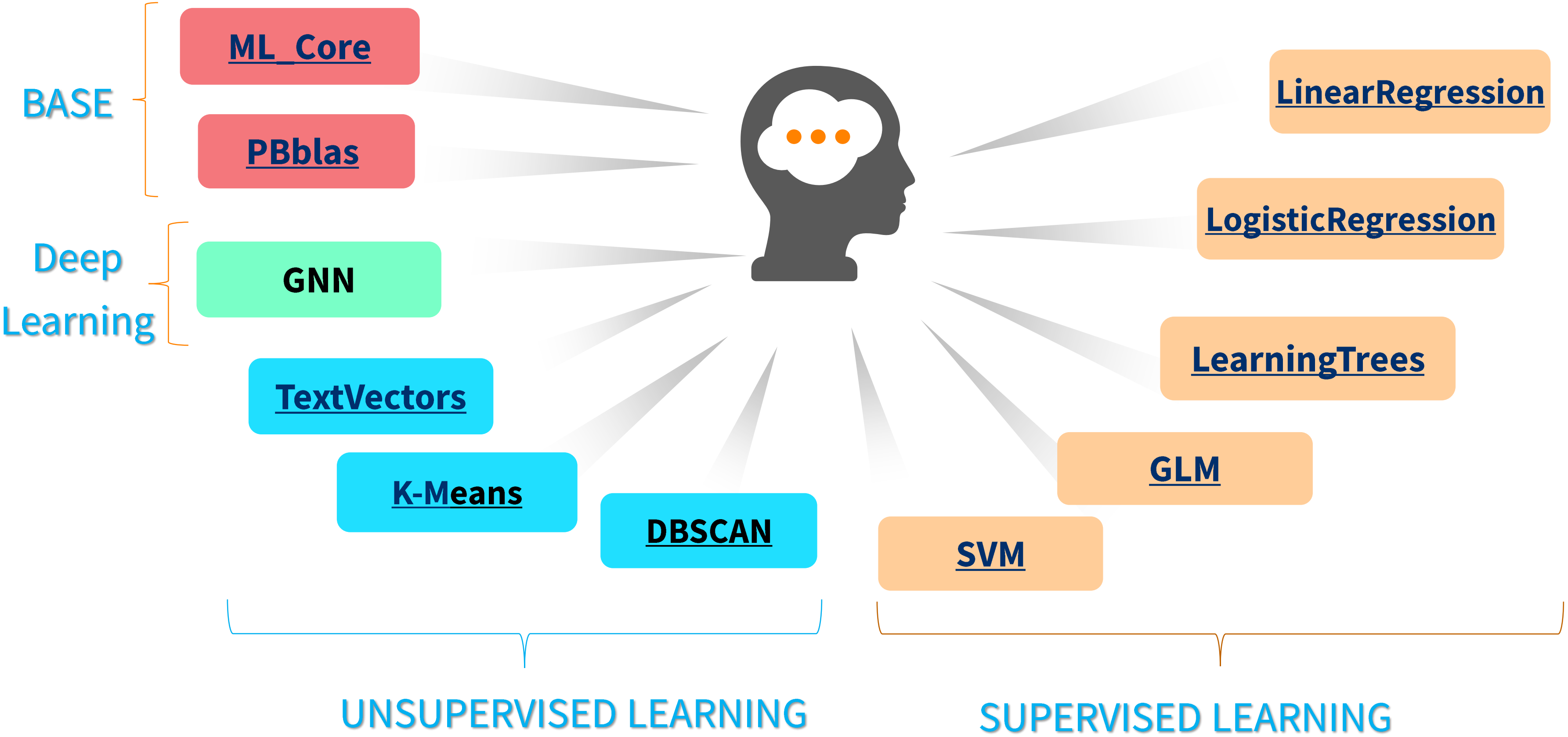
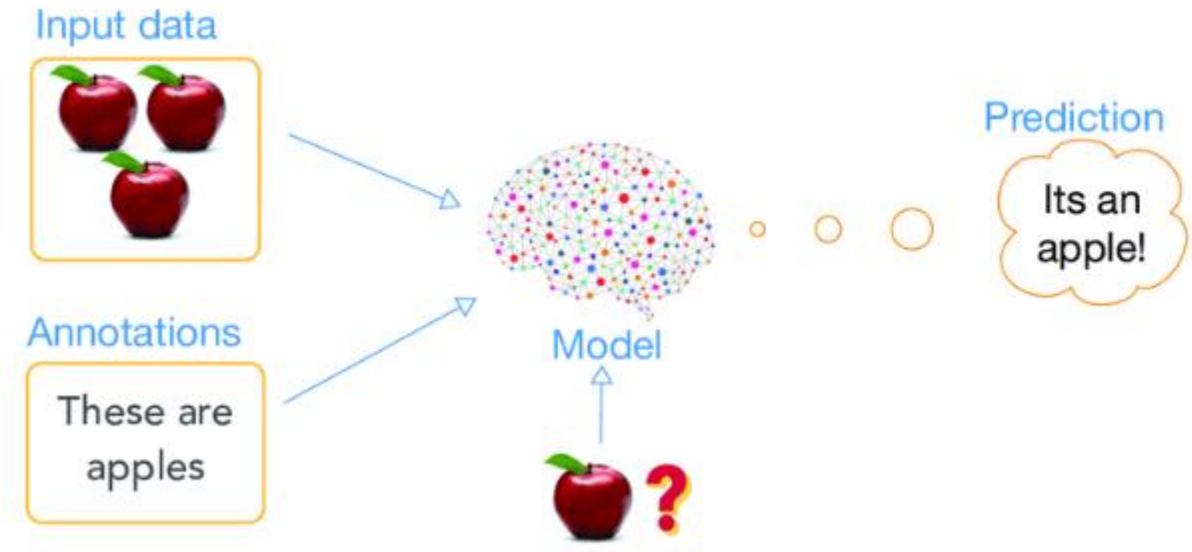
HPCC SYSTEMS MACHINE LEARNING

HPCC Systems Machine Learning Library



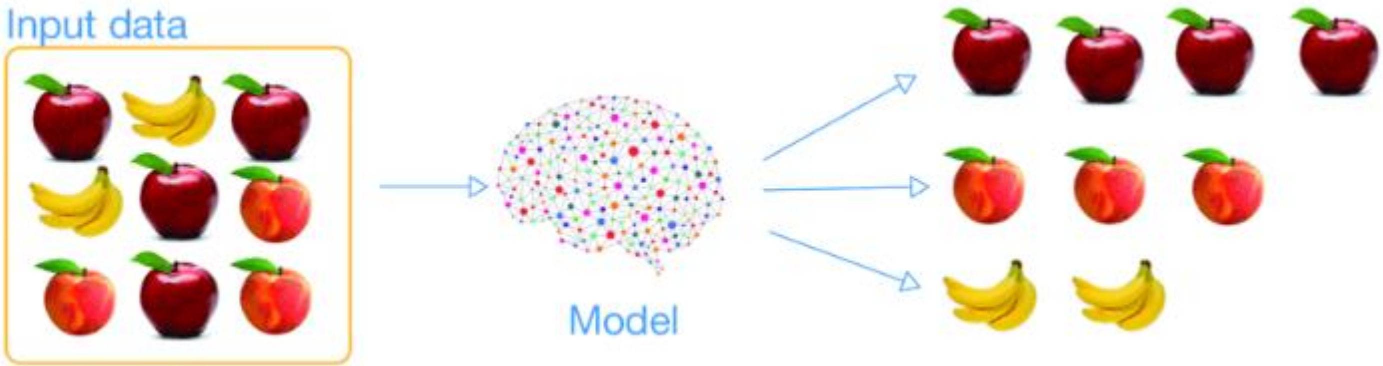
HPCC SYSTEMS®

Supervised Learning



Previous Knowledge

Unsupervised Learning



Machine Learning Data Structures

Dataset

a	a1	a2
b	b1	b2
c	c1	c2

Transform

ML DataFrame

a	a1
a	a2
b	b1
b	b2
c	c1
c	c2

*Transform record-based dataset to cell-based dataframes

HPCC SYSTEMS®

Machine Learning Data Structures

Numeric Dataset

a	a1	a2
b	b1	b2
c	c1	c2

Transform

ML_Core.ToField()

ML DataFrame

a	a1
a	a2
b	b1
b	b2
c	c1
c	c2

ML_Core.Types.NumericTypes

Machine Learning Data Structures

Discrete Dataset

a	a1	a2
b	b1	b2
c	c1	c2

ML_Core.ToField →

ML DataFrame

a	a1
a	a2
b	b1
b	b2
c	c1
c	c2

ML_Core.Types.NumericFields

PROJECT →

ML DataFrame

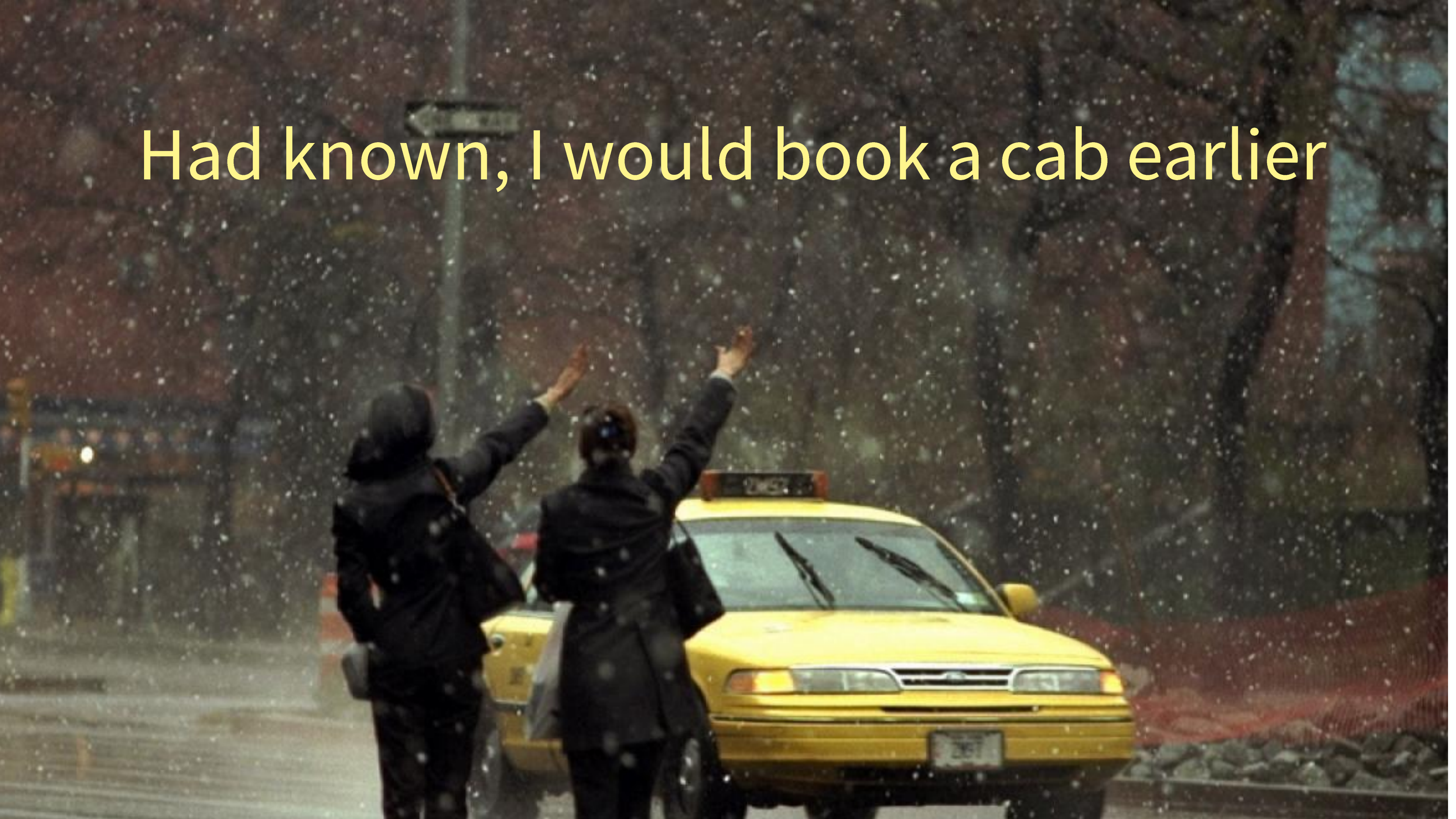
a	a1
a	a2
b	b1
b	b2
c	c1
c	c2

ML_Core.Types.DiscreteFields

ML_Core Bundle

- Prerequisite for all HPCC Systems production machine learning bundles
- Main attributes:
  - Definitions for common data types
    - ML_Core.Types
  - Data manipulation utilities
    - ML_Core.ToField
    - ML_Core.Discretize
  - Data examination via Statistical Tools
    - ML_Core. FieldAggregates
- ML_Core Bundle: https://github.com/hpcc-systems/ML_Core

New York City Taxi Trips


Had known, I would book a cab earlier

Prediction

NYC Taxi Data

48 GB
241M RECORDS
JAN 2015 – JUN 2016
16 MONTH
W/ WEATHER INFO

HPCC SYSTEMS®

HPCC Systems Machine Learning

Step 1: Setup the model

- Model := LogisticRegression(100 , 0.001);

Step 2: Train the model

- Training := Model.Fit(Training Data)

Step 3: Test the model

- Testing := Model.Predict(Testing Data)

Feature Engineer

wi	Classifier	Class	Precision	Recall	FPR
1	1	1	0.428571429	0.486486486	0.533333333
1	1	0	0.525	0.466666667	0.513513514

HPCC SYSTEMS®

Take Action Early!

wi	Classifier	Class	Precision	Recall	FPR
1	1	0	0.333	0.031	0.040
1	1	1	0.608	0.960	0.969

HPCC SYSTEMS®

```
//Reading Taxi_Weather Data
EXPORT A_Data_Ingestion := MODULE

EXPORT Layout := RECORD
  STD.Date.Date_t date;
  REAL8 precipintensity;
  INTEGER trip_counts;
END;

EXPORT raw := DATASET('~thor::taxi::traindata', Layout, THOR);

END;
```

Read Data

Sample Dataset

date	precipintensity	trip_counts
20150106	0.001289982	374040
20150118	0.057181148	416962
20150126	0.008881908	224097
20150130	0.001424809	471812
20150202	0.015235616	329387

LexisNexis® RISK SOLUTIONS

```
//Reading Taxi_Weather Data
raw := A_Data_Ingestion.raw;

//Data Profiling
Taxi_Weather_profile:= DataPatterns.Profile(raw);
OUTPUT(Taxi_Weather_profile);
```

Profile

attribute	rec_count	given_attribute_type	fill_rate	fill_count	best_attribute_type	popular_patterns		
						data_pattern	rec_count	example
date	217	unsigned4	100	217	unsigned4	99999999	217	20150109
precipintensity	217	real8	100	217	real8	9.999999999999999999	88	0.000125459220669412
						9.99999999999999999	58	0.01052740004885596
						9.9999999999999999999	45	0.000143063173374906l
						9.999999999999999a-99	16	7.661709965969239e-08
						9.9999999999999999	7	0.0177025134215619
						9.99999999999999a-99	3	3.01063264221159e-05
trip_counts	217	integer8	100	217	unsigned3	999999	216	132693

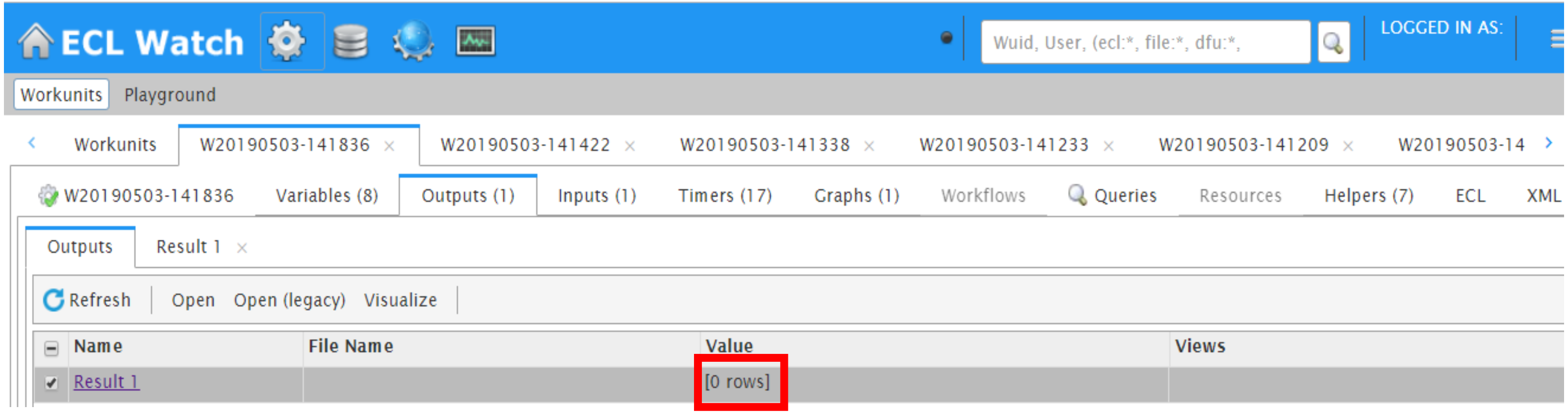
Profile Result Example

HPCC SYSTEMS®

```
//Reading Taxi_Weather Data
raw := A_Data_Ingestion.raw;

//Data Validation
validSet := raw( date < 20000101 AND date > 20190501 );
OUTPUT(validSet);
```

Validation



Validation Result Example

Enhance

```
EXPORT D_Data_Enhancement := MODULE

//Reading Taxi_Weather Data
SHARED raw := A_Data_Ingestion.raw;

//Enhance raw data
EXPORT enhancedLayout := RECORD
  INTEGER id;
  INTEGER month_of_year;
  INTEGER day_of_week;
  REAL8   precipintensity;
  INTEGER trip_counts;
END;

EXPORT enhancedData := PROJECT(raw, TRANSFORM(enhancedLayout,
                                    SELF.id := COUNTER,
                                    SELF.day_of_week := (INTEGER) Std.Date.DayOfWeek(LEFT.date),
                                    SELF.month_of_year := (INTEGER) LEFT.date[5..6],
                                    SELF.precipintensity := LEFT.precipintensity,
                                    SELF.trip_counts := LEFT.trip_counts));
END;
```

id	month_of_year	day_of_week	precipintensity	trip_counts
1	1	3	0.001289982354828361	374040
2	1	1	0.05718114840201266	416962
3	1	2	0.008881908280789124	224097
4	1	6	0.001424809034106805	471812
5	2	2	0.01523561646330912	329387
6	2	3	0.0001006261990082166	414405

Enhancement Result Example

Linear Regression

```
//Reading enhanced data
enhancedData := D_Data_Enhancement.enhancedData;

//Transform to Machine Learning Dataframe, such as NumericField
ML_Core.ToField(enhancedData, train);

// split into input (X) and output (Y) variables
X := train(number < 4);
Y := train(number = 4);

//Training LinearRegression Model
lr := LROLS.OLS(X, Y);

//Prediction
predict := lr.predict(X);
OUTPUT(predict);
```

wi	id	number	value
1	1	1	1
1	1	2	3
1	1	3	0.001289982354828361
1	1	4	374040
1	2	1	1
1	2	2	1
1	2	3	0.05718114840201266
1	2	4	416962
1	3	1	1
1	3	2	2
1	3	3	0.008881908280789124
1	3	4	224097

ML Dataframe: NumericField

wi	id	number	value
1	1	4	383492.0584366489
1	2	4	358001.6615743856

Linear Regression Result Example

HPCC SYSTEMS®

Logistic Regression

```
//Reading enhanced data
enhancedData := D_Data_Enhancement.enhancedData;
//Average trips per day
avgTrip := AVE(enhancedData, trip_counts);
//Add trend layout
trainLayout := RECORD
  INTEGER id;
  INTEGER month_of_year;
  INTEGER day_of_week;
  REAL8   precipintensity;
  INTEGER trend;
END;
trainData := PROJECT(enhancedData, TRANSFORM(trainLayout,
                      SELF.trend := IF(LEFT.trip_counts < avgTrip, 0, 1),
                      SELF := LEFT));
//Transform to Machine Learning Dataframe, such as NumericField
ML_Core.ToField(trainData, NFtrain);
//Independent and Dependent data
DStrainInd := NFtrain(number < 4);
DStrainDpt := PROJECT(NFtrain(number = 4), TRANSFORM(Types.DiscreteField, SELF.number := 1, SELF := LEFT));
//Training LogisticRegression Model
mod_bi := LR.BinomialLogisticRegression(100,0.00001).getModel(DStrainInd, DStrainDpt);
//Prediction
predict_bi := LR.BinomialLogisticRegression().Classify(mod_bi, DStrainInd);
OUTPUT(predict_bi);
```

wi	id	number	value
1	1	1	1
1	1	2	3
1	1	3	0.001289982354828361
1	1	4	0
1	2	1	1
1	2	2	1
1	2	3	0.05718114840201266
1	2	4	1
1	3	1	1
1	3	2	2
1	3	3	0.00881908280789124
1	3	4	0

ML Dataframe: DiscreteField

wi	id	number	value	conf
1	1	1	1	0.295929167271116
1	2	1	0	0.3592422652164307

Logistic Regression Result Example

Let’s Play With The Code

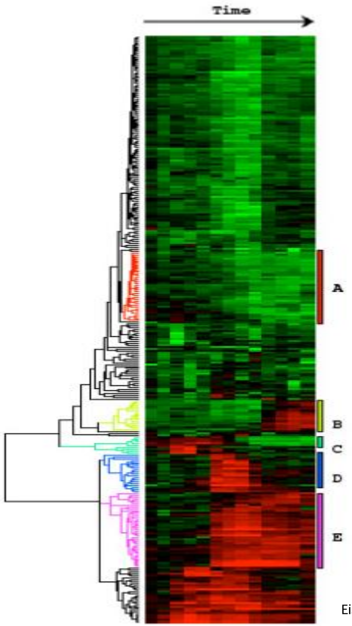
Clustering Methods in HPCC Systems : KMeans & DBSCAN

- Unsupervised Machine Learning (ML) algorithms
- Automatically find the clusters/groups of the data without previous knowledge
- Highly Scalable Parallelized for Big Data machine learning challenge

Applications

Claim\Customer segmentation

Image segmentation



Eisen et al, PNAS 1998

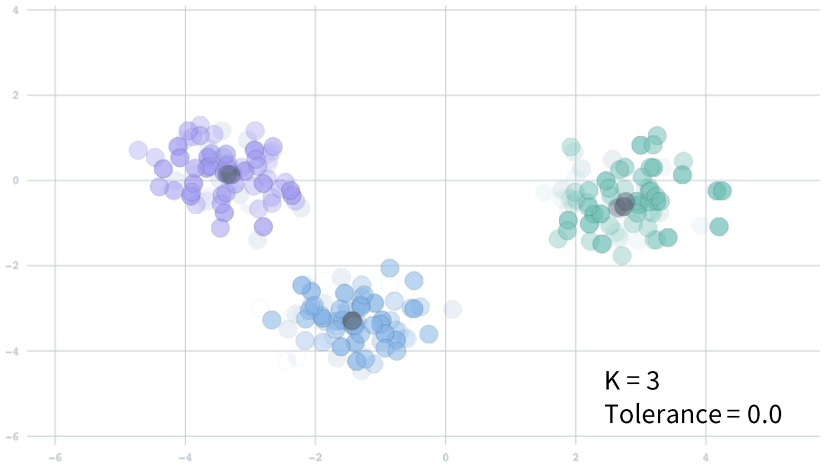
Clustering gene expressions

HPCC SYSTEMS®

KMeans vs. DBSCAN

➢ KMEANS

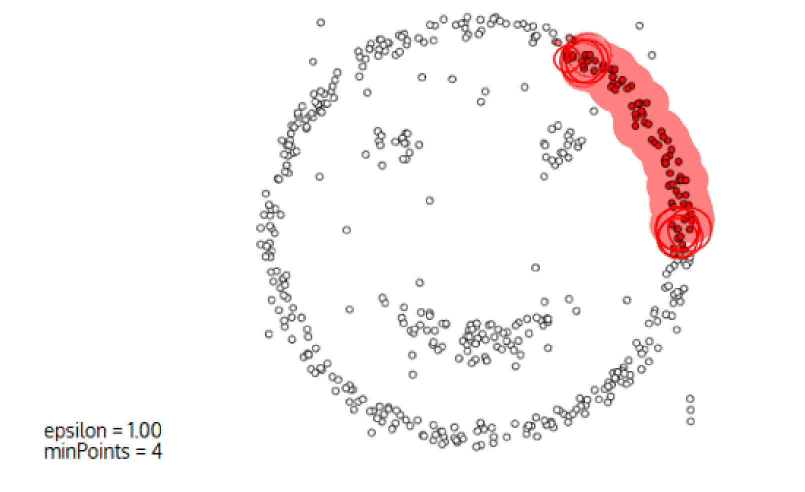
- Most popular clustering method [3]
- Highly Scalable Parallelized
- Parametric: K, Tolerance
- Sensitive to Initialization
- Spherical Clusters
- Sensitive to Outliers
- Curse of Dimensionality



KMeans vs. DBSCAN

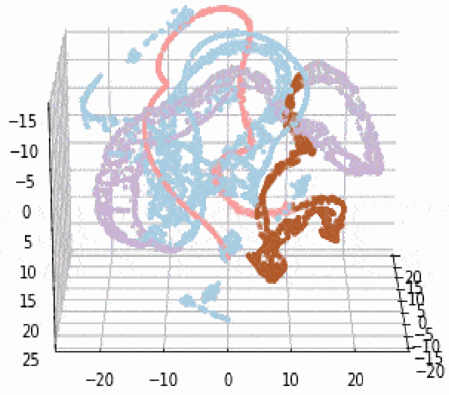
➢ DBSCAN

- Density-Based Clustering Metho
- Highly Scalable Parallelized
- Parametric: epsilon, minPoints
- Sensitive to Initialization
- Random Shapes Clusters
- Outliers Detection
- Sensitive to Density Variance
- Curse of Dimensionality



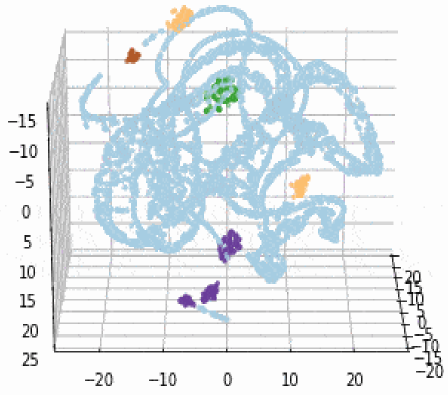
HPCC SYSTEMS®

KMeans vs. DBSCAN



KMeans

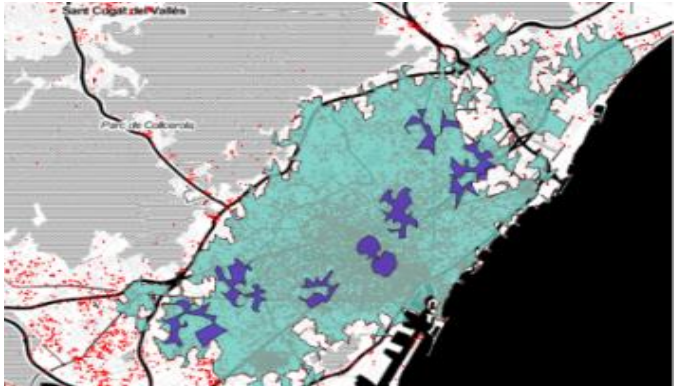
- Clusters Shape
- Cluster Size
- Model Parameters
- Number of Clusters (Fixed vs. Variable)
- Outlier Detection
- Curse of Dimensionality



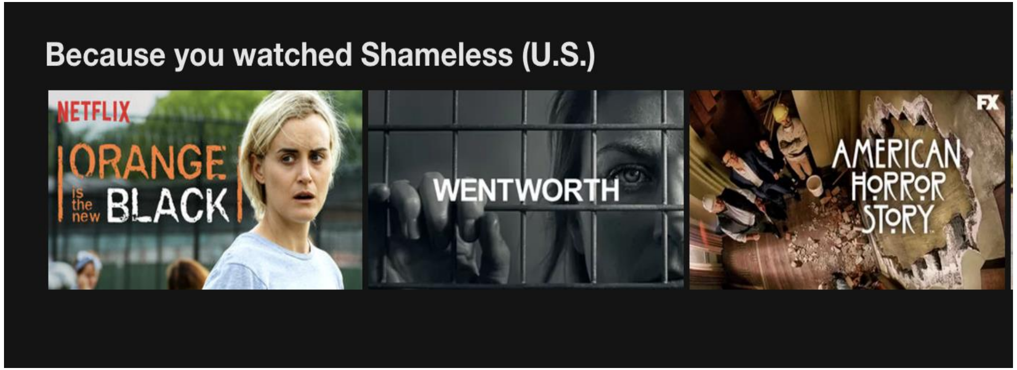
DBSCAN

HPCC SYSTEMS®

Application Domains



Clustering Demographic/Geospatial Data



Recommendation System

Easy to use

Step 1 Import K-Means bundle

```
IMPORT KMeans as KM;
```

Step 2 Train K-Means Model

```
Model := KM.KMeans(Max_iterations,Tolerance).Fit( Samples, InitialCentroids));
```

$\underbrace{\text{Max\_iterations,Tolerance}}_{\text{Optional}}$ $\underbrace{\text{Samples, InitialCentroids}}_{\text{Required}}$

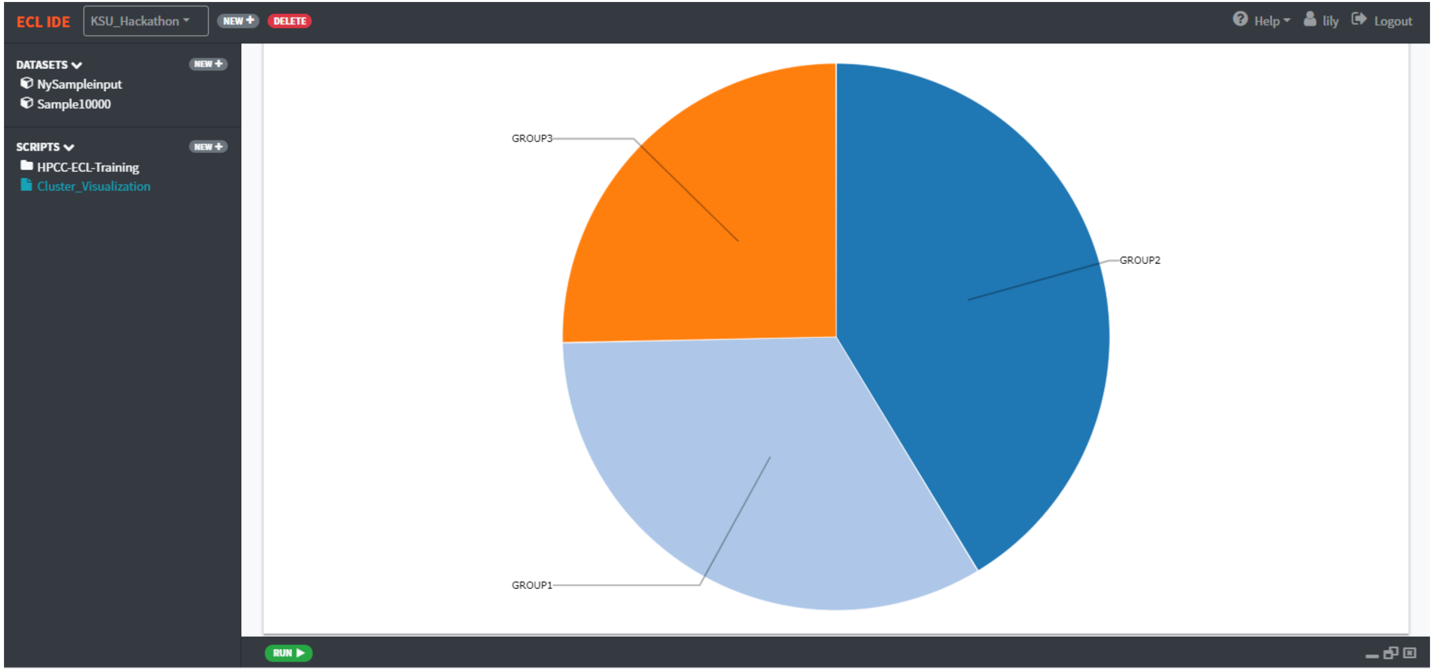
Step 3 Predict the cluster index of the new samples (Optional)

```
Labels := KM.KMeans().Predict(Model, NewSamples);
```

HPCC SYSTEMS®

Easy to use – Cont.

Step 4 Visualization (Optional)



ECL Cloud IDE: KMeans Visualization

For more details

- Tutorial:

  Automatically Cluster your Data with Massively Scalable K-Means

  Link: https://hpccsystems.com/blog/kmeans

- Research Publication:

  Massively Scalable Parallel KMeans on the HPCC Systems Platform

  Lili Xu, Amy Apon, Flavio Villanustre, Roger Dev, Arjuna Chala

Can you apply the Machine Learning models you just learnt to Flight Data?

Let’s Play With The Code

LexisNexis®
RISK SOLUTIONS

Q&A

Reference

Introduction of HPCC Systems Machine Learning :
https://hpccsystems.com/download/free-modules/machine-learning-library

Official Github: https://github.com/hpcc-systems

Forum: http://hpccsystems.com/bb/viewforum.php?f=23

Contact me:

Lili Xu
Software Engineer III
HPCC Systems
Lili.xu@lexisnexisrisk.com