ThoughtWorks®

@dhianadeva

MACHINE LEARNING FOR EVERYONE

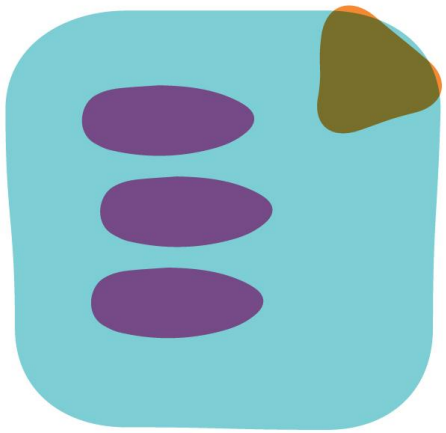
Demystifying machine learning!

AGENDA

Goal:

Encourage you to start a machine learning project. Today!

- About me
- About you
- Machine Learning
- Problems
- Design
- Algorithms
- Evaluation
- Code snippets
- Pay-as-you-go
- Competitions



ThoughtWorks®

ABOUT ME

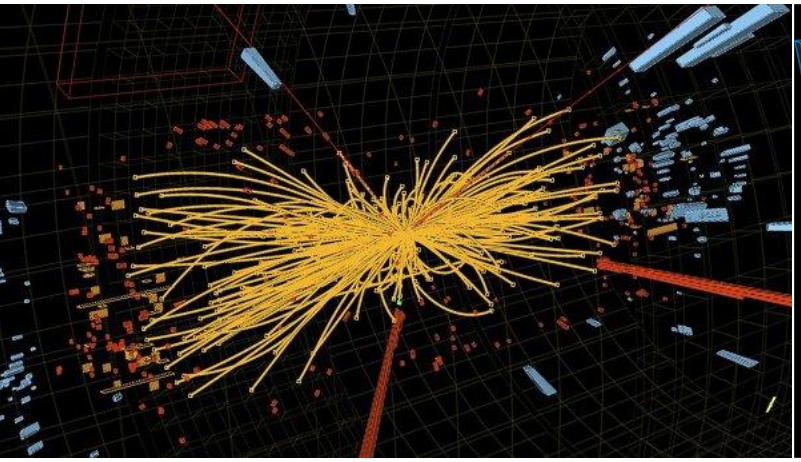
Electronics Engineering, Software Development and Data Science... Why not?

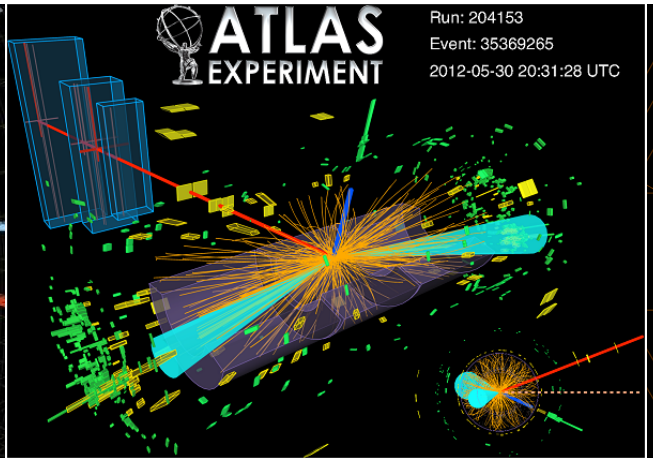
ThoughtWorks®

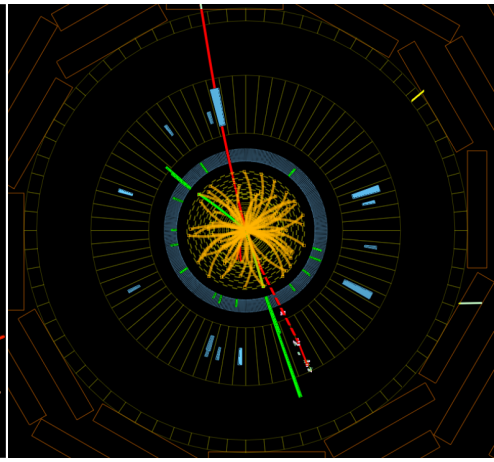
DHIANA DEVA

NEURALTB

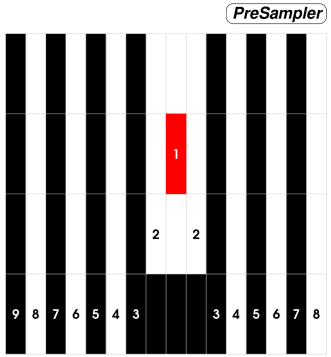
CERN

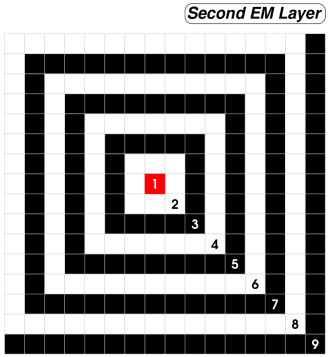


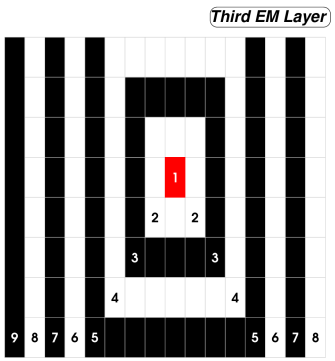

ATLAS
EXPERIMENT
Run: 204153
Event: 35369265
2012-05-30 20:31:28 UTC

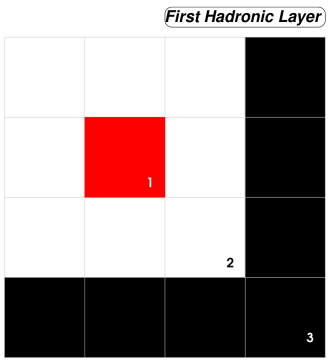


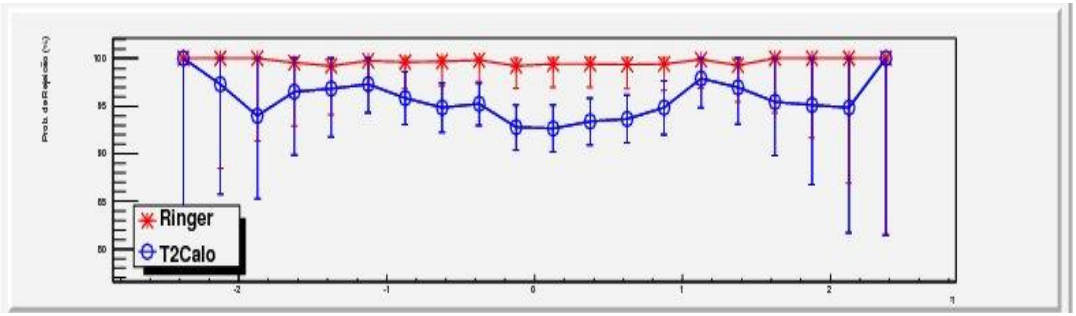
NEURALRINGER

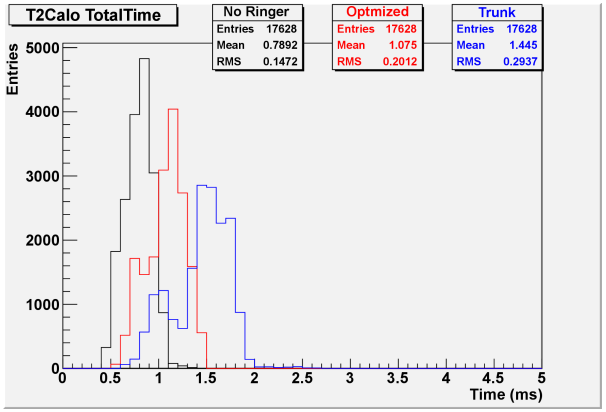






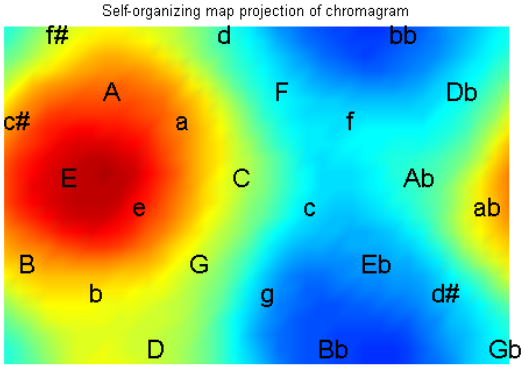


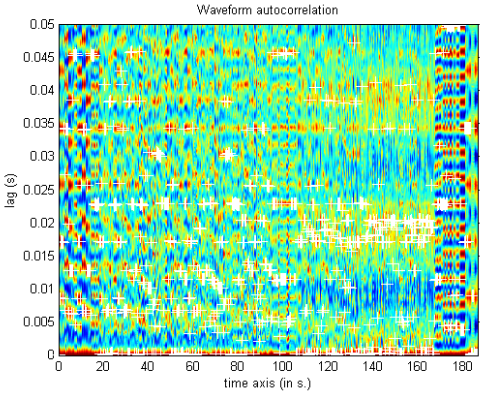


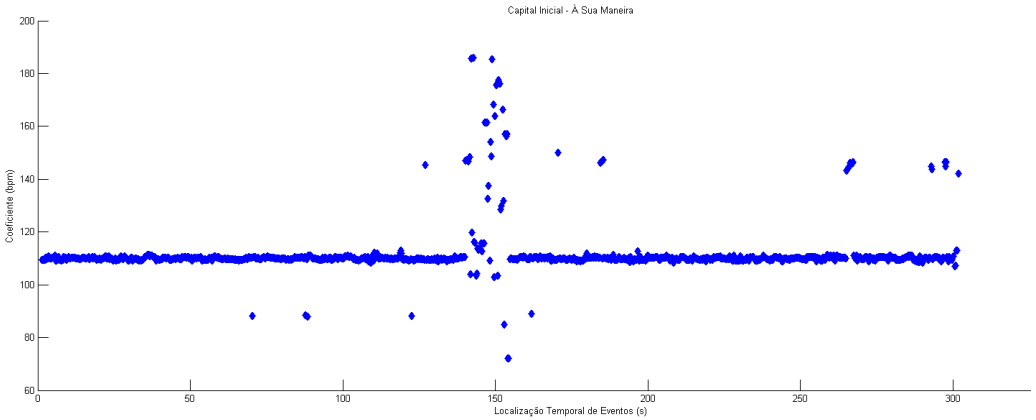


DJBRAZIL


djbrazil
a rádio do futuro


Self-organizing map projection of chromagram
f#
d
bb
A
F
Db
c#
a
f
E
C
Ab
e
c
ab
B
G
Eb
b
g
d#
D
Bb
Gb


Waveform autocorrelation
lag (s)
time axis (in s.)


Capital Inicial - À Sua Maneira
Coeficiente (bpm)
Localização Temporal de Eventos (s)

HIGGS CHALLENGE

Winner announcement

The Higgs Machine Learning Challenge has completed the 15th September 2014, gathering 1785 teams and 1942 participants.

After due verifications, we are pleased to announce the three winners, with the three best scores on the private leaderboard when disclosed.

1 : Gabor Melis : 7000 dollars
2 : Tim Salimans : 4000 dollars
3 : Pierre Courtiol (nhlx5haze) : 2000 dollars

All three have been invited (at the HiggsML organisation expense) to NIPS conference at Montreal, where a special workshop is organised the 13th December 2014, to discuss machine learning techniques applications to high energy physics, and specific developments made for this challenge.
https://nips.cc/Conferences/2014/Program/event.php?ID=4292

In addition, documented software was scrutinized, and the special HEP meets ML award is given to :

crowwork (Tianqi Chen and Tong Hel)

They have developed XGBoost **https://github.com/tqchen/xgboost** and made it available to other participants early, and it was indeed used by many of them ; while not giving the very best score, it appears to be an excellent compromise between performance and simplicity, which makes it a promising improvements to tools currently used by high energy physicists.

The team will be invited at CERN in 2015 for a workshop (being organised) where machine learning techniques application to high energy physics, in particular as they emerged in this challenge, will be discussed further.

The organizers would like to make special mention of CSE_TEAM_0 (Chamila Wijayarathna, Dimuthu Upeksha, Maduranga Siriwardena, Sachith Withana) for the detailed documentation of their optimisation, and Dhiana Deva an enthusiastic undergraduate.

ThoughtWorks®

ABOUT YOU

You can do it!

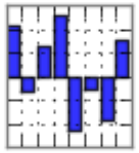
FOR ALL

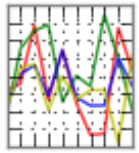
MASSIVE ONLINE OPEN COURSES

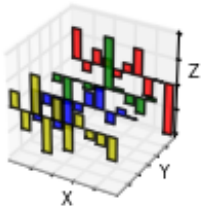
OPEN SOURCE TOOLS

pandas

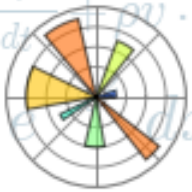
$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$





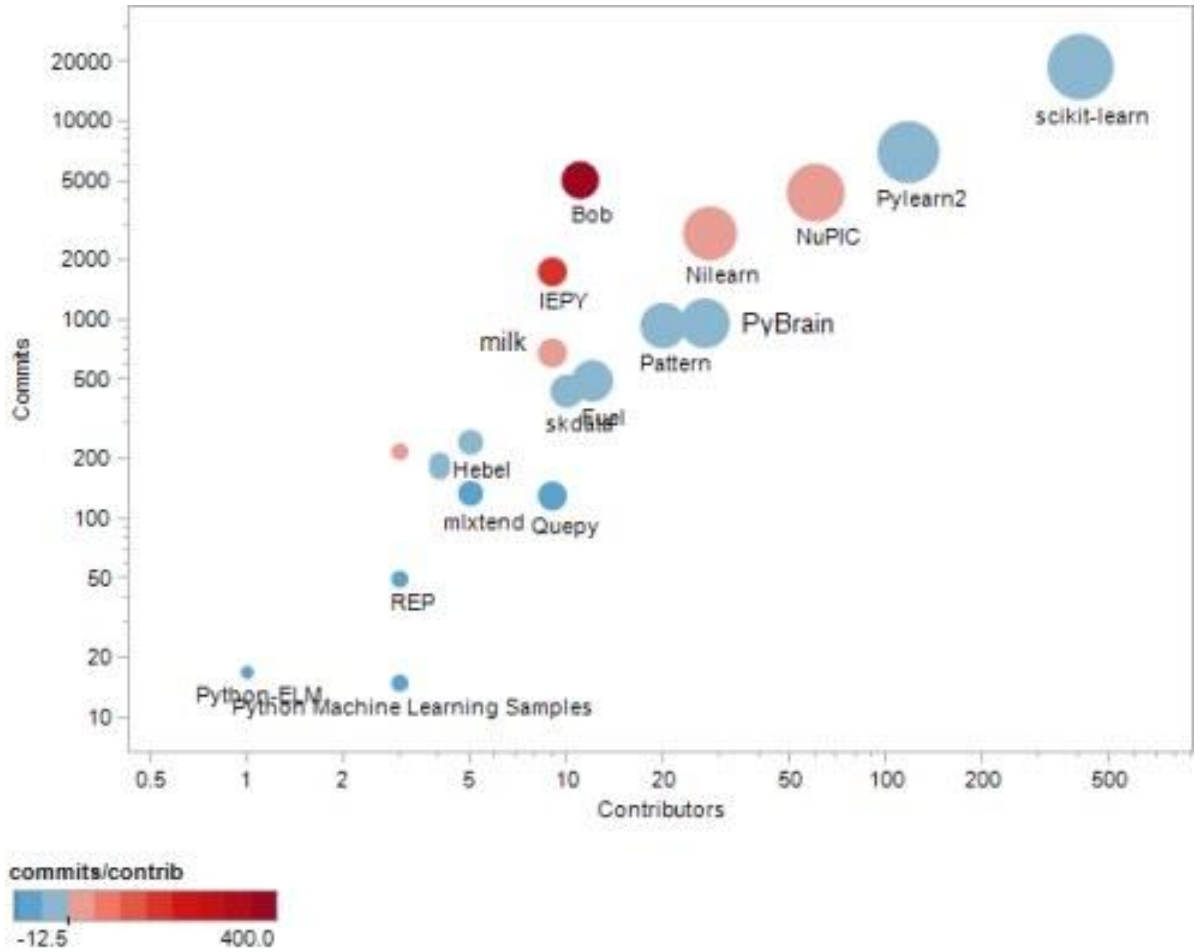


WEKA
The University of Waikato



matplotlib

OPEN SOURCE PYTHON TOOLS



PAY-AS-YOU-GO SERVICES

amazon web services™

Microsoft Azure

Google Cloud Platform

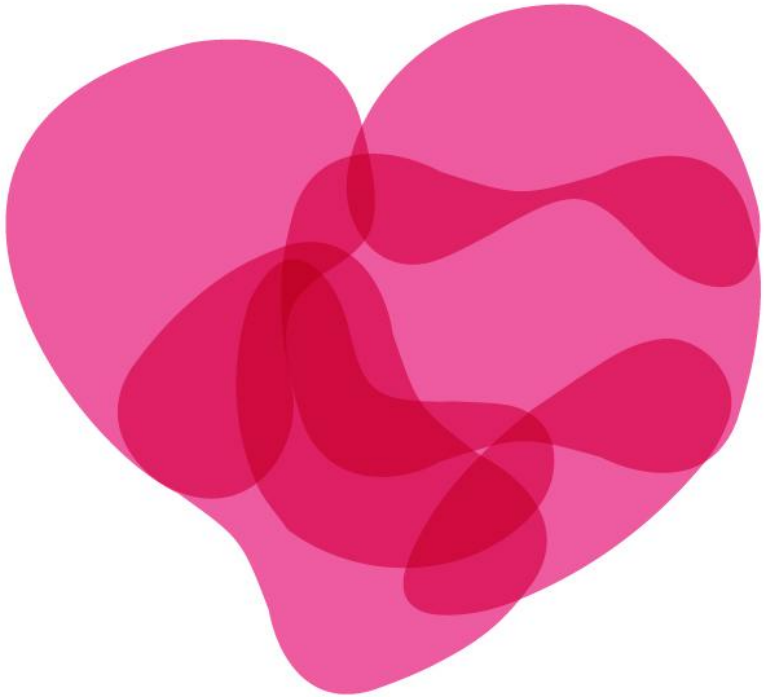
ThoughtWorks®

MACHINE LEARNING

Learning, machine learning!

EXPECTATIONS

REALITY



FEATURE EXTRACTION

Item

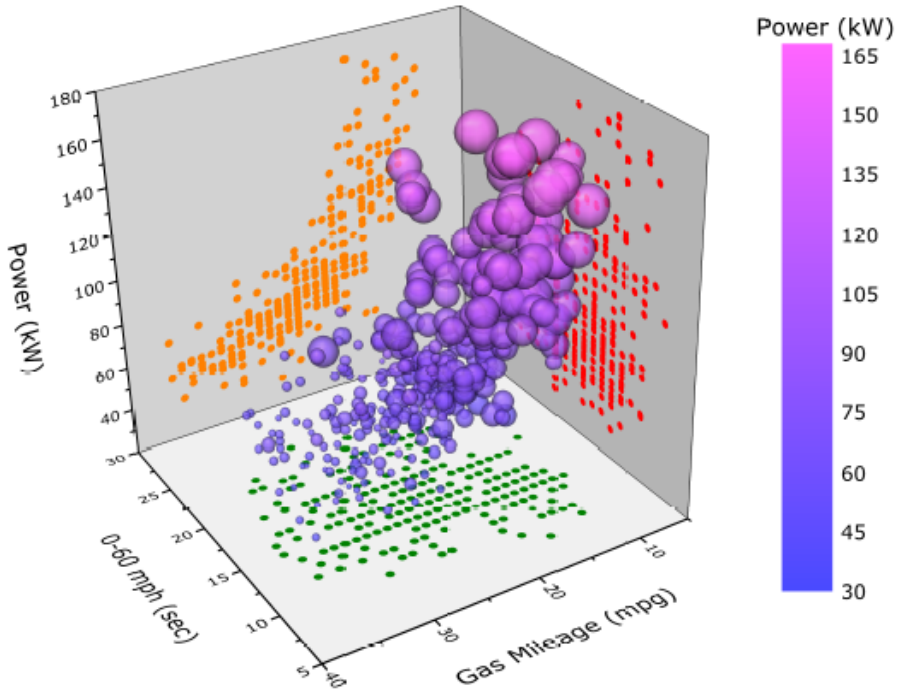
{

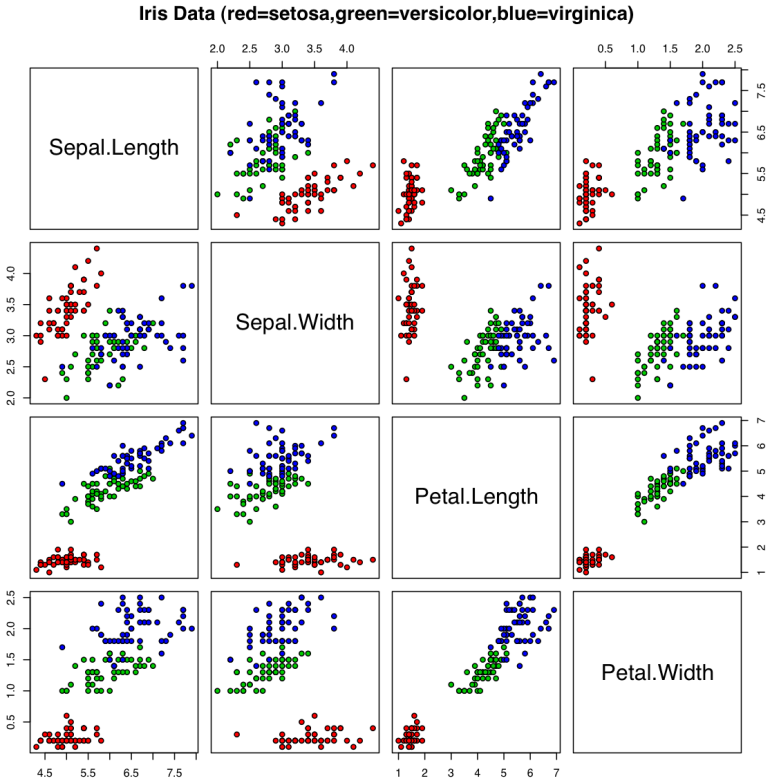
Feature 1
Feature 2

…

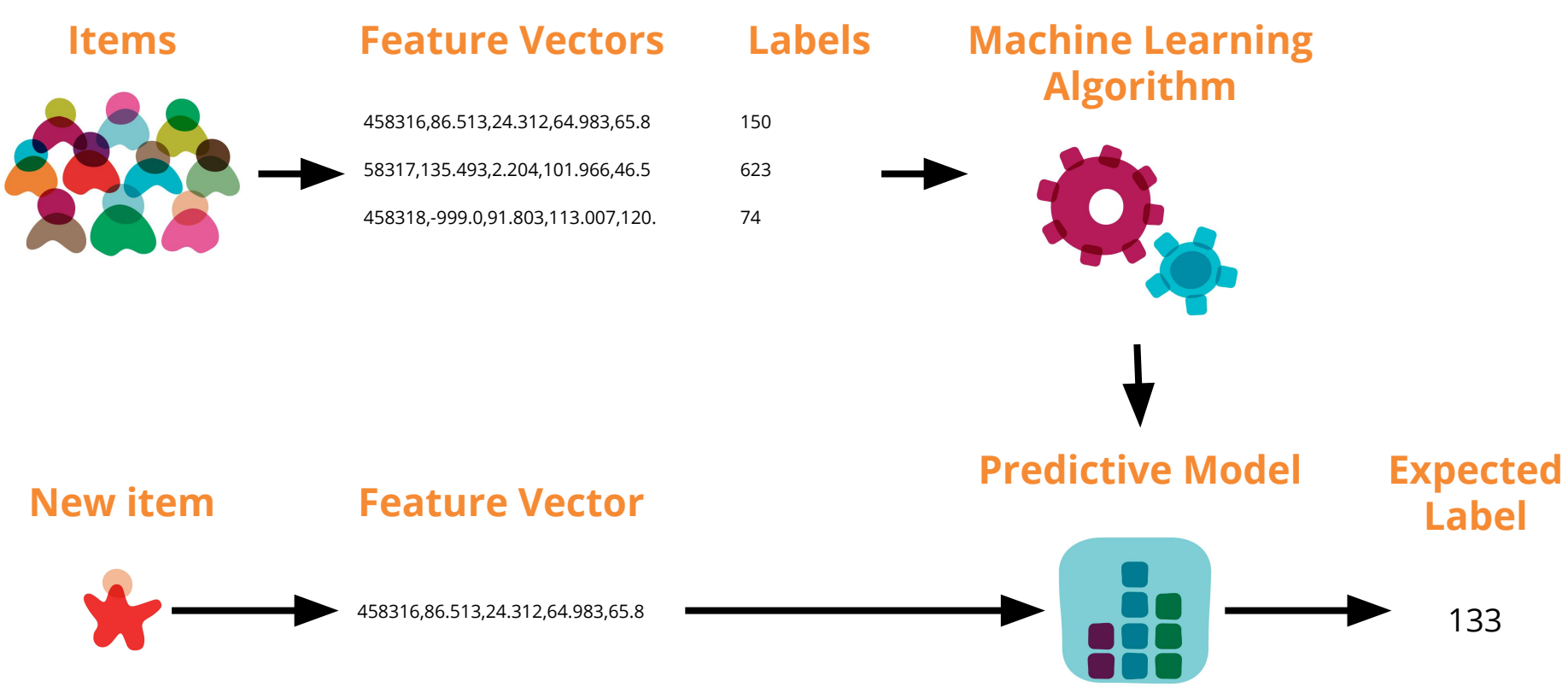
Feature N

FEATURE SPACE

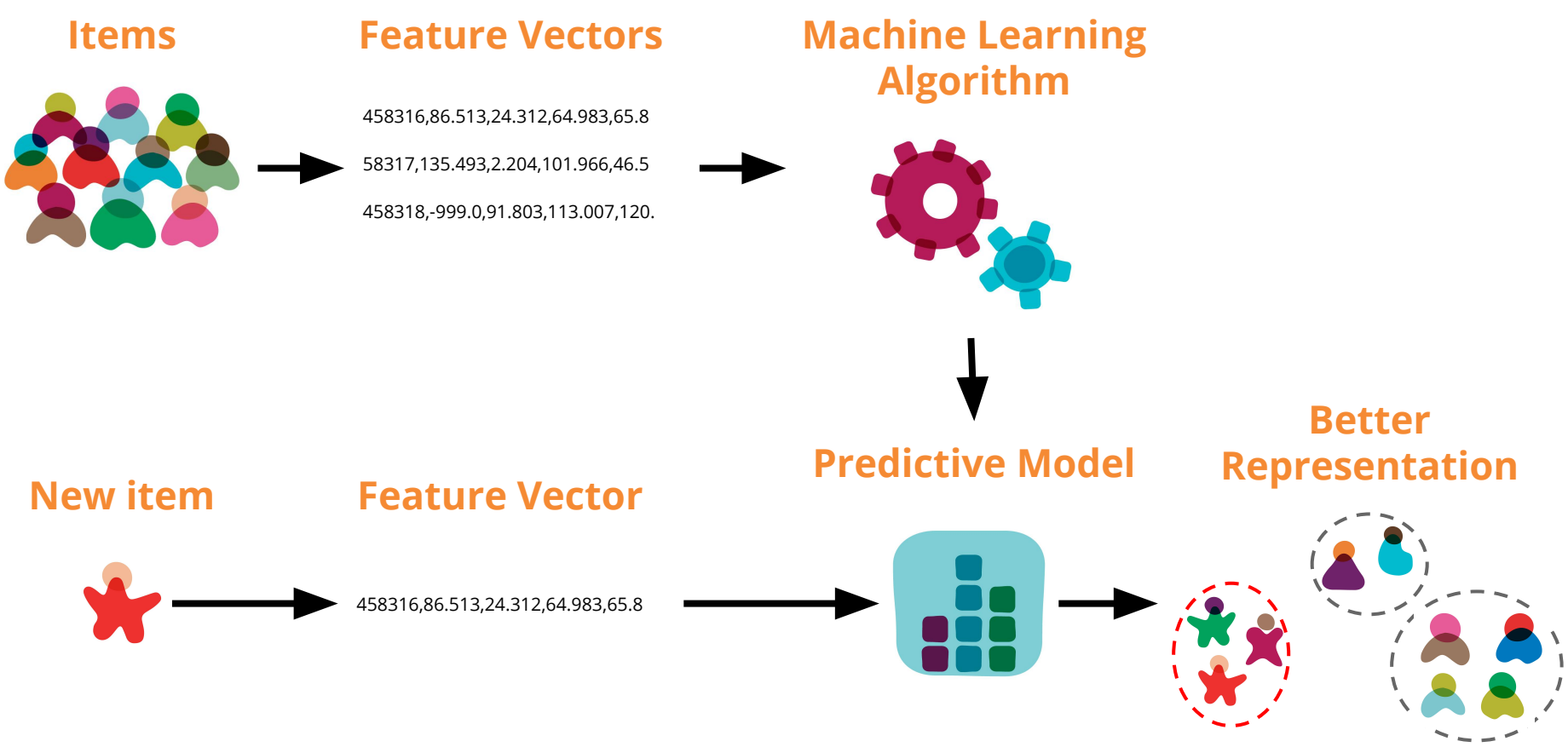




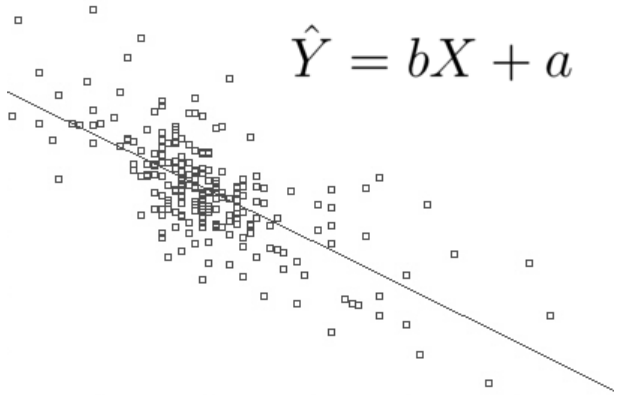
SUPERVISED LEARNING

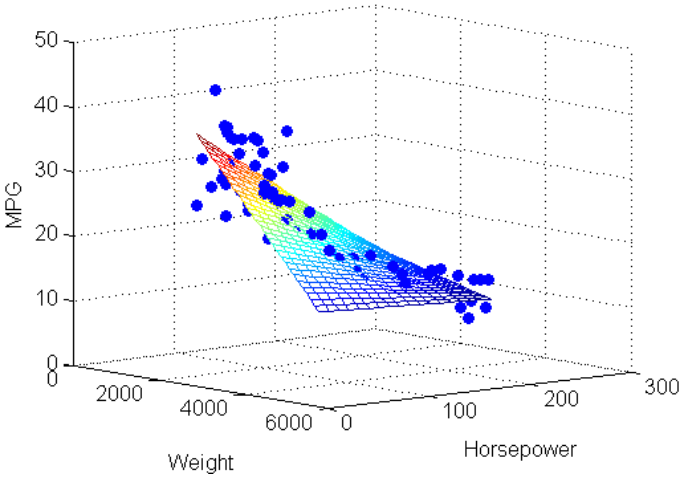


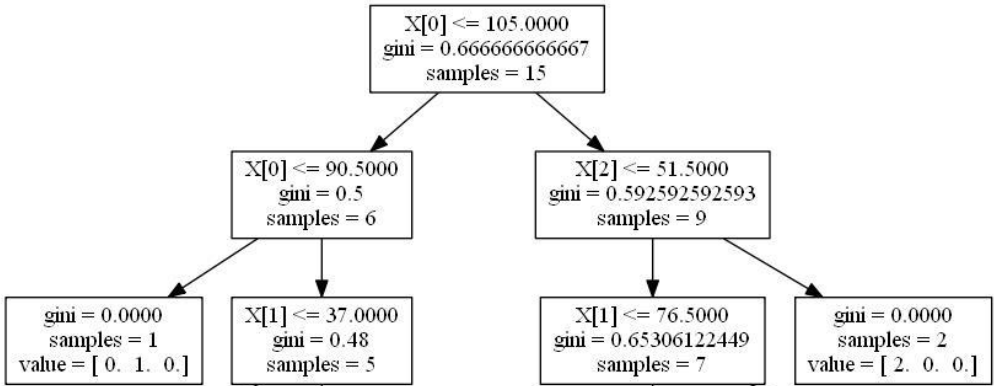
UNSUPERVISED LEARNING

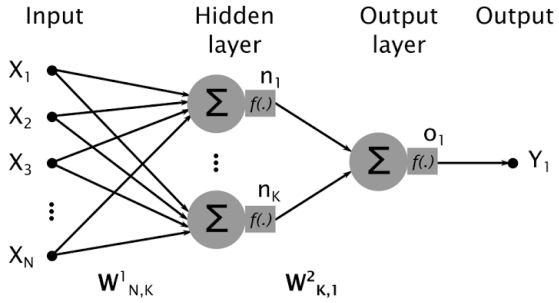


MODELS

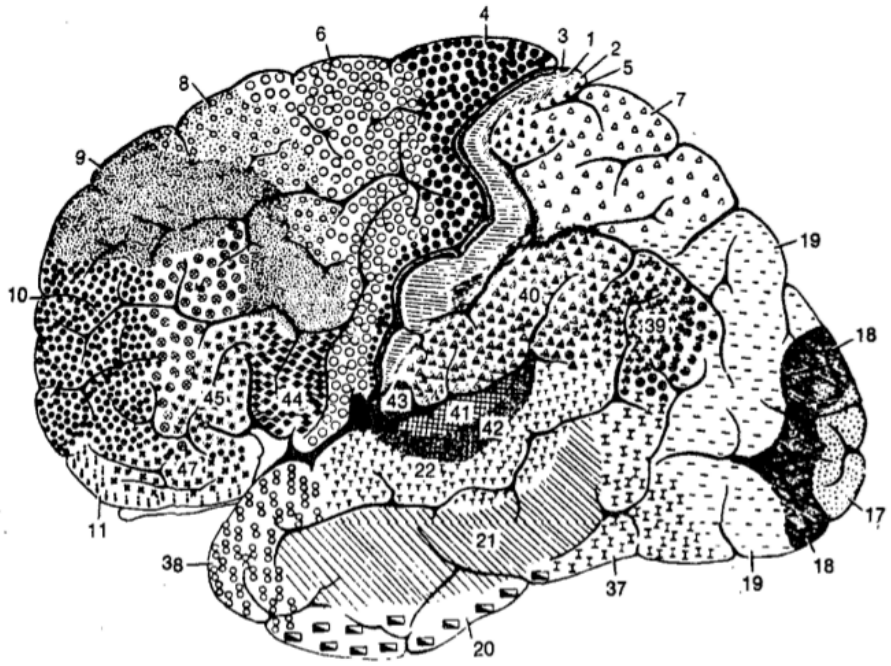

$\hat{Y} = bX + a$

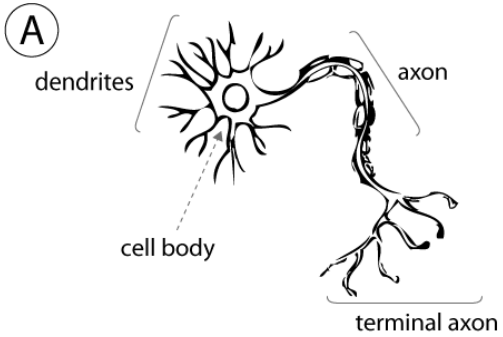

MPG
50
40
30
20
10
0
0
2000
4000
6000
Weight
0
100
200
300
Horsepower

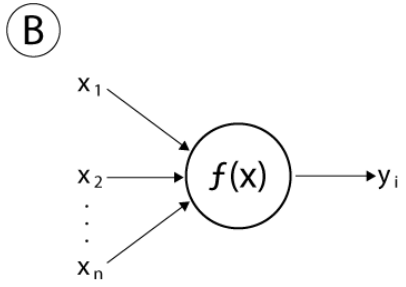

X[0] <= 105.0000
gini = 0.666666666667
samples = 15
X[0] <= 90.5000
gini = 0.5
samples = 6
X[2] <= 51.5000
gini = 0.592592592593
samples = 9
gini = 0.0000
samples = 1
value = [0. 1. 0.]
X[1] <= 37.0000
gini = 0.48
samples = 5
X[1] <= 76.5000
gini = 0.65306122449
samples = 7
gini = 0.0000
samples = 2
value = [2. 0. 0.]

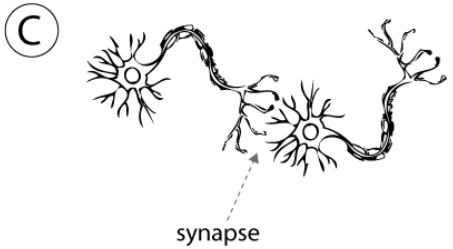

Input
Hidden layer
Output layer
Output
X1
X2
X3
XN
n1
nK
f(.)
o1
Y1
W1 N,K
W2 K,1

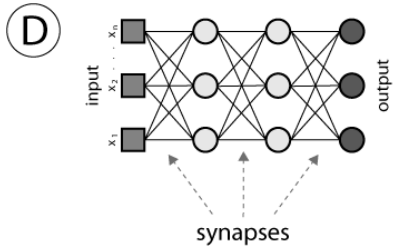
BIOLOGICAL MOTIVATION











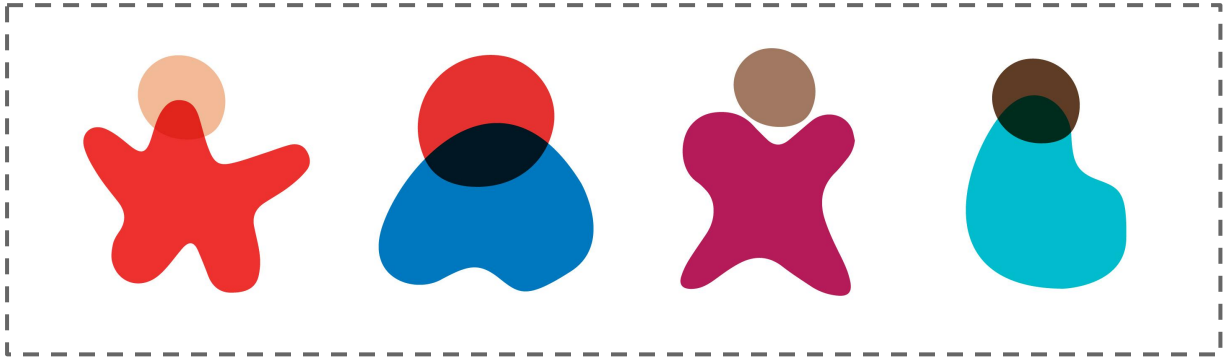
ThoughtWorks®

PROBLEMS

I've got 99 problems, but machine learning ain't one!

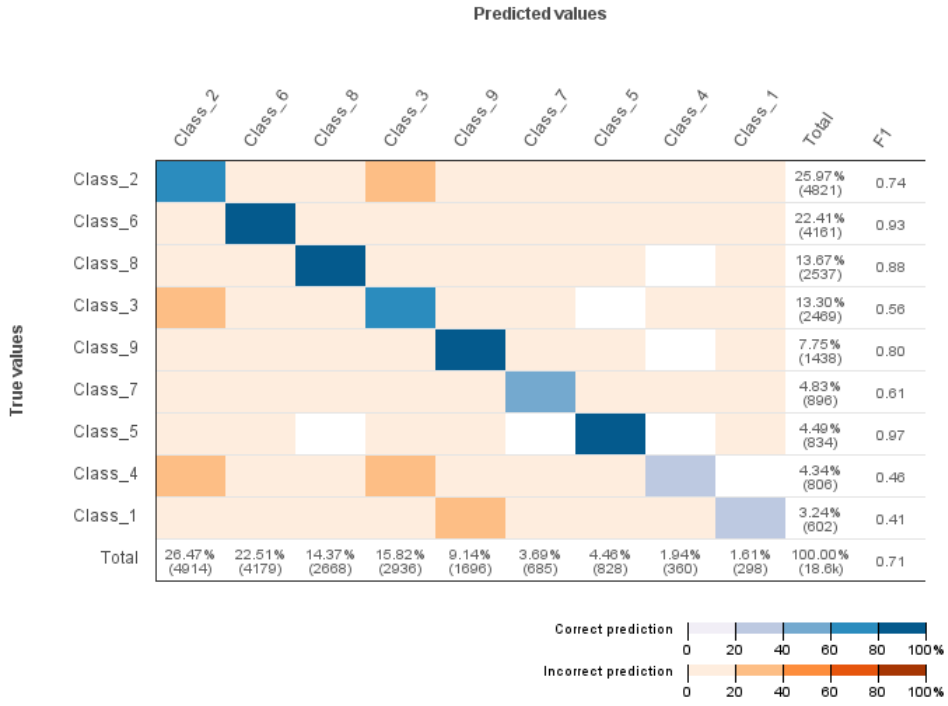
CLASSIFICATION

A



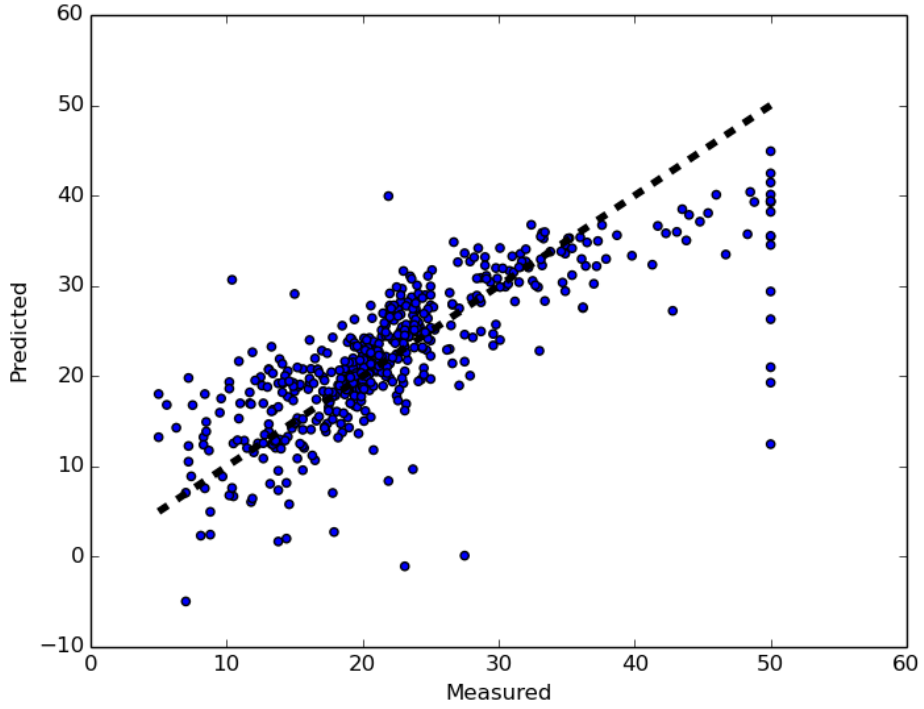
B

CLASSIFICATION



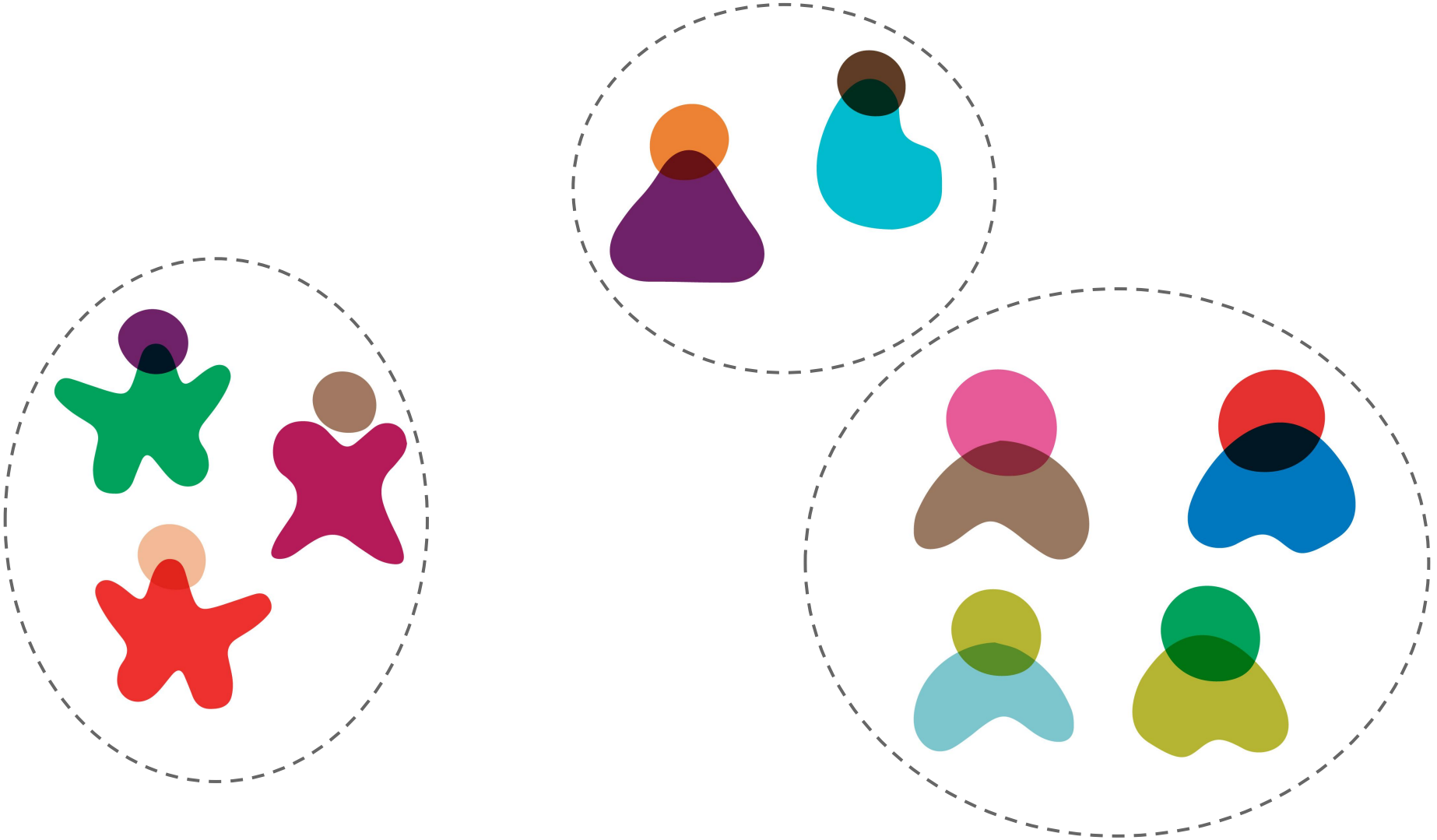
REGRESSION

REGRESSION

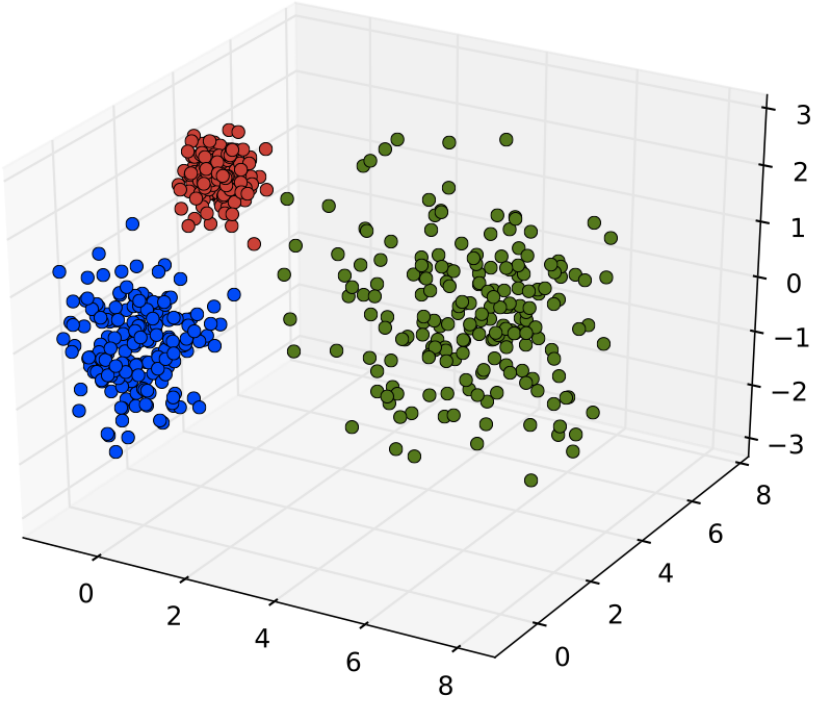


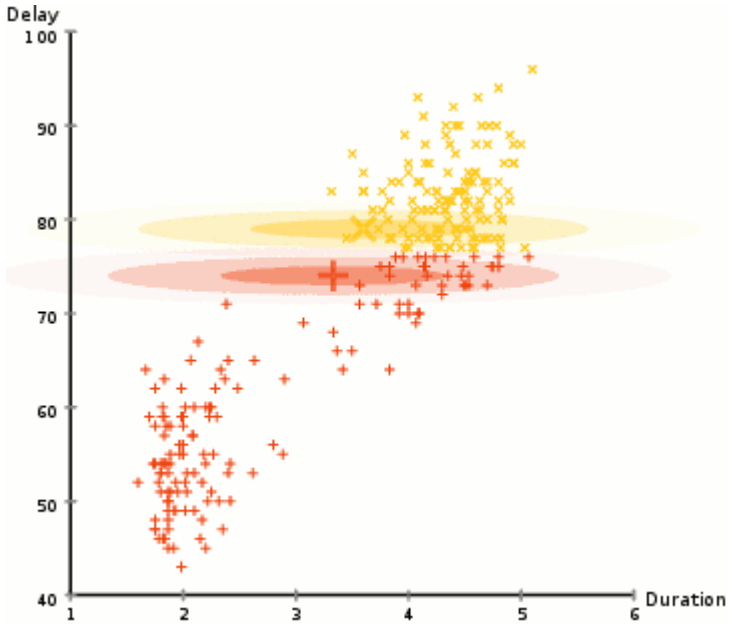
CRIM	Per capita crime rate by town
ZN	Proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS	Proportion of non-retail business acres per town
CHAS	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
NOX	Nitric oxides concentration (parts per 10 million)
RM	Average number of rooms per dwelling
AGE	Proportion of owner-occupied units built prior to 1940
DIS	Weighted distances to five Boston employment centers
RAD	Index of accessibility to radial highways
TAX	Full-value property-tax rate per $10,000
PTRATIO	Pupil-teacher ratio by town
B	1000(Bk - 0.63)^2 where Bk is the proportion of African-Americans by town
LSTAT	% Lower status of the population
MEDV	Median value of owner-occupied homes in $1000's
CAT.MEDV	Binary variable that indicates based on the MEDV variable. If MEDV > 30, CAT.MEDV = 1

CLUSTERING

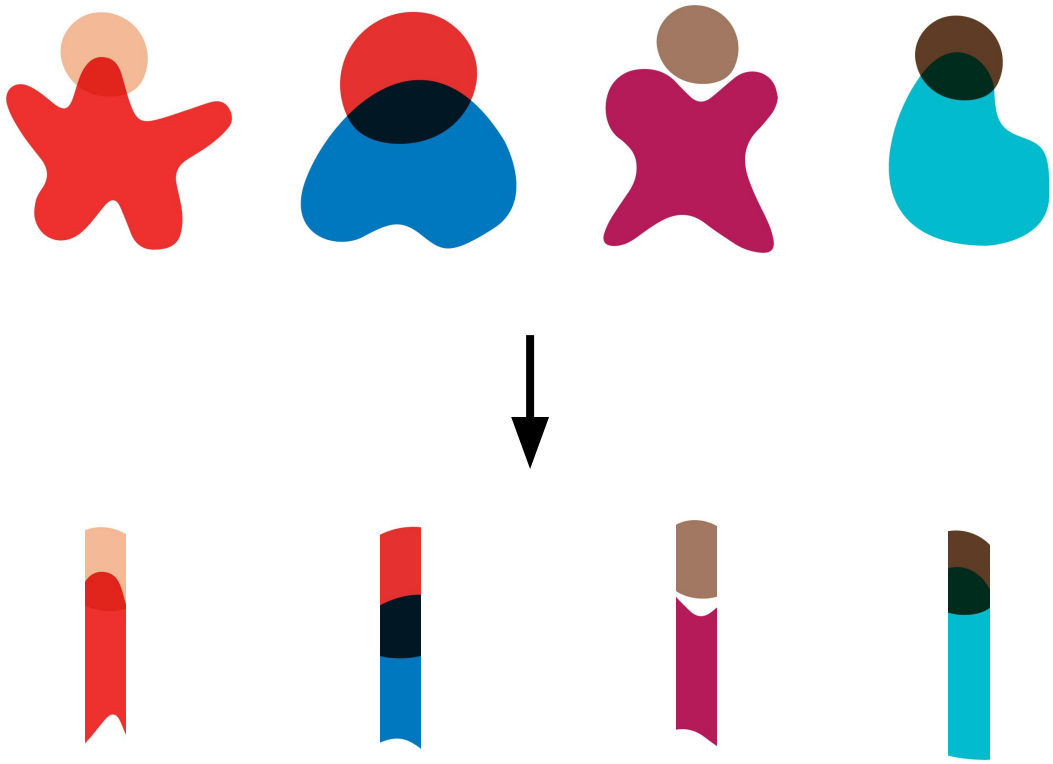


CLUSTERING

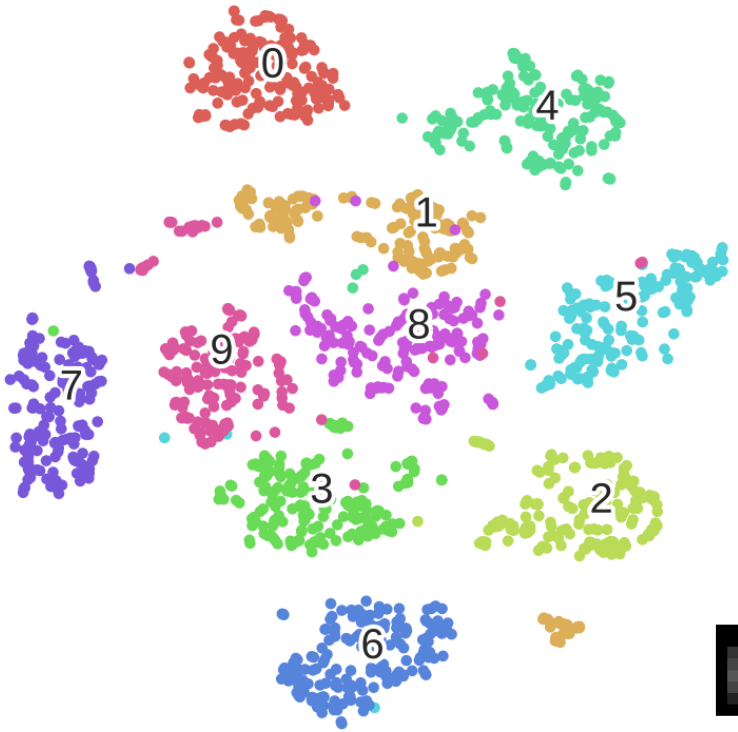

3
2
1
0
−1
−2
−3
0
2
4
6
8
0
2
4
6
8

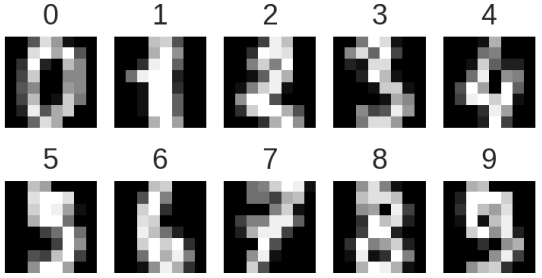

Delay
100
90
80
70
60
50
40
1
2
3
4
5
6
Duration

DIMENSIONALITY REDUCTION



DIMENSIONALITY REDUCTION



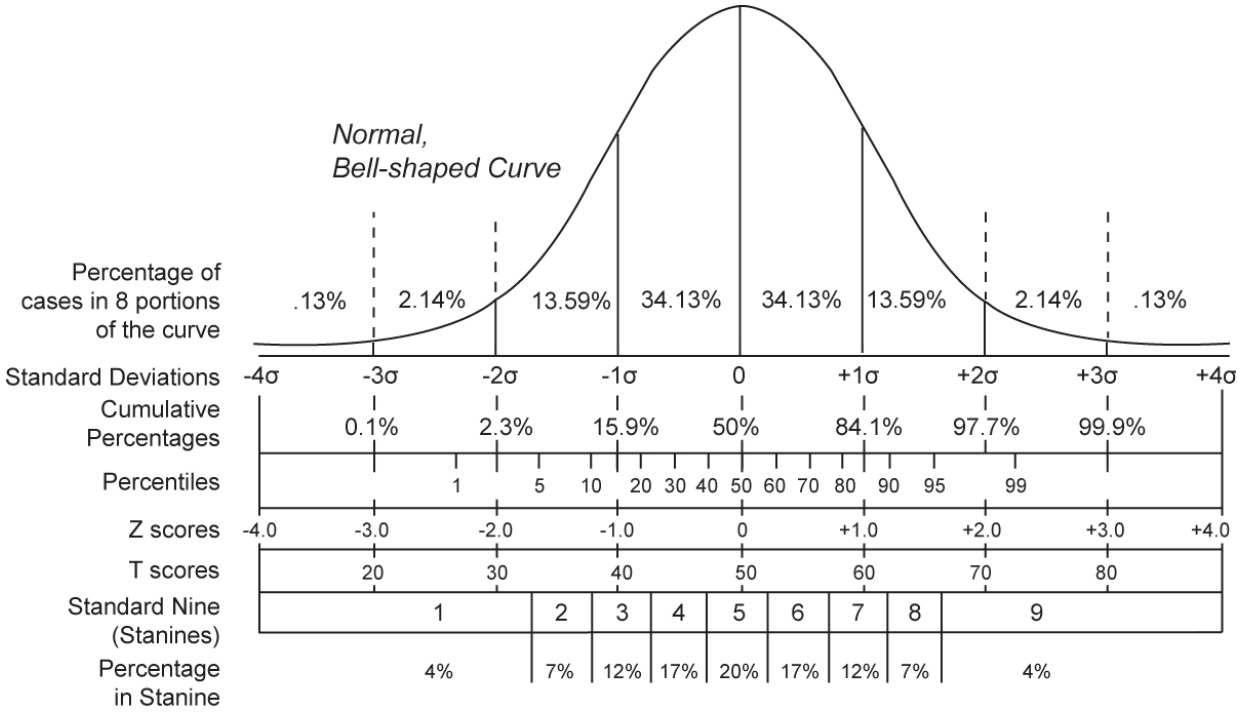


ThoughtWorks®

DESIGN DECISIONS

1, 2 steps!

NORMALIZATION



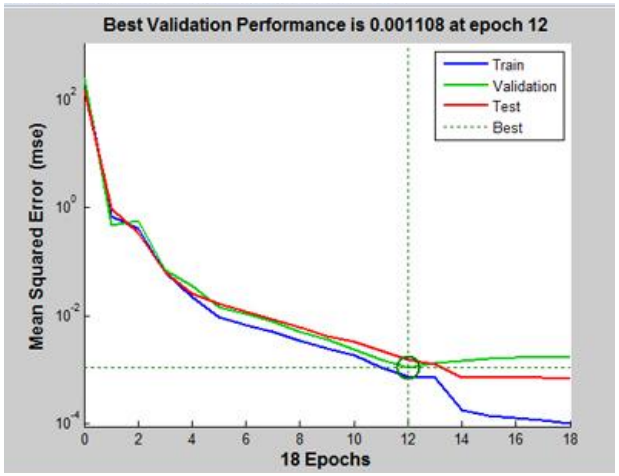
- z-score

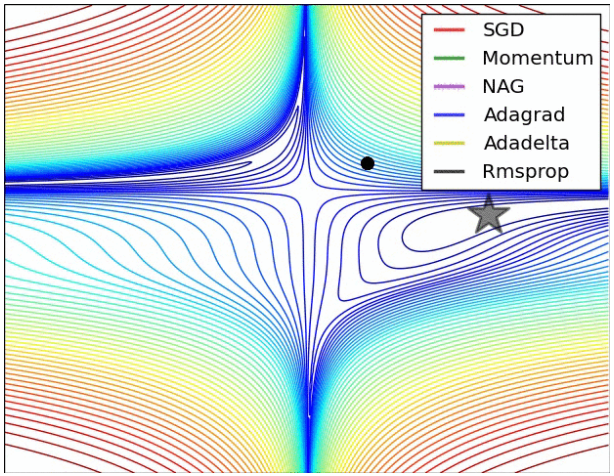
$$X' = \frac{X - \mu}{\sigma}$$

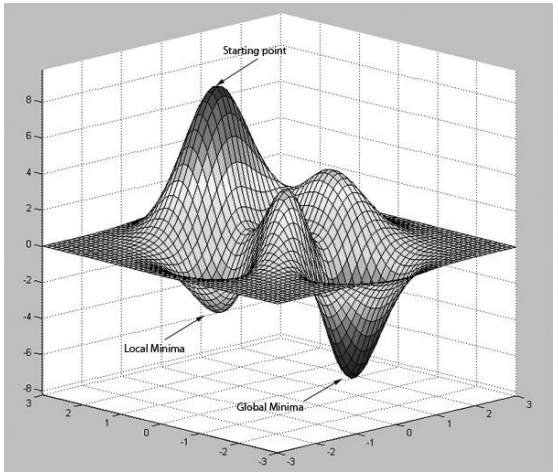
- min-max

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

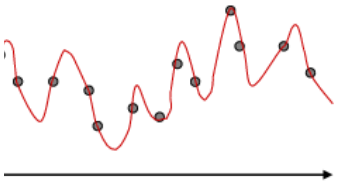
TRAINING


Best Validation Performance is 0.001108 at epoch 12
Train
Validation
Test
Best
Mean Squared Error (mse)
18 Epochs

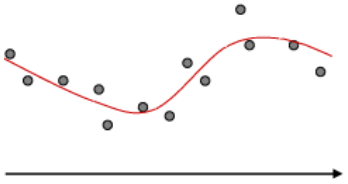

SGD
Momentum
NAG
Adagrad
Adadelta
Rmsprop


Starting point
Local Minima
Global Minima

REGULARIZATION

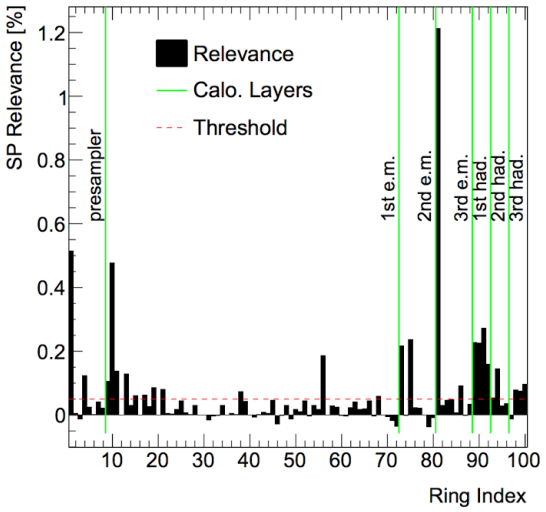


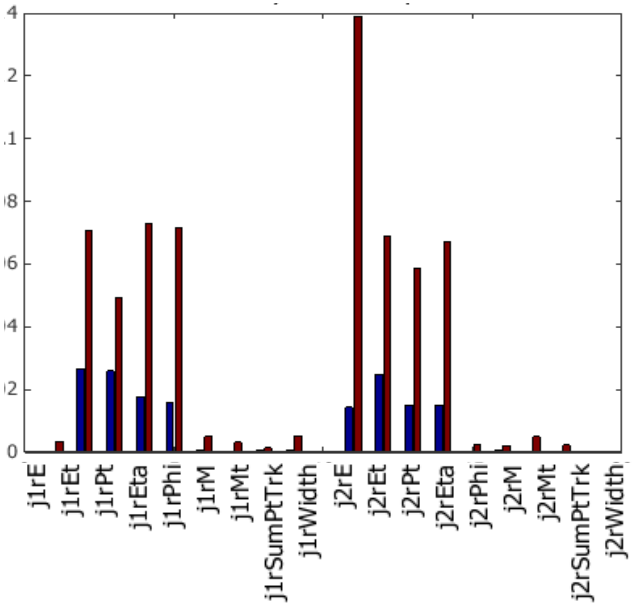
$$\min_f |Y_i - f(X_i)|^2$$

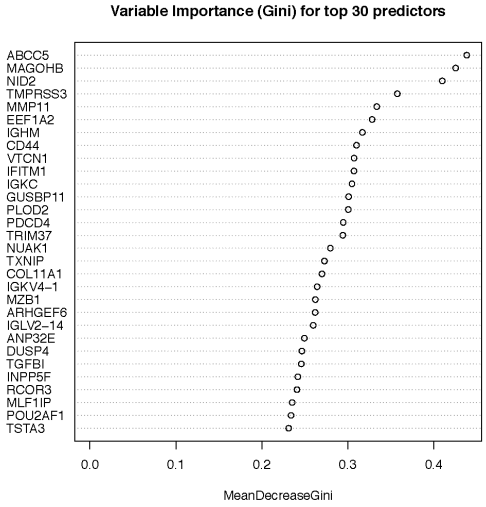


$$\min_{f \in H} \sum_{i=1}^{n} |Y_i - f(X_i)|^2 + \lambda \|f\|_H^2$$

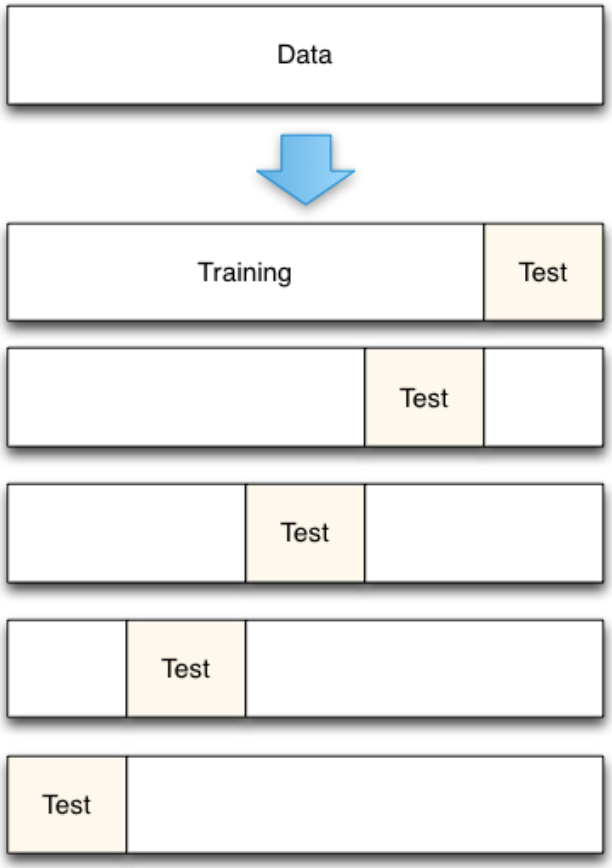
RELEVANCE ANALYSIS







CROSS VALIDATION

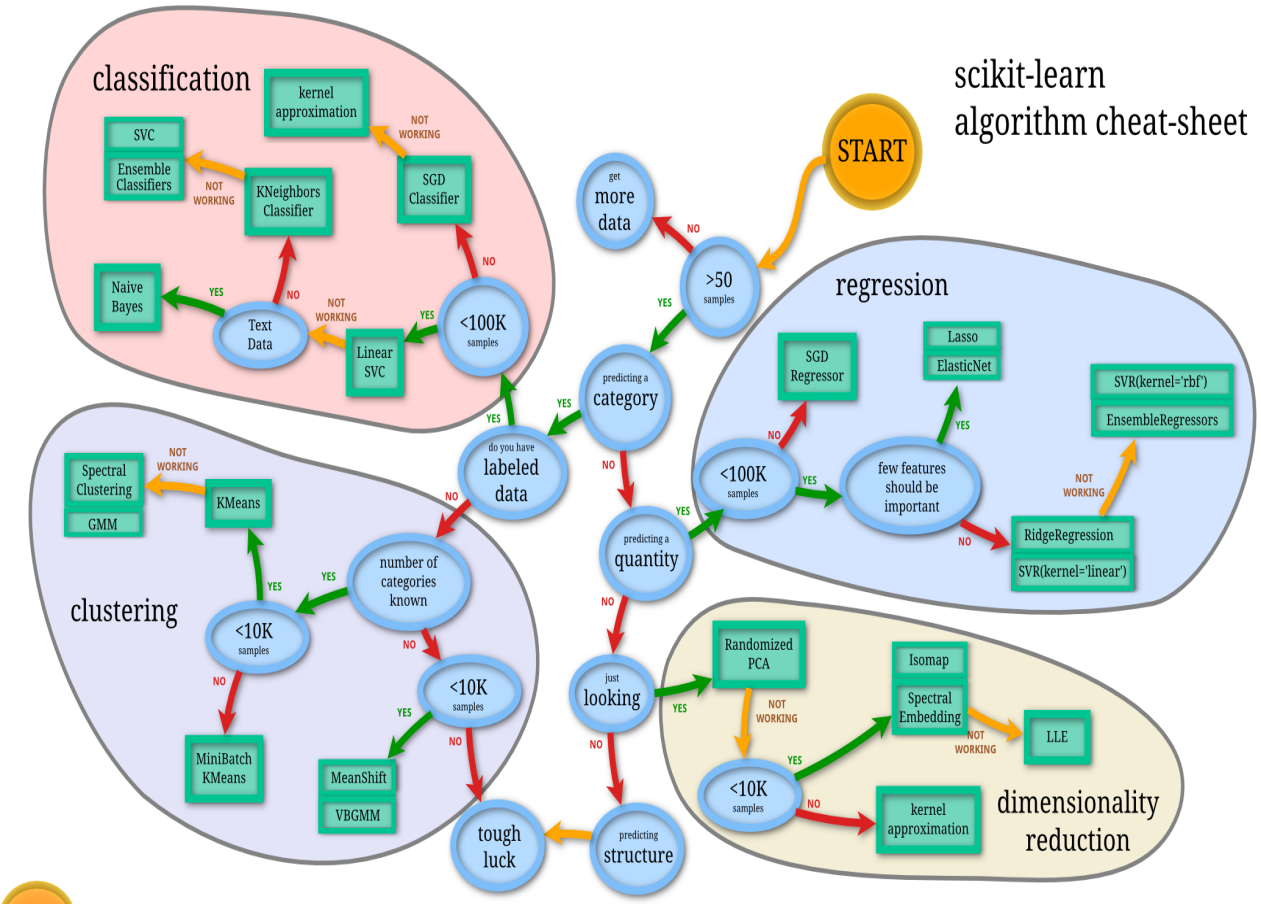


ThoughtWorks®

ALGORITHMS

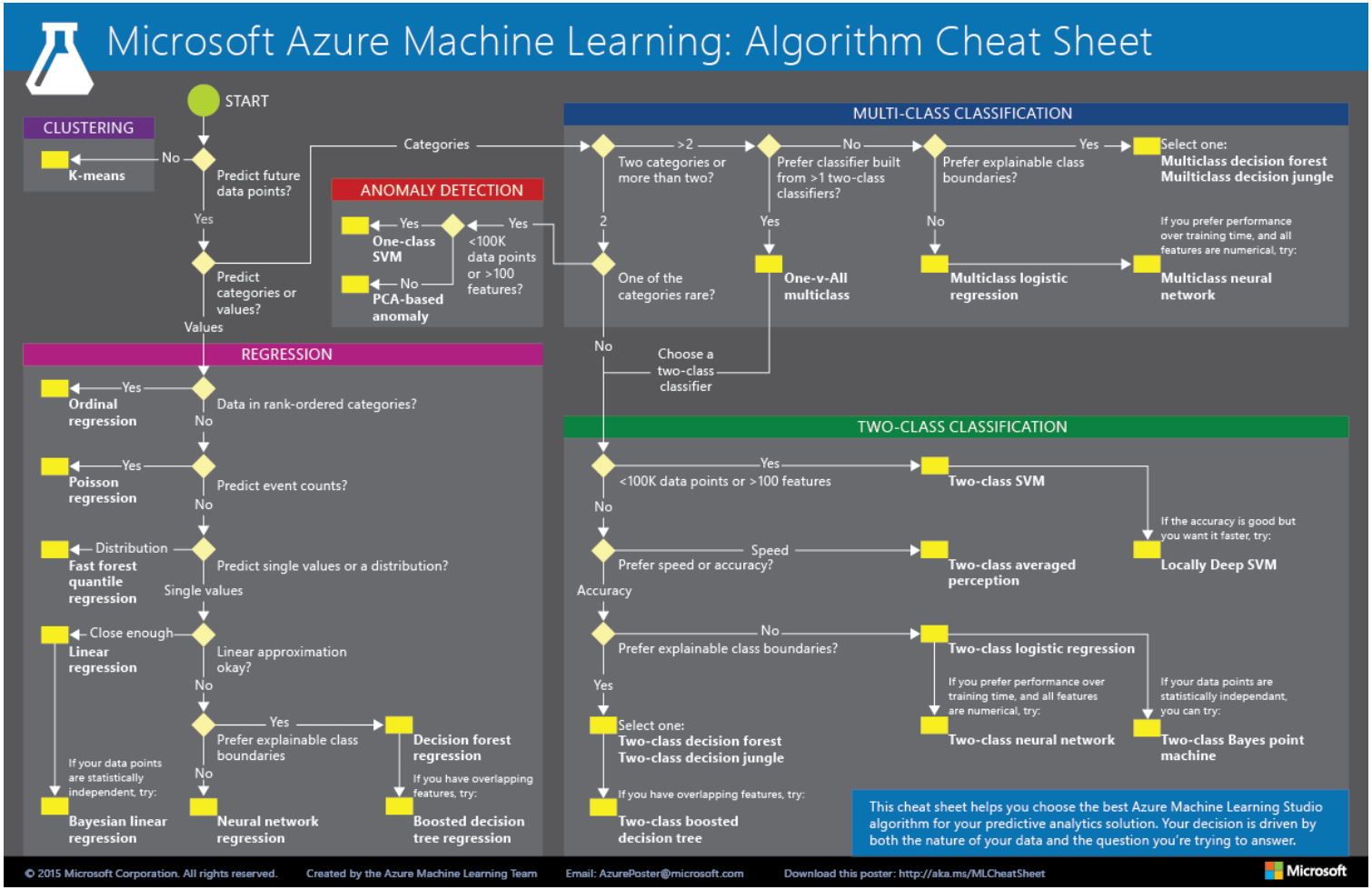
Cheat sheet included!

CHEAT SHEET



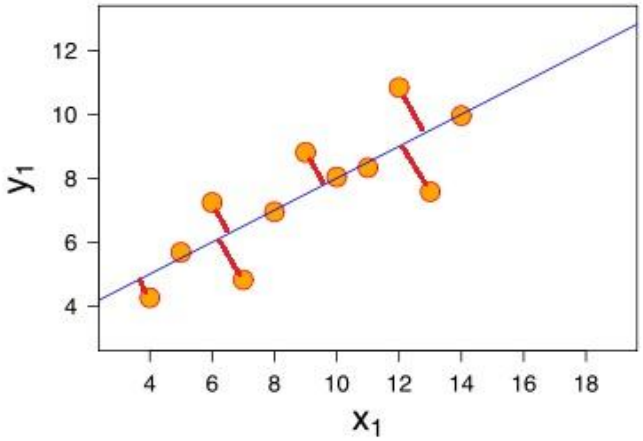
Back

CHEAT SHEET

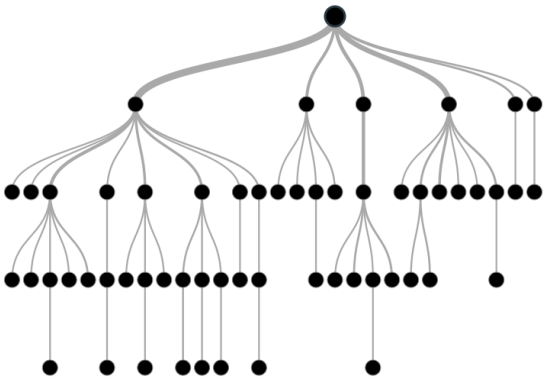

Microsoft Azure Machine Learning: Algorithm Cheat Sheet
START
CLUSTERING
No
K-means
Predict future data points?
Yes
Predict categories or values?
Categories
Values
ANOMALY DETECTION
Yes
One-class SVM
No
PCA-based anomaly
Yes
<100K data points or >100 features?
REGRESSION
Yes
Ordinal regression
Data in rank-ordered categories?
No
Yes
Poisson regression
Predict event counts?
No
Distribution
Fast forest quantile regression
Predict single values or a distribution?
Single values
Close enough
Linear regression
Linear approximation okay?
No
Yes
Prefer explainable class boundaries
Decision forest regression
If you have overlapping features, try:
No
If your data points are statistically independent, try:
Bayesian linear regression
Neural network regression
MULTI-CLASS CLASSIFICATION
Two categories or more than two?
>2
2
Prefer classifier built from >1 two-class classifiers?
No
Yes
Prefer explainable class boundaries?
Yes
No
Select one:
Multiclass decision forest
Muilticlass decision jungle
If you prefer performance over training time, and all features are numerical, try:
Multiclass neural network
One of the categories rare?
Yes
One-v-All multiclass
No
Multiclass logistic regression
No
Choose a two-class classifier
TWO-CLASS CLASSIFICATION
Yes
<100K data points or >100 features
Two-class SVM
No
Speed
Prefer speed or accuracy?
Two-class averaged perception
If the accuracy is good but you want it faster, try:
Locally Deep SVM
Accuracy
No
Prefer explainable class boundaries?
Two-class logistic regression
If you prefer performance over training time, and all features are numerical, try:
Two-class neural network
If your data points are statistically independant, you can try:
Two-class Bayes point machine
Yes
Select one:
Two-class decision forest
Two-class decision jungle
If you have overlapping features, try:
Two-class boosted decision tree
This cheat sheet helps you choose the best Azure Machine Learning Studio algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the question you're trying to answer.
© 2015 Microsoft Corporation. All rights reserved.
Created by the Azure Machine Learning Team
Email: AzurePoster@microsoft.com
Download this poster: http://aka.ms/MLCheatSheet
Microsoft

ALGORITHMS PT. I

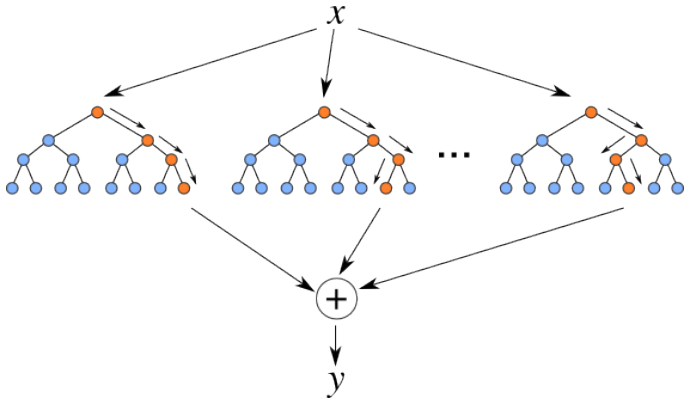
Linear Regression



Decision Trees

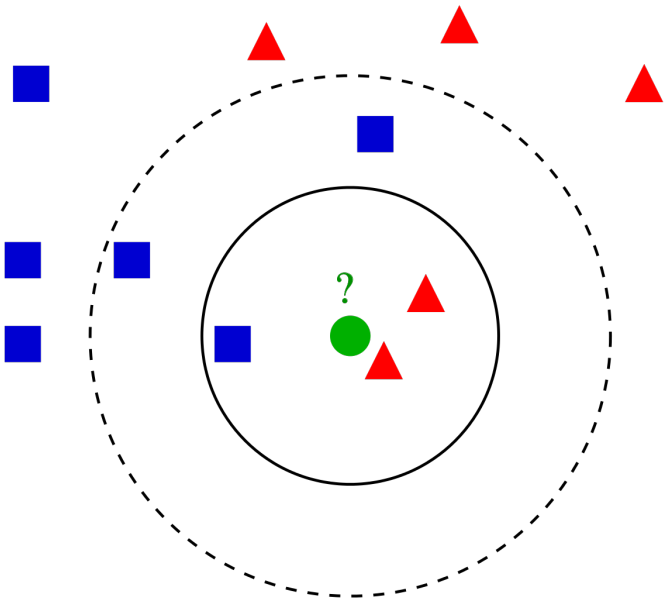


Random Forest

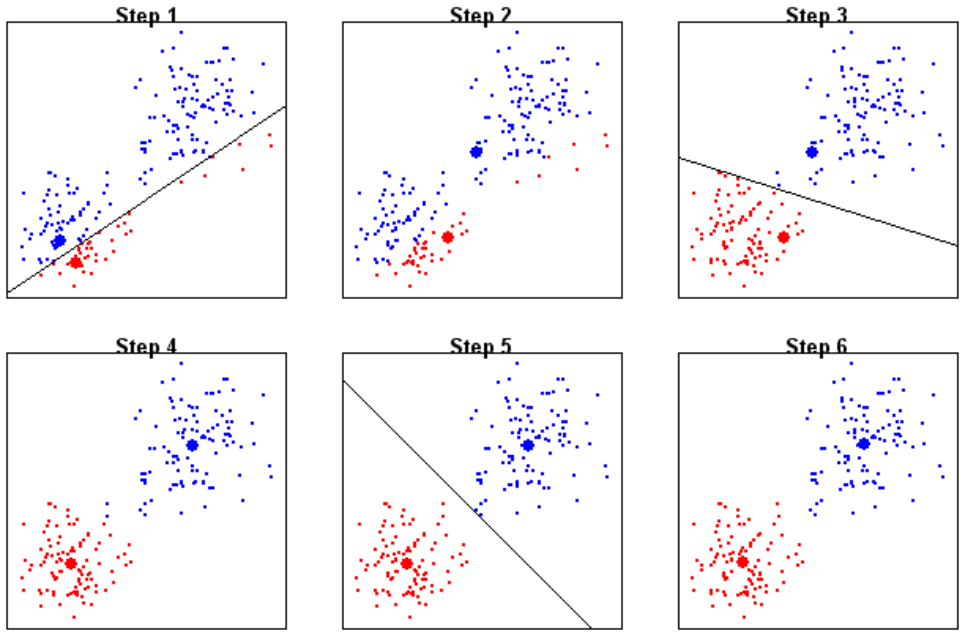


ALGORITHMS PT. II

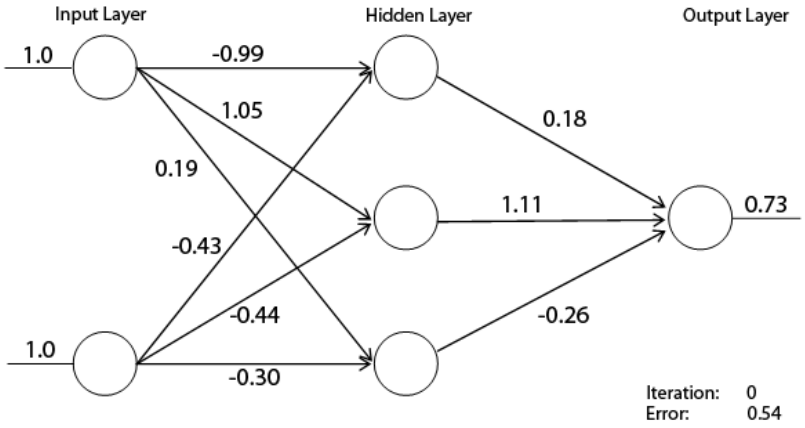
K-Nearest Neighbors

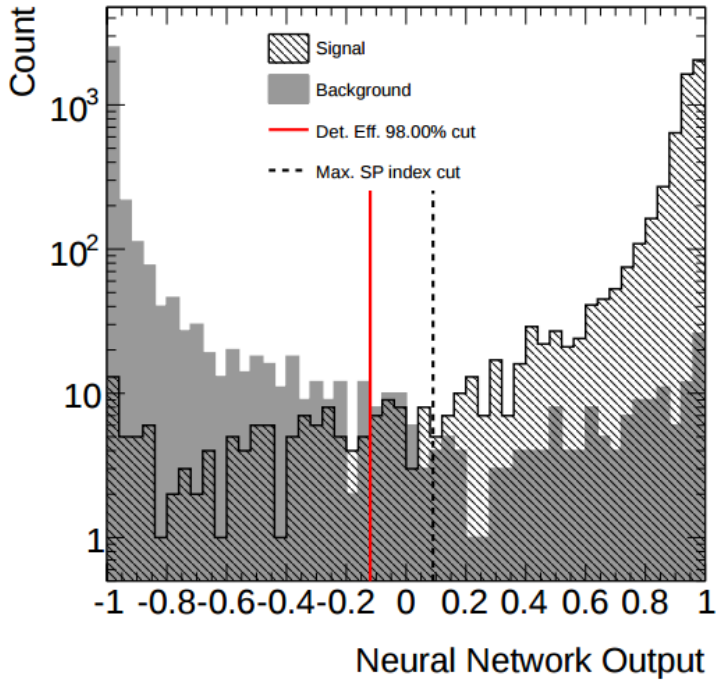


K-Means

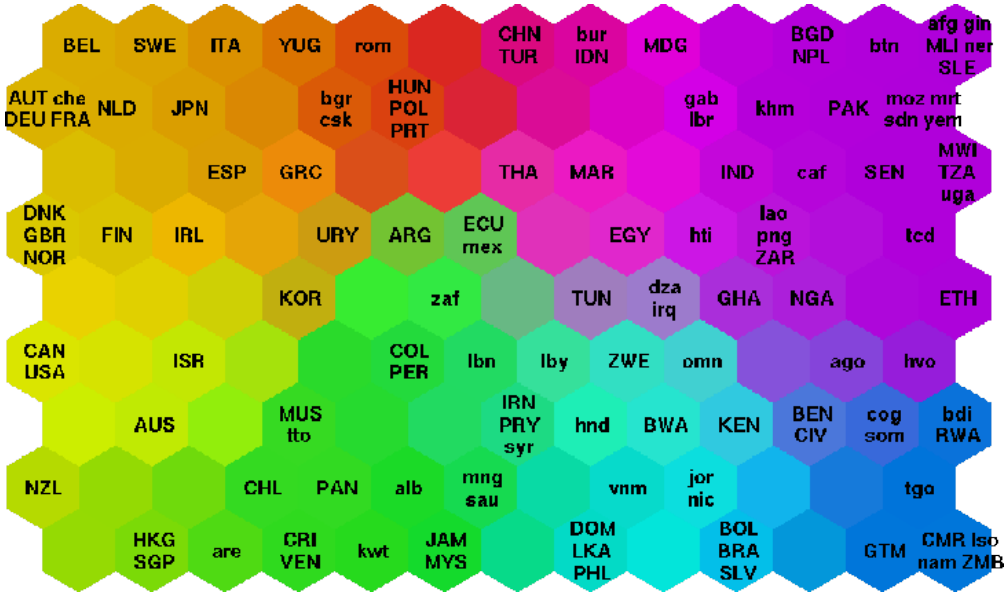


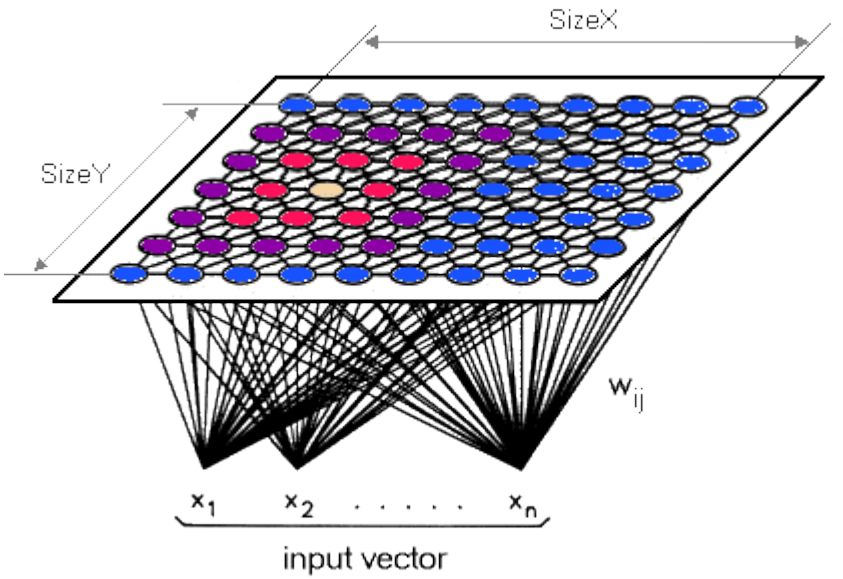
NEURAL NETWORKS



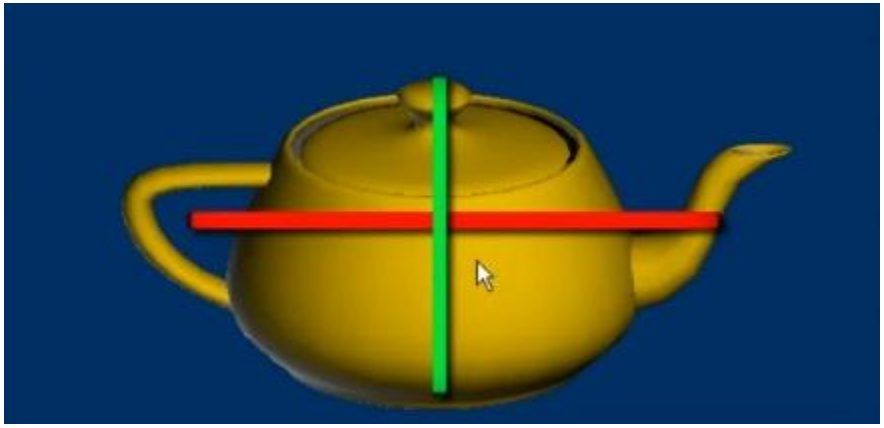


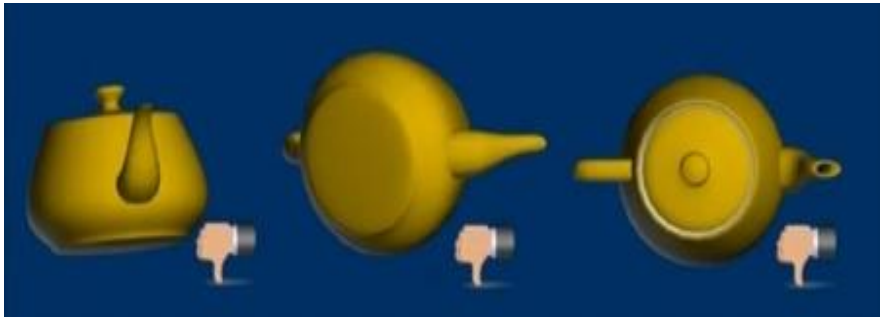
SELF-ORGANIZING MAPS

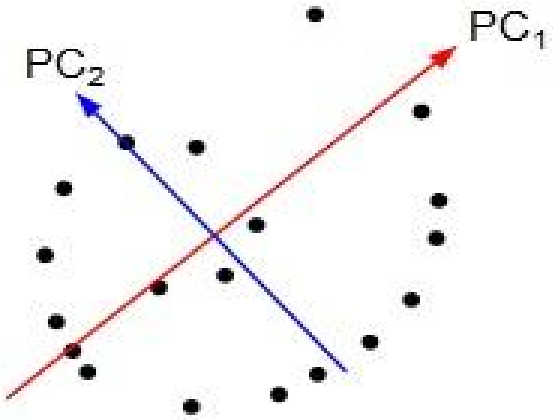




PRINCIPAL COMPONENTS ANALYSIS







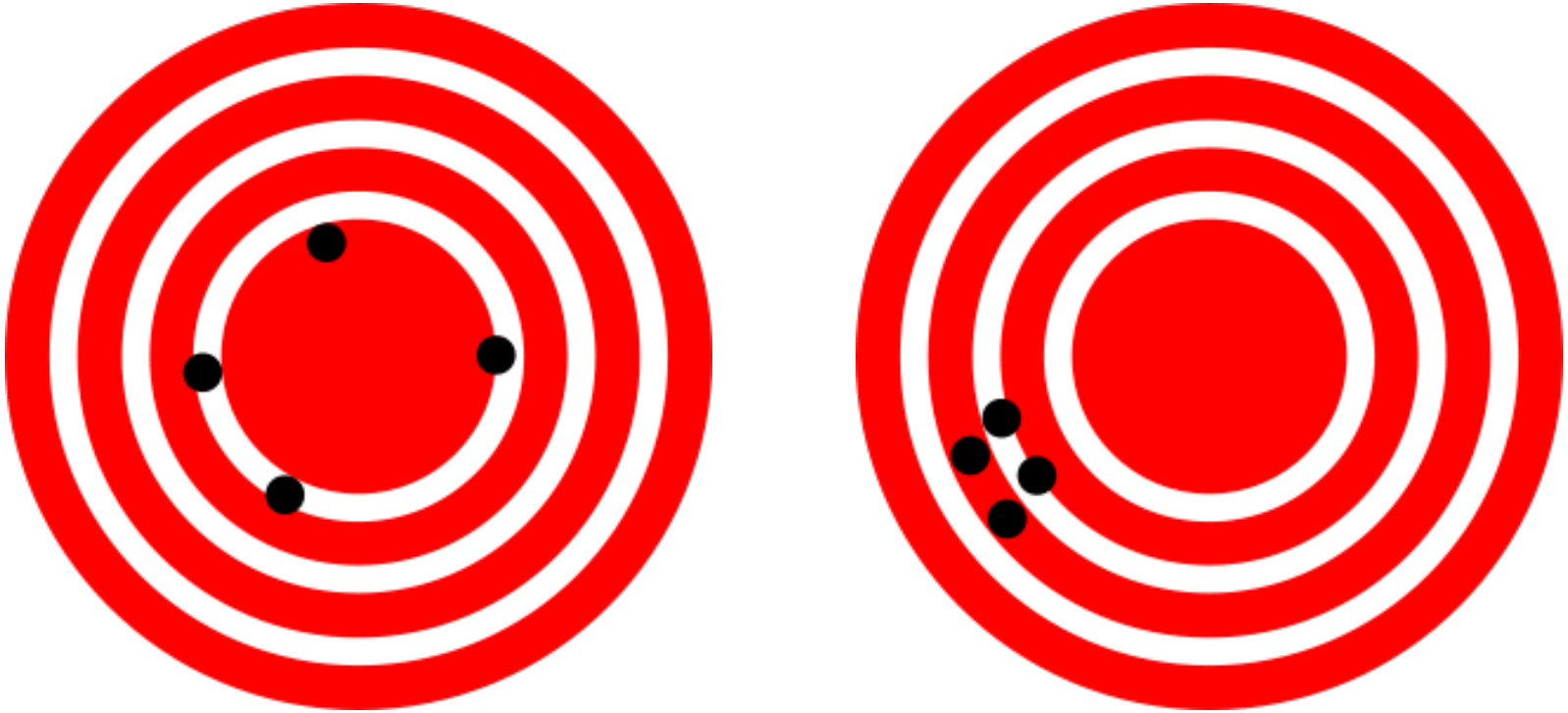
T-SNE

ThoughtWorks®

EVALUATION

How you doin'?

PRECISION AND ACCURACY



CONFUSION MATRIX

TP = True Positives
TN = True Negatives
FP = False Positives
FN = False Negatives

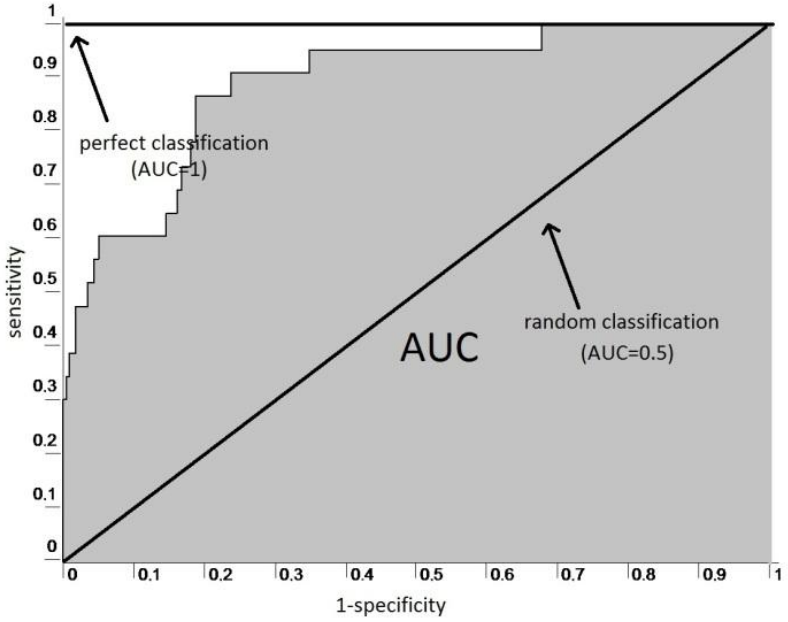
	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

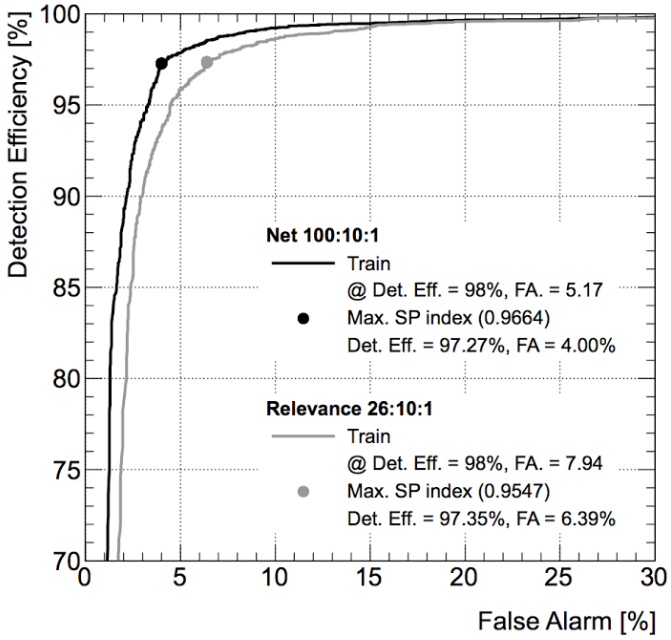
$$\textbf{Precision} = \frac{TP}{TP + FP}$$

$$\textbf{Recall} = \frac{TP}{TP + FN}$$

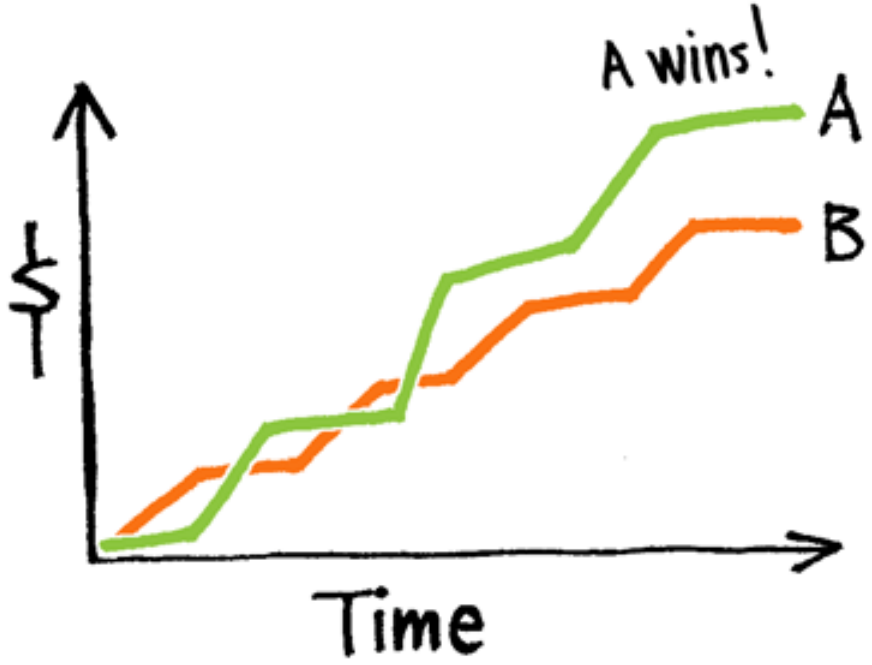
$$\textbf{F1-score} = \frac{2 * precision * recall}{precision + recall}$$

ROC CURVE


perfect classification
(AUC=1)
AUC
random classification
(AUC=0.5)
sensitivity
1-specificity
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Detection Efficiency [%]
False Alarm [%]
Net 100:10:1
Train
@ Det. Eff. = 98%, FA. = 5.17
Max. SP index (0.9664)
Det. Eff. = 97.27%, FA = 4.00%
Relevance 26:10:1
Train
@ Det. Eff. = 98%, FA. = 7.94
Max. SP index (0.9547)
Det. Eff. = 97.35%, FA = 6.39%

A/B TESTS



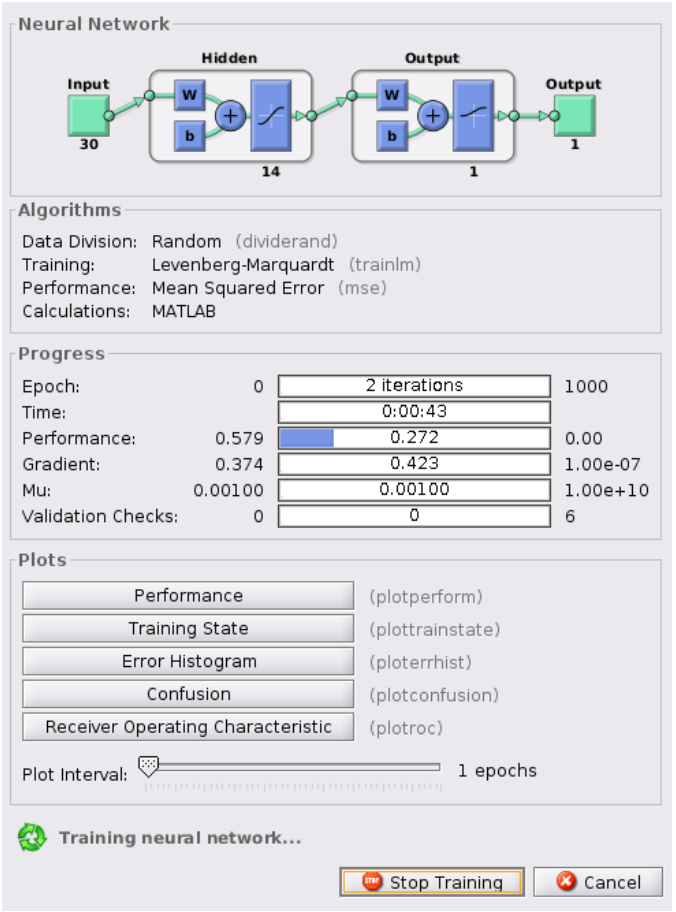
ThoughtWorks®

Code Snippets

"Hello, Machine Learning"

MATLAB 101

```
[x,y] = ovarian_dataset;
net = patternnet(5);
[net,tr] = train(net,x,y);
testX = x(:,tr.testInd);
testY = net(testX);
```

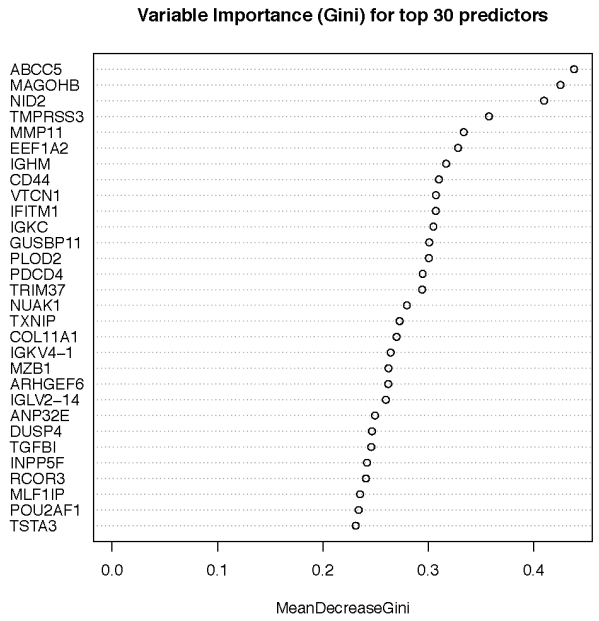


MATLAB 201

```
net = patternnet(14);
net.input.processFcns = {'mapminmax', 'fixunknowns', 'processpca'};
net.inputs{1}.processParams{3}.maxfrac = 0.02;
net.trainFcn = 'trainlm';
net.performFcn = 'mse';
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
[net, tr] = train(net_config, test_inputs, train_targets);
outputs = net(test_inputs);
```

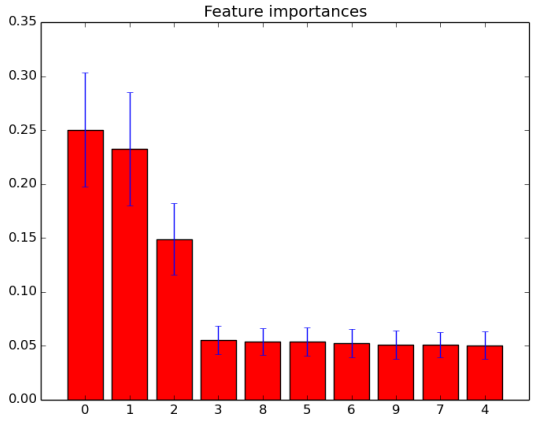
R

```
library(randomForest)
raw.orig < - read.csv(file="train.txt", header=T, sep="\t")
frmla = Metal ~ OTW + AirDecay + Koc
fit.rf = randomForest(frmla, data=raw)
print(fit.rf)
importance(fit.rf)
```



SCIKIT LEARN

```
dataset = pd.read_csv('Data/train.csv')
target = dataset.Activity.values
train = dataset.drop('Activity', axis=1).values
test = pd.read_csv('Data/test.csv').values
rf = RandomForestClassifier(n_estimators=100, n_jobs=-1)
rf.fit(train, target)
predicted_probs = [x[1] for x in rf.predict_proba(test)]
importances = rf.feature_importances_
```



ThoughtWorks®

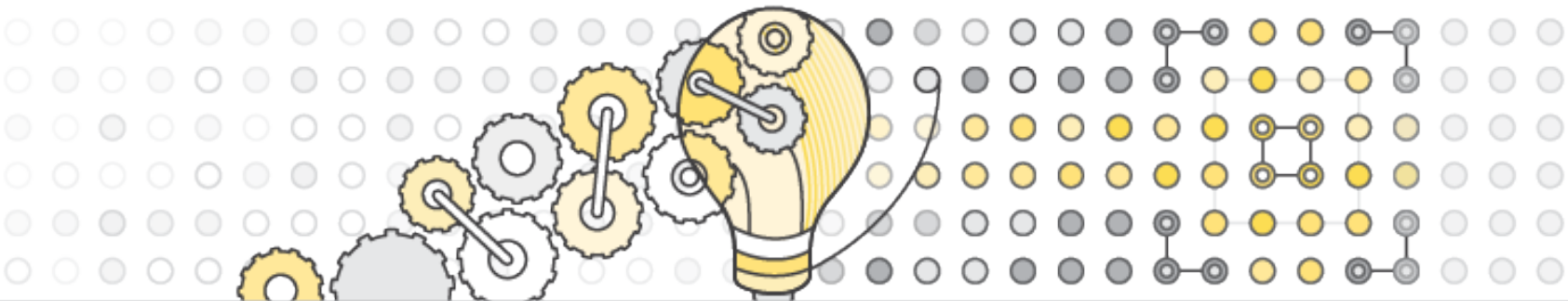
PAY-AS-YOU-GO SERVICES

Amazon Machine Learning

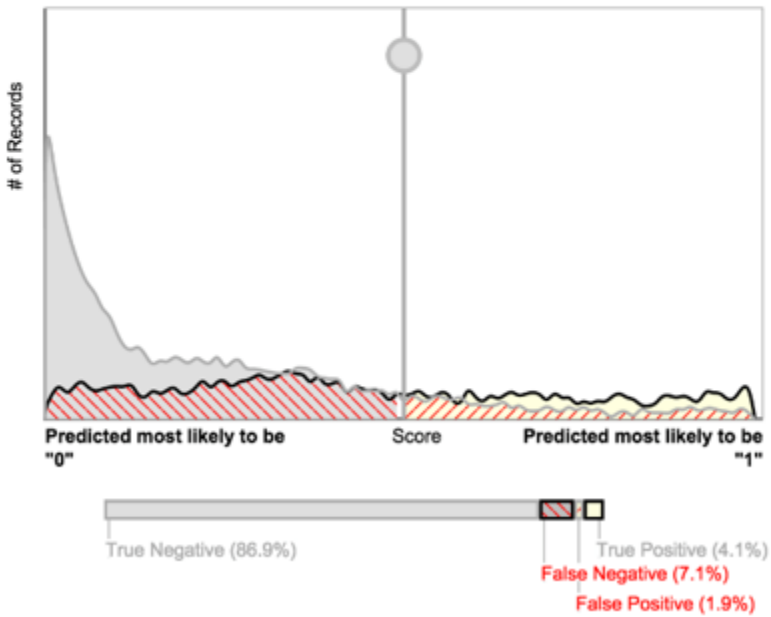
AMAZON MACHINE LEARNING

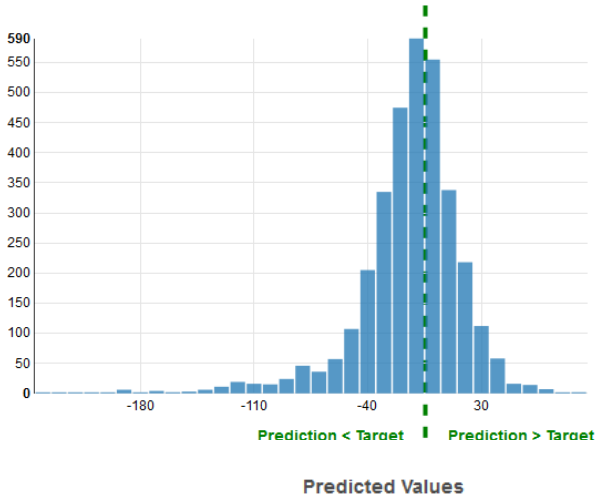
Five easy steps

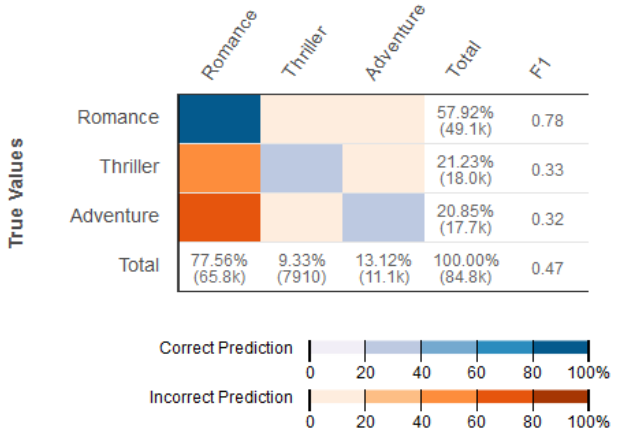
1. Upload csv dataset to Amazon S3
2. Create **Datasource** with metadata about uploaded dataset
3. Create **ML Model** with configurations for model training
4. Create **Evaluation** to analyse and tune model efficiency
5. Create **Prediction** to use trained model with new data



EVALUATION


# of Records
Predicted most likely to be "0"
Score
Predicted most likely to be "1"
True Negative (86.9%)
True Positive (4.1%)
False Negative (7.1%)
False Positive (1.9%)


590
550
500
450
400
350
300
250
200
150
100
50
0
-180
-110
-40
30
Prediction < Target
Prediction > Target
Predicted Values


Romance
Thriller
Adventure
Total
F1
True Values
Romance
Thriller
Adventure
Total
57.92% (49.1k)
21.23% (18.0k)
20.85% (17.7k)
77.56% (65.8k)
9.33% (7910)
13.12% (11.1k)
100.00% (84.8k)
0.78
0.33
0.32
0.47
Correct Prediction
Incorrect Prediction
0
20
40
60
80
100%

SDKs

AWS SDK for Java

AWS SDK for .NET

AWS SDK for Python

AWS SDK for PHP

AWS SDK for JavaScript in Node.js

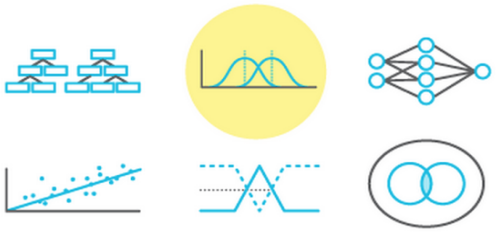
AWS SDK for Ruby

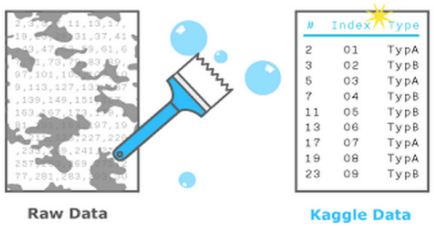
ThoughtWorks®

DATA SCIENCE COMPETITIONS

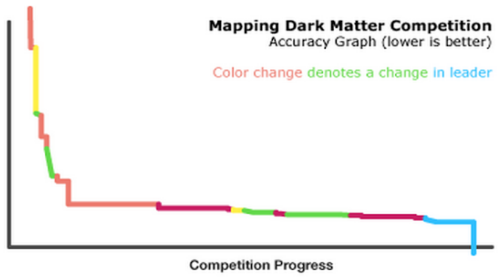
Challenge accepted!

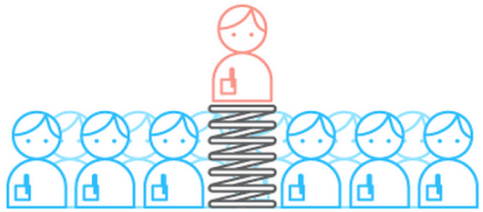
KAGGLE




Raw Data
Index Type
2 01 TypA
3 02 TypB
5 03 TypA
7 04 TypB
11 05 TypB
13 06 TypB
17 07 TypA
19 08 TypA
23 09 TypB
Kaggle Data


Training Data
Sample Entry


Mapping Dark Matter Competition
Accuracy Graph (lower is better)
Color change denotes a change in leader
Competition Progress



SPONSORED


EMC²

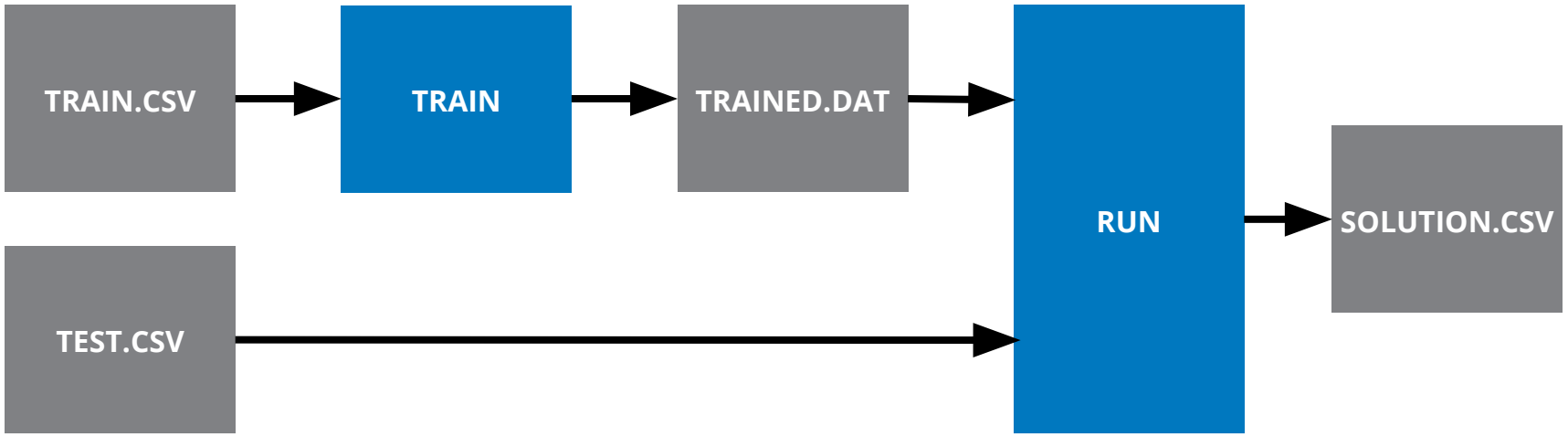

Walmart


information security
amazon


stackoverflow


imagination at work

END TO END



THANK YOU

Questions?

Dhiana Deva

ddeva@thoughtworks.com

ThoughtWorks®