	392138				1	-1	-1
8	66	0	-8	0	3258940	1121306				-1	-1	-1
0	18	0	-10	0	0	0				-1	-1	-1
12	0	531	36	0	491345	56742	ELof	mj86		-1	-1	-1
0	12	16803	12	0	201110	1693090				1	-1	-1
14	0	25740		0	2932660	313200	xG3x			-1	-1	1



Data

KDD Cup 2009: Customer relationship prediction

Data Download

Training and test data matrices and practice target values

The large dataset archives are available since the onset of the challenge. The small dataset will be made available at the end of the fast challenge. Both training and test sets contain **50,000 examples**. The data are split similarly for the small and large versions, but the samples are ordered differently within the training and within the test sets. Both small and large datasets have numerical and categorical variables. For the large dataset, the first **14,740 variables are numerical** and the last **260 are categorical**. For the small dataset, the first **190 variables are numerical** and the last **40 are categorical**. Toy target values are available only for practice purpose. The prediction of the toy target values will not be part of the final evaluation.

Small version (230 var.):

- orange_small_train.data.zip (8.2 Mbytes)
- orange_small_test.data.zip (8.2 Mbytes)

Large version (15,000 var.):

- orange_large_train.data.chunk1.zip (52.7 Mbytes)
- orange_large_train.data.chunk2.zip (52.7 Mbytes)
- orange_large_train.data.chunk3.zip (52.6 Mbytes)
- orange_large_train.data.chunk4.zip (52.5 Mbytes)
- orange_large_train.data.chunk5.zip (52.6 Mbytes)

- orange_large_test.data.chunk1.zip (52.8 Mbytes)
- orange_large_test.data.chunk2.zip (52.5 Mbytes)
- orange_large_test.data.chunk3.zip (52.6 Mbytes)
- orange_large_test.data.chunk4.zip (52.6 Mbytes)
- orange_large_test.data.chunk5.zip (52.6 Mbytes)

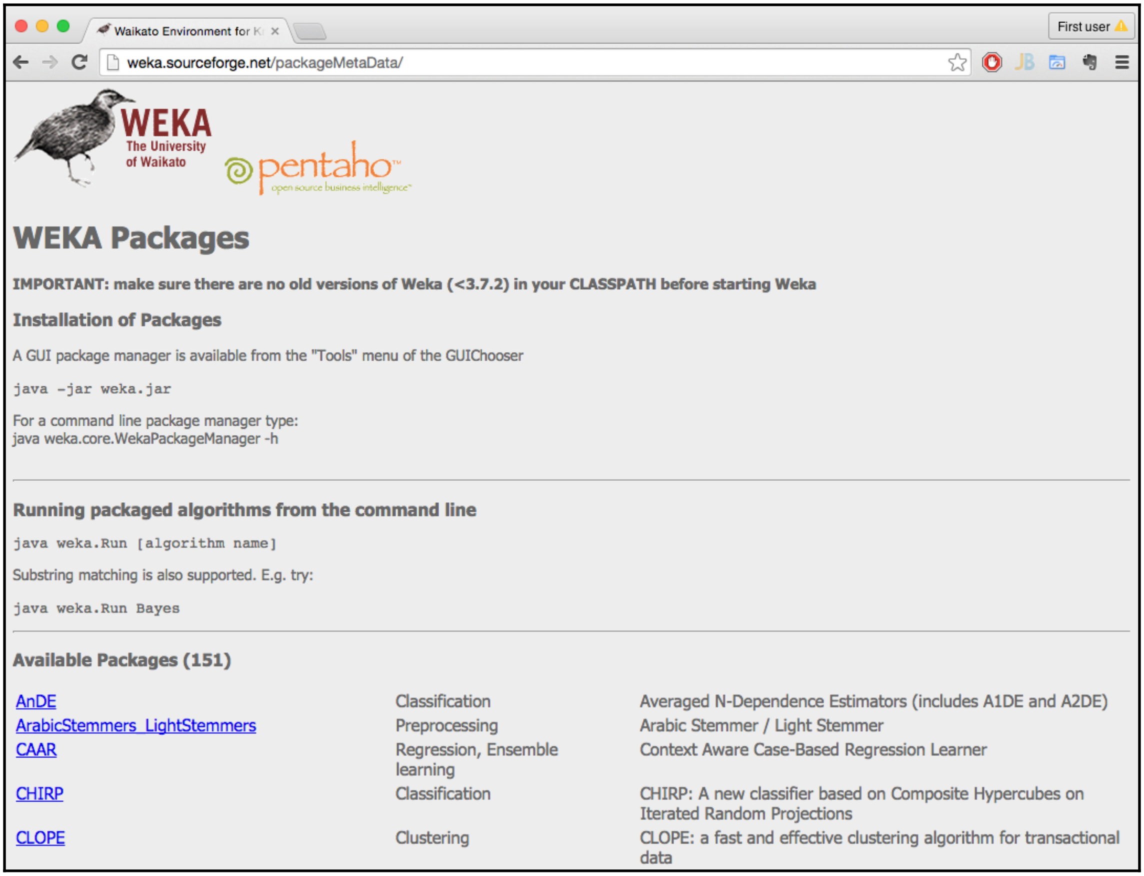
Toy targets (large):

- orange_large_train_toy.labels

True task labels

Real binary targets (small):

- orange_small_train_appentency.labels
- orange_small_train_churn.labels
- orange_small_train_upselling.labels



Waikato Environment for K
First user
weka.sourceforge.net/packageMetaData/
WEKA
The University of Waikato
pentaho
open source business intelligence

WEKA Packages

IMPORTANT: make sure there are no old versions of Weka (<3.7.2) in your CLASSPATH before starting Weka

Installation of Packages

A GUI package manager is available from the "Tools" menu of the GUIChooser

```
java -jar weka.jar
```

For a command line package manager type:
java weka.core.WekaPackageManager -h

Running packaged algorithms from the command line

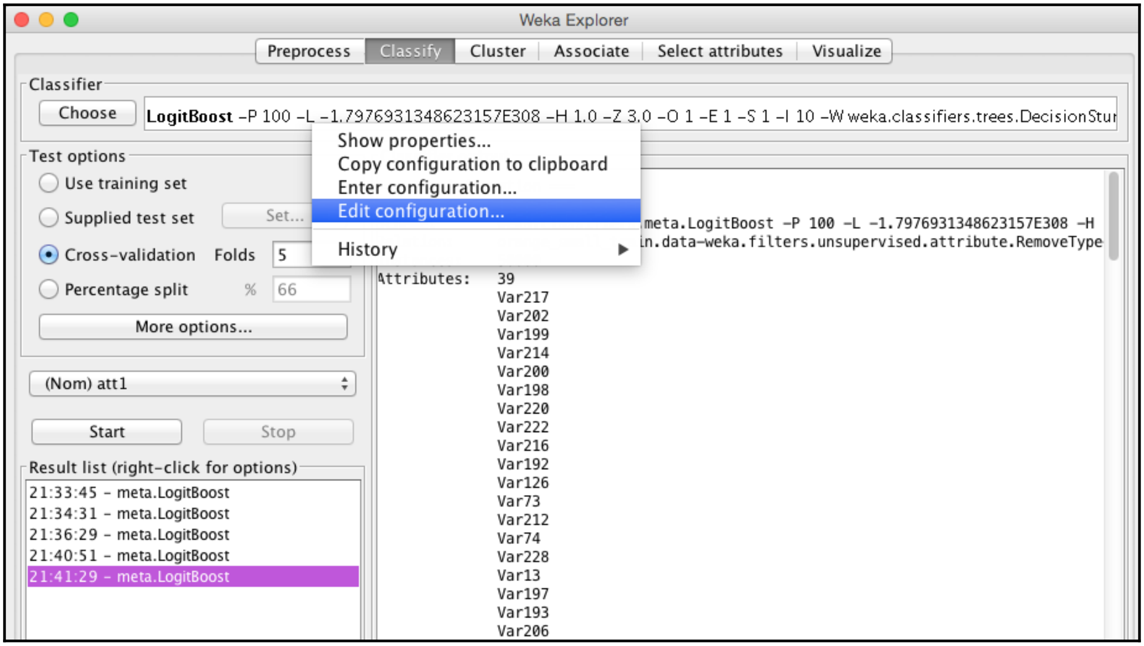
```
java weka.Run [algorithm name]
```

Substring matching is also supported. E.g. try:

```
java weka.Run Bayes
```

Available Packages (151)

AnDE	Classification	Averaged N-Dependence Estimators (includes A1DE and A2DE)
ArabicStemmers_LightStemmers	Preprocessing	Arabic Stemmer / Light Stemmer
CAAR	Regression, Ensemble learning	Context Aware Case-Based Regression Learner
CHIRP	Classification	CHIRP: A new classifier based on Composite Hypercubes on Iterated Random Projections
CLOPE	Clustering	CLOPE: a fast and effective clustering algorithm for transactional data



LogitBoost -P 100 -L -1.7976931348623157E308 -H 1.0 -Z 3.0 -O 1 -E 1 -S 1 -I 10 -W weka.classifiers.trees.DecisionStur

○ Use training set

○ Supplied test set Set...

● Cross-validation Folds 5

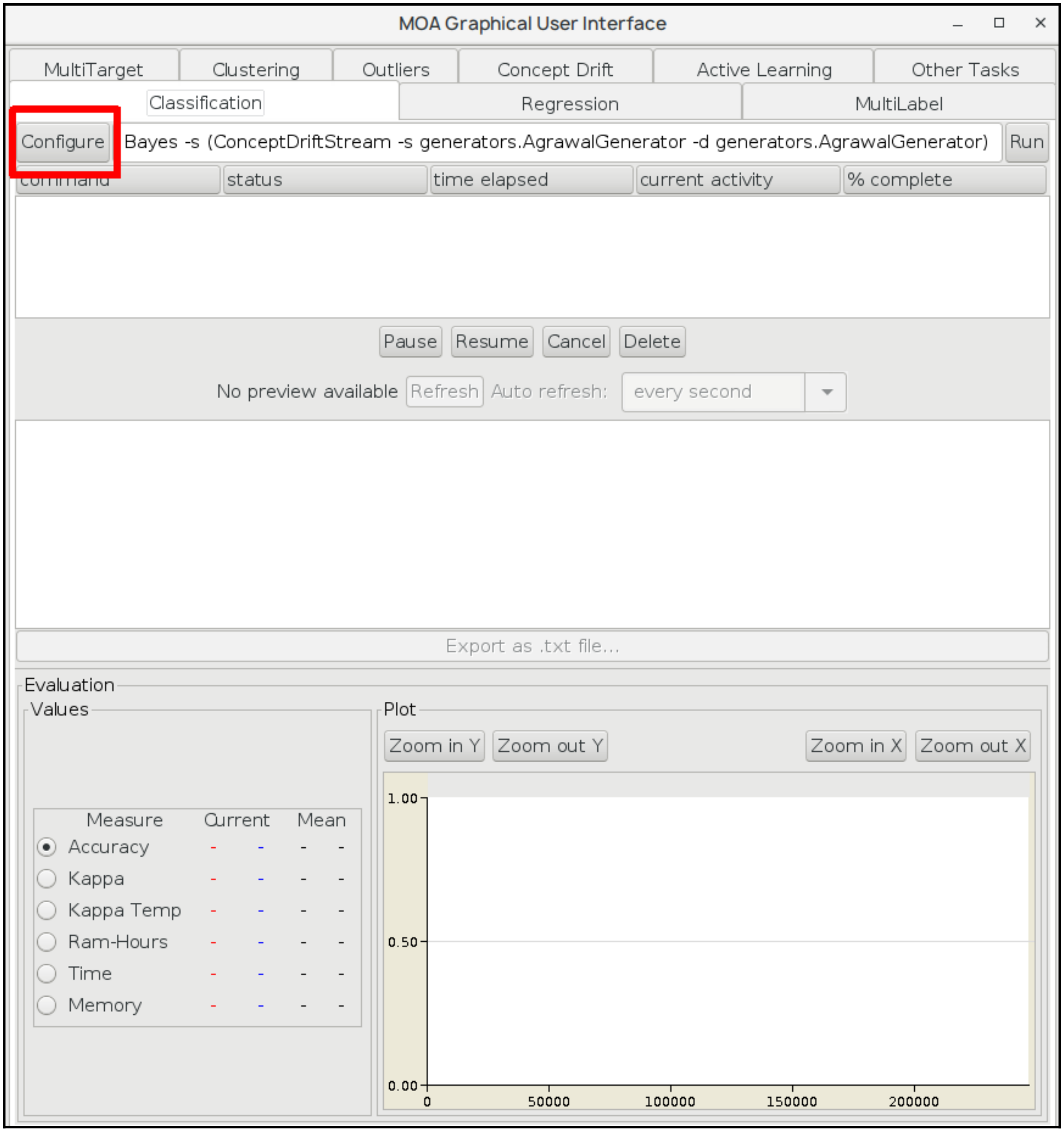
○ Percentage split % 66

(Nom) att1

Result list (right-click for options)

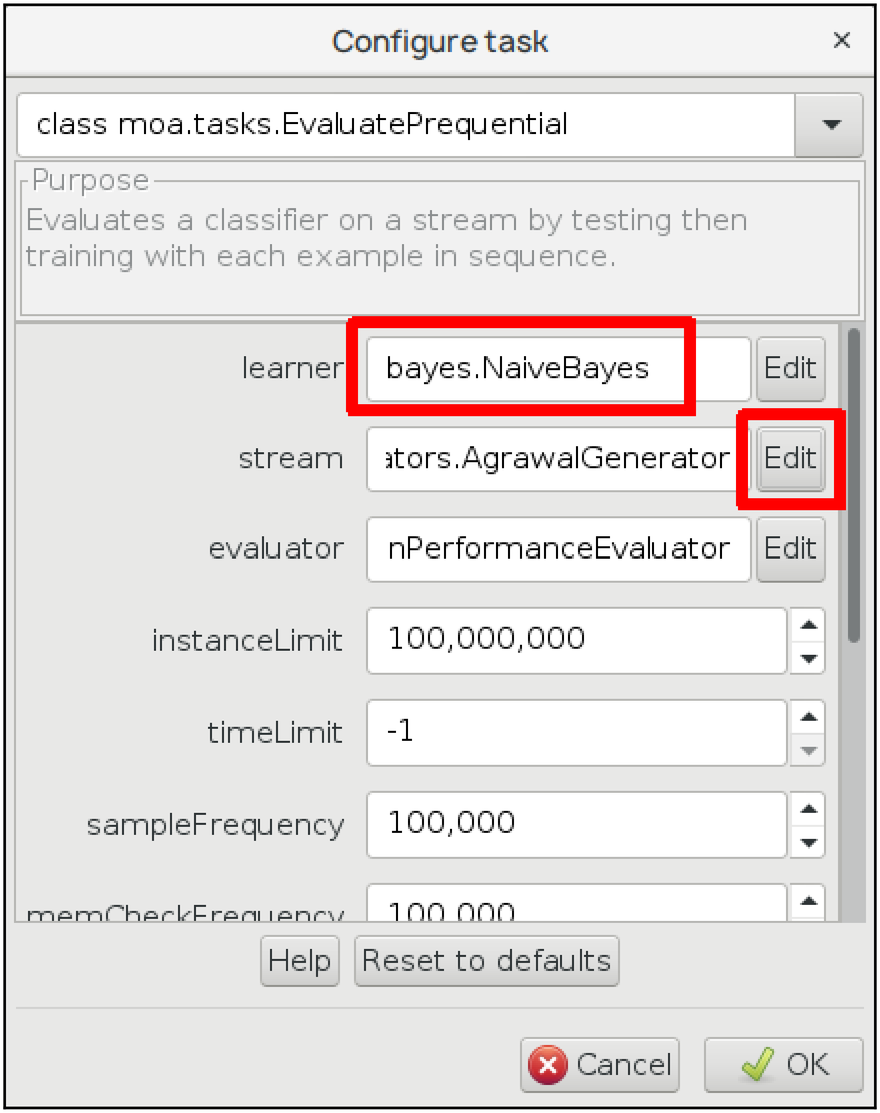
21:33:45 - meta.LogitBoost
21:34:31 - meta.LogitBoost
21:36:29 - meta.LogitBoost
21:40:51 - meta.LogitBoost
21:41:29 - meta.LogitBoost

```
meta.LogitBoost -P 100 -L -1.7976931348623157E308 -H
in.data-weka.filters.unsupervised.attribute.RemoveType
Attributes:   39
              Var217
              Var202
              Var199
              Var214
              Var200
              Var198
              Var220
              Var222
              Var216
              Var192
              Var126
              Var73
              Var212
              Var74
              Var228
              Var13
              Var197
              Var193
              Var206
```



Bayes -s (ConceptDriftStream -s generators.AgrawalGenerator -d generators.AgrawalGenerator)

	Measure	Current		Mean	
●	Accuracy	-	-	-	-
○	Kappa	-	-	-	-
○	Kappa Temp	-	-	-	-
○	Ram-Hours	-	-	-	-
○	Time	-	-	-	-
○	Memory	-	-	-	-



Configure task

class moa.tasks.EvaluatePrequential

Purpose

Evaluates a classifier on a stream by testing then training with each example in sequence.

learner bayes.NaiveBayes

stream ɪtors.AgrawalGenerator

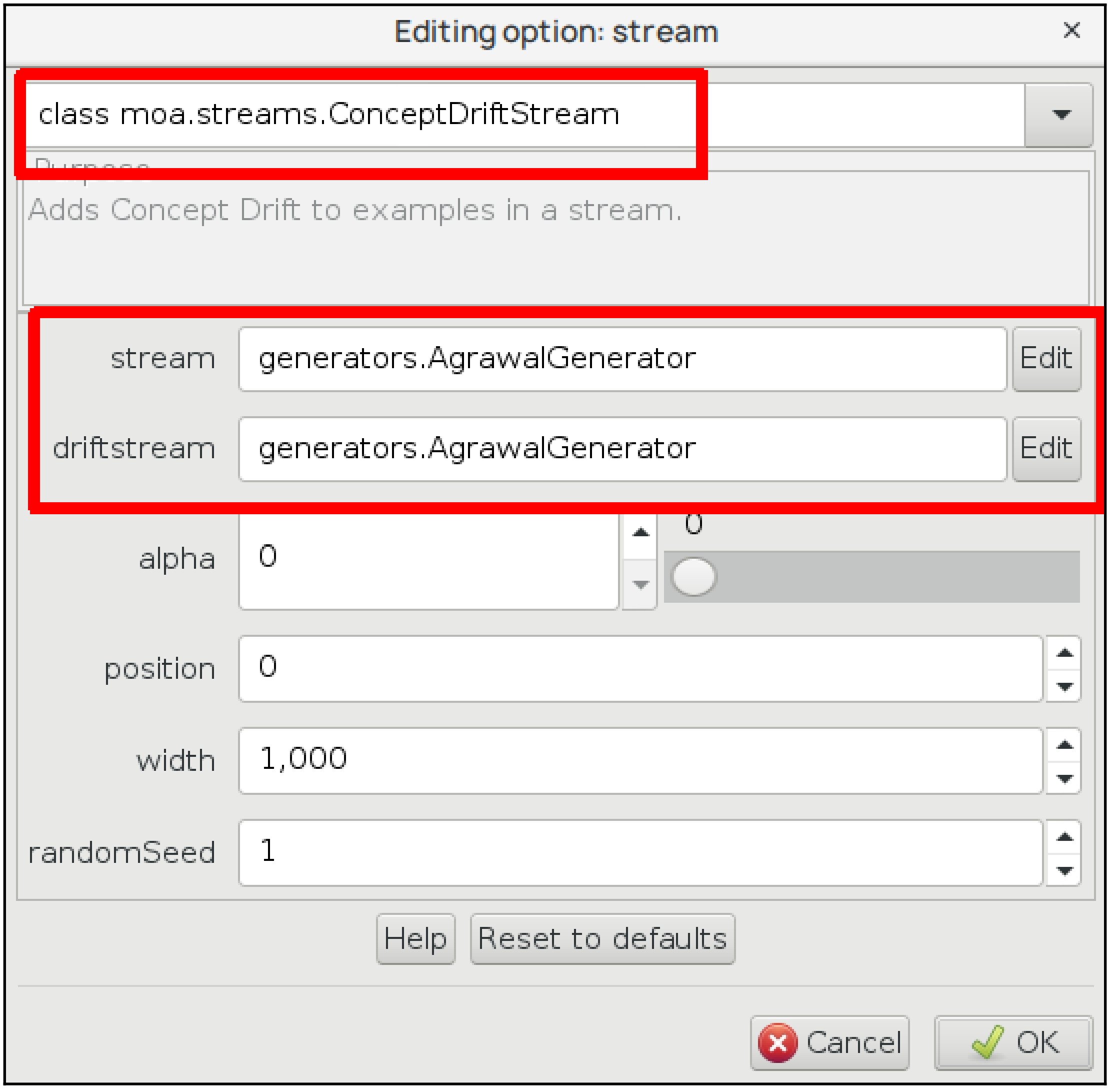
evaluator nPerformanceEvaluator

instanceLimit 100,000,000

timeLimit -1

sampleFrequency 100,000

memCheckFrequency 100,000



Editing option: stream

class moa.streams.ConceptDriftStream

Purpose

Adds Concept Drift to examples in a stream.

stream generators.AgrawalGenerator

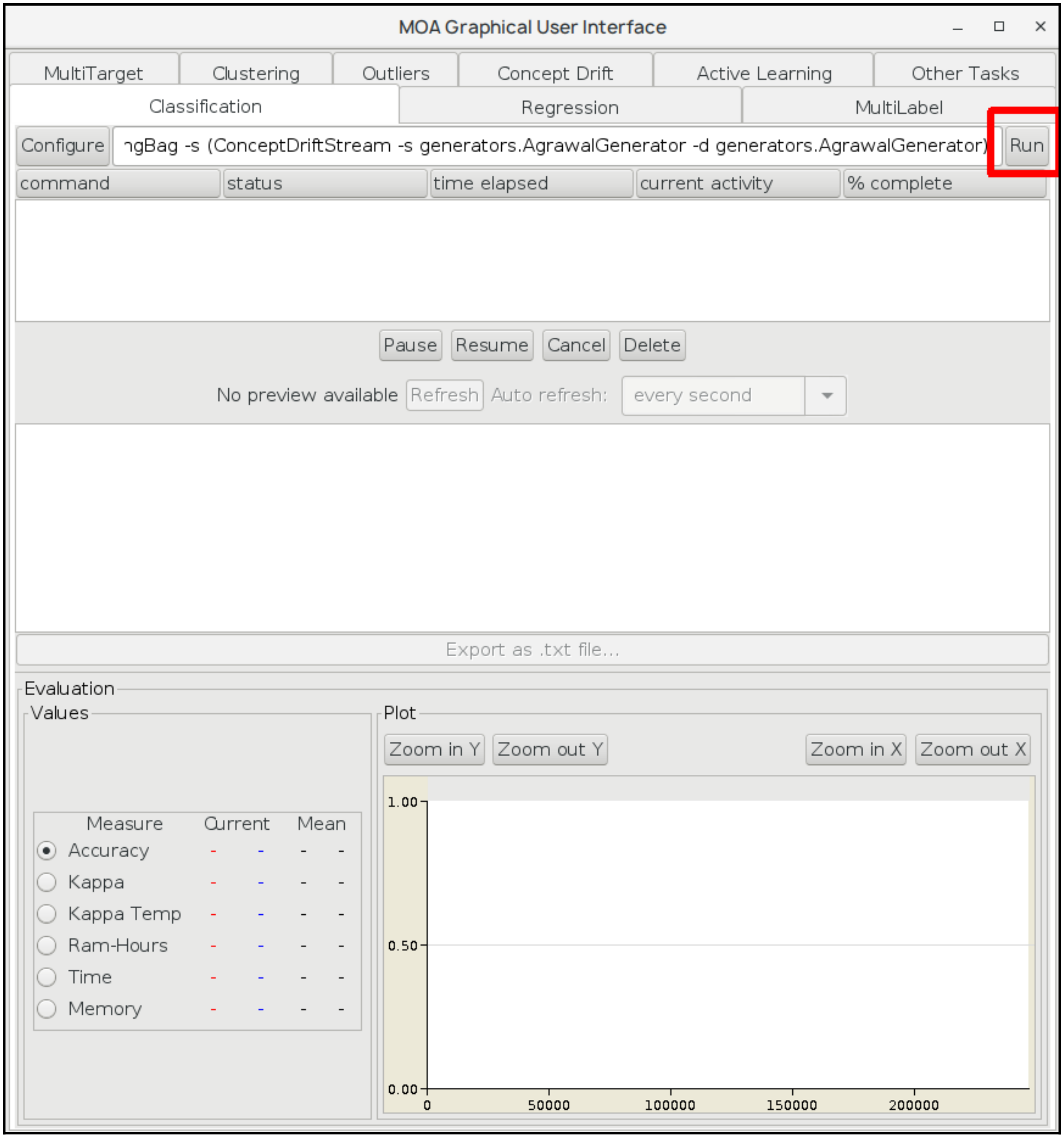
driftstream generators.AgrawalGenerator

alpha 0

position 0

width 1,000

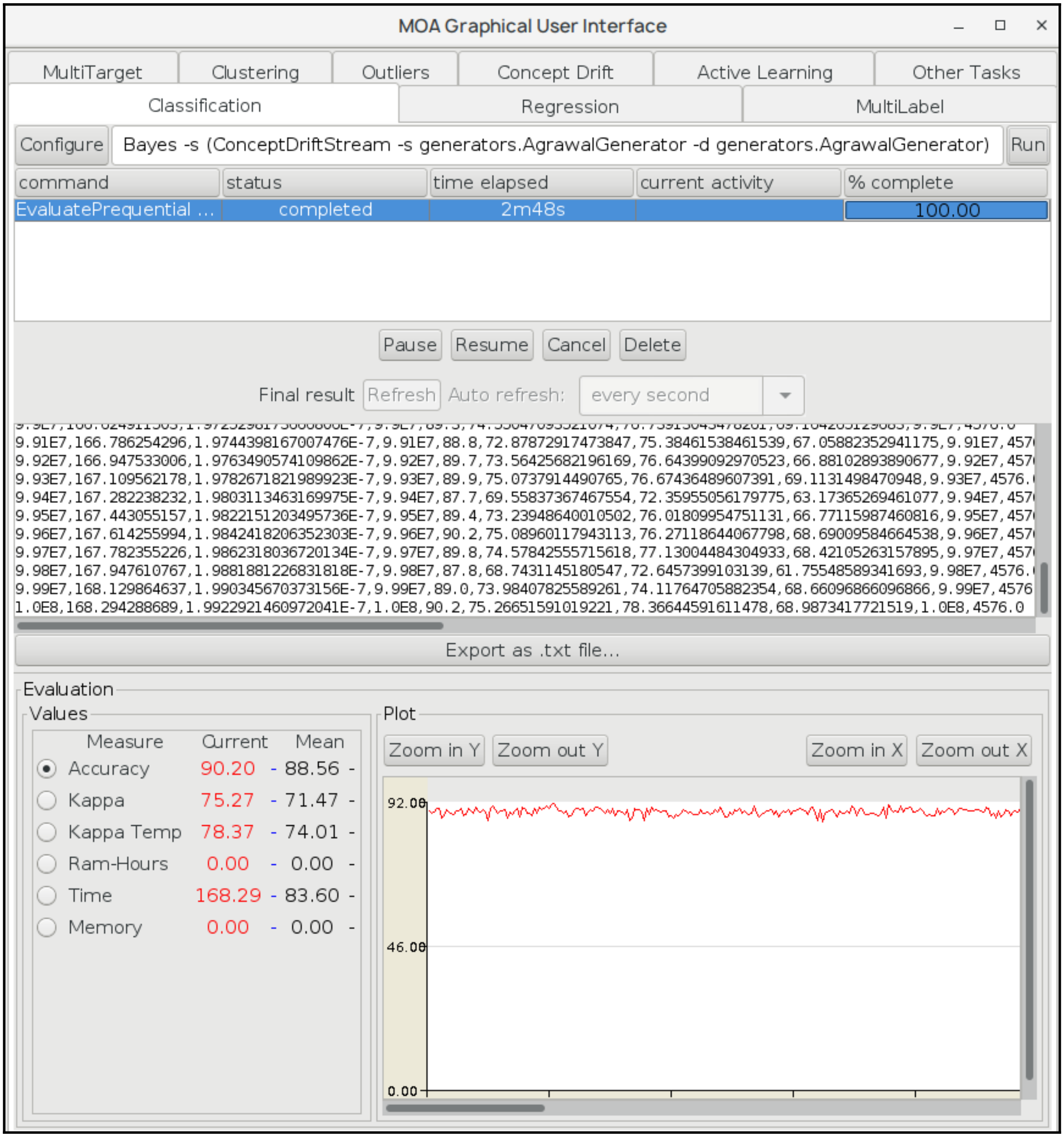
randomSeed 1



```
ngBag -s (ConceptDriftStream -s generators.AgrawalGenerator -d generators.AgrawalGenerator)
```

command	status	time elapsed	current activity	% complete

	Measure	Current		Mean	
●	Accuracy	-	-	-	-
○	Kappa	-	-	-	-
○	Kappa Temp	-	-	-	-
○	Ram-Hours	-	-	-	-
○	Time	-	-	-	-
○	Memory	-	-	-	-



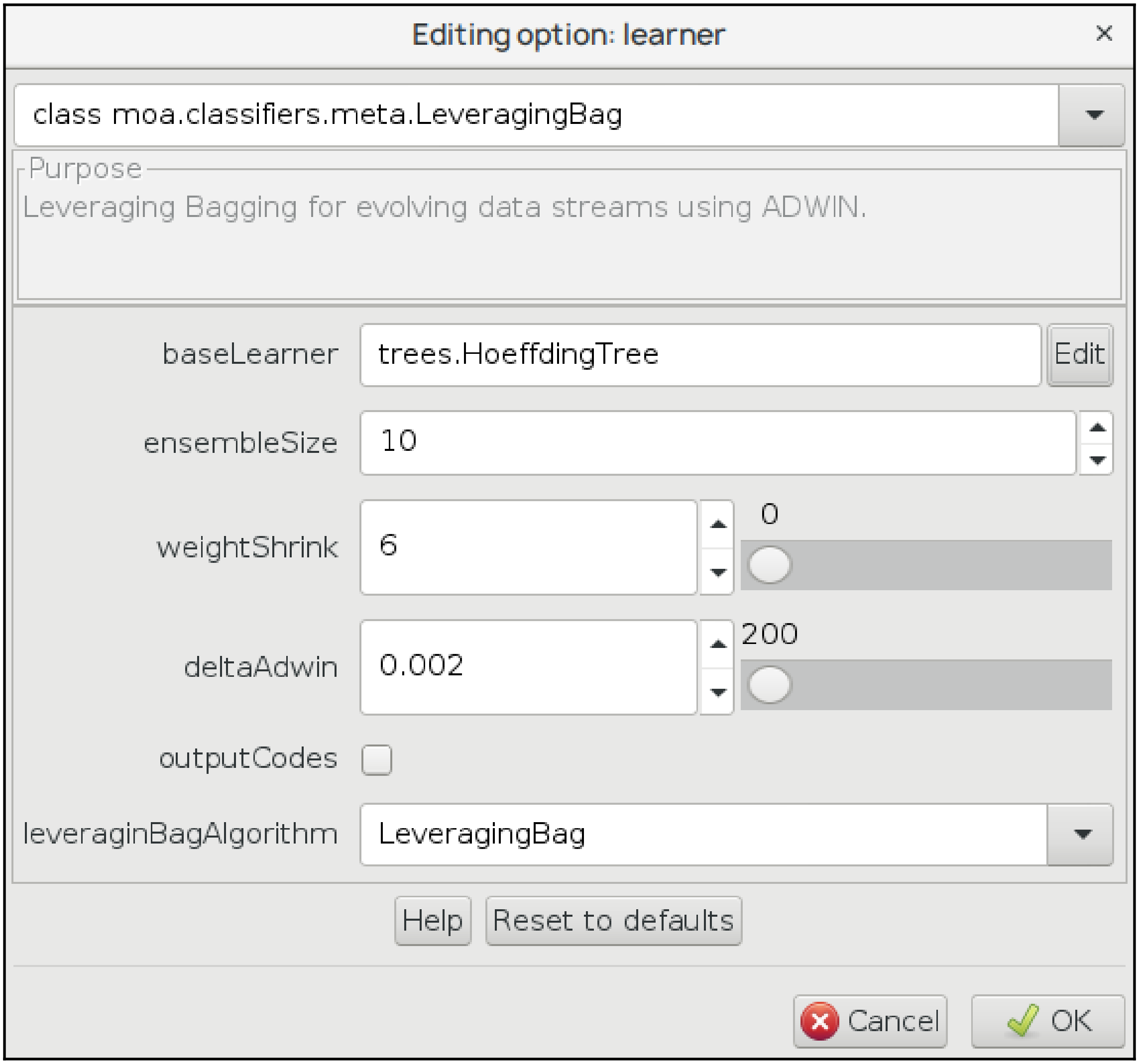
Bayes -s (ConceptDriftStream -s generators.AgrawalGenerator -d generators.AgrawalGenerator)

command	status	time elapsed	current activity	% complete
EvaluatePrequential ...	completed	2m48s		100.00

Final result Auto refresh: every second

```
9.9E7,166.624911503,1.9725298173060806E-7,9.9E7,89.5,74.55047093521074,76.73913043478261,69.164205129683,9.9E7,4576.0
9.91E7,166.786254296,1.9744398167007476E-7,9.91E7,88.8,72.87872917473847,75.38461538461539,67.05882352941175,9.91E7,457
9.92E7,166.947533006,1.9763490574109862E-7,9.92E7,89.7,73.56425682196169,76.64399092970523,66.88102893890677,9.92E7,457
9.93E7,167.109562178,1.9782671821989923E-7,9.93E7,89.9,75.0737914490765,76.67436489607391,69.1131498470948,9.93E7,4576.
9.94E7,167.282238232,1.9803113463169975E-7,9.94E7,87.7,69.55837367467554,72.35955056179775,63.17365269461077,9.94E7,457
9.95E7,167.443055157,1.9822151203495736E-7,9.95E7,89.4,73.23948640010502,76.01809954751131,66.77115987460816,9.95E7,457
9.96E7,167.614255994,1.9842418206352303E-7,9.96E7,90.2,75.0896011794313,76.27118644067798,68.69009584664538,9.96E7,457
9.97E7,167.782355226,1.986231803672O134E-7,9.97E7,89.8,74.57842555715618,77.13004484304933,68.42105263157895,9.97E7,457
9.98E7,167.947610767,1.9881881226831818E-7,9.98E7,87.8,68.7431145180547,72.6457399103139,61.75548589341693,9.98E7,4576.
9.99E7,168.129864637,1.9903456703731560E-7,9.99E7,89.0,73.98407825589261,74.11764705882354,68.66096866096866,9.99E7,4576
1.0E8,168.294288689,1.9922921460972041E-7,1.0E8,90.2,75.26651591019221,78.36644591611478,68.9873417721519,1.0E8,4576.0
```

	Measure	Current		Mean	
●	Accuracy	90.20	-	88.56	-
○	Kappa	75.27	-	71.47	-
○	Kappa Temp	78.37	-	74.01	-
○	Ram-Hours	0.00	-	0.00	-
○	Time	168.29	-	83.60	-
○	Memory	0.00	-	0.00	-



class moa.classifiers.meta.LeveragingBag

Purpose

Leveraging Bagging for evolving data streams using ADWIN.

baseLearner: trees.HoeffdingTree

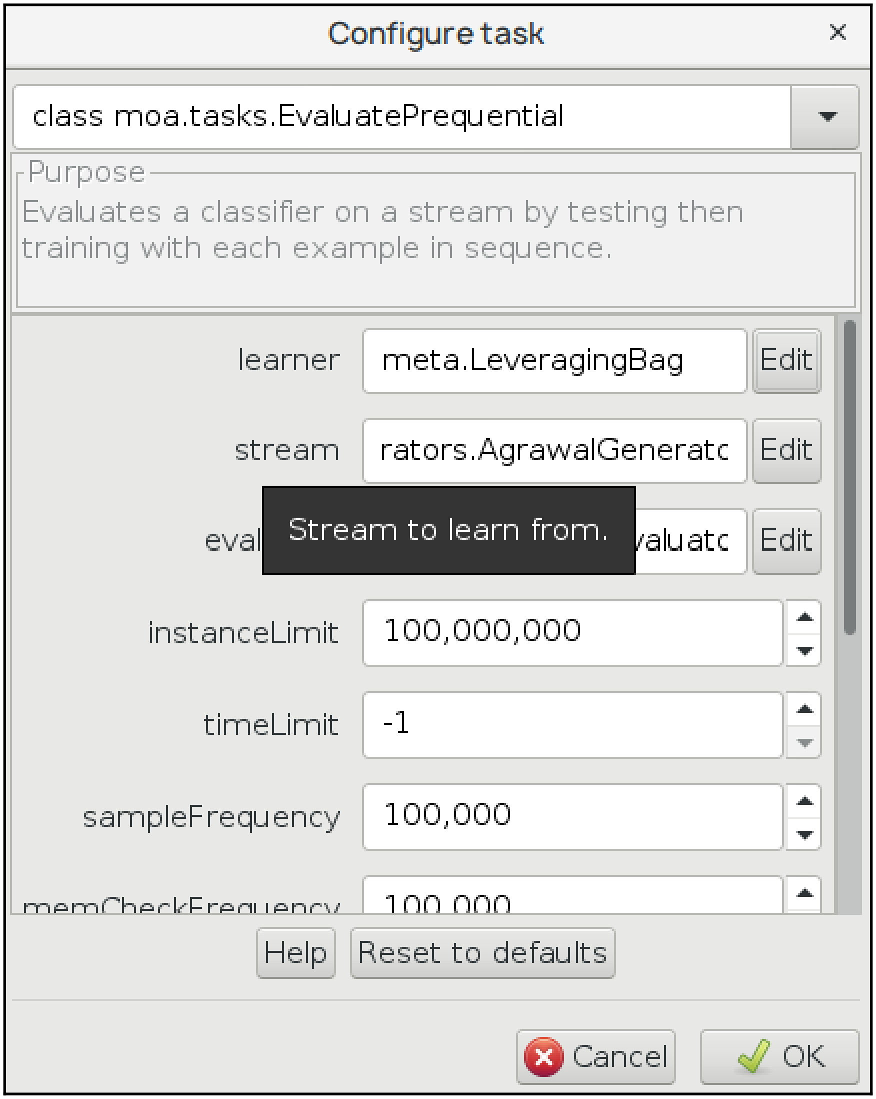
ensembleSize: 10

weightShrink: 6

deltaAdwin: 0.002

outputCodes: □

leveraginBagAlgorithm: LeveragingBag



Configure task

class moa.tasks.EvaluatePrequential

Purpose

Evaluates a classifier on a stream by testing then training with each example in sequence.

learner meta.LeveragingBag

stream rators.AgrawalGeneratc

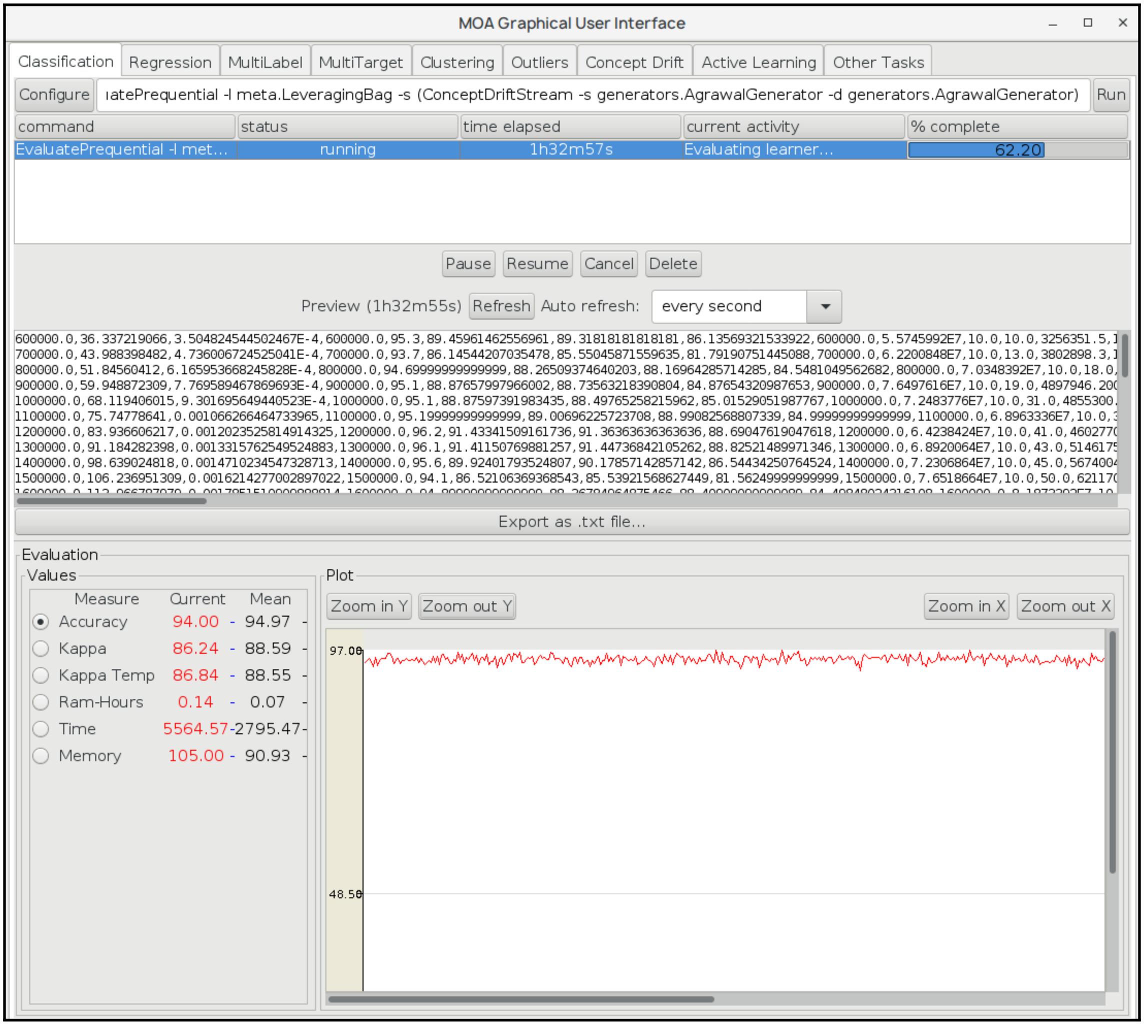
eval Stream to learn from. valuatc

instanceLimit 100,000,000

timeLimit -1

sampleFrequency 100,000

memCheckFrequency 100,000



ıatePrequential -l meta.LeveragingBag -s (ConceptDriftStream -s generators.AgrawalGenerator -d generators.AgrawalGenerator)

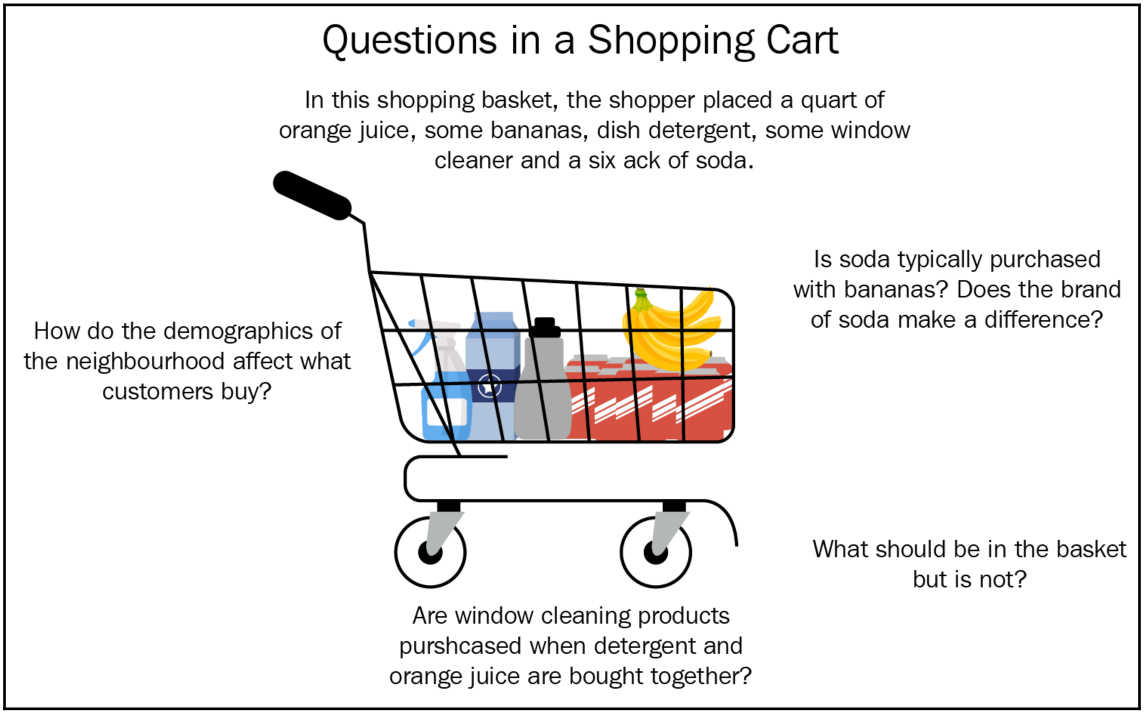
command	status	time elapsed	current activity	% complete
EvaluatePrequential -l met...	running	1h32m57s	Evaluating learner...	62.20

Preview (1h32m55s) Auto refresh: every second

```
600000.0,36.337219066,3.504824544502467E-4,600000.0,95.3,89.45961462556961,89.31818181818181,86.13569321533922,600000.0,5.5745992E7,10.0,10.0,3256351.5,
700000.0,43.988398482,4.736006724525041E-4,700000.0,93.7,86.14544207035478,85.55045871559635,81.79190751445088,700000.0,6.2200848E7,10.0,13.0,3802898.3,
800000.0,51.84560412,6.165953668245828E-4,800000.0,94.69999999999999,88.26509374640203,88.16964285714285,84.5481049562682,800000.0,7.0348392E7,10.0,18.0,
900000.0,59.948872309,7.769589467869693E-4,900000.0,95.1,88.87657997966002,88.73563218390804,84.87654320987653,900000.0,7.6497616E7,10.0,19.0,4897946.200
1000000.0,68.119406015,9.301695649440523E-4,1000000.0,95.1,88.87597391983435,88.49765258215962,85.01529051987767,1000000.0,7.2483776E7,10.0,31.0,4855300.
1100000.0,75.74778641,0.001066266464733965,1100000.0,95.19999999999999,89.00696225723708,88.99082568807339,84.99999999999999,1100000.0,6.8963336E7,10.0,3
1200000.0,83.936606217,0.0012023525814914325,1200000.0,96.2,91.43341509161736,91.36363636363636,88.69047619047618,1200000.0,6.4238424E7,10.0,41.0,460277
1300000.0,91.184282398,0.0013315762549524883,1300000.0,96.1,91.41150769881257,91.44736842105262,88.82521489971346,1300000.0,6.8920064E7,10.0,43.0,514617
1400000.0,98.639024818,0.0014710234547328713,1400000.0,95.6,89.92401793524807,90.17857142857142,86.54434250764524,1400000.0,7.2306864E7,10.0,45.0,567400
1500000.0,106.236951309,0.0016214277002897022,1500000.0,94.1,86.52106369368543,85.53921568627449,81.56249999999999,1500000.0,7.6518664E7,10.0,50.0,62117
```

	Measure	Current		Mean	
●	Accuracy	94.00	-	94.97	-
○	Kappa	86.24	-	88.59	-
○	Kappa Temp	86.84	-	88.55	-
○	Ram-Hours	0.14	-	0.07	-
○	Time	5564.57	-	2795.47	-
○	Memory	105.00	-	90.93	-

Chapter 5: Affinity Analysis



Questions in a Shopping Cart

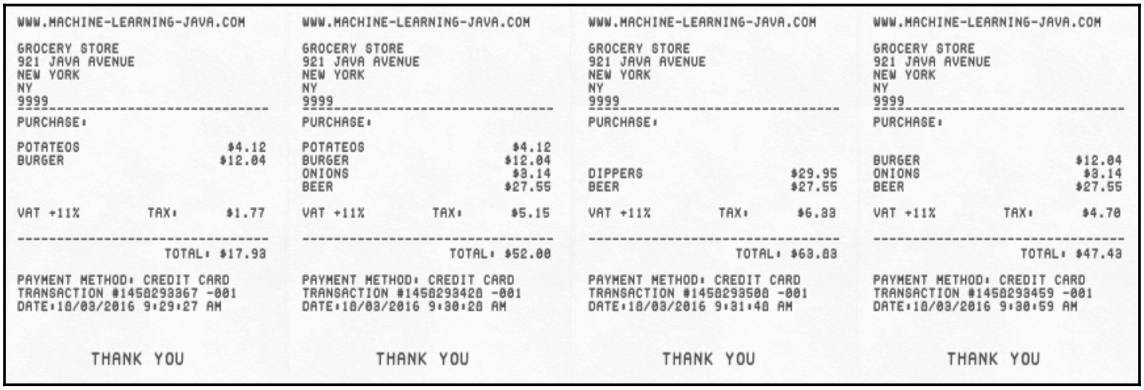
In this shopping basket, the shopper placed a quart of orange juice, some bananas, dish detergent, some window cleaner and a six ack of soda.

How do the demographics of the neighbourhood affect what customers buy?

Is soda typically purchased with bananas? Does the brand of soda make a difference?

What should be in the basket but is not?

Are window cleaning products purshcased when detergent and orange juice are bought together?

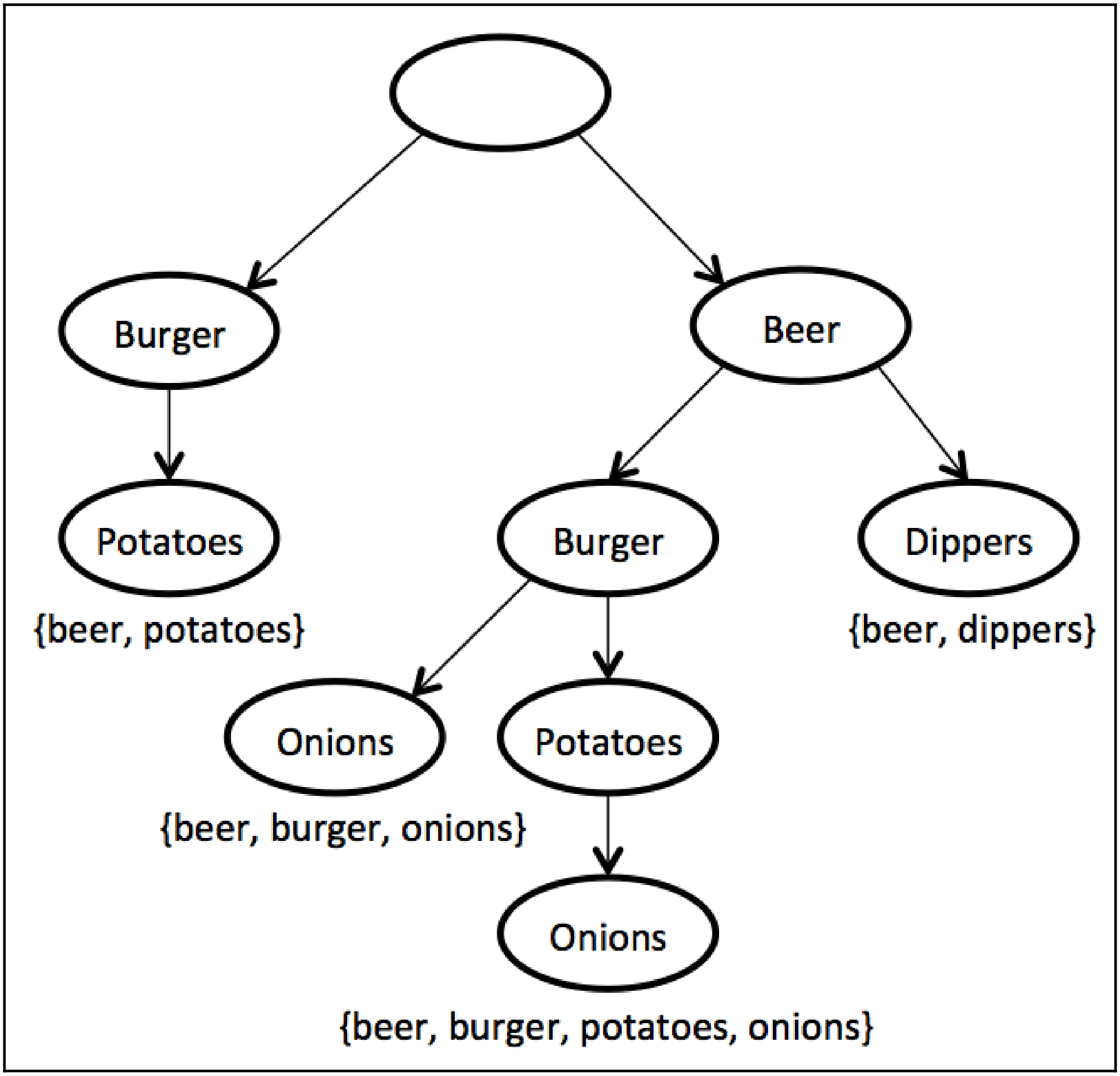


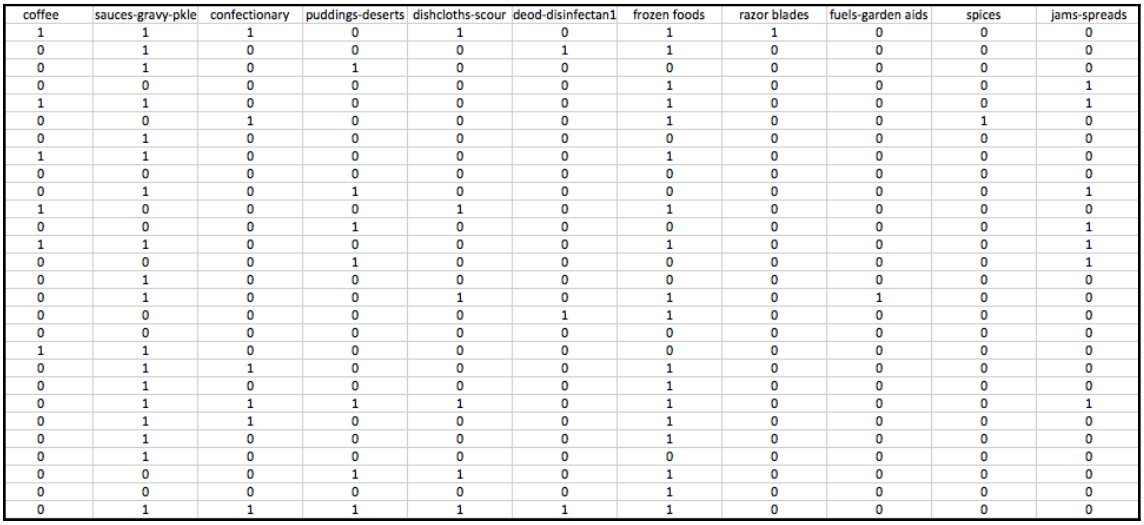
```
WWW.MACHINE-LEARNING-JAVA.COM

GROCERY STORE
921 JAVA AVENUE
NEW YORK
NY
9999_____________________________
PURCHASE:

POTATEOS                    $4.12
BURGER                     $12.04


VAT +11%         TAX:       $1.77

---------------------------------
                  TOTAL: $17.93

PAYMENT METHOD: CREDIT CARD
TRANSACTION #1458293367 -001
DATE:18/03/2016 9:29:27 AM


         THANK YOU
```

```
WWW.MACHINE-LEARNING-JAVA.COM

GROCERY STORE
921 JAVA AVENUE
NEW YORK
NY
9999_____________________________
PURCHASE:

POTATEOS                    $4.12
BURGER                     $12.04
ONIONS                      $3.14
BEER                       $27.55

VAT +11%         TAX:       $5.15

---------------------------------
                  TOTAL: $52.00

PAYMENT METHOD: CREDIT CARD
TRANSACTION #1458293428 -001
DATE:18/03/2016 9:30:28 AM


         THANK YOU
```

```
WWW.MACHINE-LEARNING-JAVA.COM

GROCERY STORE
921 JAVA AVENUE
NEW YORK
NY
9999_____________________________
PURCHASE:


DIPPERS                    $29.95
BEER                       $27.55

VAT +11%         TAX:       $6.33

---------------------------------
                  TOTAL: $63.83

PAYMENT METHOD: CREDIT CARD
TRANSACTION #1458293508 -001
DATE:18/03/2016 9:31:48 AM


         THANK YOU
```

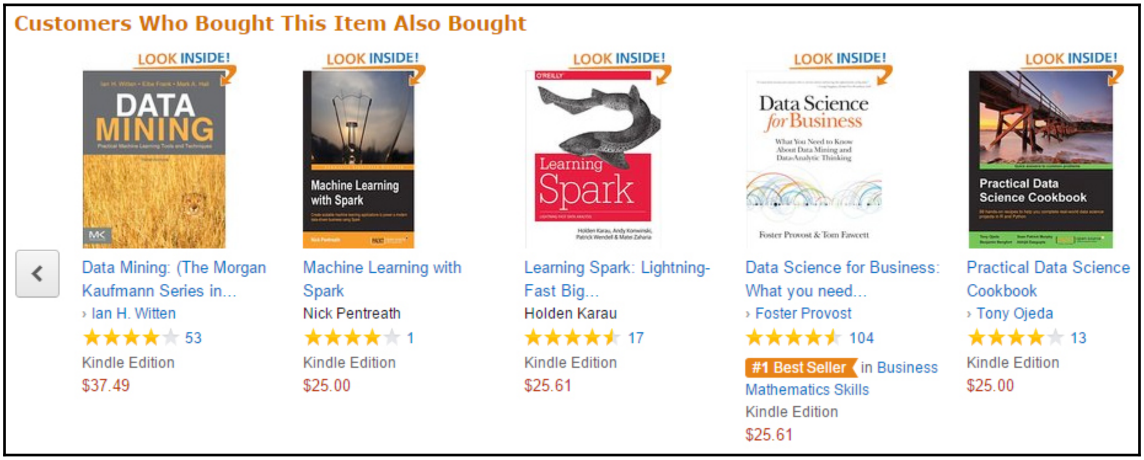
```
WWW.MACHINE-LEARNING-JAVA.COM

GROCERY STORE
921 JAVA AVENUE
NEW YORK
NY
9999_____________________________
PURCHASE:

BURGER                     $12.04
ONIONS                      $3.14
BEER                       $27.55

VAT +11%         TAX:       $4.70

---------------------------------
                  TOTAL: $47.43

PAYMENT METHOD: CREDIT CARD
TRANSACTION #1458293459 -001
DATE:18/03/2016 9:30:59 AM


         THANK YOU
```


Burger
Beer
Potatoes
Burger
Dippers
{beer, potatoes}
{beer, dippers}
Onions
Potatoes
{beer, burger, onions}
Onions
{beer, burger, potatoes, onions}



coffee	sauces-gravy-pkle	confectionary	puddings-deserts	dishcloths-scour	deod-disinfectan1	frozen foods	razor blades	fuels-garden aids	spices	jams-spreads
1	1	1	0	1	0	1	1	0	0	0
0	1	0	0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	1
1	1	0	0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0	0	1	0
0	1	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	1
1	0	0	0	1	0	1	0	0	0	0
0	0	0	1	0	0	0	0	0	0	1
1	1	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	1	0	1	0	0
0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
0	1	0	0	0	0	1	0	0	0	0
0	1	1	1	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0	0	0	0
0	1	0	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0
0	1	1	1	1	1	1	0	0	0	0

Chapter 6: Recommendation Engines with Apache Mahout



Customers Who Bought This Item Also Bought

Data Mining: (The Morgan Kaufmann Series in...
› Ian H. Witten
★★★★☆ 53
Kindle Edition
$37.49

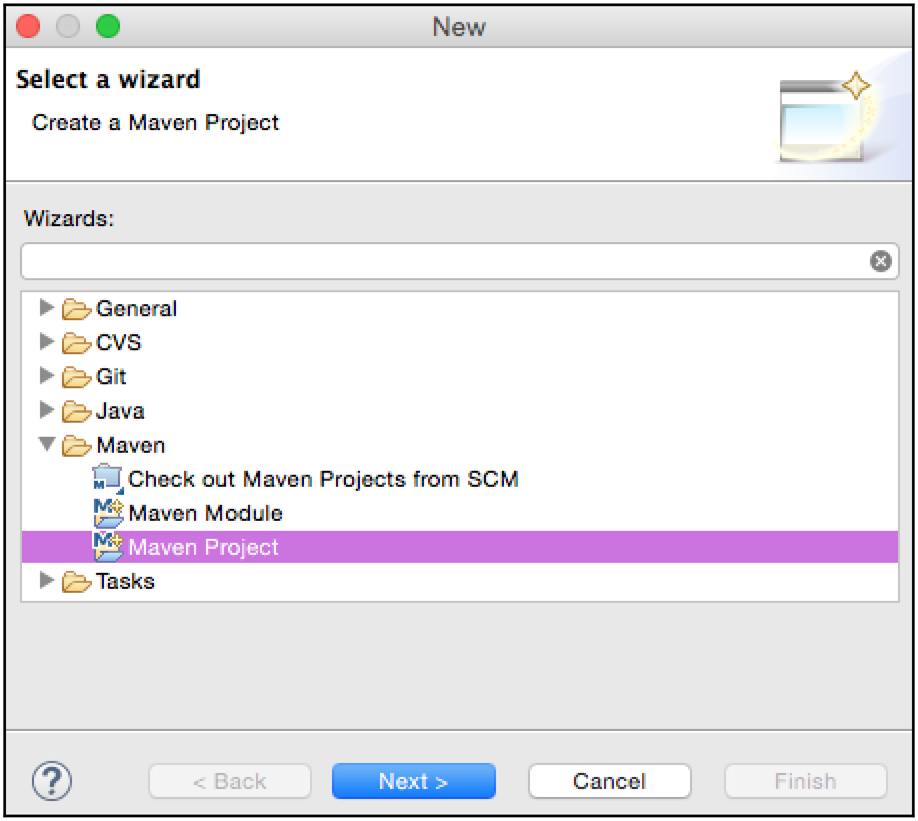
Machine Learning with Spark
Nick Pentreath
★★★★☆ 1
Kindle Edition
$25.00

Learning Spark: Lightning-Fast Big...
Holden Karau
★★★★½ 17
Kindle Edition
$25.61

Data Science for Business: What you need...
› Foster Provost
★★★★½ 104
#1 Best Seller in Business Mathematics Skills
Kindle Edition
$25.61

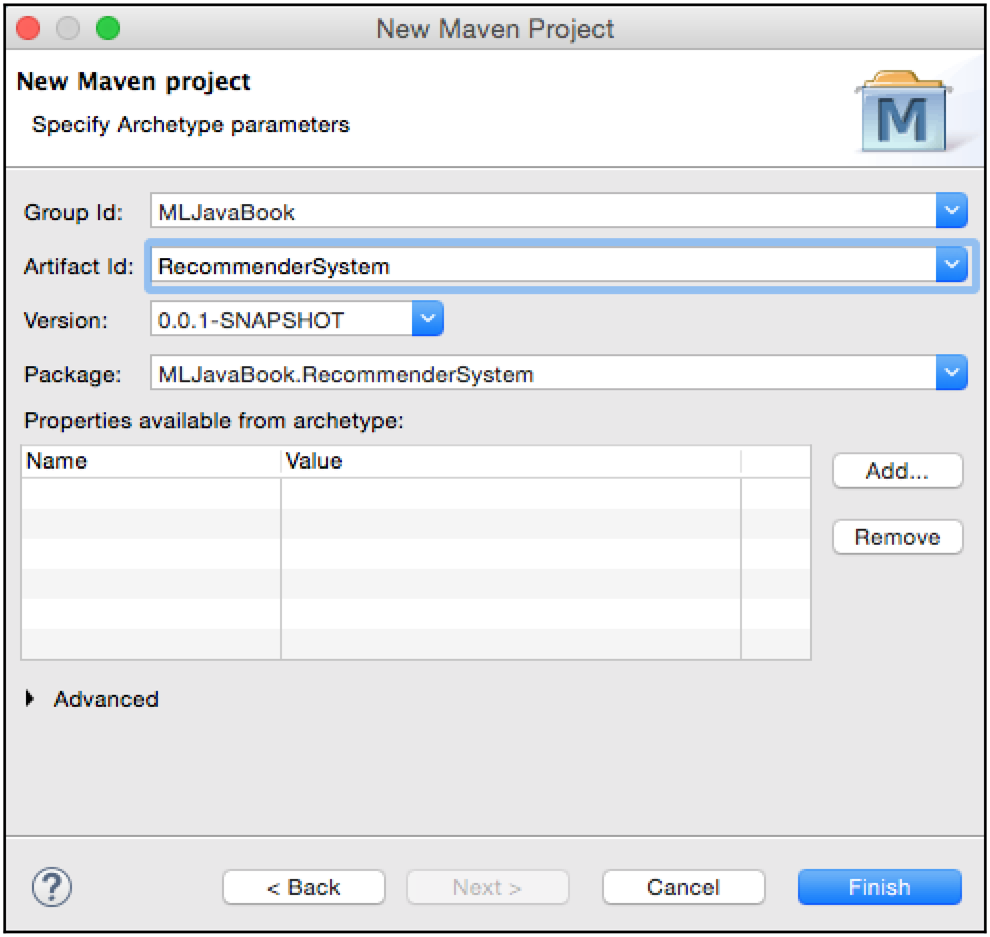
Practical Data Science Cookbook
› Tony Ojeda
★★★★☆ 13
Kindle Edition
$25.00

LOOK INSIDE!
DATA MINING
Machine Learning with Spark
Learning Spark
Data Science for Business
Practical Data Science Cookbook



Select a wizard

Create a Maven Project



New Maven Project

New Maven project

Specify Archetype parameters

Group Id: MLJavaBook

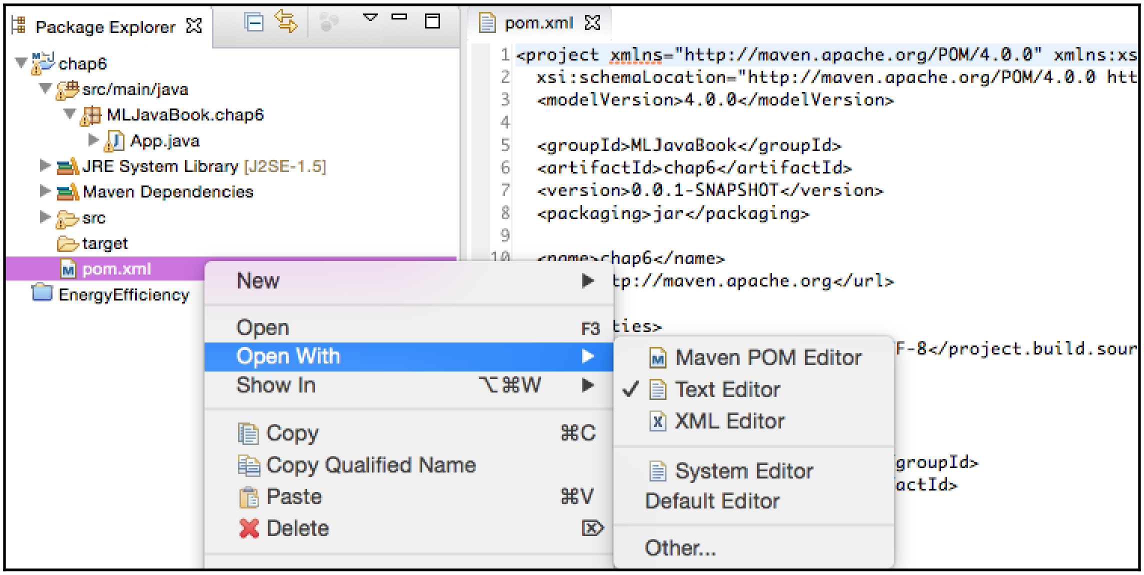
Artifact Id: RecommenderSystem

Version: 0.0.1-SNAPSHOT

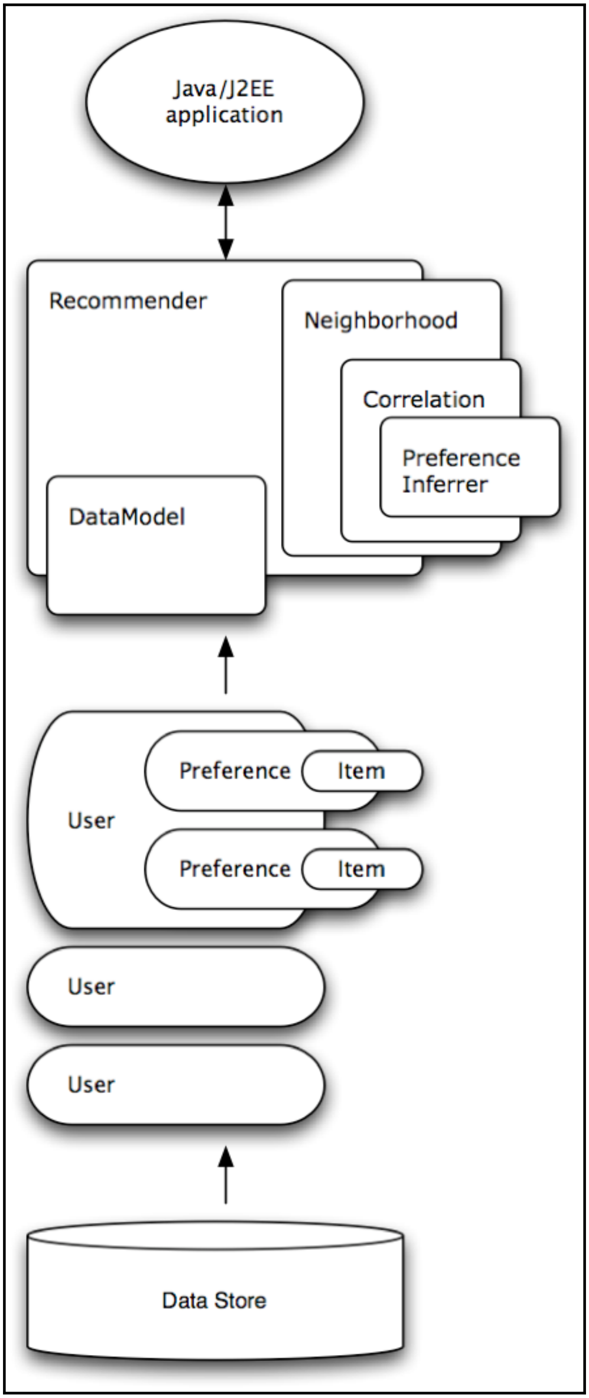
Package: MLJavaBook.RecommenderSystem

Properties available from archetype:

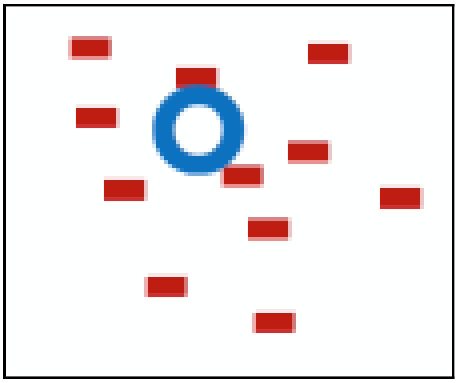
Name	Value

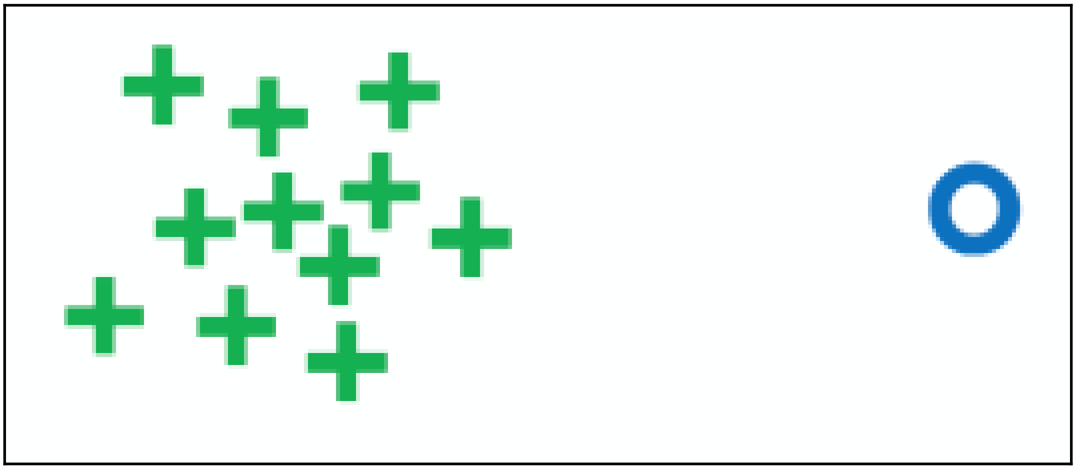


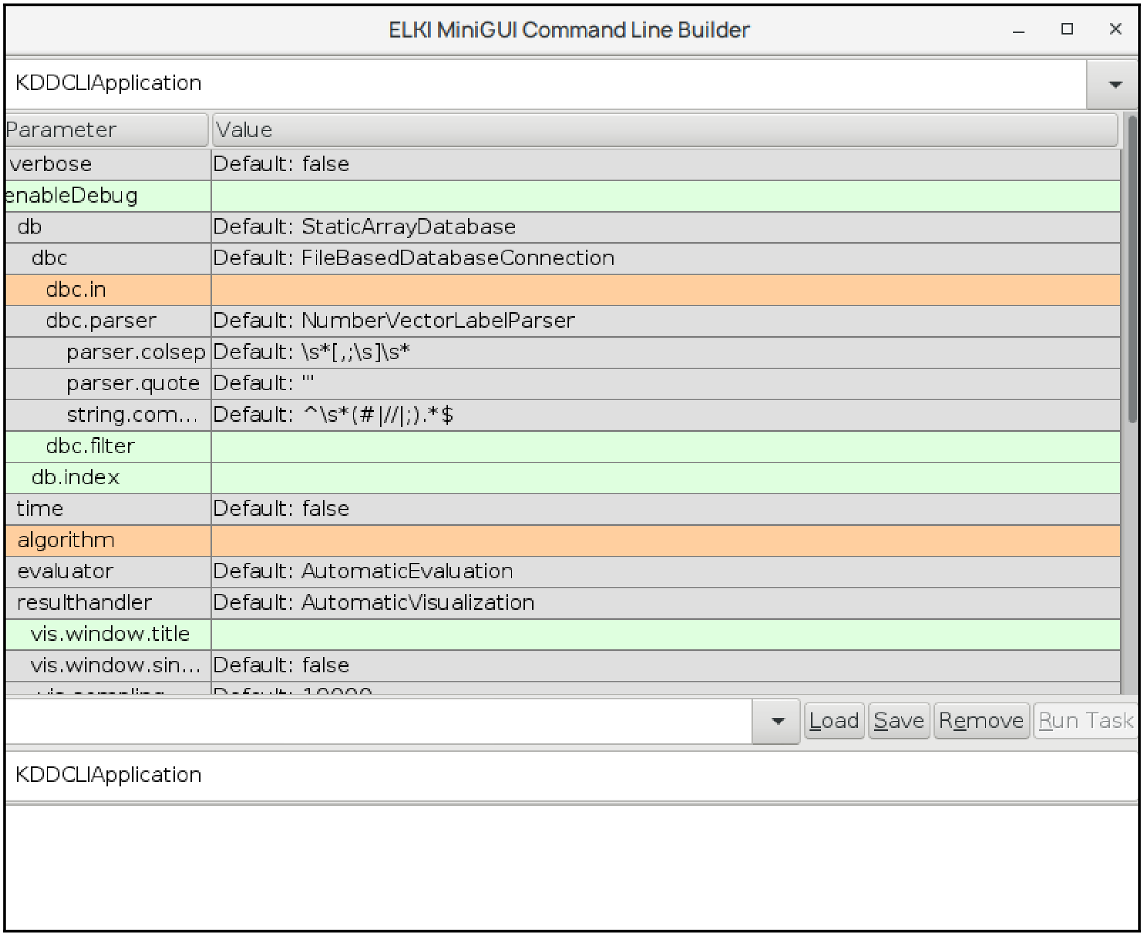
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xs
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 htt
  <modelVersion>4.0.0</modelVersion>

  <groupId>MLJavaBook</groupId>
  <artifactId>chap6</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>chap6</name>
                tp://maven.apache.org</url>

                ties>
                                          F-8</project.build.sour





                                          groupId>
                                          actId>
```


Java/J2EE application
Recommender
Neighborhood
Correlation
Preference Inferrer
DataModel
User
Preference
Item
Preference
Item
User
User
Data Store

Chapter 7: Fraud and Anomaly Detection

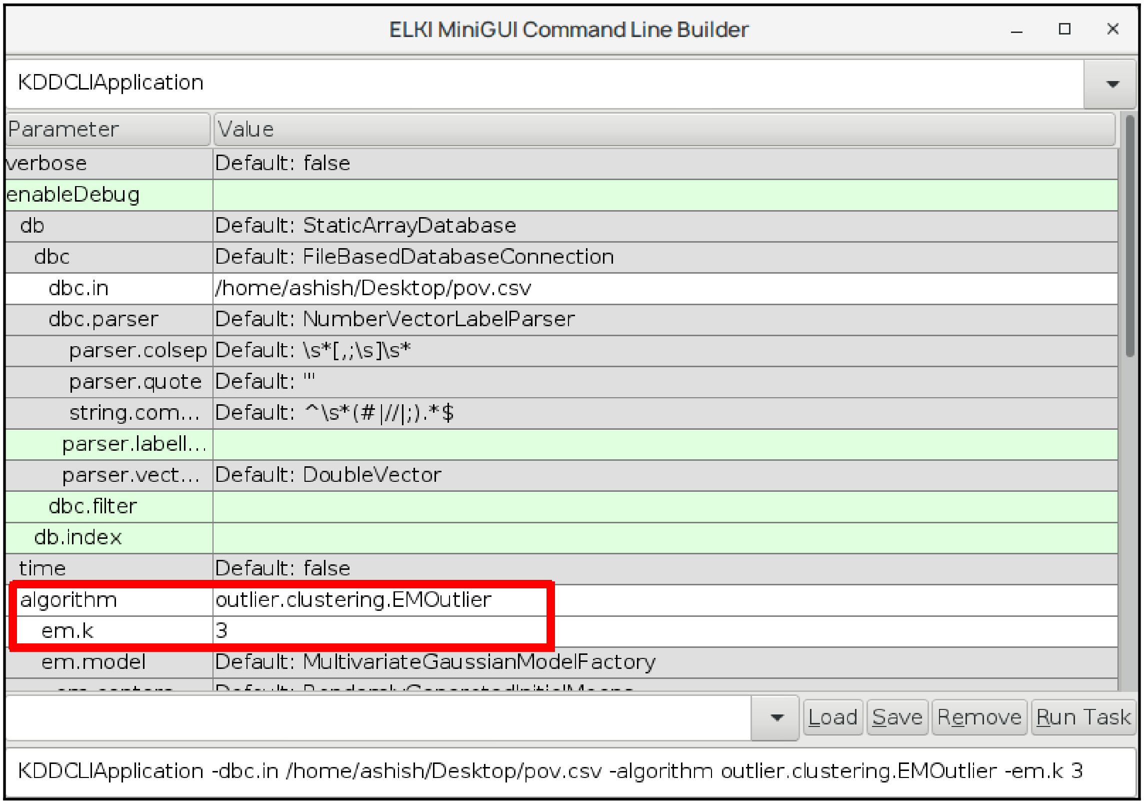






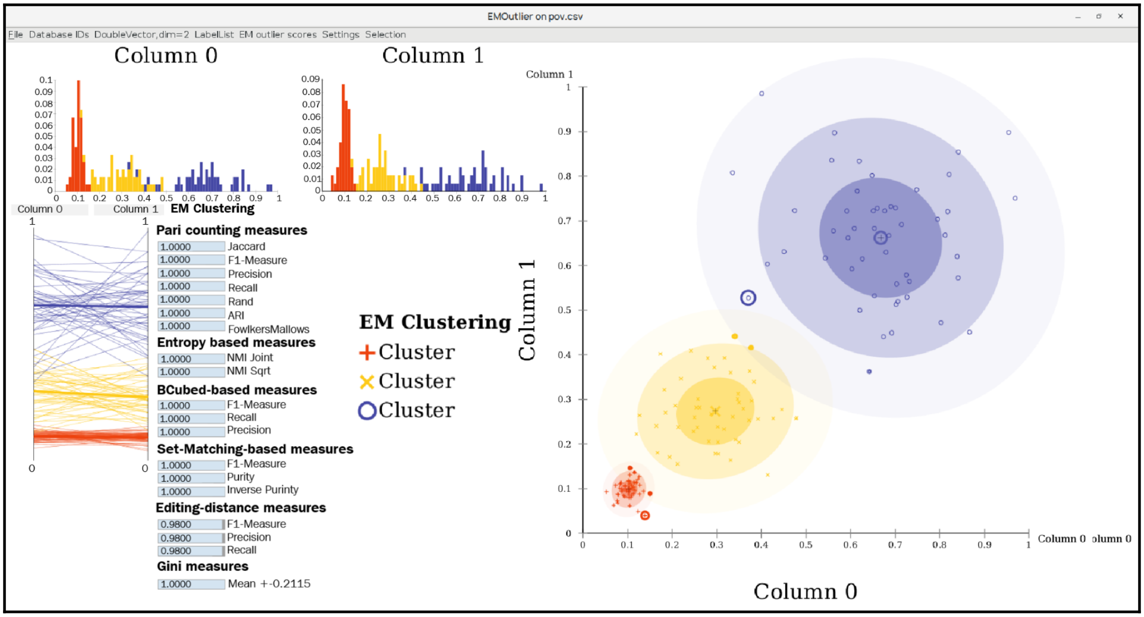
Parameter	Value
verbose	Default: false
enableDebug	
db	Default: StaticArrayDatabase
dbc	Default: FileBasedDatabaseConnection
dbc.in	
dbc.parser	Default: NumberVectorLabelParser
parser.colsep	Default: \s*[,;\s]\s*
parser.quote	Default: "'
string.com...	Default: ^\s*(#|//|;).*$
dbc.filter	
db.index	
time	Default: false
algorithm	
evaluator	Default: AutomaticEvaluation
resulthandler	Default: AutomaticVisualization
vis.window.title	
vis.window.sin...	Default: false
vis.sampling	Default: 10000

KDDCLIApplication



Parameter	Value
verbose	Default: false
enableDebug	
db	Default: StaticArrayDatabase
dbc	Default: FileBasedDatabaseConnection
dbc.in	/home/ashish/Desktop/pov.csv
dbc.parser	Default: NumberVectorLabelParser
parser.colsep	Default: \s*[,;\s]\s*
parser.quote	Default: "'
string.com...	Default: ^\s*(#|//|;).*$
parser.labell...	
parser.vect...	Default: DoubleVector
dbc.filter	
db.index	
time	Default: false
algorithm	outlier.clustering.EMOutlier
em.k	3
em.model	Default: MultivariateGaussianModelFactory

```
KDDCLIApplication -dbc.in /home/ashish/Desktop/pov.csv -algorithm outlier.clustering.EMOutlier -em.k 3
```



Pari counting measures

```
1.0000 Jaccard
1.0000 F1-Measure
1.0000 Precision
1.0000 Recall
1.0000 Rand
1.0000 ARI
1.0000 FowlkersMallows
```

Entropy based measures

```
1.0000 NMI Joint
1.0000 NMI Sqrt
```

BCubed-based measures

```
1.0000 F1-Measure
1.0000 Recall
1.0000 Precision
```

Set-Matching-based measures

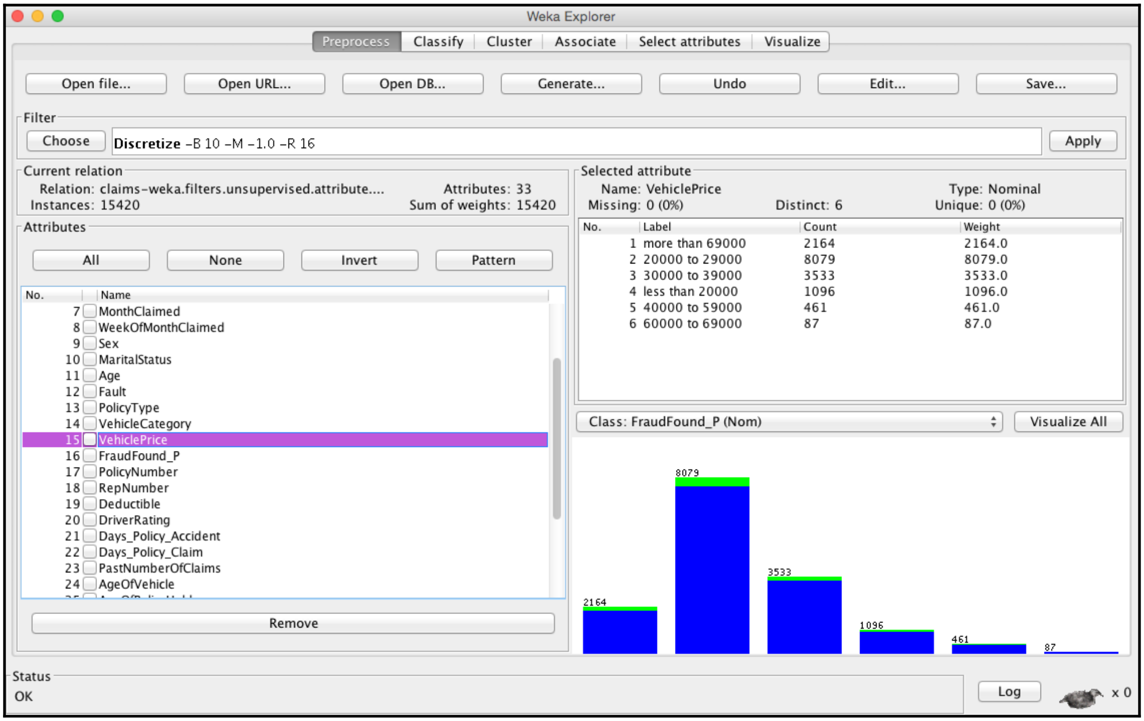
```
1.0000 F1-Measure
1.0000 Purity
1.0000 Inverse Purinty
```

Editing-distance measures

```
0.9800 F1-Measure
0.9800 Precision
0.9800 Recall
```

Gini measures

```
1.0000 Mean +-0.2115
```



Filter

Discretize -B 10 -M -1.0 -R 16

Current relation

Relation: claims-weka.filters.unsupervised.attribute.... Attributes: 33
Instances: 15420 Sum of weights: 15420

Attributes

No.	Name
7	MonthClaimed
8	WeekOfMonthClaimed
9	Sex
10	MaritalStatus
11	Age
12	Fault
13	PolicyType
14	VehicleCategory
15	VehiclePrice
16	FraudFound_P
17	PolicyNumber
18	RepNumber
19	Deductible
20	DriverRating
21	Days_Policy_Accident
22	Days_Policy_Claim
23	PastNumberOfClaims
24	AgeOfVehicle

Selected attribute

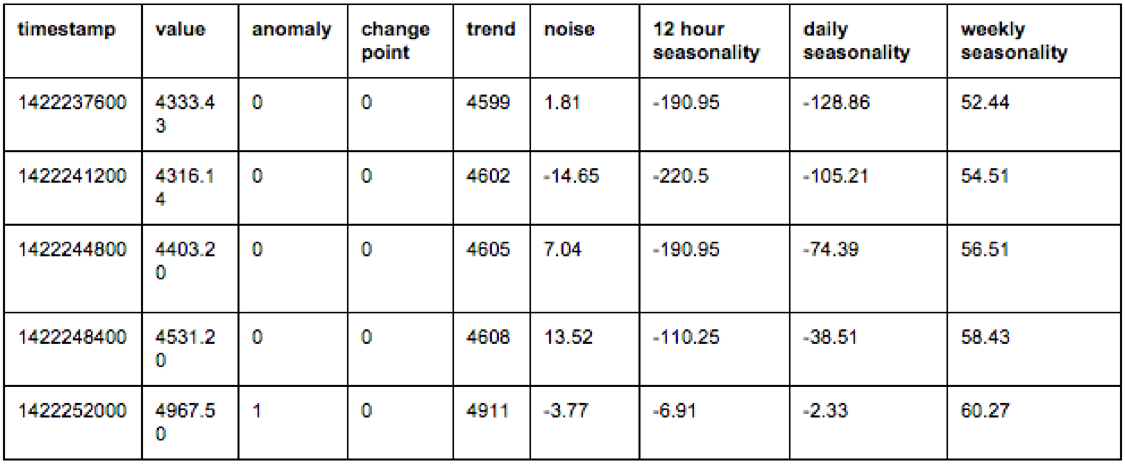
Name: VehiclePrice Type: Nominal
Missing: 0 (0%) Distinct: 6 Unique: 0 (0%)

No.	Label	Count	Weight
1	more than 69000	2164	2164.0
2	20000 to 29000	8079	8079.0
3	30000 to 39000	3533	3533.0
4	less than 20000	1096	1096.0
5	40000 to 59000	461	461.0
6	60000 to 69000	87	87.0

Class: FraudFound_P (Nom)

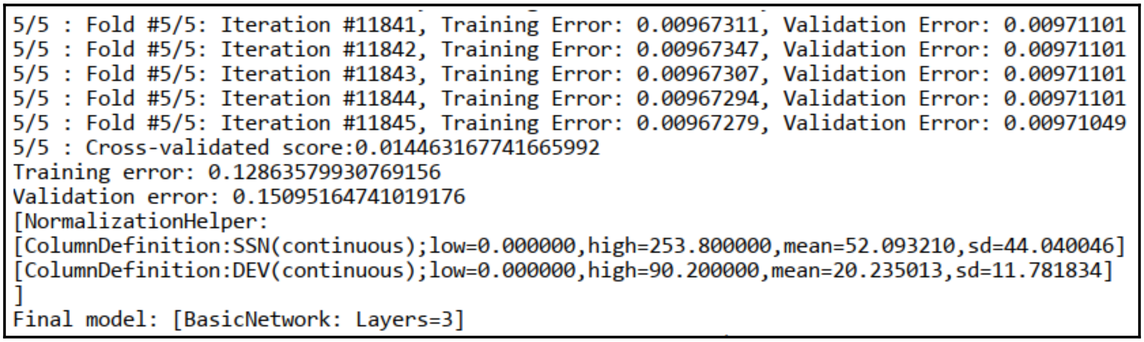
Status
OK

x 0

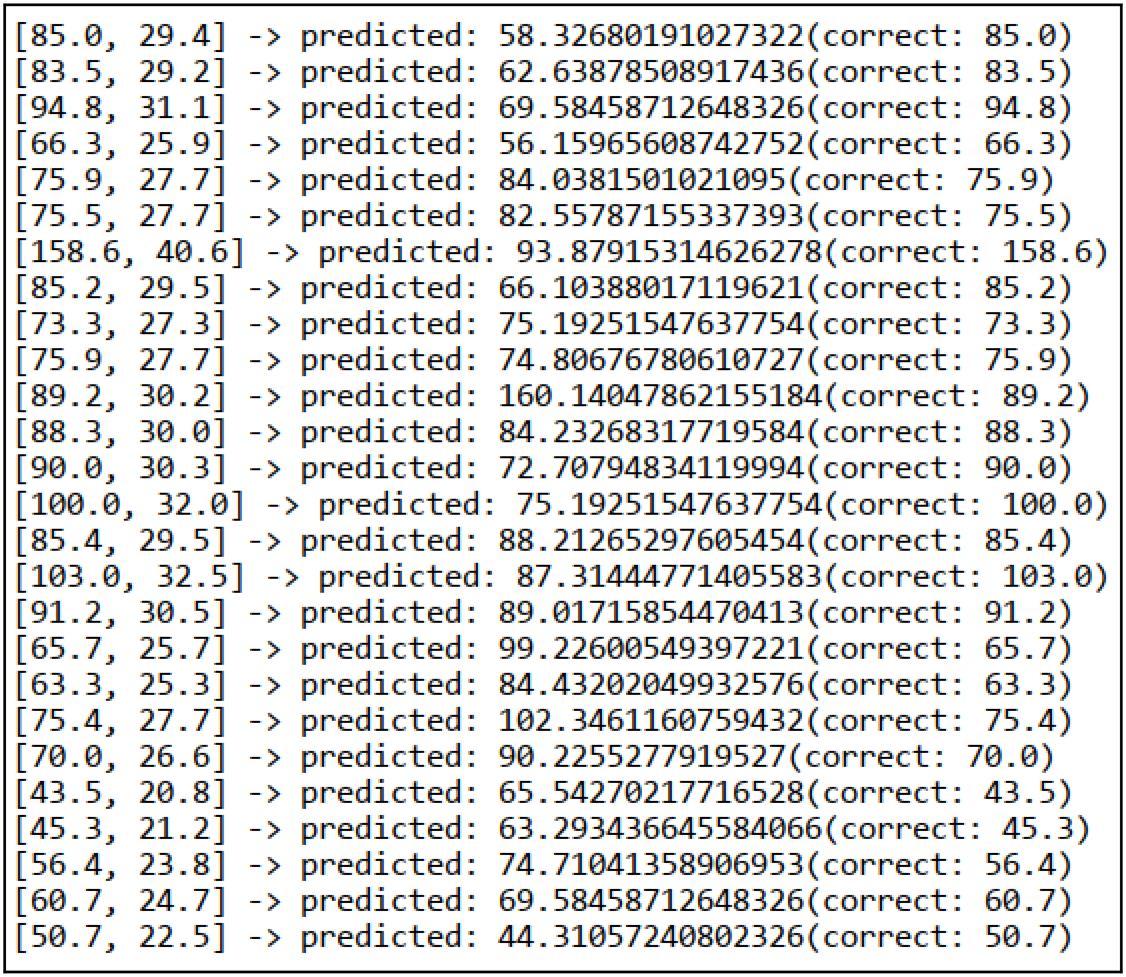


timestamp	value	anomaly	change point	trend	noise	12 hour seasonality	daily seasonality	weekly seasonality
1422237600	4333.43	0	0	4599	1.81	-190.95	-128.86	52.44
1422241200	4316.14	0	0	4602	-14.65	-220.5	-105.21	54.51
1422244800	4403.20	0	0	4605	7.04	-190.95	-74.39	56.51
1422248400	4531.20	0	0	4608	13.52	-110.25	-38.51	58.43
1422252000	4967.50	1	0	4911	-3.77	-6.91	-2.33	60.27

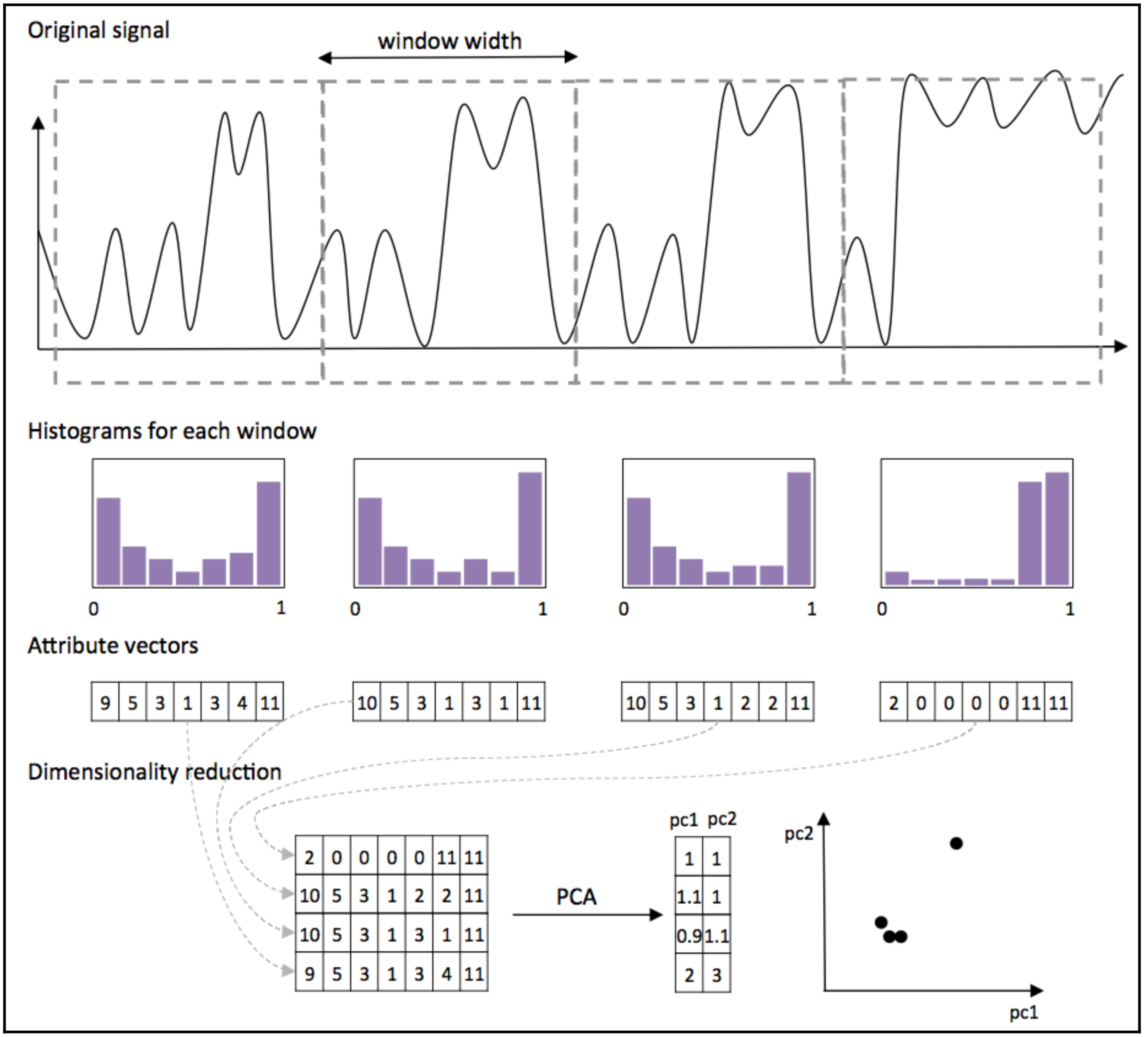
Snippet of the synthetic time-series data



```
5/5 : Fold #5/5: Iteration #11841, Training Error: 0.00967311, Validation Error: 0.00971101
5/5 : Fold #5/5: Iteration #11842, Training Error: 0.00967347, Validation Error: 0.00971101
5/5 : Fold #5/5: Iteration #11843, Training Error: 0.00967307, Validation Error: 0.00971101
5/5 : Fold #5/5: Iteration #11844, Training Error: 0.00967294, Validation Error: 0.00971101
5/5 : Fold #5/5: Iteration #11845, Training Error: 0.00967279, Validation Error: 0.00971049
5/5 : Cross-validated score:0.014463167741665992
Training error: 0.12863579930769156
Validation error: 0.15095164741019176
[NormalizationHelper:
[ColumnDefinition:SSN(continuous);low=0.000000,high=253.800000,mean=52.093210,sd=44.040046]
[ColumnDefinition:DEV(continuous);low=0.000000,high=90.200000,mean=20.235013,sd=11.781834]
]
Final model: [BasicNetwork: Layers=3]
```



```
[85.0, 29.4] -> predicted: 58.32680191027322(correct: 85.0)
[83.5, 29.2] -> predicted: 62.63878508917436(correct: 83.5)
[94.8, 31.1] -> predicted: 69.58458712648326(correct: 94.8)
[66.3, 25.9] -> predicted: 56.15965608742752(correct: 66.3)
[75.9, 27.7] -> predicted: 84.0381501021095(correct: 75.9)
[75.5, 27.7] -> predicted: 82.55787155337393(correct: 75.5)
[158.6, 40.6] -> predicted: 93.87915314626278(correct: 158.6)
[85.2, 29.5] -> predicted: 66.10388017119621(correct: 85.2)
[73.3, 27.3] -> predicted: 75.19251547637754(correct: 73.3)
[75.9, 27.7] -> predicted: 74.80676780610727(correct: 75.9)
[89.2, 30.2] -> predicted: 160.14047862155184(correct: 89.2)
[88.3, 30.0] -> predicted: 84.23268317719584(correct: 88.3)
[90.0, 30.3] -> predicted: 72.70794834119994(correct: 90.0)
[100.0, 32.0] -> predicted: 75.19251547637754(correct: 100.0)
[85.4, 29.5] -> predicted: 88.21265297605454(correct: 85.4)
[103.0, 32.5] -> predicted: 87.31444771405583(correct: 103.0)
[91.2, 30.5] -> predicted: 89.01715854470413(correct: 91.2)
[65.7, 25.7] -> predicted: 99.22600549397221(correct: 65.7)
[63.3, 25.3] -> predicted: 84.43202049932576(correct: 63.3)
[75.4, 27.7] -> predicted: 102.3461160759432(correct: 75.4)
[70.0, 26.6] -> predicted: 90.2255277919527(correct: 70.0)
[43.5, 20.8] -> predicted: 65.54270217716528(correct: 43.5)
[45.3, 21.2] -> predicted: 63.293436645584066(correct: 45.3)
[56.4, 23.8] -> predicted: 74.71041358906953(correct: 56.4)
[60.7, 24.7] -> predicted: 69.58458712648326(correct: 60.7)
[50.7, 22.5] -> predicted: 44.31057240802326(correct: 50.7)
```



9	5	3	1	3	4	11

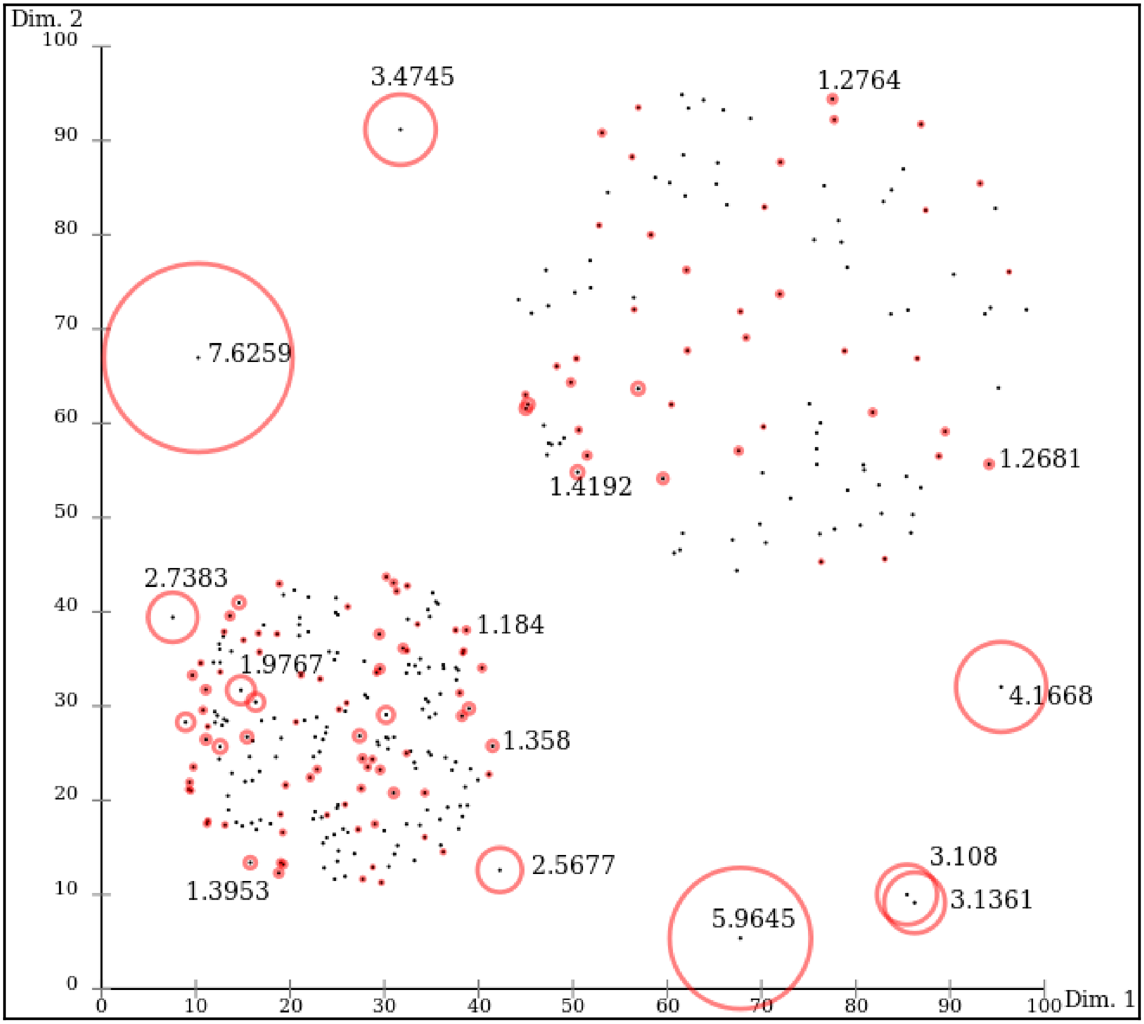
10	5	3	1	3	1	11

10	5	3	1	2	2	11

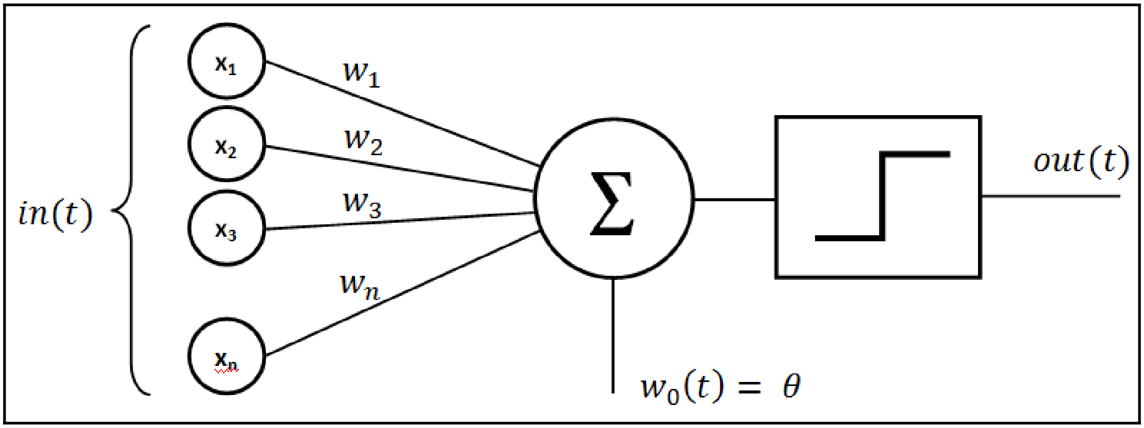
2	0	0	0	0	11	11

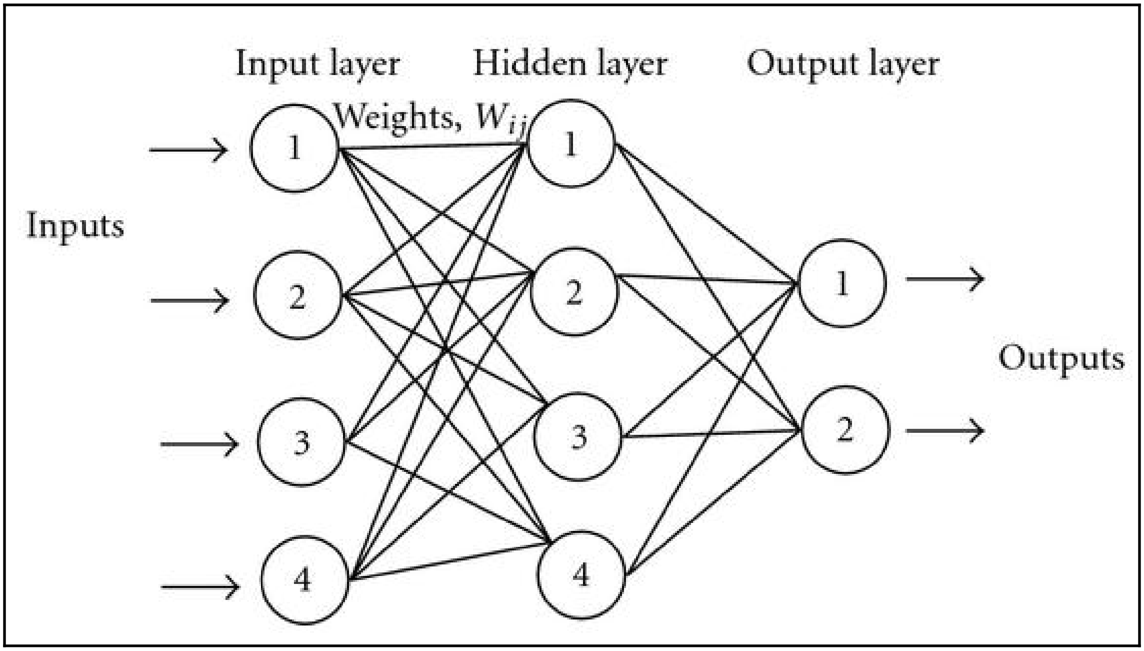
2	0	0	0	0	11	11
10	5	3	1	2	2	11
10	5	3	1	3	1	11
9	5	3	1	3	4	11

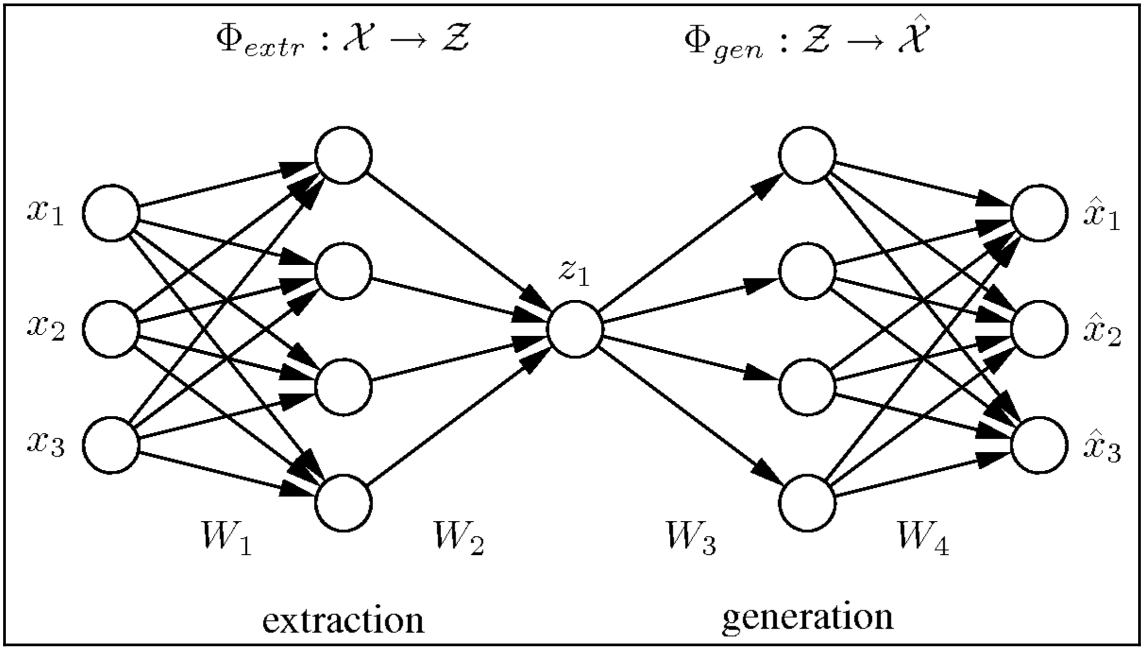
pc1	pc2
1	1
1.1	1
0.9	1.1
2	3

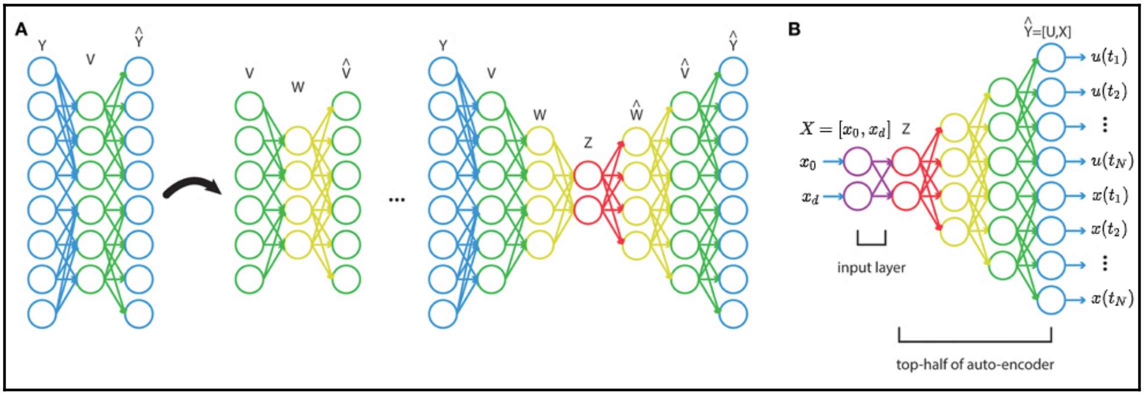

Dim. 2
100
90
80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60
70
80
90
100
Dim. 1
3.4745
1.2764
7.6259
1.2681
1.4192
2.7383
1.184
1.9767
4.1668
1.358
2.5677
3.108
1.3953
3.1361
5.9645

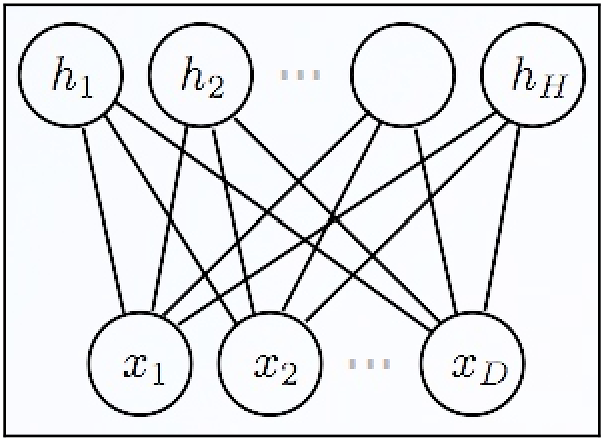
Chapter 8: Image Recognition with Deeplearning4j

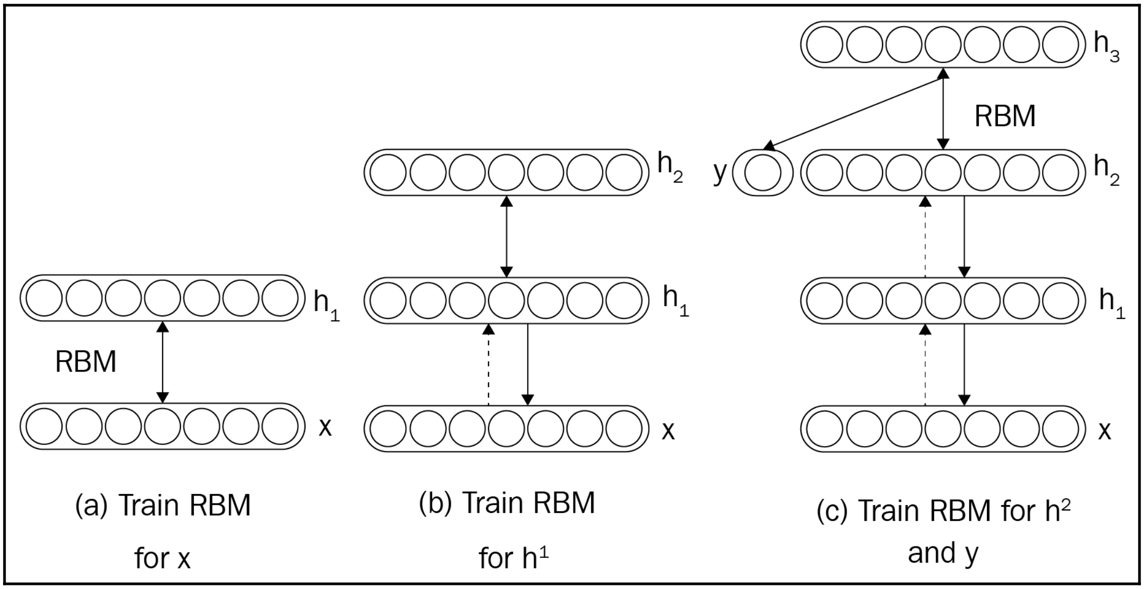





$\Phi_{extr} : \mathcal{X} \to \mathcal{Z}$
$\Phi_{gen} : \mathcal{Z} \to \hat{\mathcal{X}}$
x_1
x_2
x_3
z_1
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
W_1
W_2
W_3
W_4
extraction
generation


A
Y
V
$\hat{Y}$
V
W
$\hat{V}$
...
Y
V
W
Z
$\hat{W}$
$\hat{V}$
$\hat{Y}$
B
$\hat{Y}$=[U,X]
$u(t_1)$
$u(t_2)$
⋮
$u(t_N)$
$x(t_1)$
$x(t_2)$
⋮
$x(t_N)$
$X = [x_0, x_d]$
Z
x_0
x_d
input layer
top-half of auto-encoder

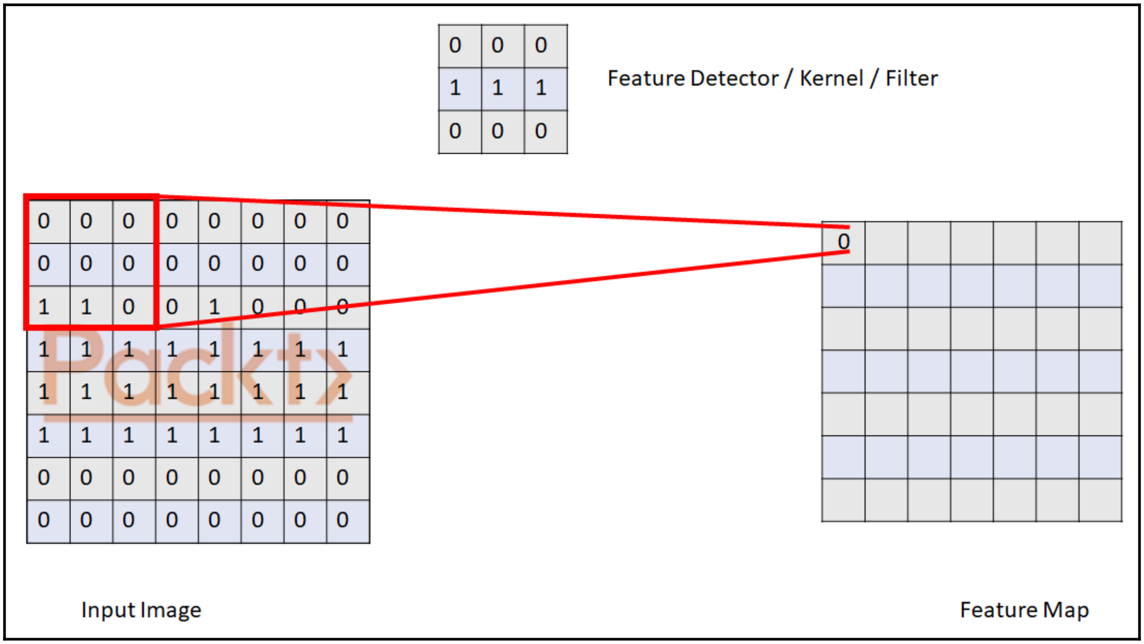




(a) Train RBM for x

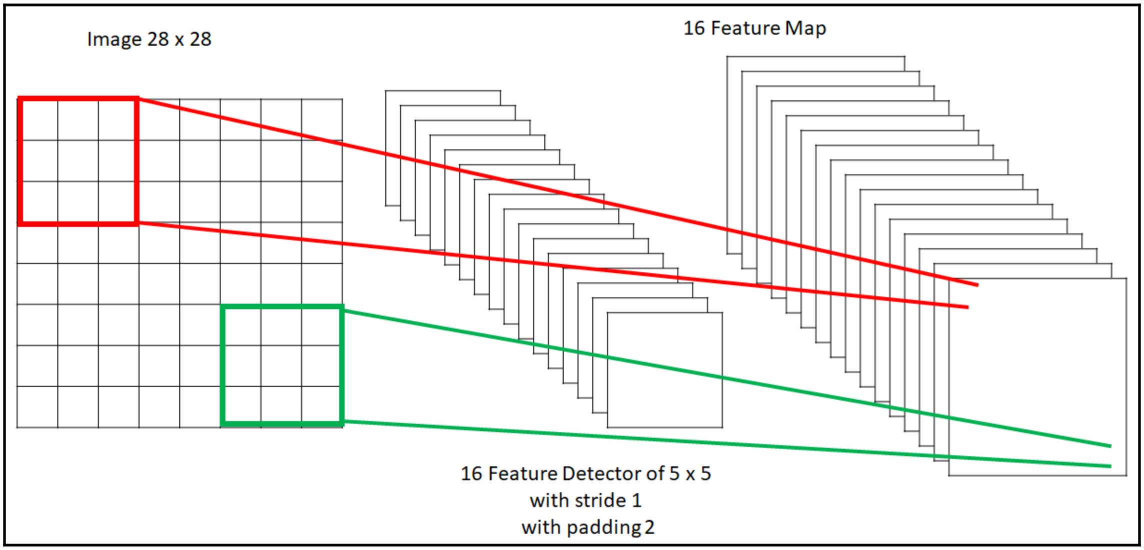
(b) Train RBM for h¹

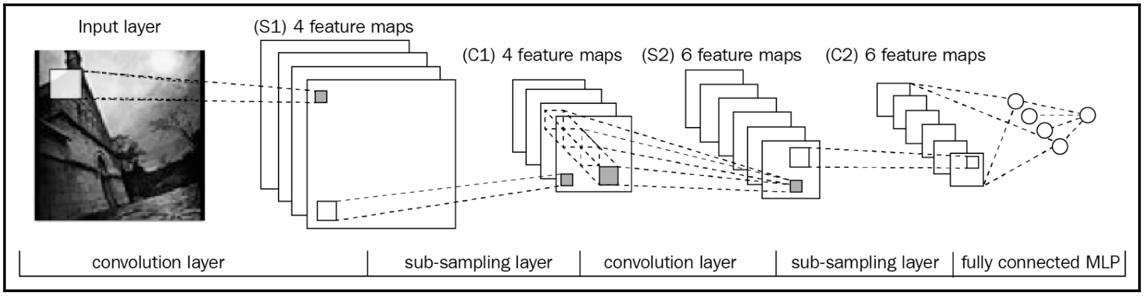
(c) Train RBM for h² and y

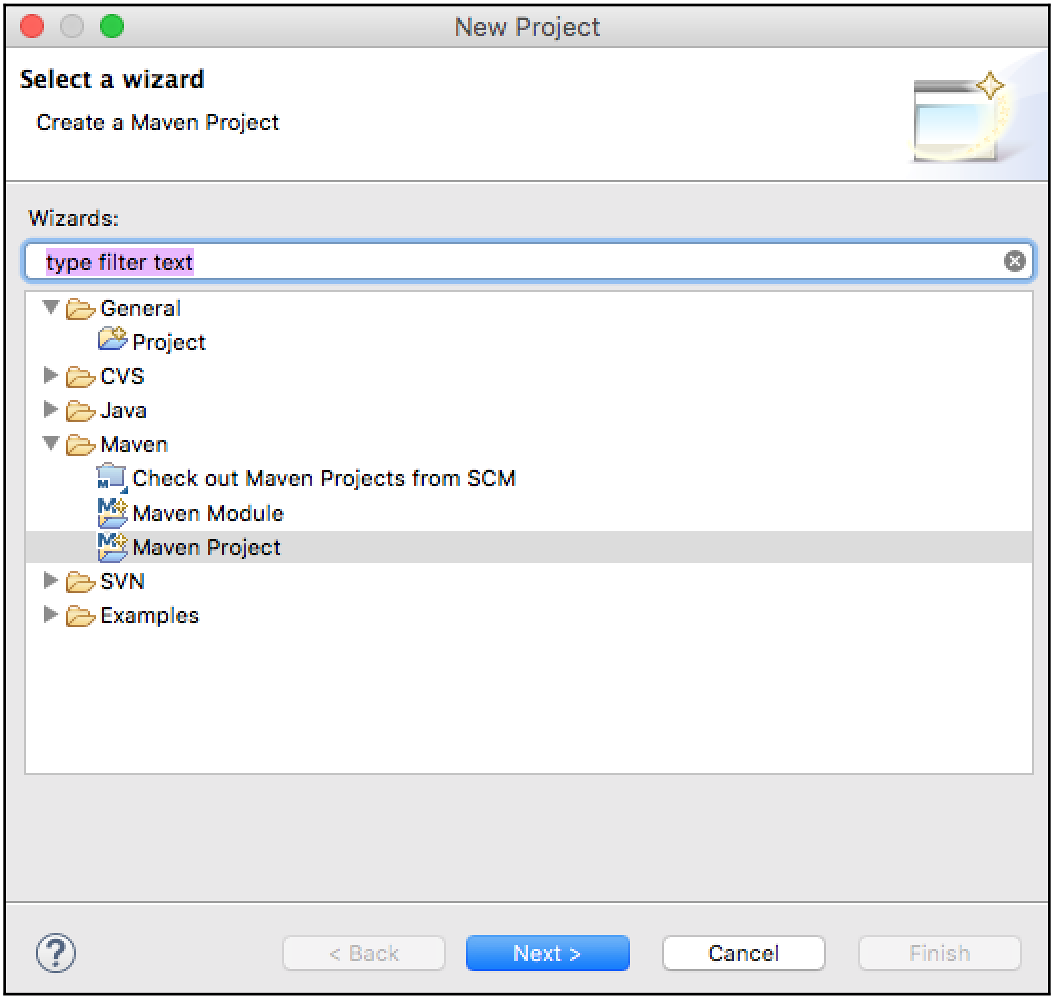


0	0	0
1	1	1
0	0	0

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0






New Project

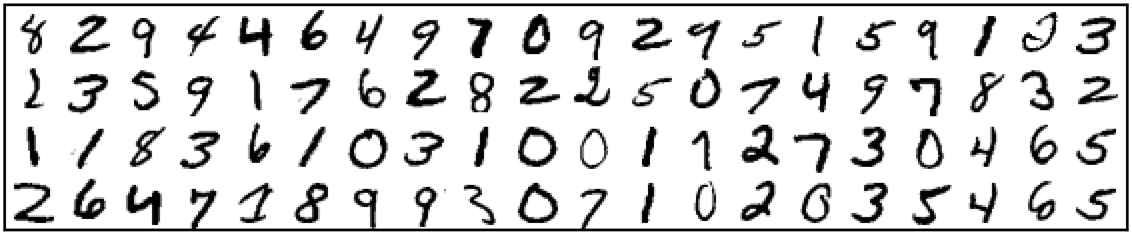
Select a wizard

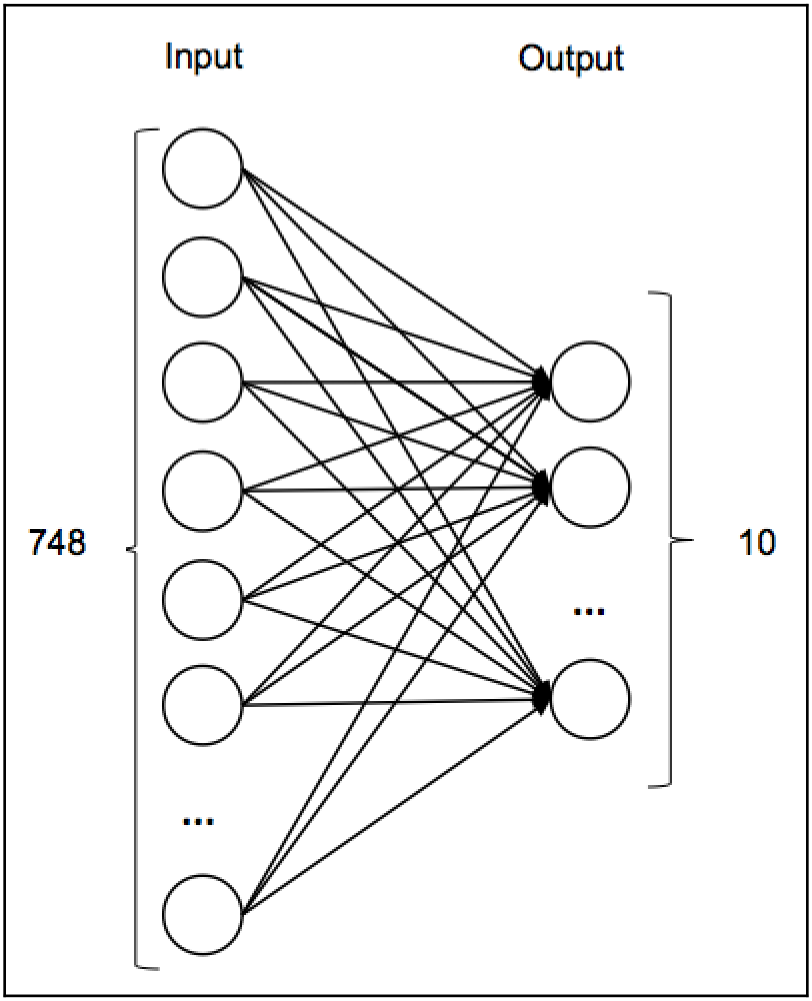
Create a Maven Project

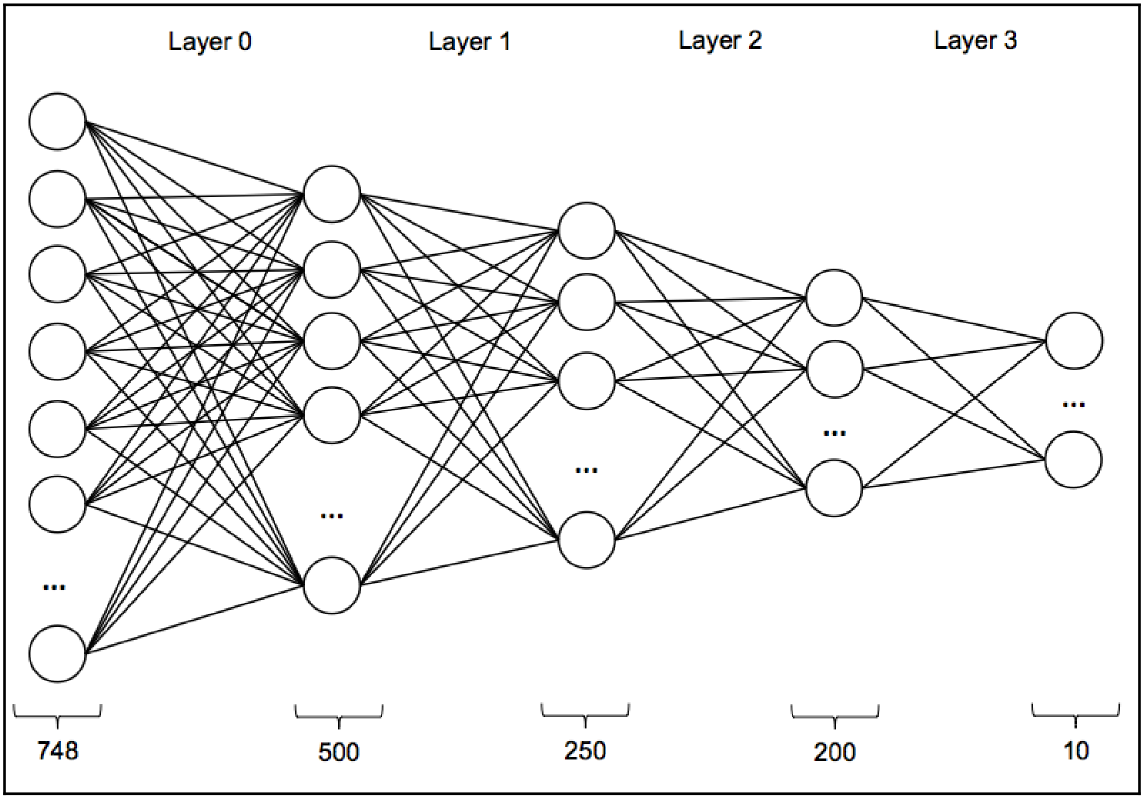
Wizards:

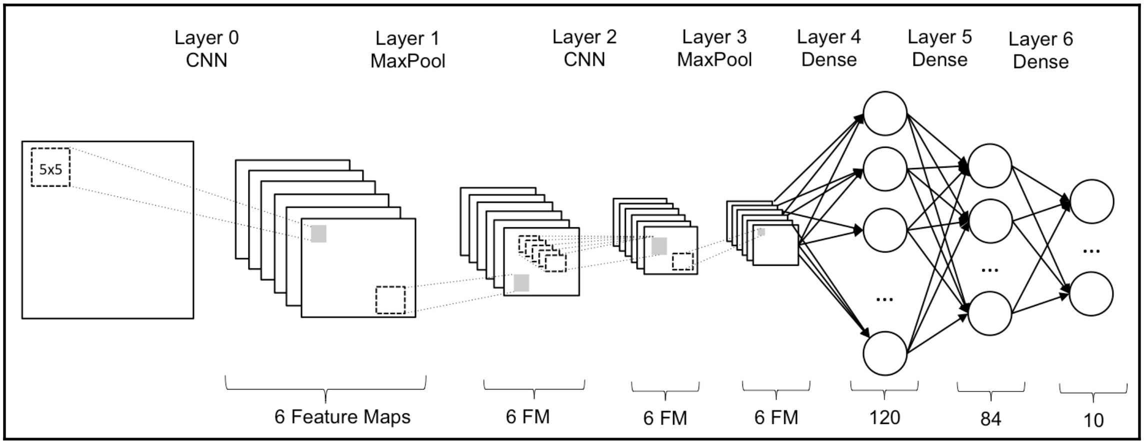
type filter text

- General
  - Project
- CVS
- Java
- Maven
  - Check out Maven Projects from SCM
  - Maven Module
  - Maven Project
- SVN
- Examples

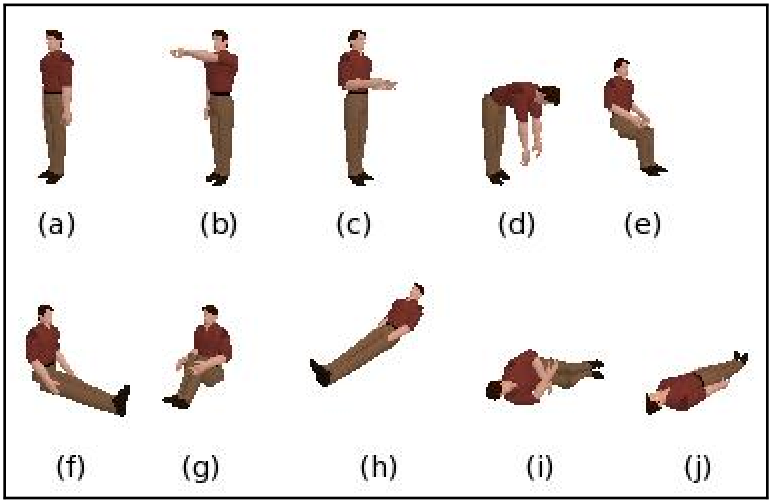


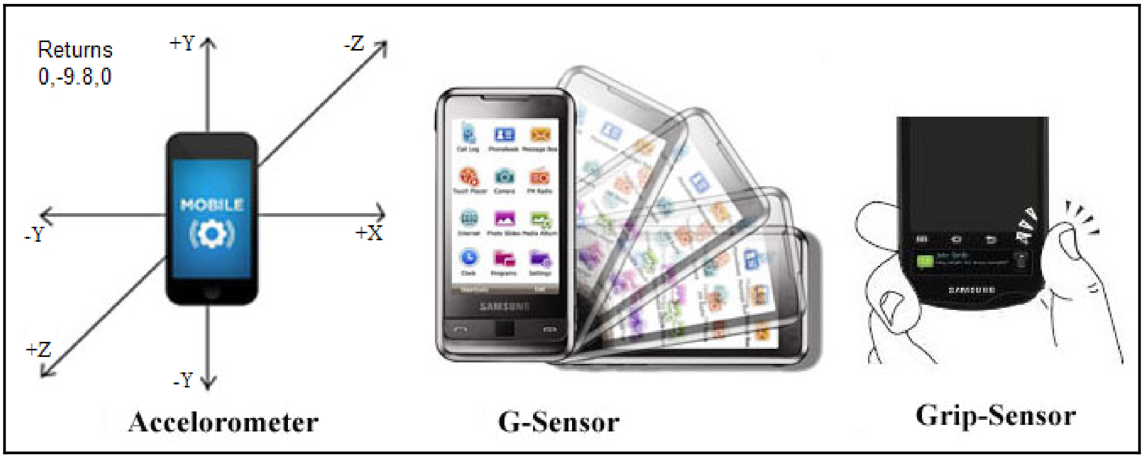

Input
Output
748
10
...
...

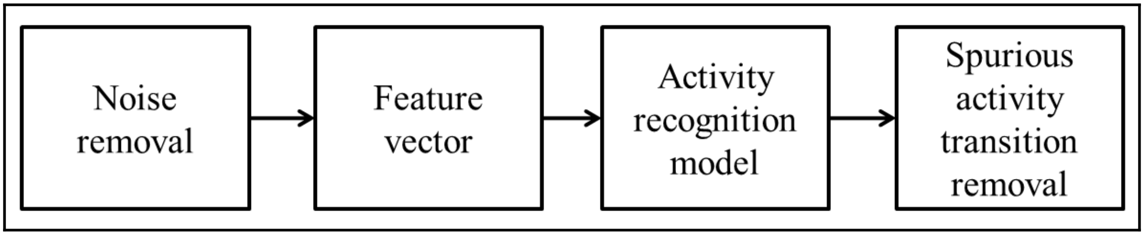

Layer 0
Layer 1
Layer 2
Layer 3
...
...
...
...
...
748
500
250
200
10

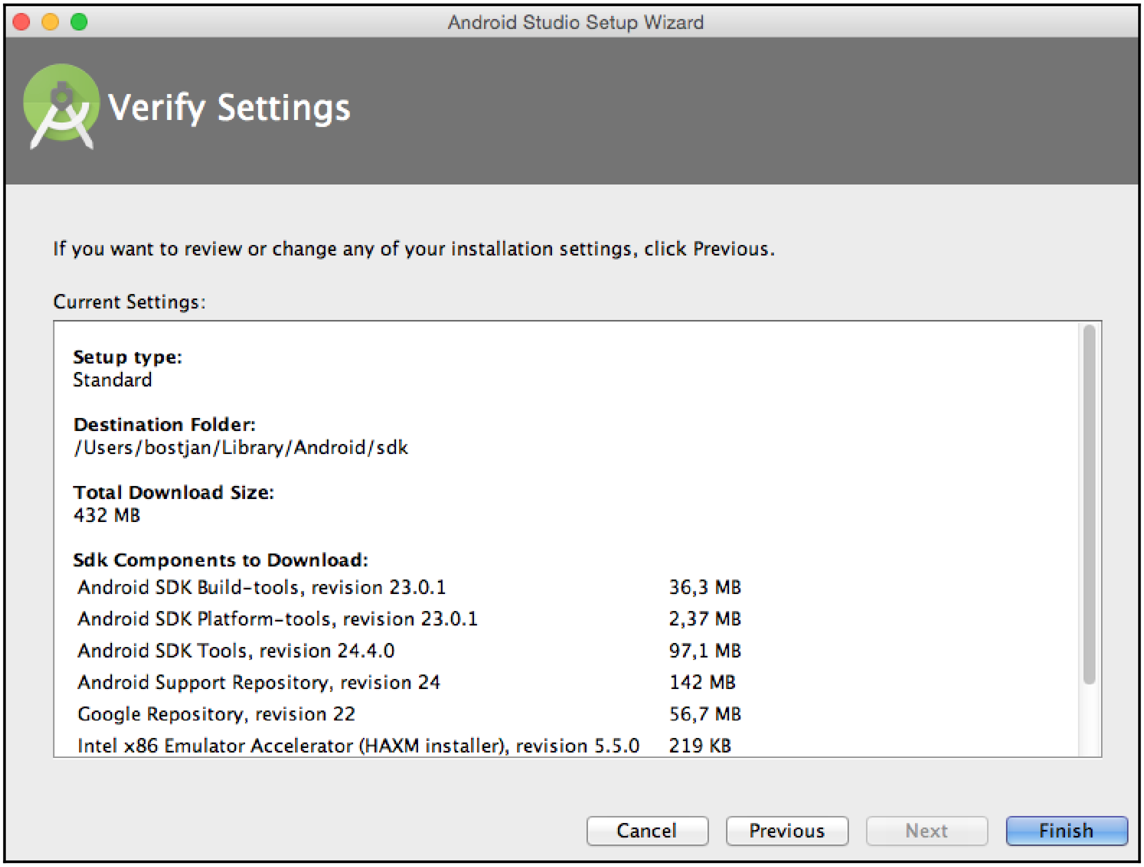

Layer 0
CNN
Layer 1
MaxPool
Layer 2
CNN
Layer 3
MaxPool
Layer 4
Dense
Layer 5
Dense
Layer 6
Dense
5x5
...
...
...
6 Feature Maps
6 FM
6 FM
6 FM
120
84
10

Chapter 9: Activity Recognition with Mobile Phone Sensors






Noise removal
Feature vector
Activity recognition model
Spurious activity transition removal



If you want to review or change any of your installation settings, click Previous.

Current Settings:

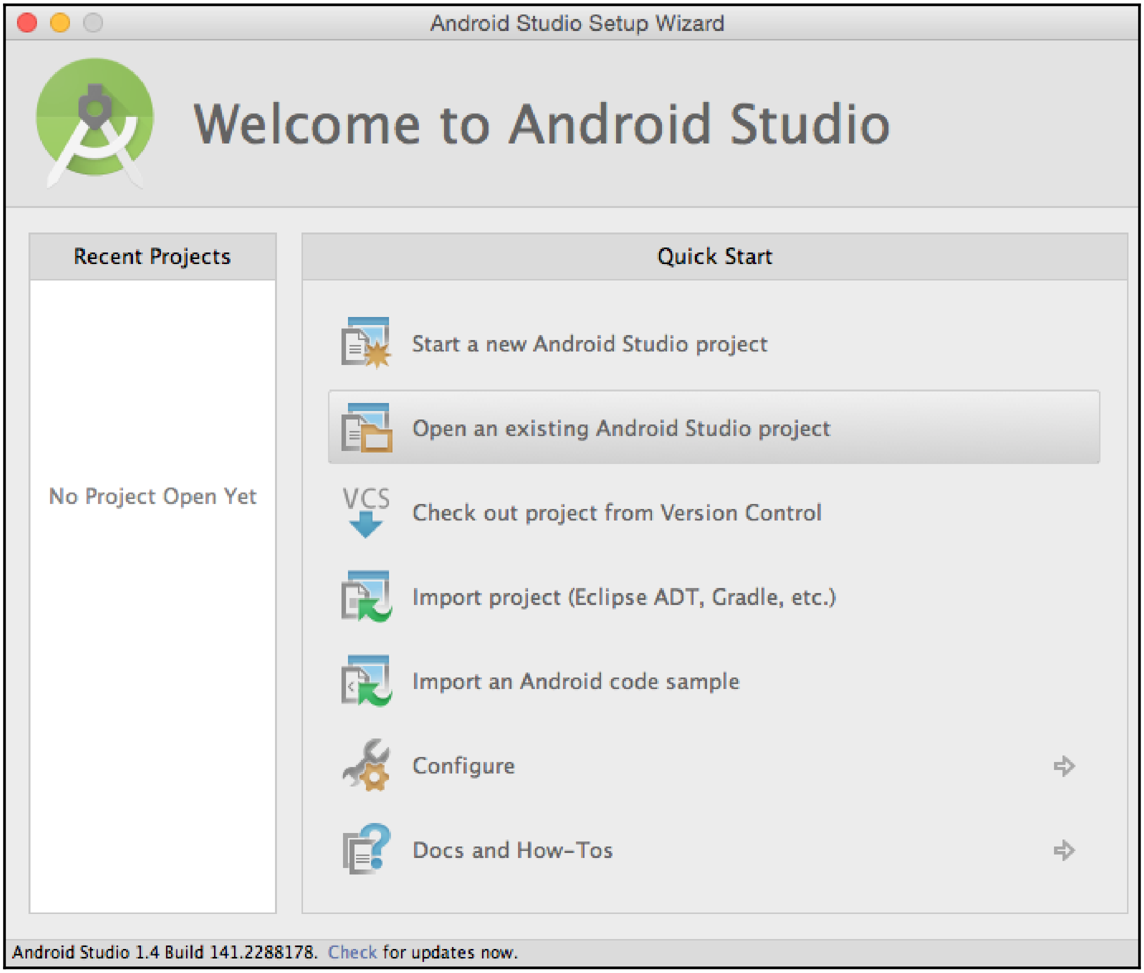
Setup type:
Standard

Destination Folder:
/Users/bostjan/Library/Android/sdk

Total Download Size:
432 MB

Sdk Components to Download:

Android SDK Build-tools, revision 23.0.1	36,3 MB
Android SDK Platform-tools, revision 23.0.1	2,37 MB
Android SDK Tools, revision 24.4.0	97,1 MB
Android Support Repository, revision 24	142 MB
Google Repository, revision 22	56,7 MB
Intel x86 Emulator Accelerator (HAXM installer), revision 5.5.0	219 KB


Android Studio Setup Wizard

Welcome to Android Studio

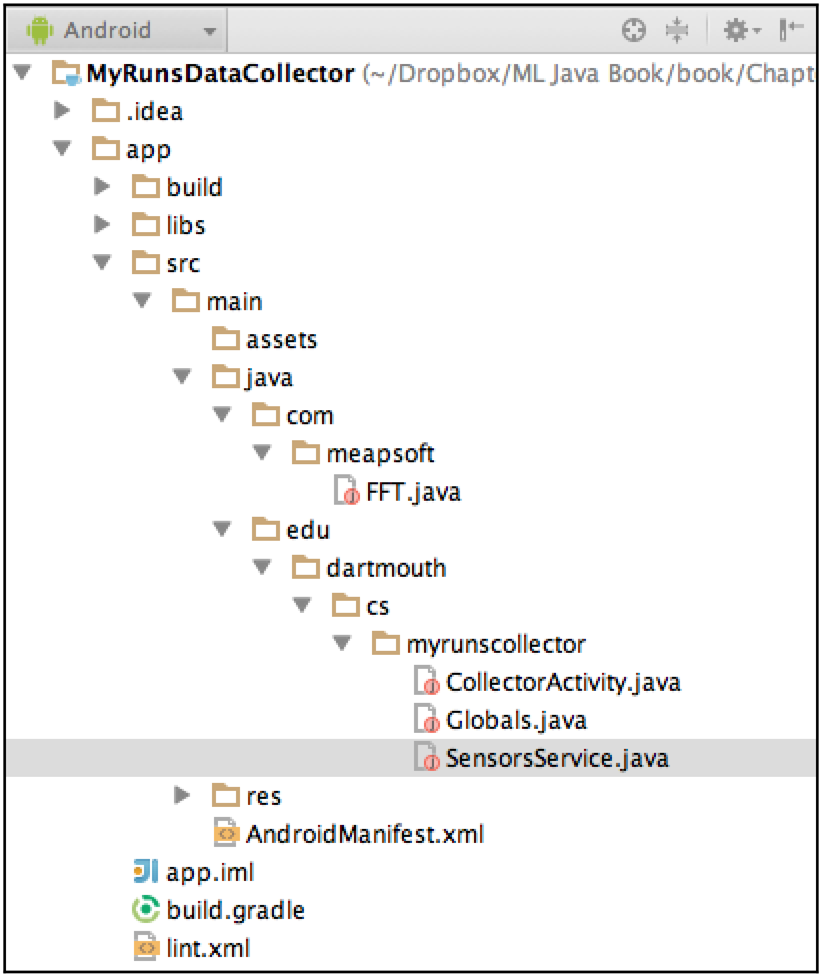
Recent Projects

No Project Open Yet

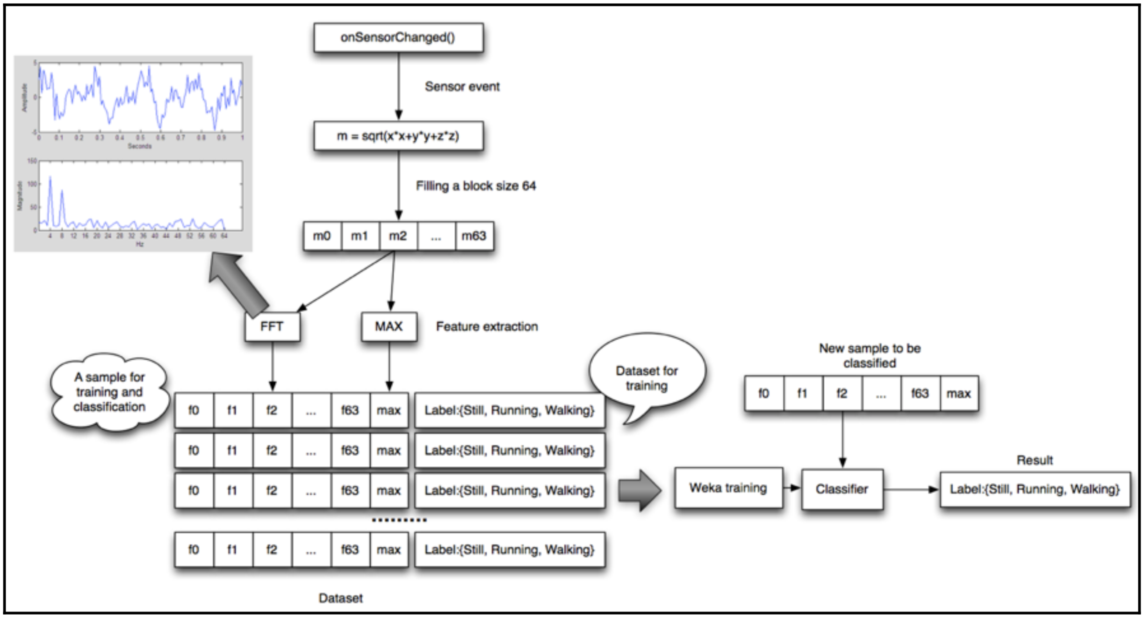
Quick Start

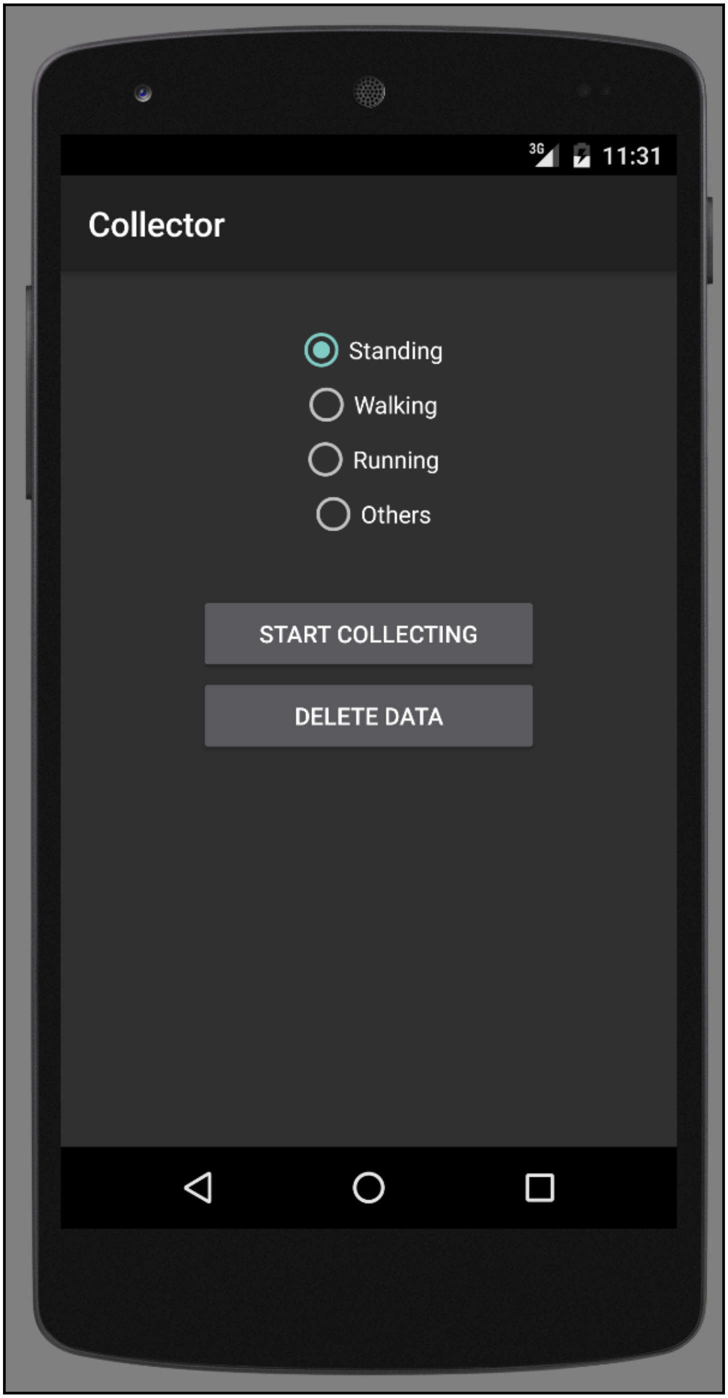
Start a new Android Studio project
Open an existing Android Studio project
VCS
Check out project from Version Control
Import project (Eclipse ADT, Gradle, etc.)
Import an Android code sample
Configure
Docs and How-Tos

Android Studio 1.4 Build 141.2288178. Check for updates now.



```
MyRunsDataCollector (~/Dropbox/ML Java Book/book/Chapt
  .idea
  app
    build
    libs
    src
      main
        assets
        java
          com
            meapsoft
              FFT.java
          edu
            dartmouth
              cs
                myrunscollector
                  CollectorActivity.java
                  Globals.java
                  SensorsService.java
        res
        AndroidManifest.xml
    app.iml
    build.gradle
    lint.xml
```


onSensorChanged()
Sensor event
m = sqrt(x*x+y*y+z*z)
Filling a block size 64
m0
m1
m2
...
m63
Amplitude
Seconds
Magnitude
Hz
FFT
MAX
Feature extraction
A sample for training and classification
f0
f1
f2
...
f63
max
Label:{Still, Running, Walking}
Dataset for training
Dataset
New sample to be classified
Weka training
Classifier
Result
Label:{Still, Running, Walking}



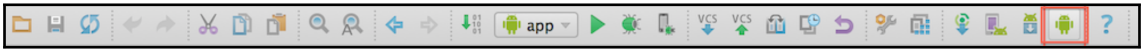
Collector

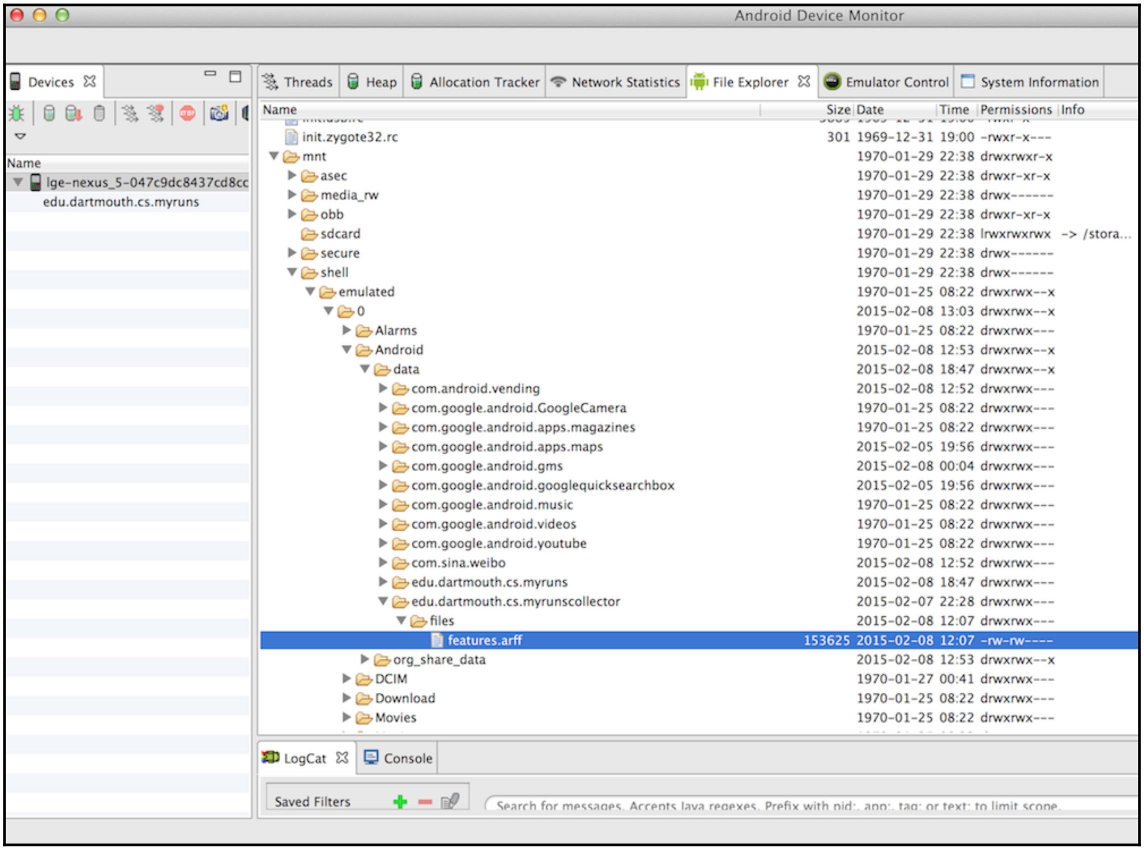
● Standing

○ Walking

○ Running

○ Others

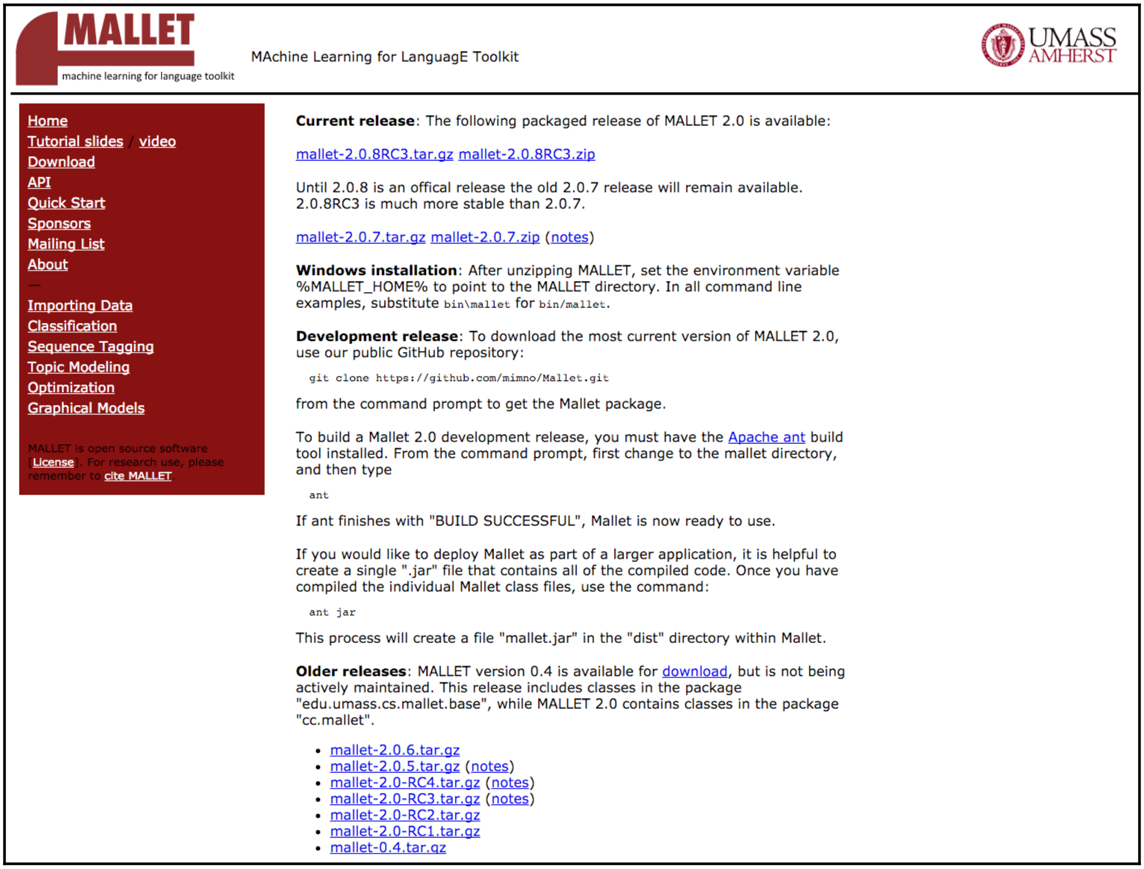




Name
lge-nexus_5-047c9dc8437cd8cc
edu.dartmouth.cs.myruns

Name	Size	Date	Time	Permissions	Info
init.zygote32.rc	301	1969-12-31	19:00	-rwxr-x---	
mnt		1970-01-29	22:38	drwxrwxr-x	
asec		1970-01-29	22:38	drwxr-xr-x	
media_rw		1970-01-29	22:38	drwx------	
obb		1970-01-29	22:38	drwxr-xr-x	
sdcard		1970-01-29	22:38	lrwxrwxrwx	-> /stora...
secure		1970-01-29	22:38	drwx------	
shell		1970-01-29	22:38	drwx------	
emulated		1970-01-25	08:22	drwxrwx--x	
0		2015-02-08	13:03	drwxrwx--x	
Alarms		1970-01-25	08:22	drwxrwx---	
Android		2015-02-08	12:53	drwxrwx--x	
data		2015-02-08	18:47	drwxrwx--x	
com.android.vending		2015-02-08	12:52	drwxrwx---	
com.google.android.GoogleCamera		1970-01-25	08:22	drwxrwx---	
com.google.android.apps.magazines		1970-01-25	08:22	drwxrwx---	
com.google.android.apps.maps		2015-02-05	19:56	drwxrwx---	
com.google.android.gms		2015-02-08	00:04	drwxrwx---	
com.google.android.googlequicksearchbox		2015-02-05	19:56	drwxrwx---	
com.google.android.music		1970-01-25	08:22	drwxrwx---	
com.google.android.videos		1970-01-25	08:22	drwxrwx---	
com.google.android.youtube		1970-01-25	08:22	drwxrwx---	
com.sina.weibo		2015-02-08	12:52	drwxrwx---	
edu.dartmouth.cs.myruns		2015-02-08	18:47	drwxrwx---	
edu.dartmouth.cs.myrunscollector		2015-02-07	22:28	drwxrwx---	
files		2015-02-08	12:07	drwxrwx---	
features.arff	153625	2015-02-08	12:07	-rw-rw----	
org_share_data		2015-02-08	12:53	drwxrwx--x	
DCIM		1970-01-27	00:41	drwxrwx---	
Download		1970-01-25	08:22	drwxrwx---	
Movies		1970-01-25	08:22	drwxrwx---	

Chapter 10: Text Mining with Mallet – Topic Modeling and Spam Detection



MAchine Learning for LanguagE Toolkit

MALLET is open source software [License]. For research use, please remember to cite MALLET.

Current release: The following packaged release of MALLET 2.0 is available:

mallet-2.0.8RC3.tar.gz mallet-2.0.8RC3.zip

Until 2.0.8 is an offical release the old 2.0.7 release will remain available. 2.0.8RC3 is much more stable than 2.0.7.

mallet-2.0.7.tar.gz mallet-2.0.7.zip (notes)

Windows installation: After unzipping MALLET, set the environment variable %MALLET_HOME% to point to the MALLET directory. In all command line examples, substitute `bin\mallet` for `bin/mallet`.

Development release: To download the most current version of MALLET 2.0, use our public GitHub repository:

```
git clone https://github.com/mimno/Mallet.git
```

from the command prompt to get the Mallet package.

To build a Mallet 2.0 development release, you must have the Apache ant build tool installed. From the command prompt, first change to the mallet directory, and then type

```
ant
```

If ant finishes with "BUILD SUCCESSFUL", Mallet is now ready to use.

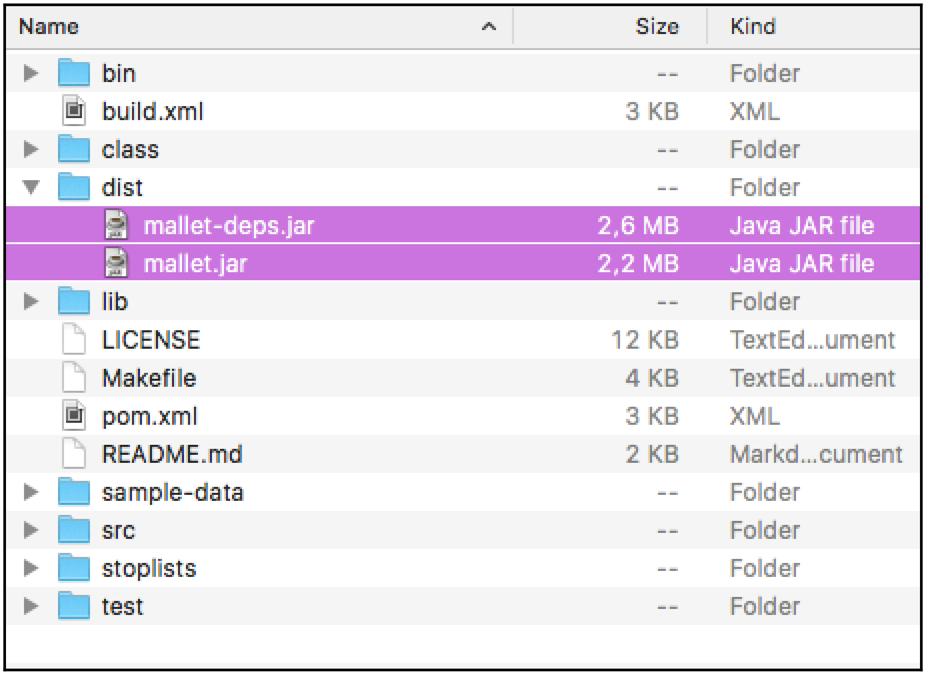
If you would like to deploy Mallet as part of a larger application, it is helpful to create a single ".jar" file that contains all of the compiled code. Once you have compiled the individual Mallet class files, use the command:

```
ant jar
```

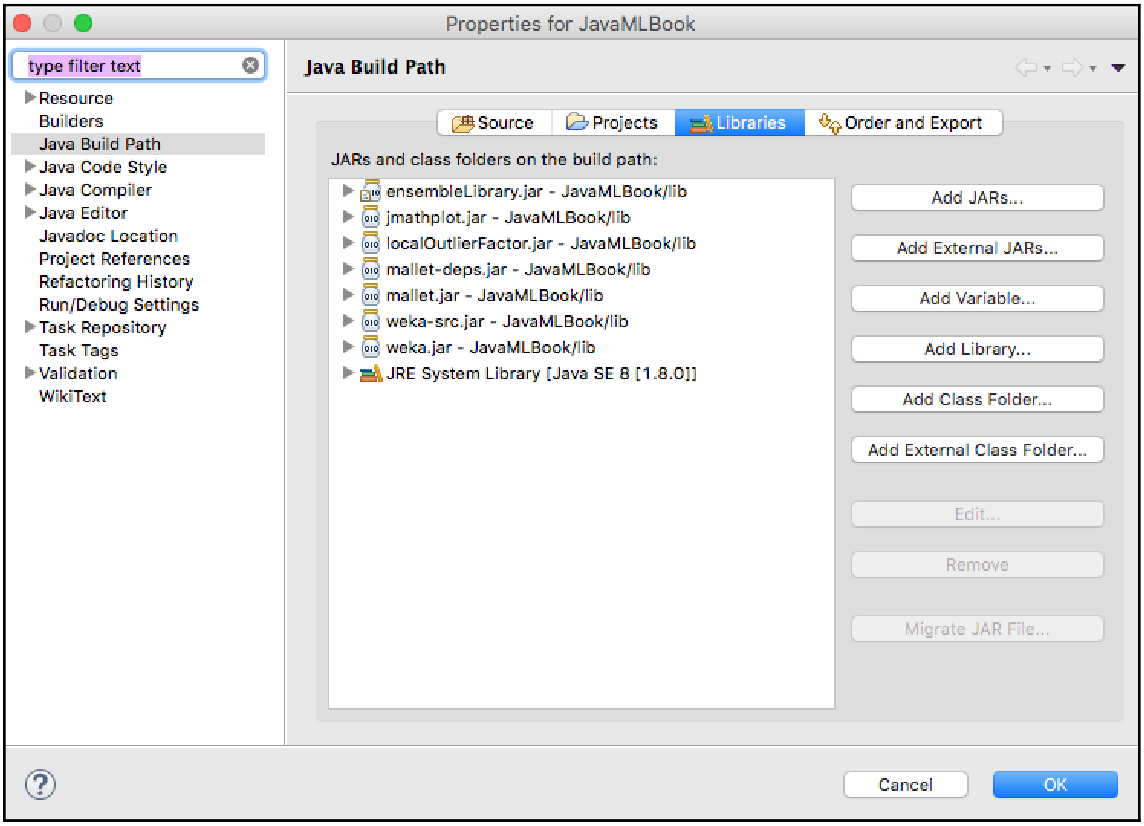
This process will create a file "mallet.jar" in the "dist" directory within Mallet.

Older releases: MALLET version 0.4 is available for download, but is not being actively maintained. This release includes classes in the package "edu.umass.cs.mallet.base", while MALLET 2.0 contains classes in the package "cc.mallet".

- mallet-2.0.6.tar.gz
- mallet-2.0.5.tar.gz (notes)
- mallet-2.0-RC4.tar.gz (notes)
- mallet-2.0-RC3.tar.gz (notes)
- mallet-2.0-RC2.tar.gz
- mallet-2.0-RC1.tar.gz
- mallet-0.4.tar.gz

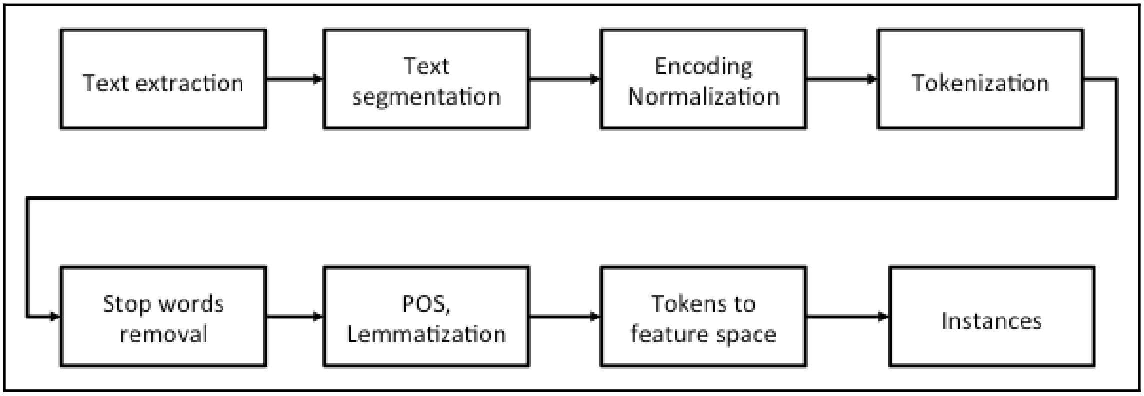


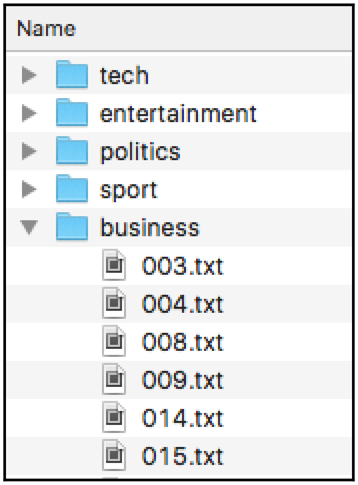
Name	Size	Kind
bin	--	Folder
build.xml	3 KB	XML
class	--	Folder
dist	--	Folder
mallet-deps.jar	2,6 MB	Java JAR file
mallet.jar	2,2 MB	Java JAR file
lib	--	Folder
LICENSE	12 KB	TextEd...ument
Makefile	4 KB	TextEd...ument
pom.xml	3 KB	XML
README.md	2 KB	Markd...cument
sample-data	--	Folder
src	--	Folder
stoplists	--	Folder
test	--	Folder



JARs and class folders on the build path:

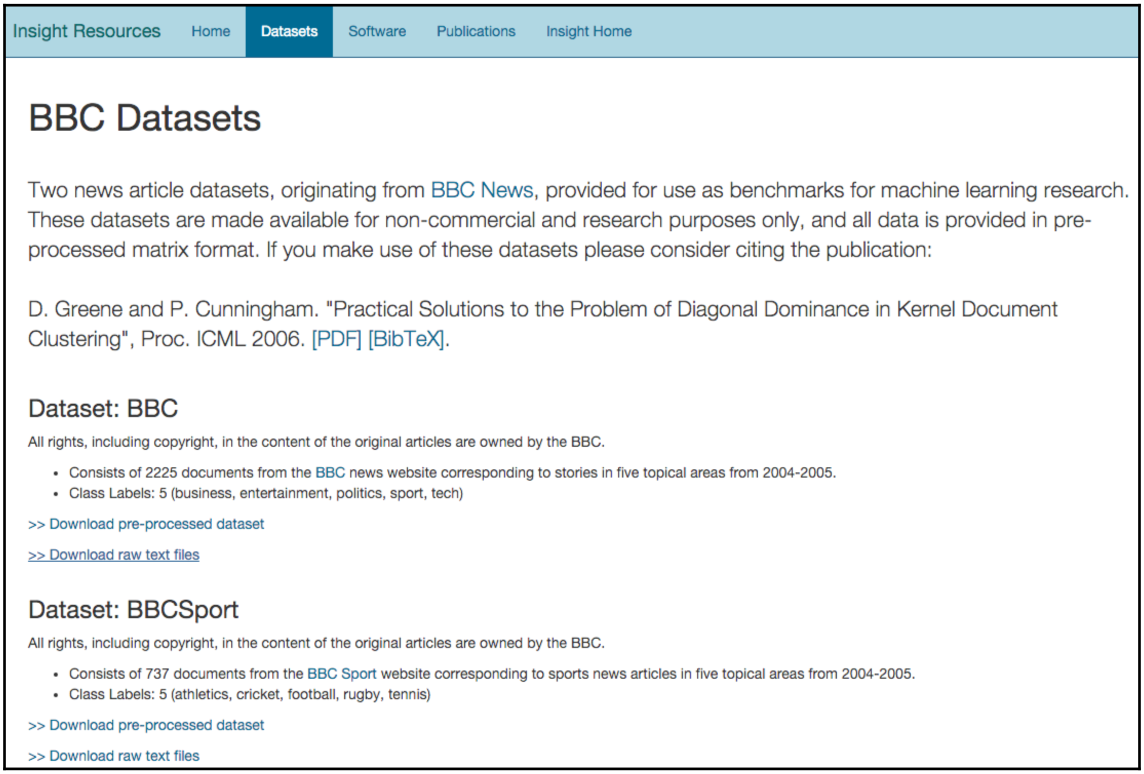
- ensembleLibrary.jar - JavaMLBook/lib
- jmathplot.jar - JavaMLBook/lib
- localOutlierFactor.jar - JavaMLBook/lib
- mallet-deps.jar - JavaMLBook/lib
- mallet.jar - JavaMLBook/lib
- weka-src.jar - JavaMLBook/lib
- weka.jar - JavaMLBook/lib
- JRE System Library [Java SE 8 [1.8.0]]





Name

- tech
- entertainment
- politics
- sport
- business
  - 003.txt
  - 004.txt
  - 008.txt
  - 009.txt
  - 014.txt
  - 015.txt



BBC Datasets

Two news article datasets, originating from BBC News, provided for use as benchmarks for machine learning research. These datasets are made available for non-commercial and research purposes only, and all data is provided in pre-processed matrix format. If you make use of these datasets please consider citing the publication:

D. Greene and P. Cunningham. "Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering", Proc. ICML 2006. [PDF] [BibTeX].

Dataset: BBC

All rights, including copyright, in the content of the original articles are owned by the BBC.

- Consists of 2225 documents from the BBC news website corresponding to stories in five topical areas from 2004-2005.
- Class Labels: 5 (business, entertainment, politics, sport, tech)

>> Download pre-processed dataset

>> Download raw text files

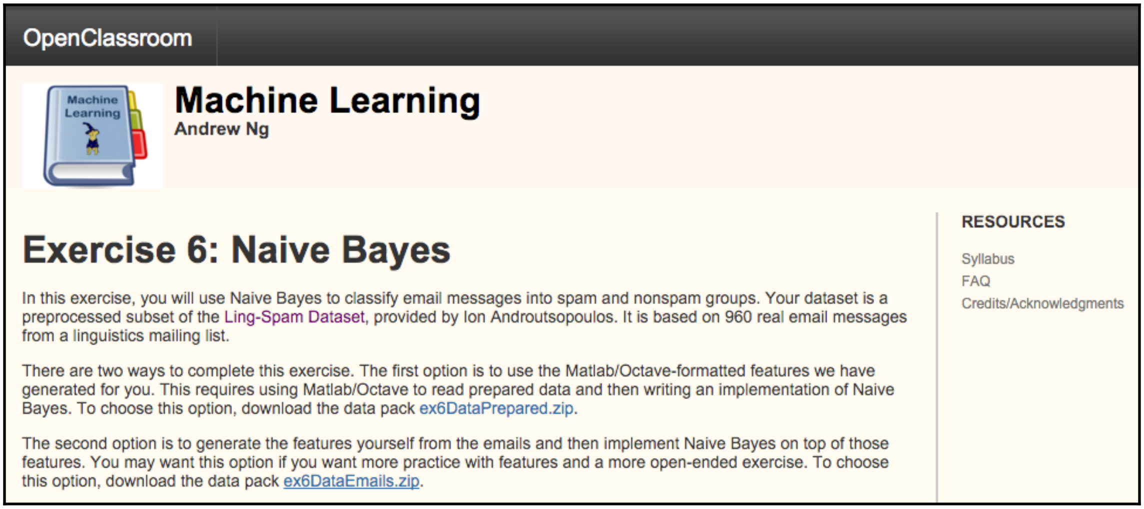
Dataset: BBCSport

All rights, including copyright, in the content of the original articles are owned by the BBC.

- Consists of 737 documents from the BBC Sport website corresponding to sports news articles in five topical areas from 2004-2005.
- Class Labels: 5 (athletics, cricket, football, rugby, tennis)

>> Download pre-processed dataset

>> Download raw text files



OpenClassroom

Machine Learning
Andrew Ng

Exercise 6: Naive Bayes

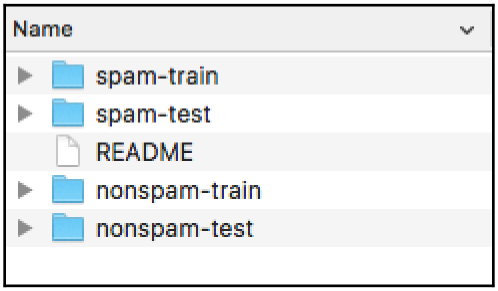
In this exercise, you will use Naive Bayes to classify email messages into spam and nonspam groups. Your dataset is a preprocessed subset of the Ling-Spam Dataset, provided by Ion Androutsopoulos. It is based on 960 real email messages from a linguistics mailing list.

There are two ways to complete this exercise. The first option is to use the Matlab/Octave-formatted features we have generated for you. This requires using Matlab/Octave to read prepared data and then writing an implementation of Naive Bayes. To choose this option, download the data pack ex6DataPrepared.zip.

The second option is to generate the features yourself from the emails and then implement Naive Bayes on top of those features. You may want this option if you want more practice with features and a more open-ended exercise. To choose this option, download the data pack ex6DataEmails.zip.

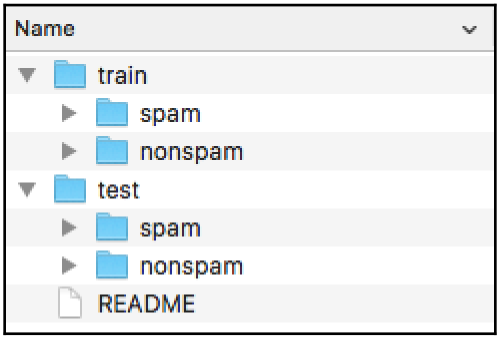
RESOURCES

Syllabus
FAQ
Credits/Acknowledgments



Name

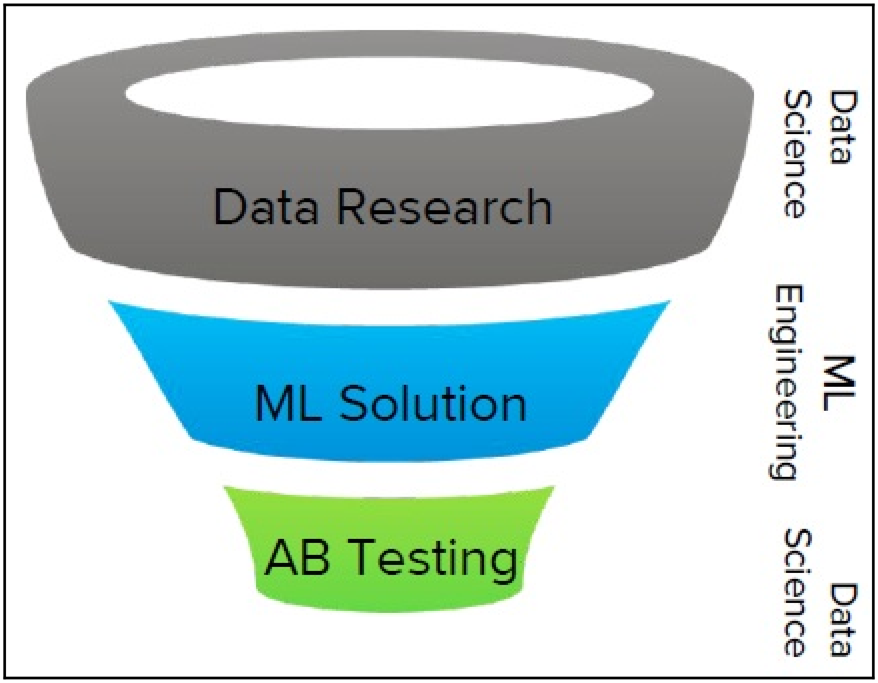
- spam-train
- spam-test
- README
- nonspam-train
- nonspam-test

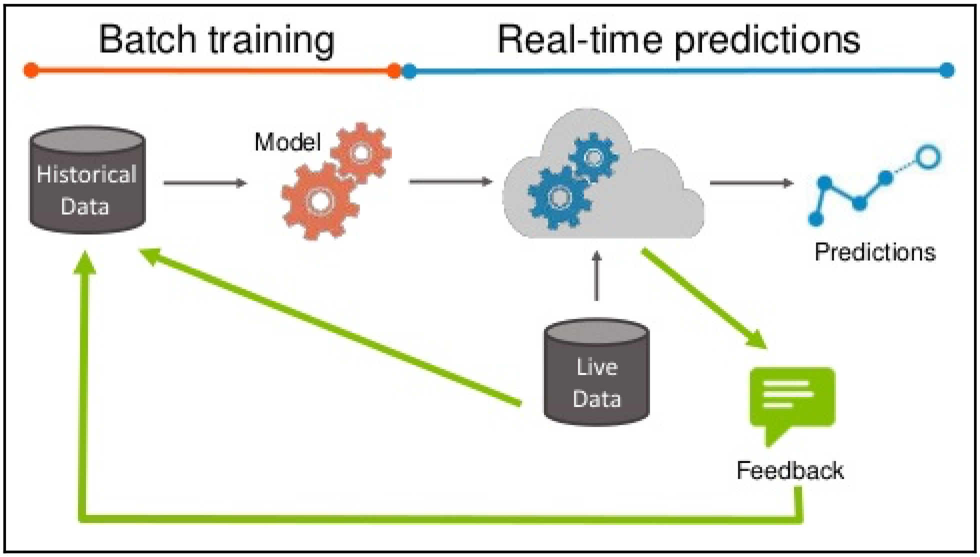


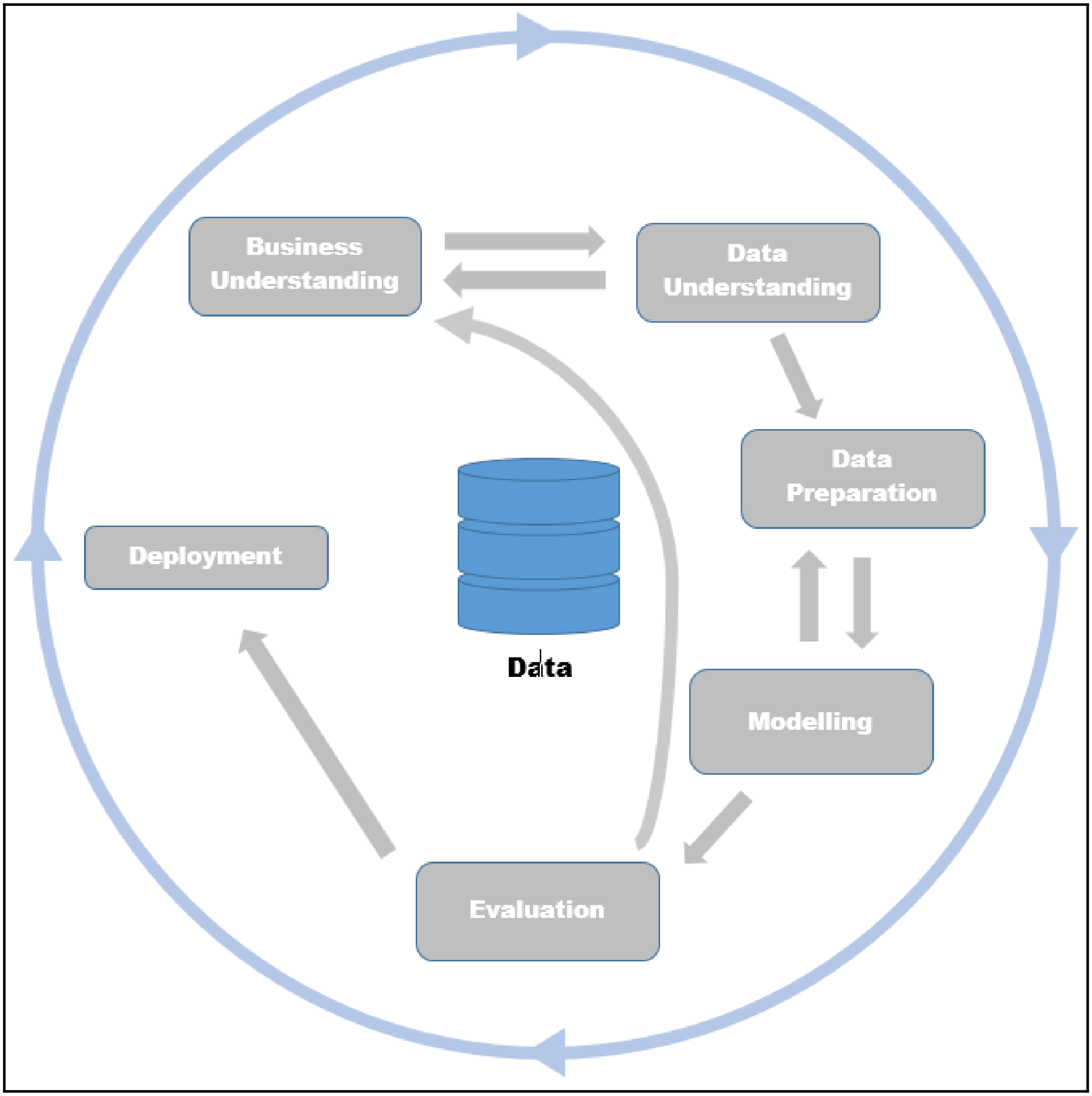
Name

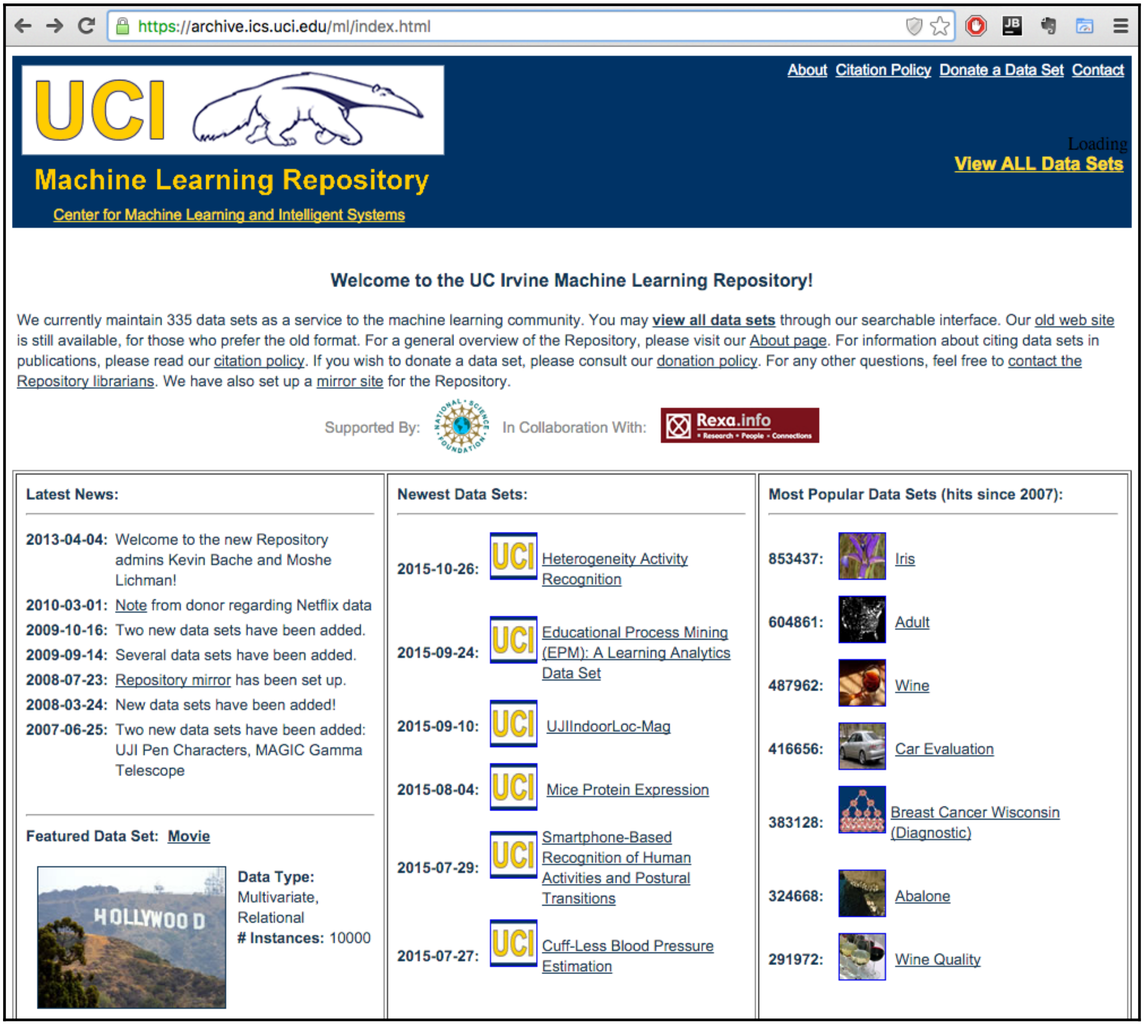
- train
  - spam
  - nonspam
- test
  - spam
  - nonspam
- README

Chapter 11: What Is Next?






Business Understanding
Data Understanding
Data Preparation
Deployment
Data
Modelling
Evaluation


https://archive.ics.uci.edu/ml/index.html
About
Citation Policy
Donate a Data Set
Contact
UCI
Machine Learning Repository
Center for Machine Learning and Intelligent Systems
Loading
View ALL Data Sets

Welcome to the UC Irvine Machine Learning Repository!

We currently maintain 335 data sets as a service to the machine learning community. You may **view all data sets** through our searchable interface. Our old web site is still available, for those who prefer the old format. For a general overview of the Repository, please visit our About page. For information about citing data sets in publications, please read our citation policy. If you wish to donate a data set, please consult our donation policy. For any other questions, feel free to contact the Repository librarians. We have also set up a mirror site for the Repository.

Supported By: In Collaboration With:

Latest News:

2013-04-04:	Welcome to the new Repository admins Kevin Bache and Moshe Lichman!
2010-03-01:	Note from donor regarding Netflix data
2009-10-16:	Two new data sets have been added.
2009-09-14:	Several data sets have been added.
2008-07-23:	Repository mirror has been set up.
2008-03-24:	New data sets have been added!
2007-06-25:	Two new data sets have been added: UJI Pen Characters, MAGIC Gamma Telescope

Featured Data Set: **Movie**

Data Type: Multivariate, Relational
# Instances: 10000

Newest Data Sets:

2015-10-26:	Heterogeneity Activity Recognition
2015-09-24:	Educational Process Mining (EPM): A Learning Analytics Data Set
2015-09-10:	UJIIndoorLoc-Mag
2015-08-04:	Mice Protein Expression
2015-07-29:	Smartphone-Based Recognition of Human Activities and Postural Transitions
2015-07-27:	Cuff-Less Blood Pressure Estimation

Most Popular Data Sets (hits since 2007):

853437:	Iris
604861:	Adult
487962:	Wine
416656:	Car Evaluation
383128:	Breast Cancer Wisconsin (Diagnostic)
324668:	Abalone
291972:	Wine Quality