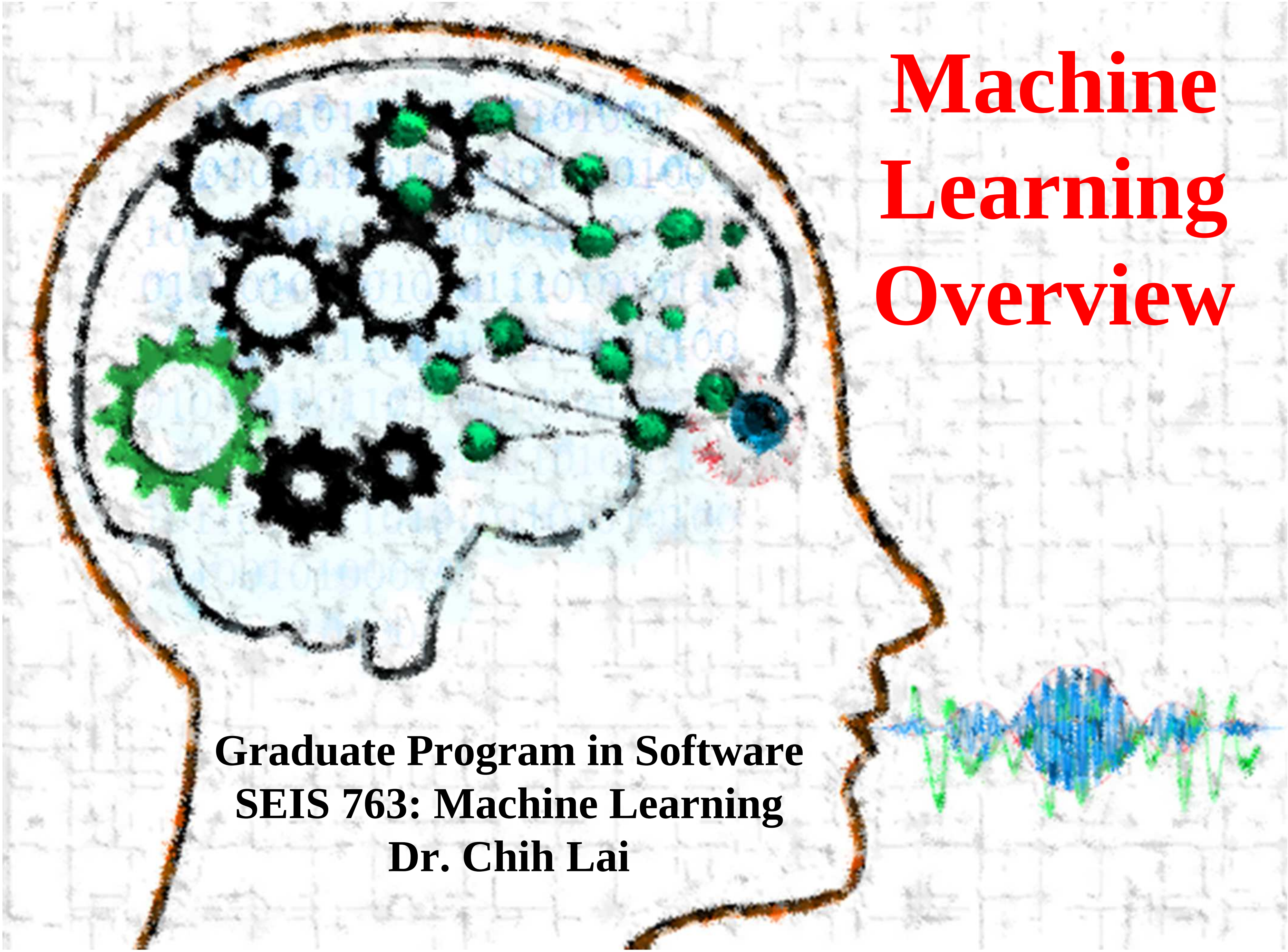
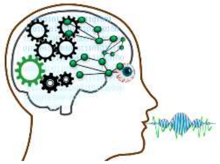


Machine Learning Overview

Graduate Program in Software
SEIS 763: Machine Learning
Dr. Chih Lai



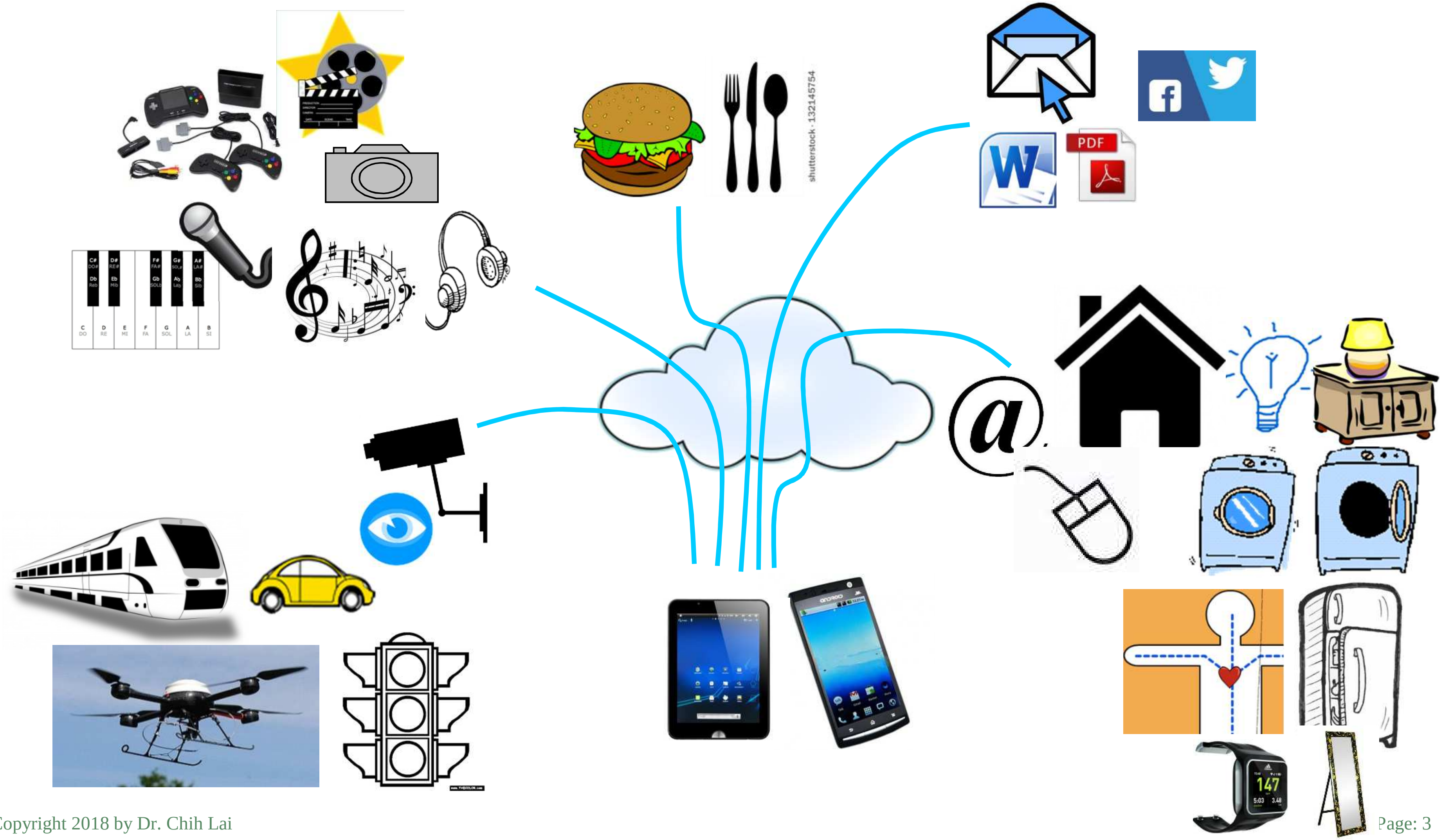
Outline

- Proliferation of Data (Structured and Unstructured) / Internet of Things.
  - Just manually query all the data? Descriptive Analytics?
- Machine Learning Tools.
- Major topics in ML.
  - Linear Regression, Logistic Regression, **Linear Discriminant Analysis (LDA)**.
  - Gradient Descent, Regularization & Overfitting.
  - Support Vector Machine (SVM) and Kernel Methods.
  - Ensemble, Boosting / Bagging, Quality Evaluation
  - Self-Organized Map (SOM), **PCA, ICA, SVD, MDS**.
  - Collaborative Filtering, Matrix Factorization, (Recommender System), Gaussian Mixture Model.
- Machine Learning Computing Platform.

2020

INTERNET
of **THINGS**

- **75-billion connected devices, 50-trillion GB data.**



Data = New Oil

INTERNET
of THINGS

- **75-billion connected devices, 50-trillion GB data.**

Data is new oil!!

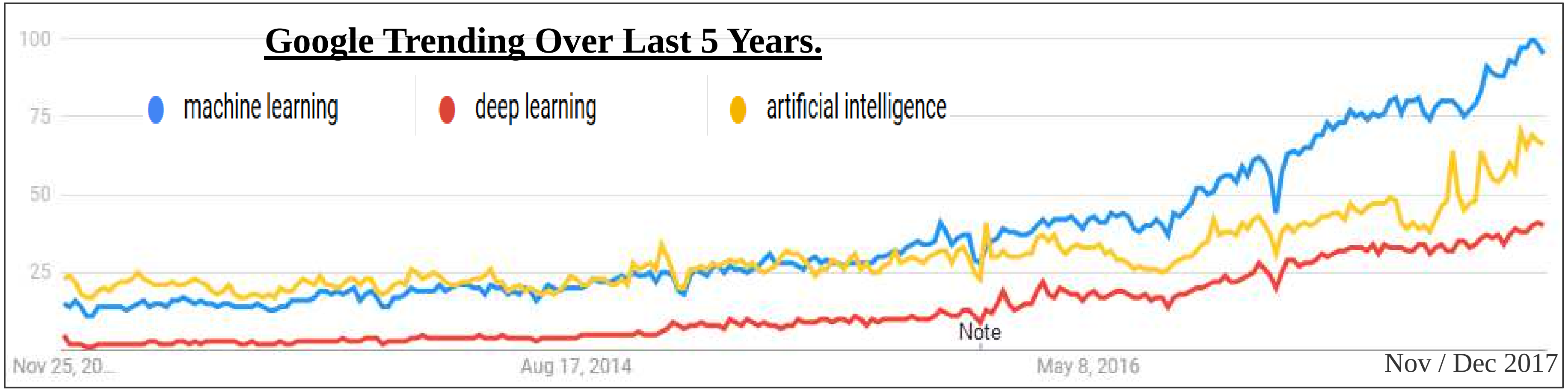
Drowning in data, but starving for knowledge.

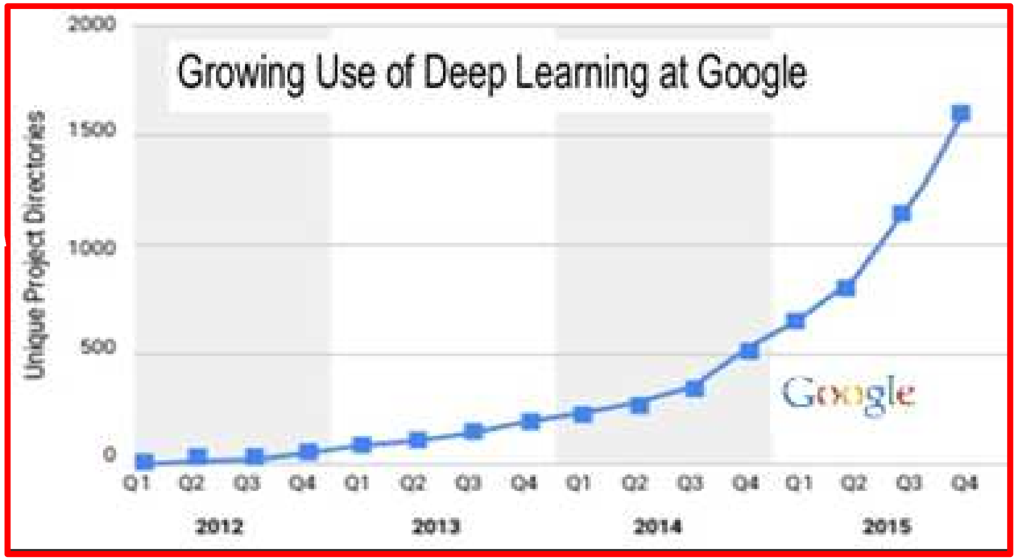
Data rich but information poor.

Methodsssss…

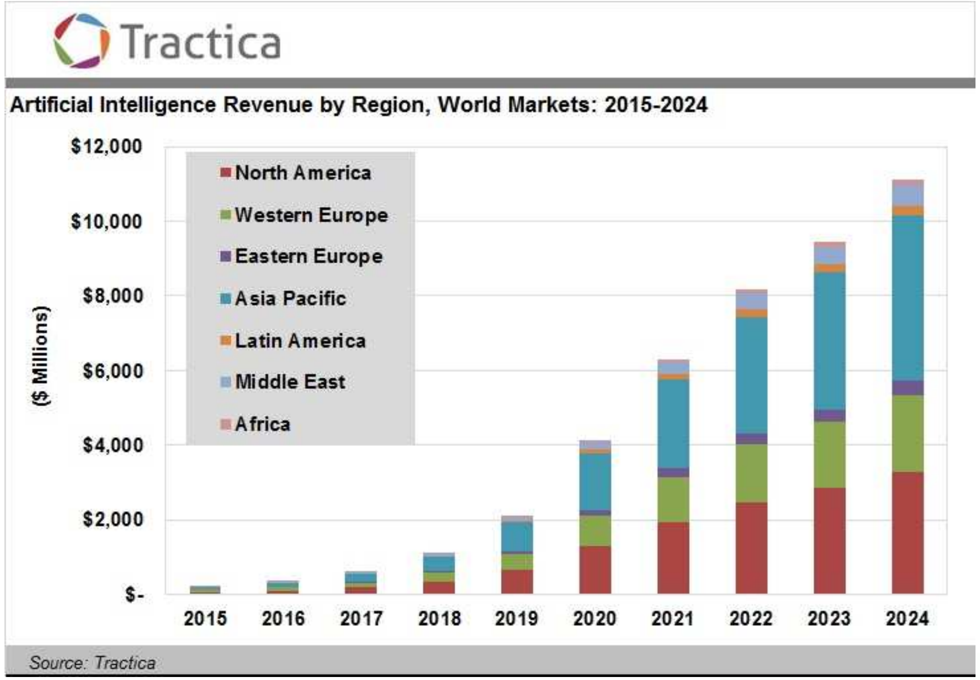
Everything Is About…

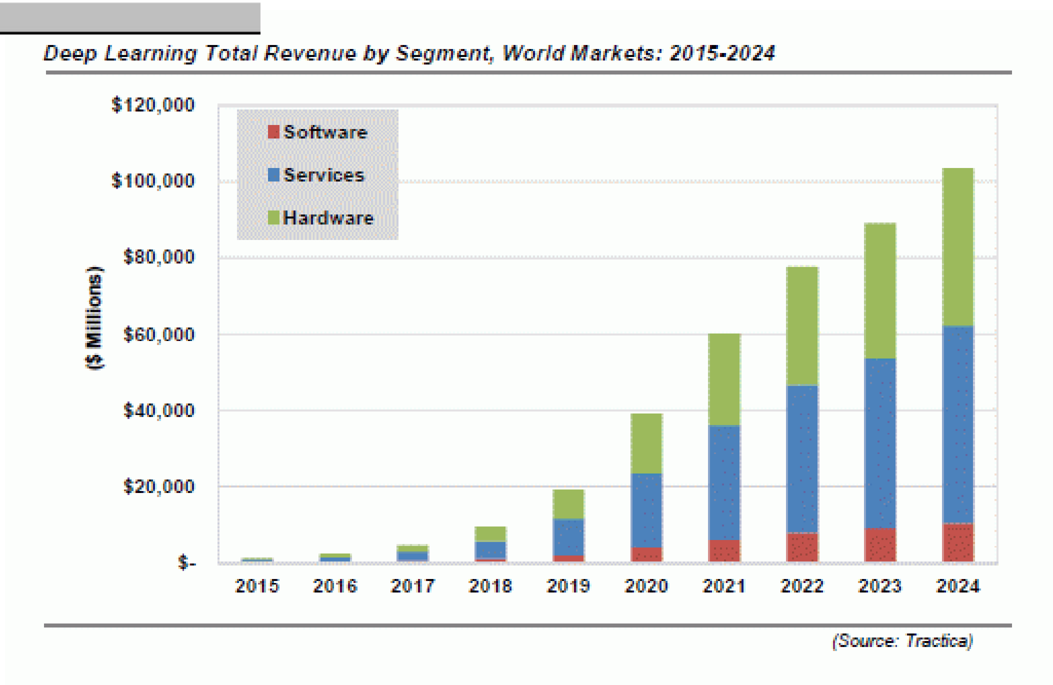
- Google is rethinking everything with ***machine learning*** at the core.
- Growing use of ***deep learning*** at Google is exponential.





Future Projection



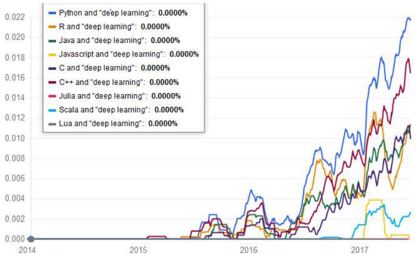


Some Popular ML+DL Tools

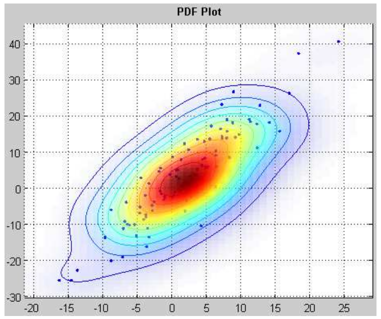
GPU??	Tool				
	Python	R	Spark	Matlab	TensorFlow
License	Open source	Open source	Open source	Proprietary	Open source
Distributed	No	No	Yes	No YES	No
Visualization	Yes	Yes	No	Yes	No
Neural nets	Yes	Yes	Multilayer perceptron classifier	Yes	Yes
Supported languages	Python	R	Scala, Java, Python, and R	Matlab	Python and C++
Variety of machine-learning models	High	High	Medium	High	Low
Suitability as a general-purpose tool	High	Medium	Medium	High	Low
Maturity	High	Very high	Medium ??	Very high	Low

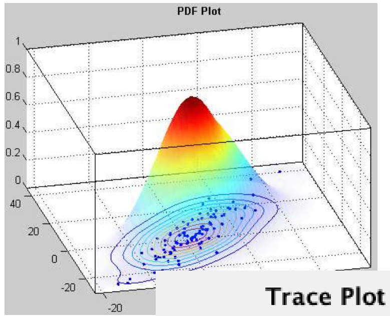
[IEEE Computing Edge April 2017, pp.12]

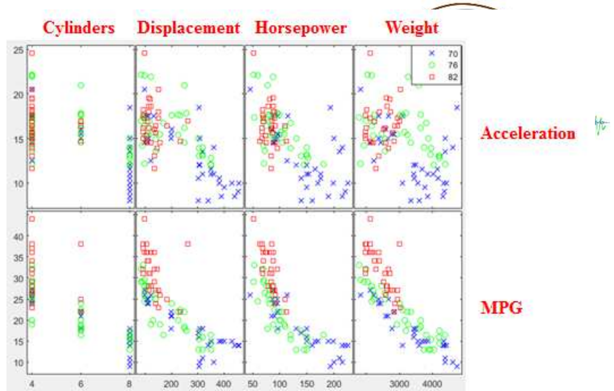
- http://www.kdnuggets.com/2017/01/most-popular-language-machine-learning-data-science.html

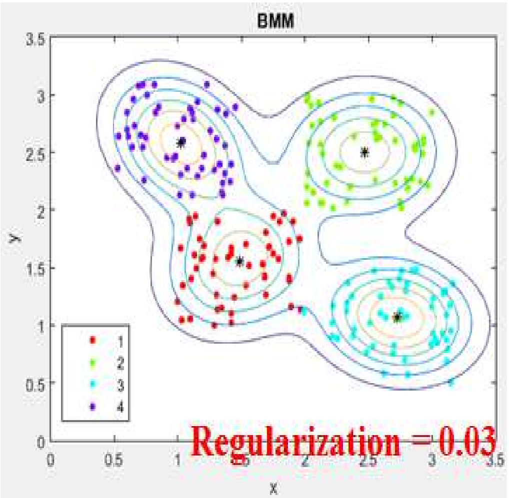


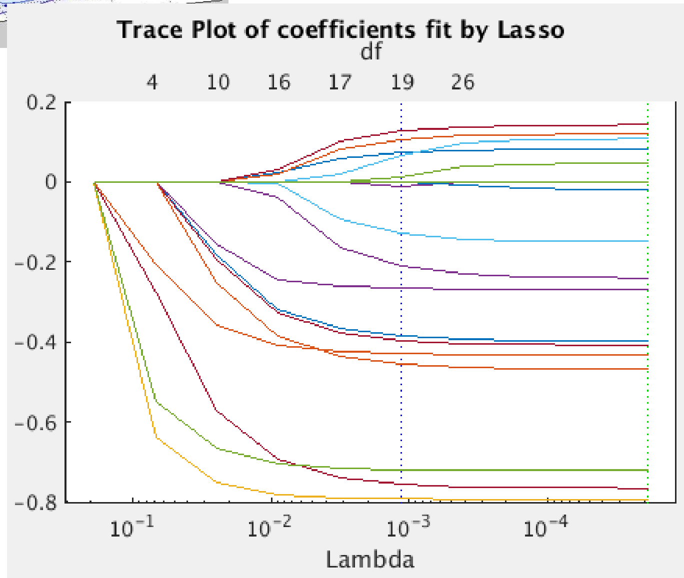
Visualizing Data

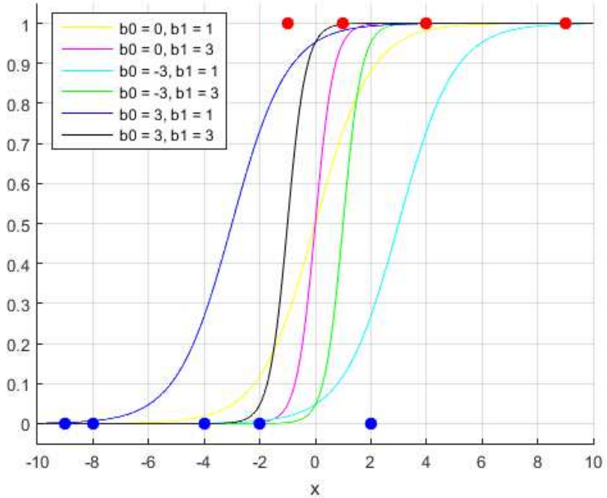


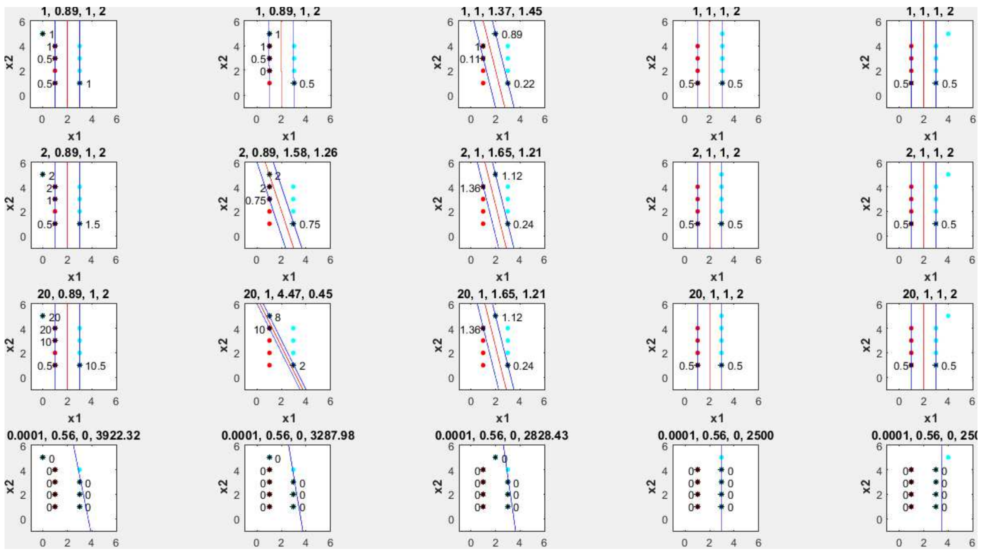


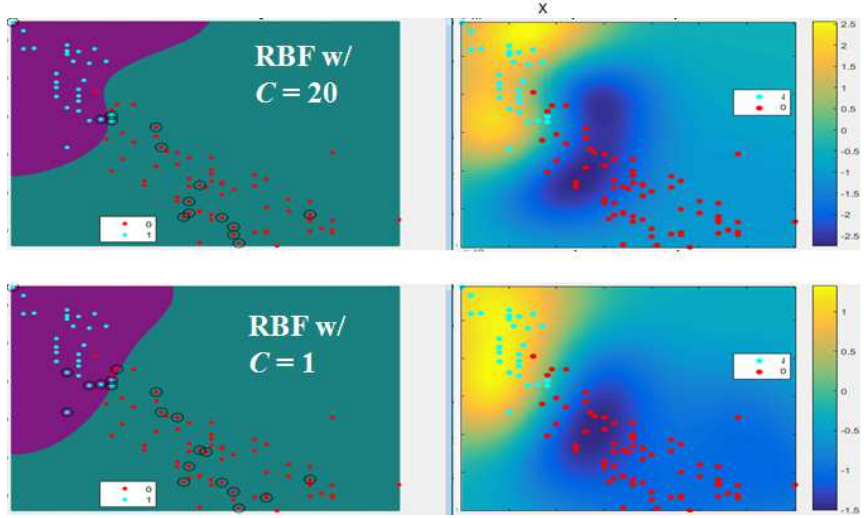












Outline

- Proliferation of Data (Structured and Unstructured) / Internet of Things.
  - Just manually query all the data? Descriptive Analytics?
- Machine Learning Tools.
- Major topics in ML.
  - Linear Regression, Logistic Regression, **Linear Discriminant Analysis (LDA)**.
  - Gradient Descent, Regularization & Overfitting.
  - Support Vector Machine (SVM) and Kernel Methods.
  - Ensemble, Boosting / Bagging, Quality Evaluation
  - Self-Organized Map (SOM), **PCA, ICA, SVD, MDS**.
  - Collaborative Filtering, Matrix Factorization, (Recommender System), Gaussian Mixture Model.
- Machine Learning Computing Platform.

Machine Learning

- Machine derives weights (importance) for each predictor to predict target.

X / Predictors / Features / Independent Variables

Weights = 0.08 -1.48 0.5 -0.01 9.7 ???

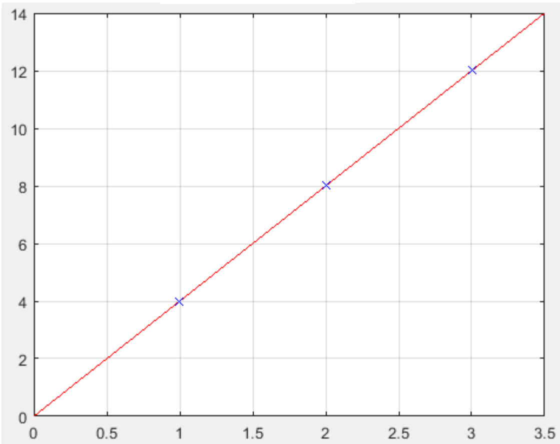
	Age	Gender	Height	Weight	Smoker	HealthStatus	Systolic
1 Smith	38	'Male'	71	176	1	'Excellent'	124
2 Johnson	43	'Male'	69	163	0	'Fair'	109
3 Williams	38	'Female'	64	131	0	'Good'	125
4 Jones	40	'Female'	67	133	0	'Fair'	117
5 Brown	49	'Female'	64	119	0	'Good'	122
6 Davis	46	'Female'	68	142	0	'Good'	121
7 Miller	33	'Female'	64	142	1	'Good'	130
8 Wilson	40	'Male'	68	180	0	'Good'	115
9 Moore	28	'Male'	68	183	0	'Excellent'	115
10 Taylor	31	'Female'	66	132	0	'Excellent'	118

Y
Target
Dependent
Response

- Initialize a random weight for each predictor, then…
- Try to reduce error (i.e. ***cost***)
  - You define "***error***".

X	Y
1	\$4
2	\$8
3	\$12

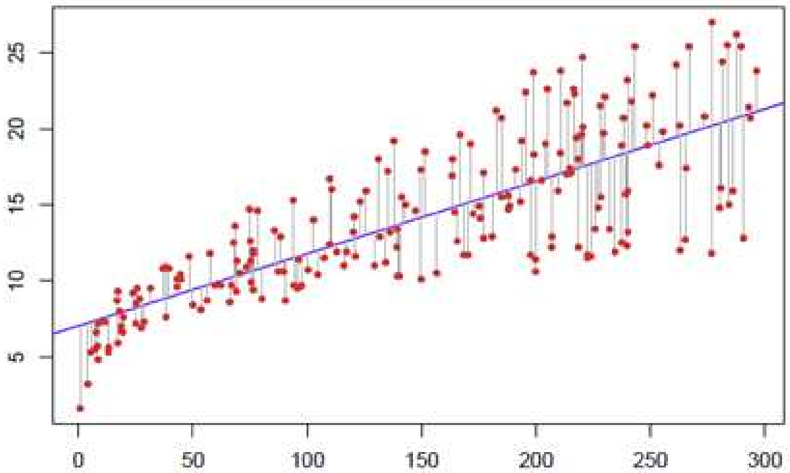
$W_1 = 4$



Linear Regression, $\hat{y} = \theta^T X$

- Linear regression
  1) Build a model by learning coefficients θ as $\hat{y} = \theta^T X$ (column) or $\hat{y} = \theta X^T$ (row)
  2) Verify $\hat{y}$ against Y

$$h(\theta) = \hat{y} = \theta^T X = \theta_0 \times 1 + \theta_1 \times x_1 + \ldots + \theta_n \times x_n$$



$\theta = \begin{bmatrix} 2 \\ 6 \end{bmatrix}$, $\quad X = \begin{bmatrix} 5 & 7 & 8 \\ 1 & 3 & 6 \end{bmatrix}$ $\quad \hat{y} = \theta^T X$

$\theta = \begin{bmatrix} 2 & 6 \end{bmatrix}$, $\quad X = \begin{bmatrix} 5 & 1 \\ 7 & 3 \\ 8 & 6 \end{bmatrix}$ $\quad \hat{y} = \theta X^T$

- Outlier effects?? Prediction quality?? Discrete attributes??

Detailed Info. Returned from Matlab LR

- Matlab **LinearModel** class
  - http://www.mathworks.com/help/stats/linearmodel-class.html

```
X = [1 2 3]';          Y = [5 8 10]';
mdl = fitlm(X, Y)
plot(mdl)      % added variable plot
figure, plotResiduals(mdl, 'fitted')
```

```
regr = linear_model.LinearRegression()
regr.fit(x, y)
```

- **MSE**
- **Residuals,** **.Raw**
- **Fitted**
- **Diagnostics,** **.Leverage** **.CooksDistance**
- **SSE, SST, SSR**
- **Coefficients** **.Estimate**
- **Rsquared**
- **LogLikelihood**

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score

# Load dataset
# Create linear regression object
regr = linear_model.LinearRegression()

# Train the model using the training sets
regr.fit(X_train, y_train)
# Make predictions using the testing set
y_pred = regr.predict(X_test)

# The coefficients
print('Coefficients: \n', regr.coef_ )
# The mean squared error
print("MSE: %.2f" %mean_squared_error(y_test, y_pred))
# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % r2_score(y_test, y_pred))

# Plot outputs
plt.scatter(X_test, y_test, color='black')
plt.plot(X_test, y_pred, color='blue', linewidth=3)

plt.xticks(());        plt.yticks(());     plt.show()
```

lm | lm.Coefficients | lm.Fitted

1x1 LinearModel

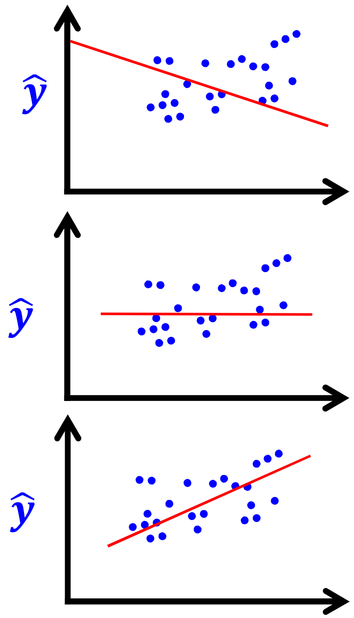
Property	Value	Min	Max
MSE	15.6785	15.6785	15.6785
Robust	[]		
Residuals	100x4 table		
Fitted	100x1 double	9.9068	32.6362
Diagnostics	100x7 table		
RMSE	3.9596	3.9596	3.9596
Steps	[]		
Formula	1x1 classreg.regr.LinearF...		
LogLikelihood	-261.2133	-261.2...	-261.2...
DFE	91	91	91
SSE	1.4267e+03	1.4267...	1.4267...
SST	6.0053e+03	6.0053...	6.0053...
SSR	4.5785e+03	4.5785...	4.5785...
CoefficientCovariance	[2.4820,-8.8216e-04,0.05...	-8.821...	2.4820
CoefficientNames	1x3 cell		
NumCoefficients	3	3	3
NumEstimatedCoefficients	3	3	3
Coefficients	3x4 table		
Rsquared	1x1 struct		
ModelCriterion	1x1 struct		
VariableInfo	3x4 table		
ObservationInfo	100x4 table		
Variables	100x3 table		
NumVariables	3	3	3
VariableNames	3x1 cell		
NumPredictors	2	2	2
PredictorNames	2x1 cell		
ResponseName	'y'		
NumObservations	94	94	94
ObservationNames	0x0 cell		

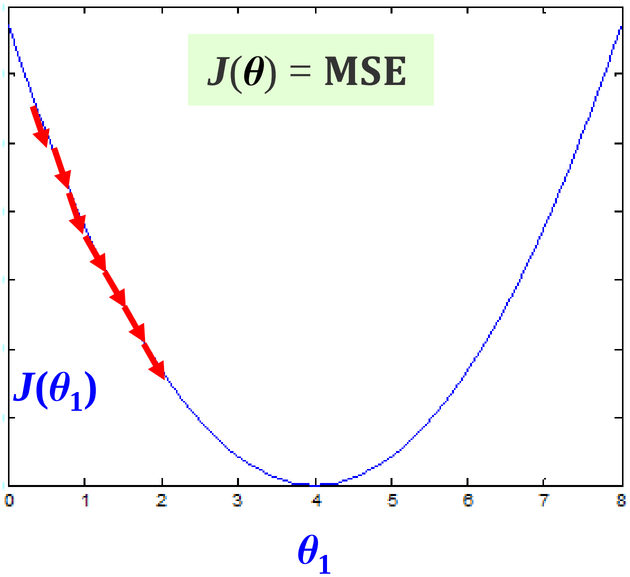
Machine Learning using **Gradient Descent** w/ Learning Rate α

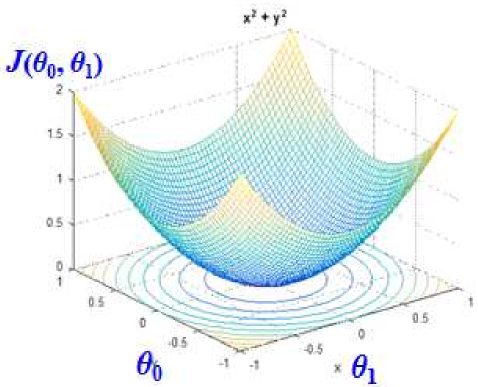
- Minimizing "***ERROR***"!!

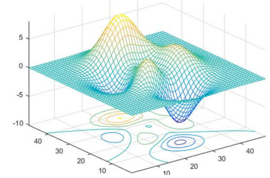
$$\boldsymbol{h}(\theta) = \hat{y} = \theta^{\mathrm{T}}\boldsymbol{X} = \theta_0 \times 1 + \theta_1 \times x_1 + \ldots + \theta_n \times x_n$$

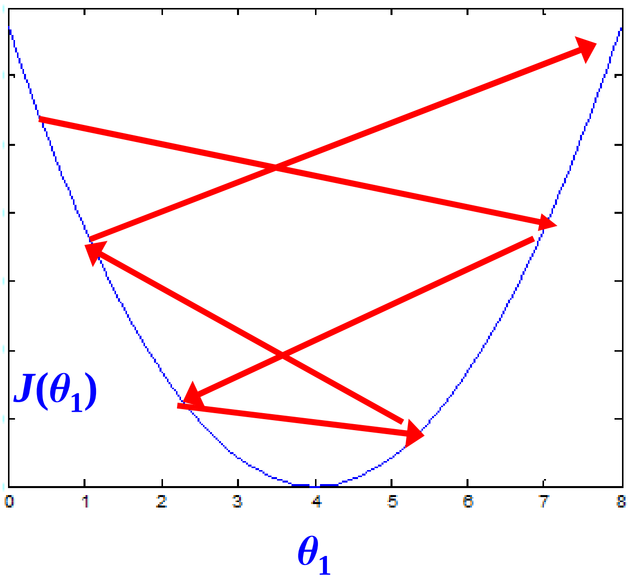
$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$









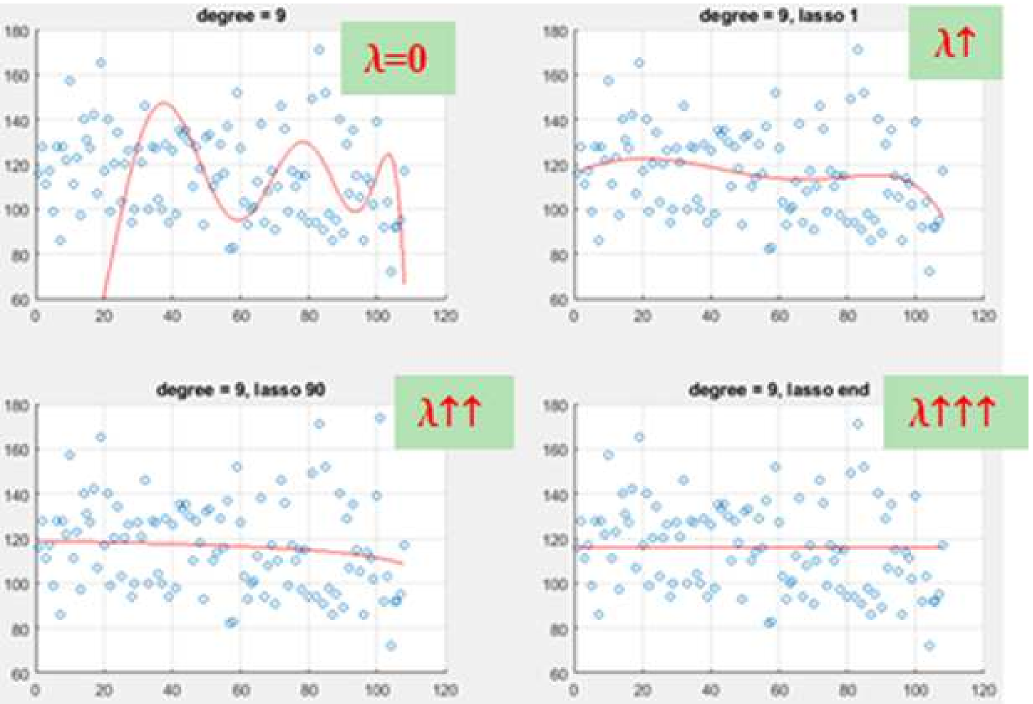


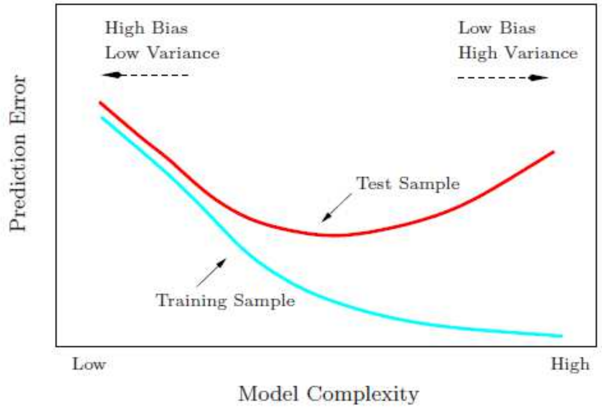
How Do We Avoid Over-Learning?

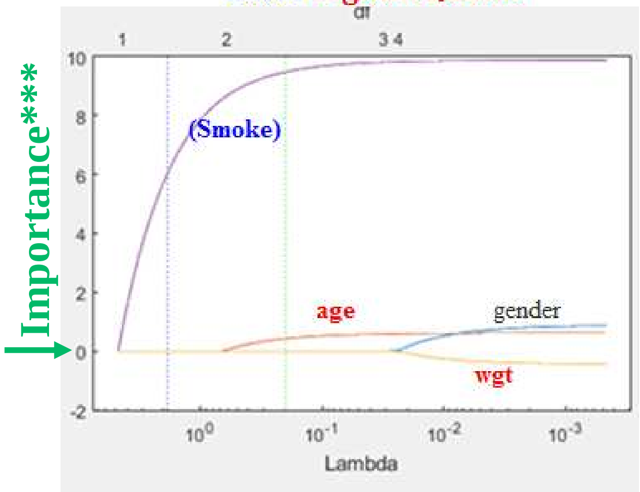
- Avoid overfitting (w/ **simpler** model) but still preserve good prediction accuracy.
  - By identifying really "important" attributes for prediction.

$$\hat{y} = \theta^T X = \theta_0 \times 1 + \theta_1 \times x_1 + \ldots + \theta_n \times x_n$$

- **Lasso** **(and Ridge)** ***regularization***.
  - **Balance error $\Leftrightarrow$ λ × model complexity.**



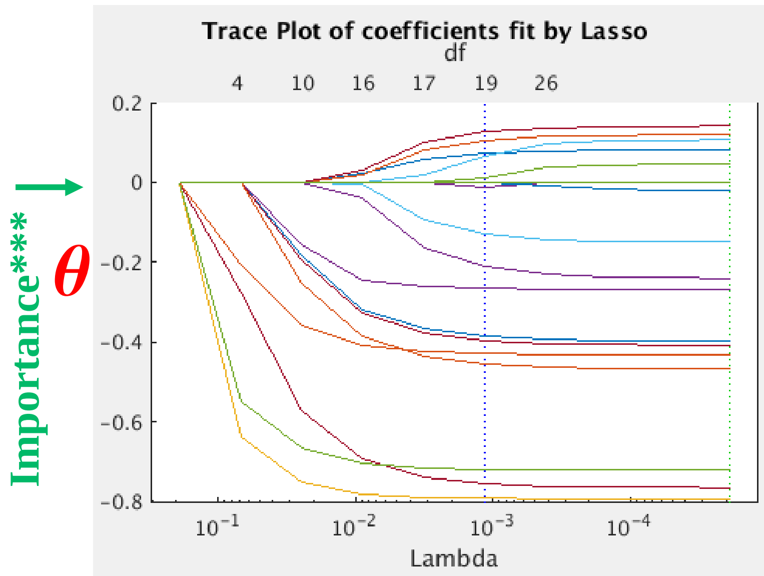




Predicting Class Grades

$$\hat{y} = \theta^{T}\boldsymbol{X} = \theta_0 \times 1 + \theta_1 \times x_1 + \ldots + \theta_n \times x_n$$

- Grades from students.
  - 10 homework assignments. **10%**
  - 1 semester project. **25%**
  - 1 midterm exam. **30%**
  - 1 final exam. **35%**
  - 1 overall weighted score. dependent variable.
- LR prediction will create a model with 13 variables.
  - Hard to understand the final model…
  - By identifying really "important" attributes to be used in the prediction.
  - Lasso and Ridge **regularization**.

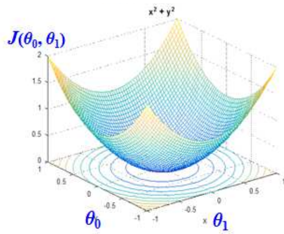


Another example:

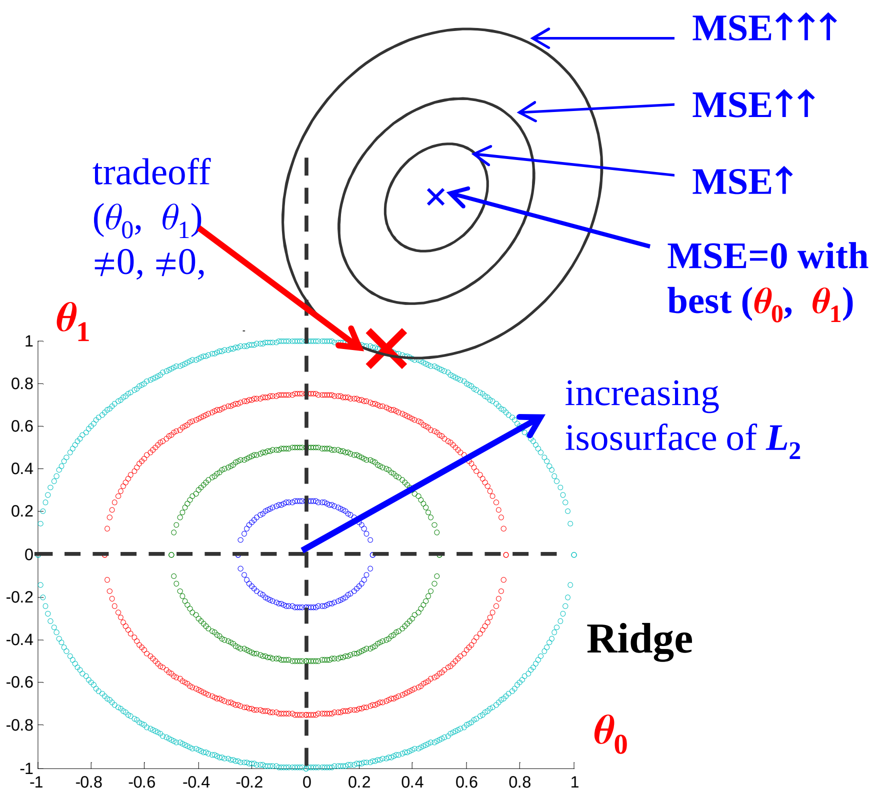
More likely to open a bank account in August.

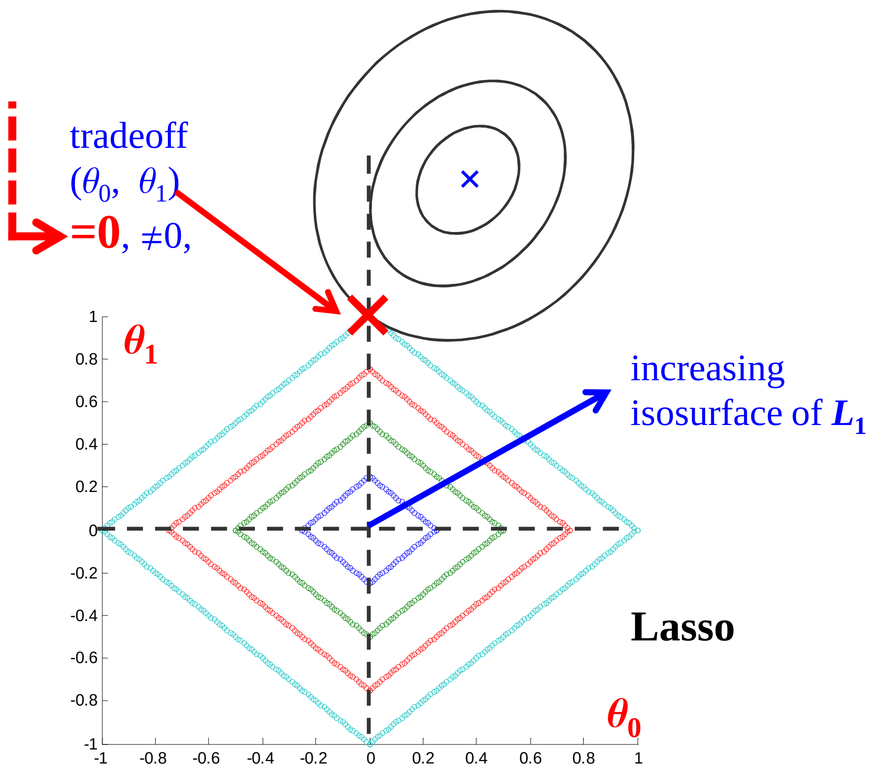
Less likely to open a bank account in September.

Interpreting Lasso and Ridge



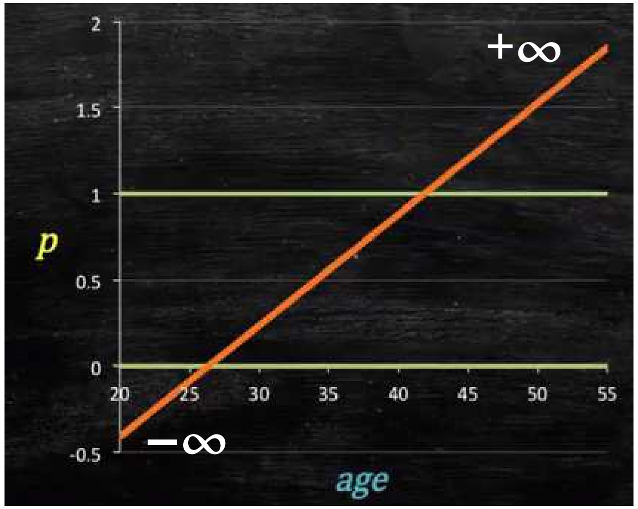
- $\boldsymbol{J(\theta)} = \frac{1}{2m}[(\boldsymbol{Y} - \boldsymbol{h(\theta)})^2 + \lambda\sum_{j=1}^{n}\boldsymbol{\theta_j}^2] = \frac{1}{2m}[(Y - X\theta)^{\mathrm{T}}(Y - X\theta) + \lambda\sum_{j=1}^{n}\boldsymbol{\theta_j}^2]$.
- isosurface of $\boldsymbol{L_p}$ regularizer = $(\sum_{j=1}^{n}\theta_j{}^p)^{1/\boldsymbol{p}}$ or $\|\theta\|_{\boldsymbol{p}}$
- Need to balance out between increasing MSE and increasing regularizer (penalty).
- Lasso
  - More parameters θ will be exactly 0 in L_1 since its isosurface is more protruding.
  - More likely to lead to sparser (simpler) model. Balance between **sparsity** & **convexity**.

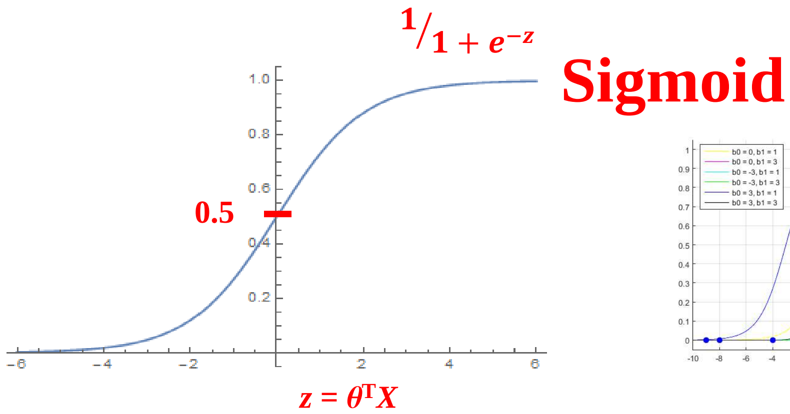


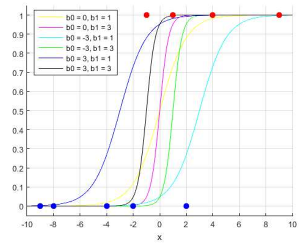


Logistic Regression…Why Not Just Linear Regression?

- Linear Regression $\boldsymbol{h}_\theta(\boldsymbol{x}) = \hat{y} = \boldsymbol{z} = \theta^{\mathrm{T}}\boldsymbol{X} \in [-\infty \,..\, +\infty]$ (i.e. **ANY** range in $\mathbb{R}$).
  - But, classification output must be categorical… Gender, Diseases, Rank, …

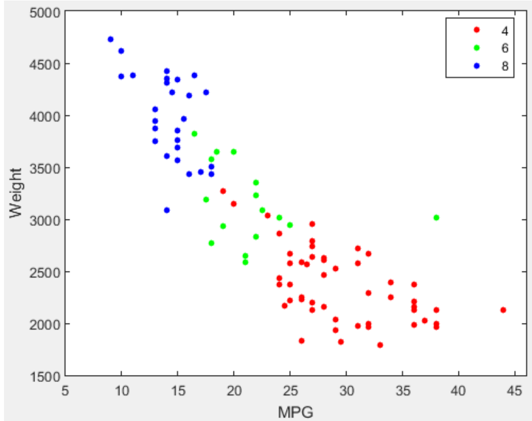




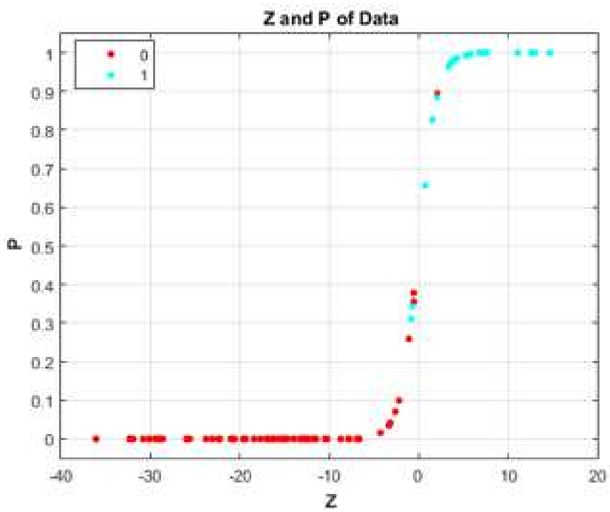


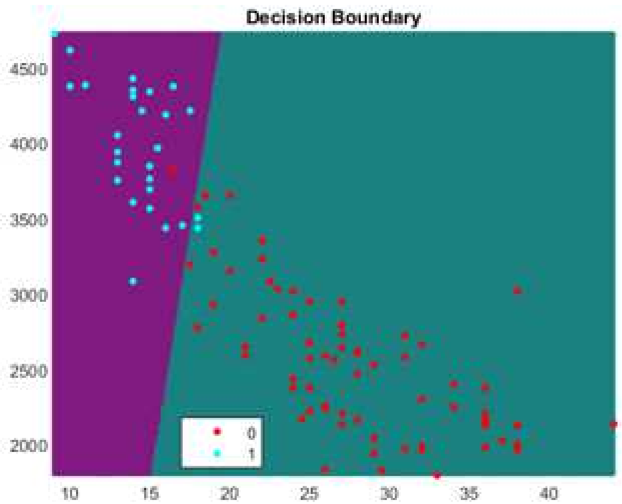
- Logistic (or logit) regression is a model where the dependent var is **categorical**.
  - Logistic regression predicts (output) ***probabilities*** between [0..1].
  - Prediction (i.e. $\theta^{\mathrm{T}}X$) of ***categorical*** dependent var must be linked to probability.
    - $0 \le \boldsymbol{P} = {}^{e^z}/_{1+e^z} = {}^{\mathbf{1}}/_{\mathbf{1}+\boldsymbol{e}^{-\boldsymbol{z}}} \le 1$, where $\boldsymbol{z} = \theta^{\mathrm{T}}\boldsymbol{X}$.

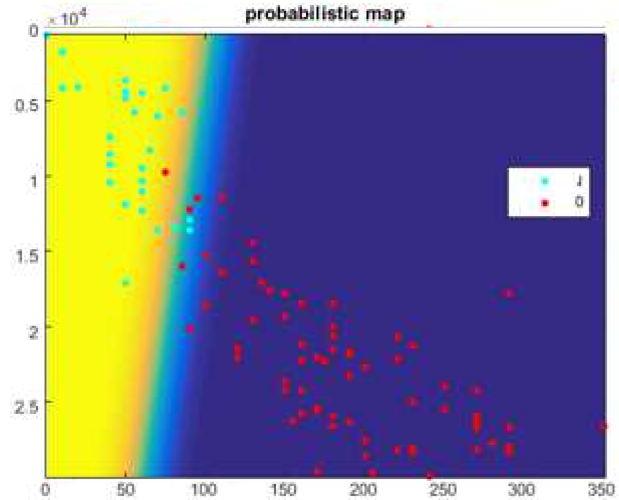
Predict # of Cylinder from MPG + Weight

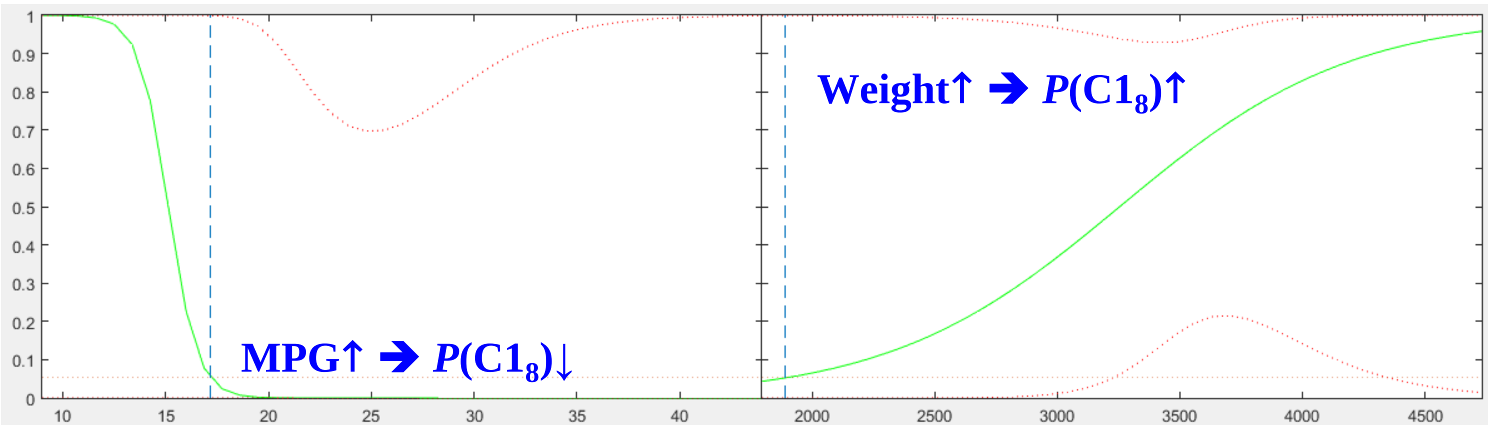


- Fuzzy Probability Boundary
  - Class 0 = cylinder 4 or 6 **(red)**
  - Class 1 = cylinder 8 **(cyan)**





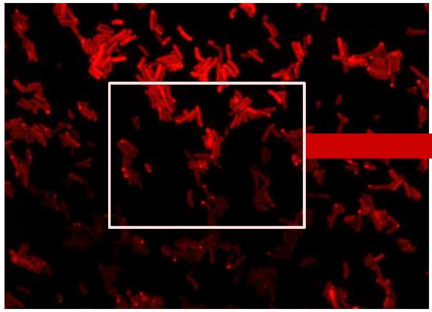




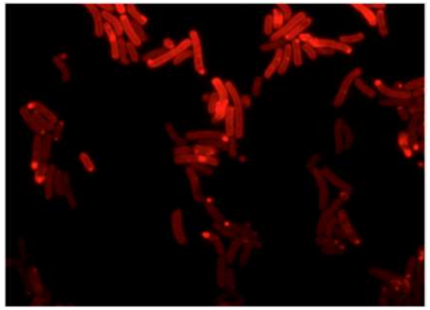
Analyzing Microscope Images for Circadian Rhythms & DNA Compaction

Joint Research by Dr. Chih Lai[2], and Dr. Jayna L. Ditty[1]
Cooper T. Rapp[1], Bethany A. Rhein[1],
Michael J. Grahl[1], Joseph W. Dubis[1],
[1]Department of Biology,
[2]Graduate Program in Software,

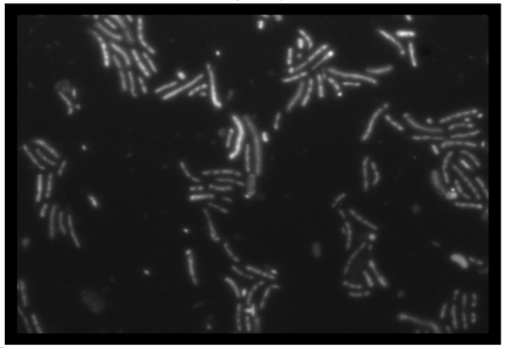
A. ***S. elongatus*** **cell membranes**



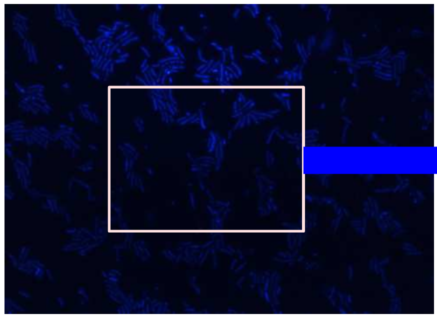
C. ***S. elongatus*** **cell membrane (enlarged image)**



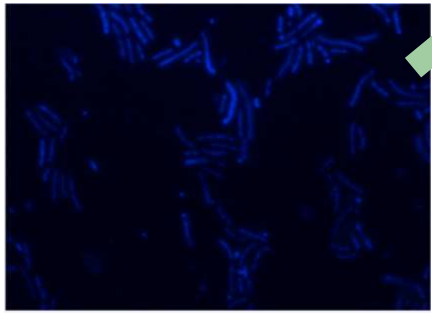
E. **Computer generated composite image**



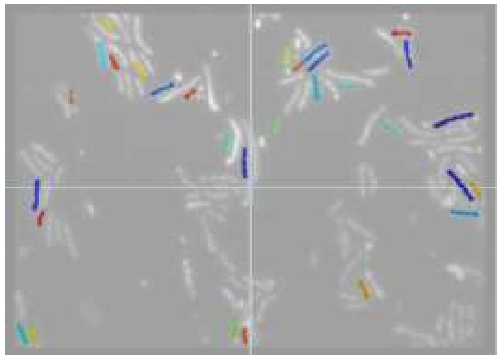
B. ***S. elongatus*** **DNA**



D. ***S. elongatus*** **DNA (enlarged image)**

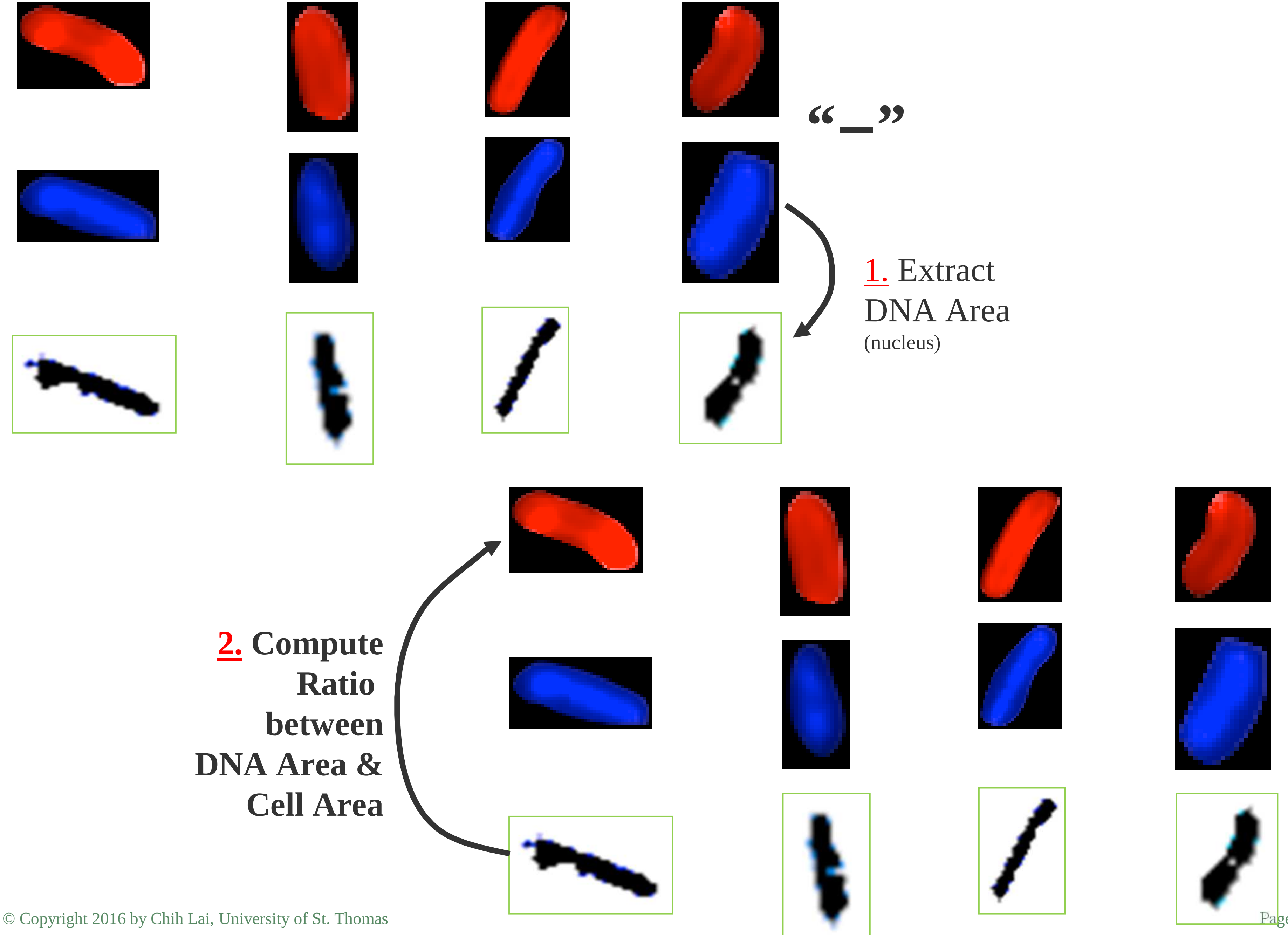


F. **Computer selected cells overlaid on composite image**

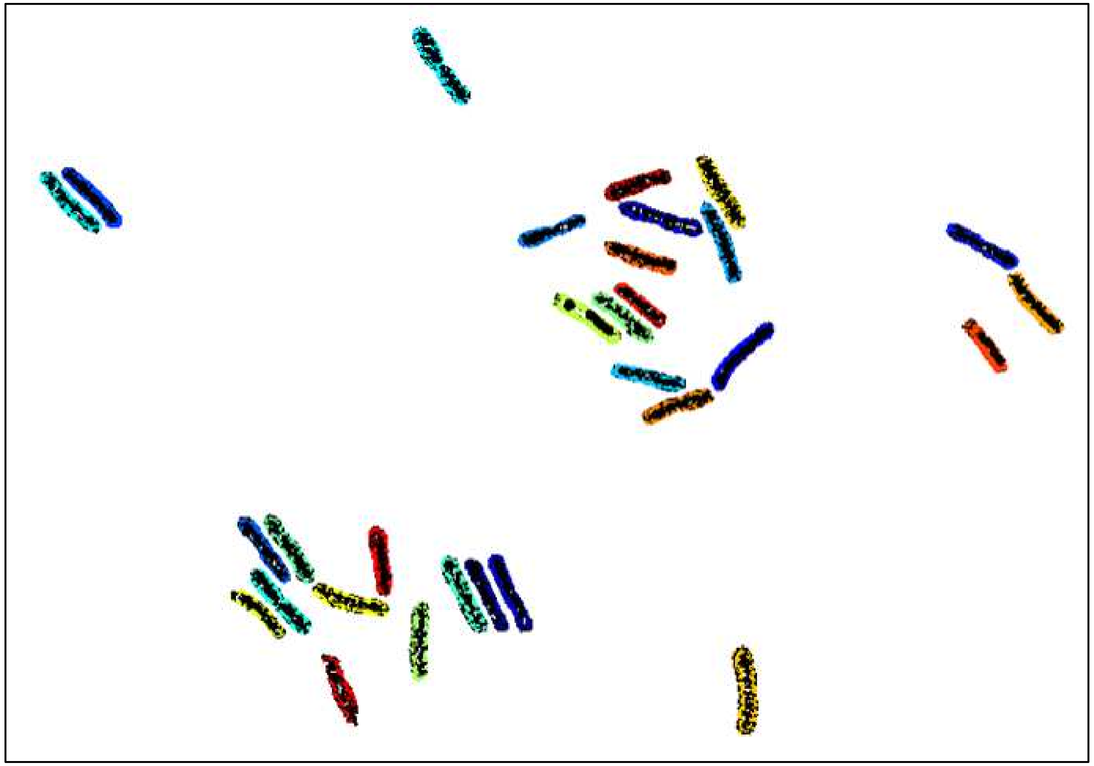


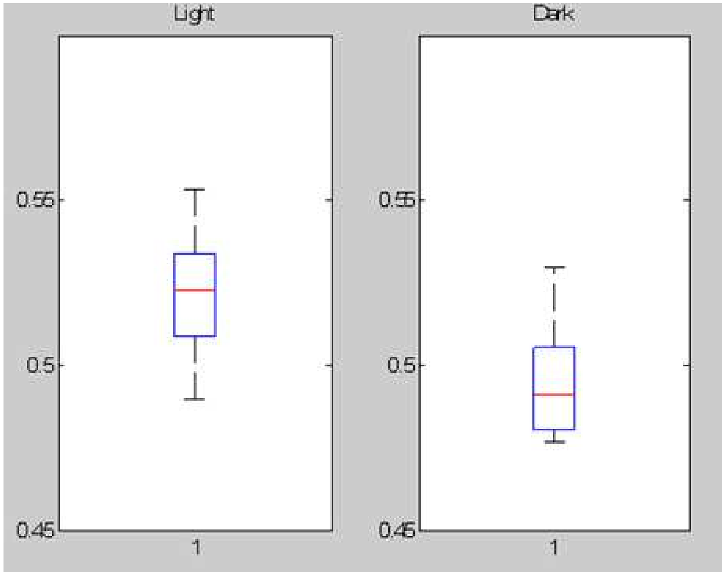
DNA was visualized under the DAPI channel at 100x magnification (excitation at 360 nm and emission at 457 nm).

Zoom in One Cell under DAPI and Rhodamine

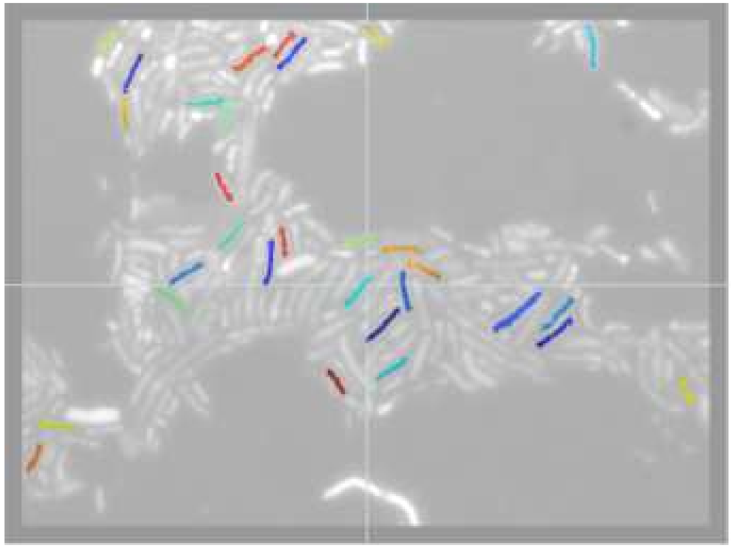


Overlap DNA Areas w/ Cells

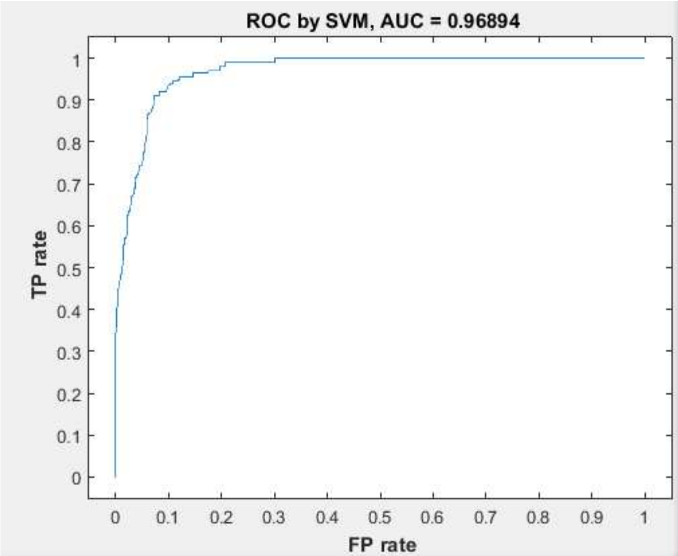


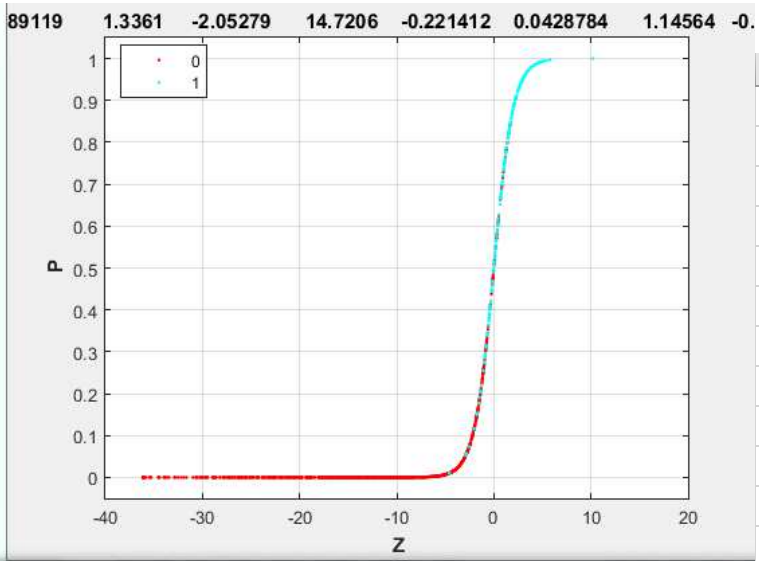


How does a computer automatically select 30 representative cells???

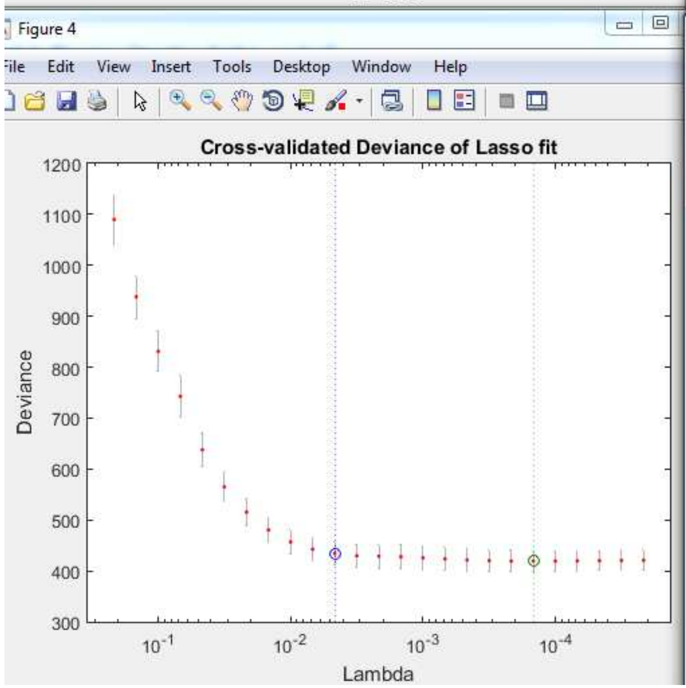


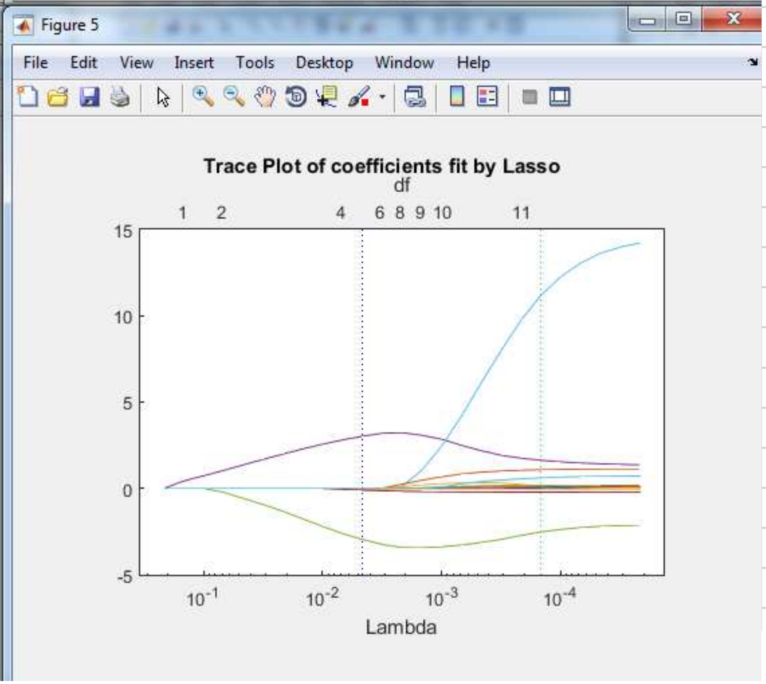
Cell DNA





L	M	N
0.578125	78.591	0
0.608333	39.217	0
0.562637	89.111	0
0.646154	43.832	0
0.252525	90.072	7
0.524775	88.623	0
0.738462	87.192	0
0.46595	99.788	0
0.316872	80.958	0
0.584795	48.141	0
0.813187	31.534	0
0.713287	43.435	0
0.422732	200.18	0
0.281159	80.85	0
0.481481	70.892	0
0.528455	97.059	4
0.227273	139.94	0
0.466905	102.856	9
0.255814	141.272	0
0.502688	82.486	0
0.646154	41.566	0
0.792517	57.279	0
0.834677	69.353	0
0.613636	58.123	0
0.530075	56.361	0
0.466667	42.633	0
0.425882	64.036	0



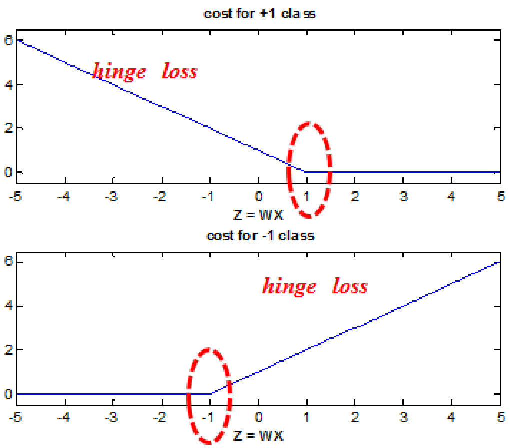


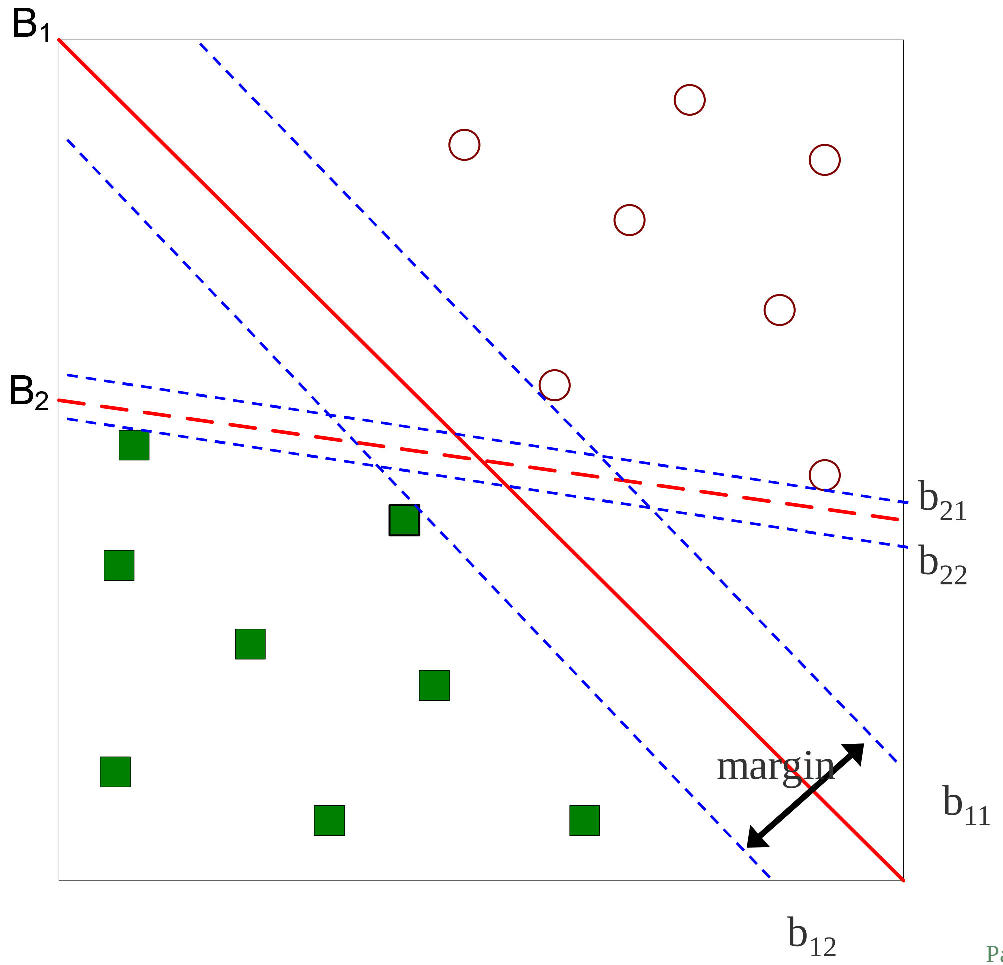
Support Vector Machine (**SVM**) = Large Margin Classifier

- Find the maximum-margin hyperplane that divides points of different classes.
  - Maximize hyperplane margin (to be safe). B1 is better than B2.

- SVM is usually referred to as ***large margin classifier***.

- But, SVM is **<u>more than</u>** just a large margin classifier.
  - Outliers.
  - ∞-dimension.

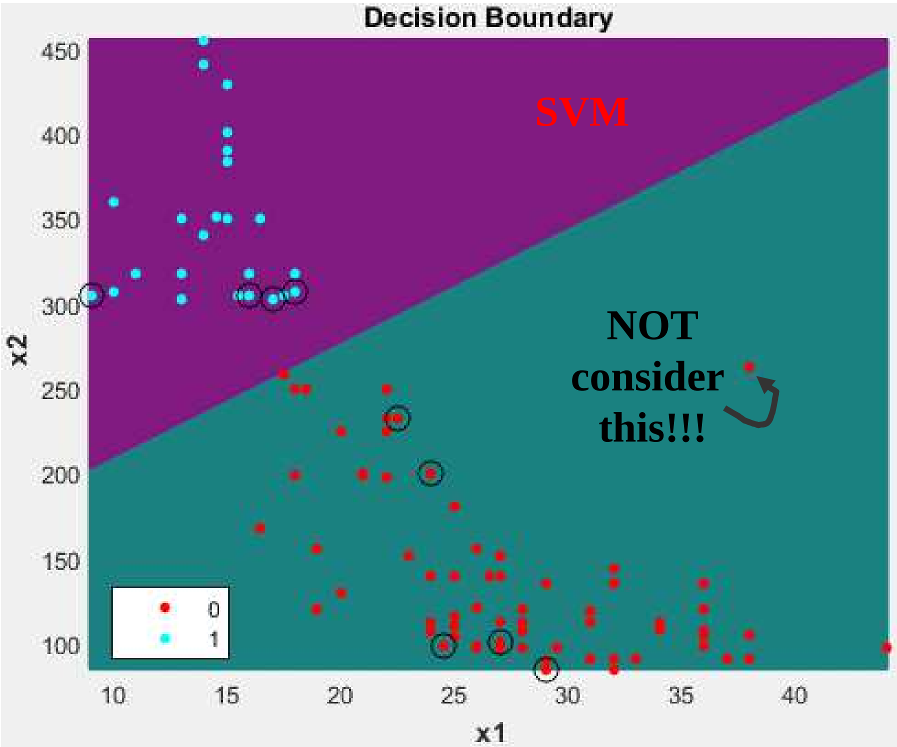


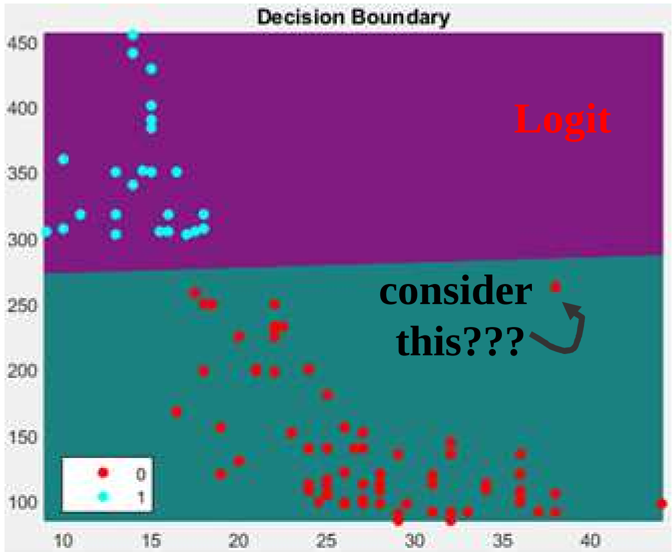


Predict # of Cylinder from MPG + Displacement

- "***Outliers***" (???) Has Much Less Impact in SVM.

- Moving outliers will have NO (or less) impact on SVM prediction.
- SVM prediction will change only if the "support vectors" move.

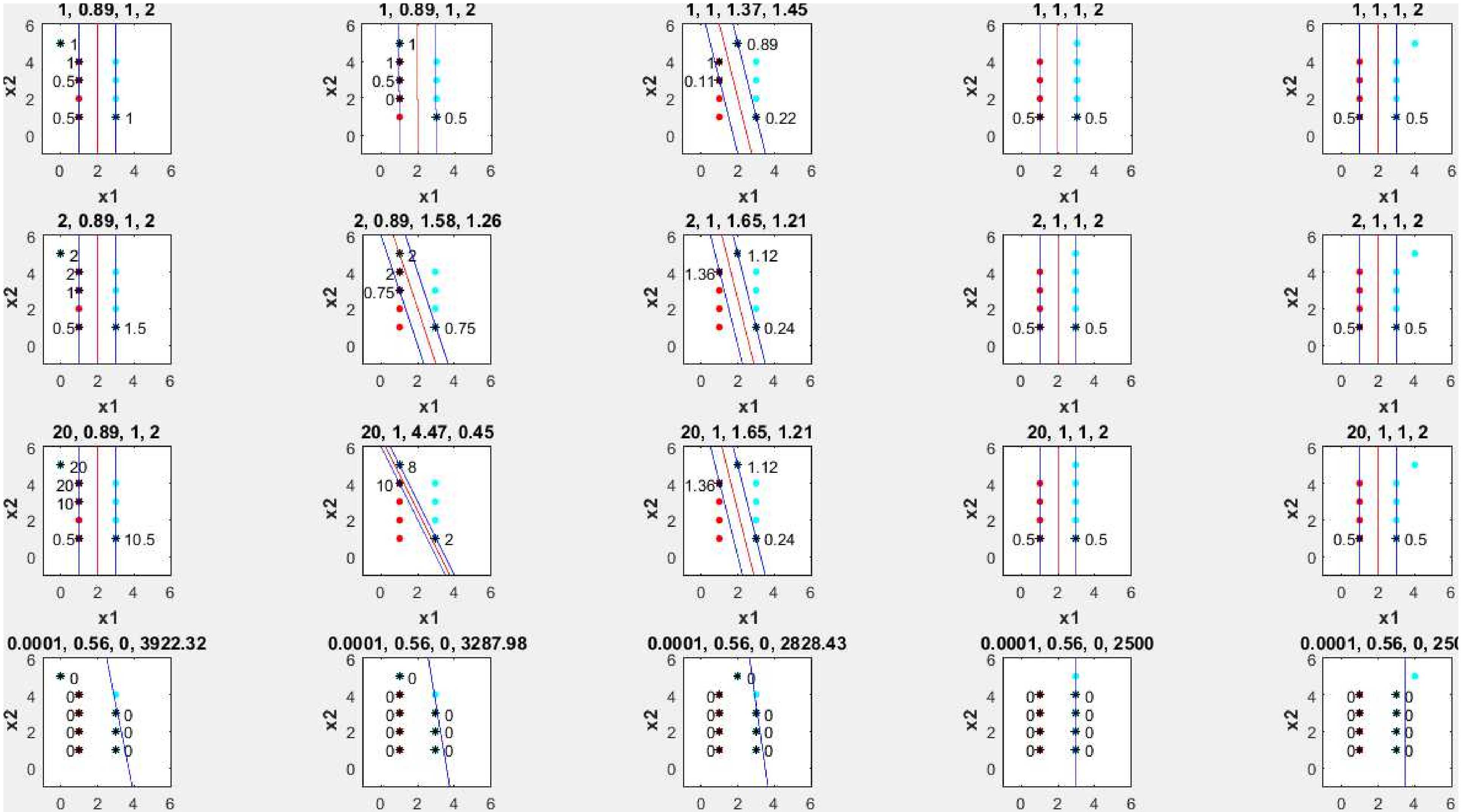




Moving One Point (with Y = 1) Across Decision Boundary, α

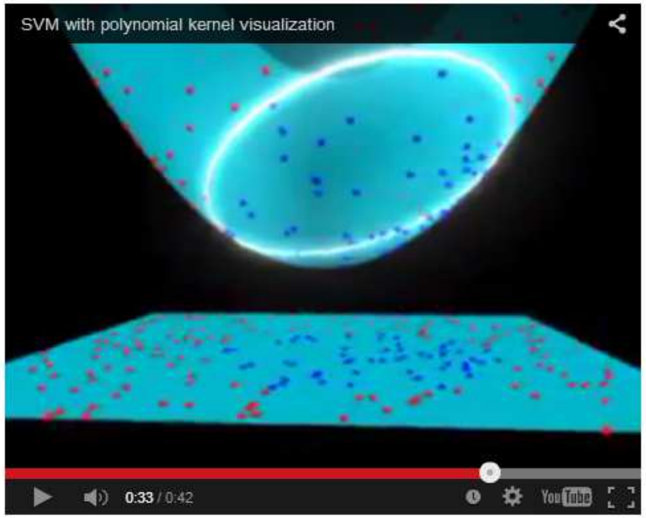
- Title = [C, Accuracy, ||w||, Margin].
- Each point = α.

$$min_{w,b}[\boldsymbol{CE} + \boldsymbol{L}]$$



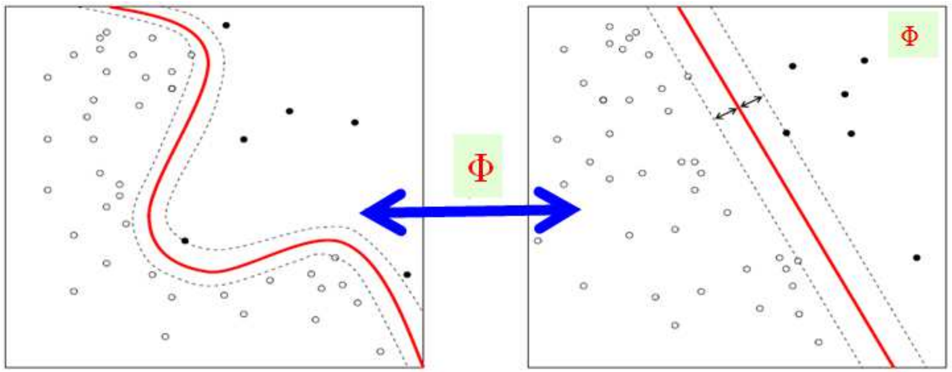
SVM in ∞–Dimensional Space

- SVM in ∞ dimensional space. **WHY???**
  - Transform data from low-D to high-D
  - We can find linear solution in high-D
  - = **non-linear** D.B. in low-D.
    - Linear model = simpler model.
- Dimension↑ ➔ Accuracy↑



https://www.youtube.com/watch?v=3liCbRZPrZA&feature=youtu.be

- **But, computationally more expensive???**
- **Use SVM Kernel!!**

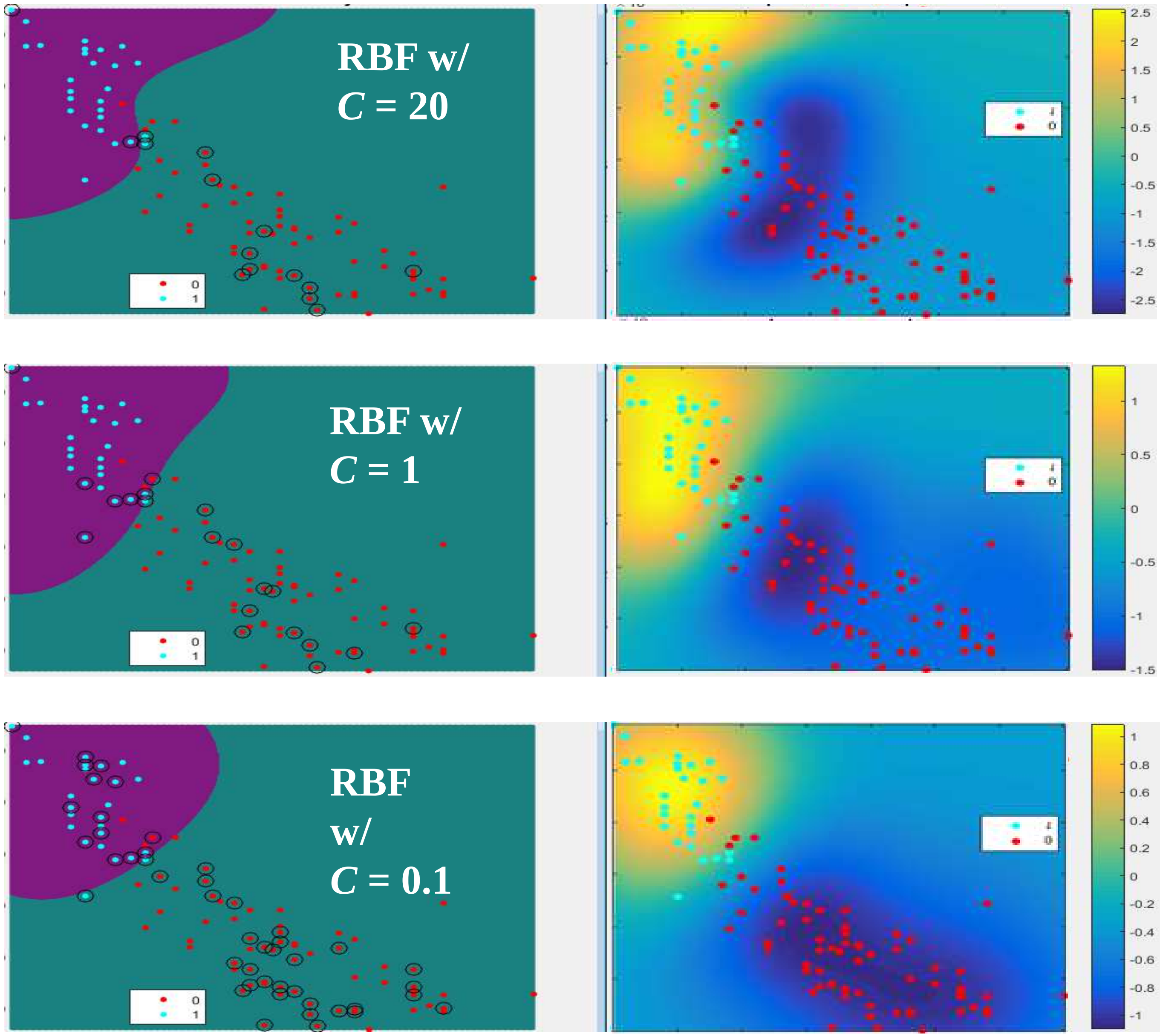


https://en.wikipedia.org/wiki/Support_vector_machine

MPG + Weight, RBF Kernel and Regularization

- Balance between **error ⇔ model complexity**

$min_{w,b}[\boldsymbol{CE} + \boldsymbol{L}]$



LDA (*Linear Discriminant Analysis*) Conceptual View

- Better separation?
  - Projecting data to W_1 line has greatest centroid separation.
    - But, projected data has large overlap due to covariance.
  - Projecting data to W_2 discriminant line (or direction) minimizes the overlapping.

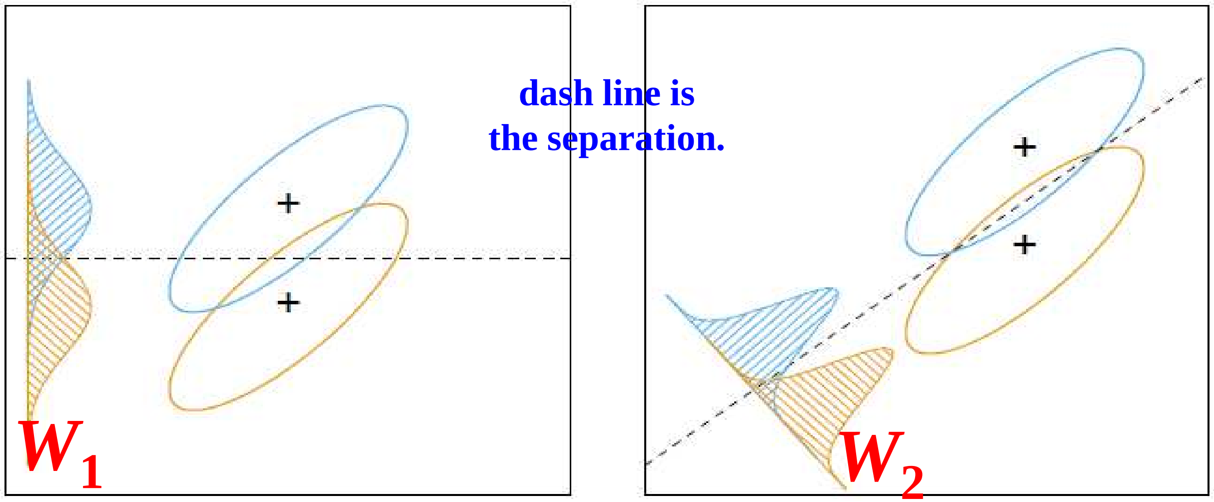
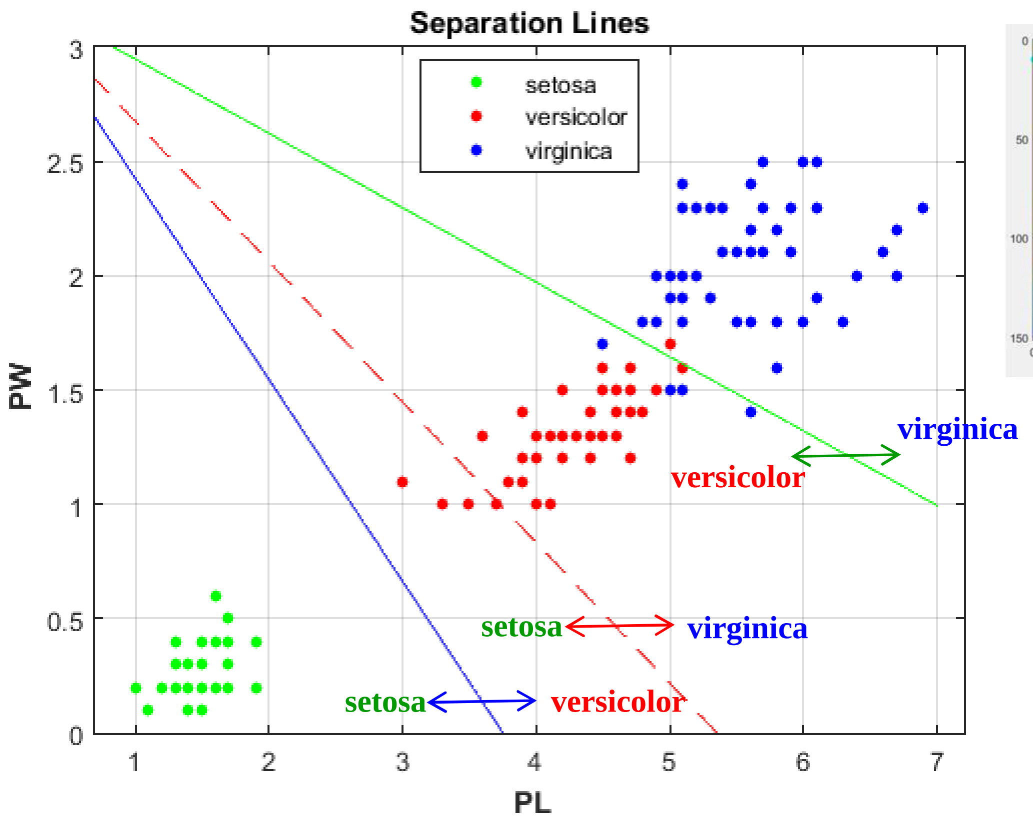


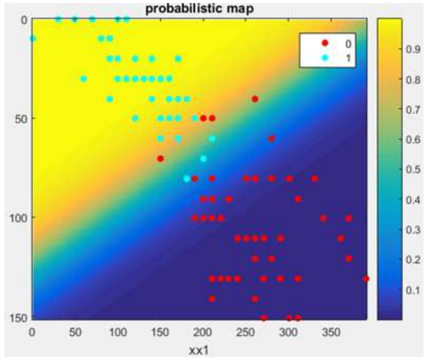
Figure 4.9, P116 of ESL book.

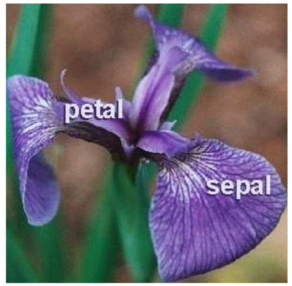
Multiclass Problem under LDA

- One-vs-One classification
  - Turn the problem with ***n*** classes into ***n*(*n*-1) / 2** separated binary-classification problems.
  - It creates ***n*(*n*-1) / 2** models, where ***n*** = 3 in this case.

- One-vs-All (or one-vs-rest) classification.
  - Turn the problem with ***n*** classes into ***n*** separated binary-classification problems.







http://oranchak.com/?p=535

iris virginica iris versicolor iris setosa

Outline

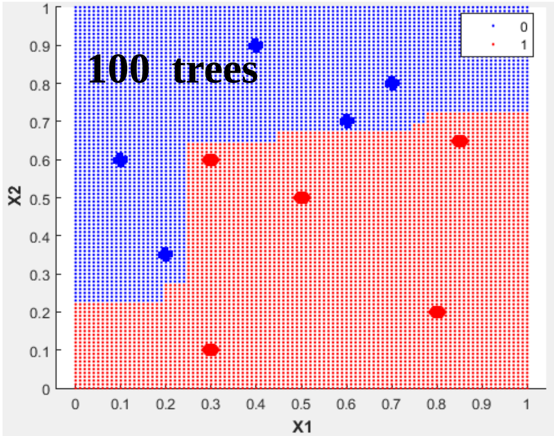
- Proliferation of Data (Structured and Unstructured) / Internet of Things.
  - Just manually query all the data? Descriptive Analytics?

- Machine Learning Tools.

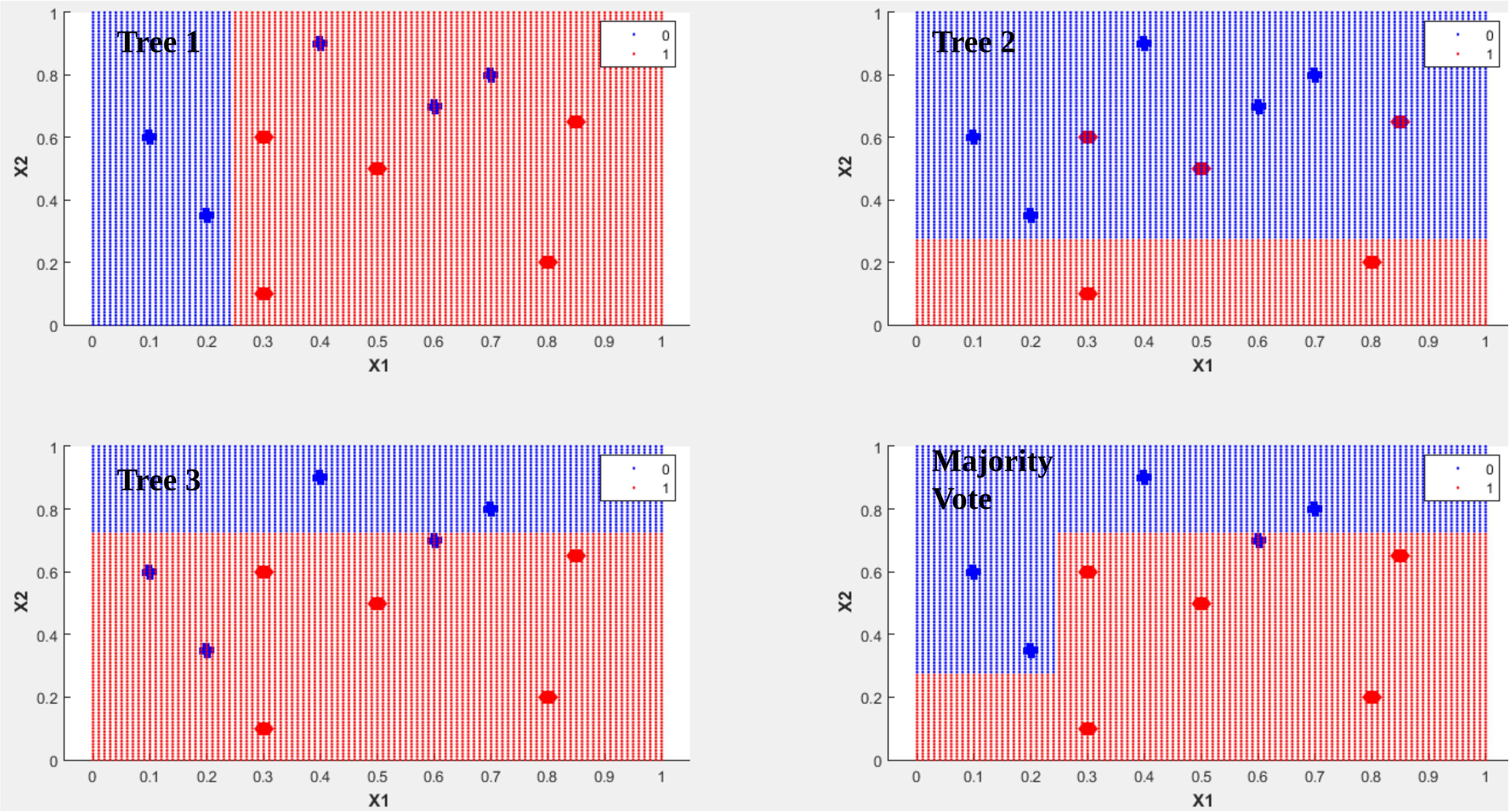
- Major topics in ML.
  - Linear Regression, Logistic Regression, **Linear Discriminant Analysis (LDA)**.
  - Gradient Descent, Regularization & Overfitting.
  - Support Vector Machine (SVM) and Kernel Methods.
  - Ensemble, Boosting / Bagging, Quality Evaluation
  - Self-Organized Map (SOM), **PCA, ICA, SVD, MDS**.
  - Collaborative Filtering, Matrix Factorization, (Recommender System), Gaussian Mixture Model.

- Machine Learning Computing Platform.

Ensemble (Bagging) Learning

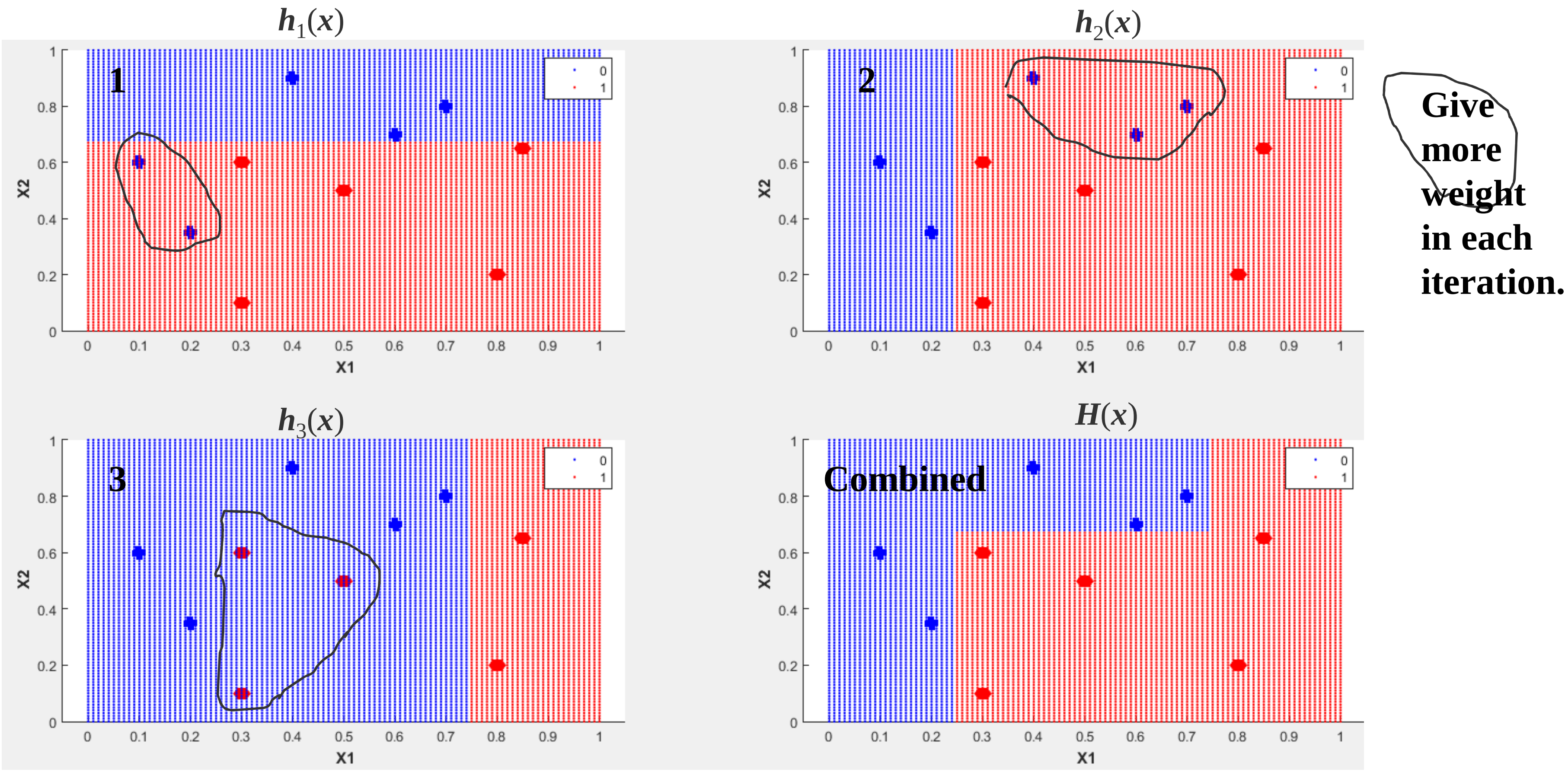


- Combine multiple weak models ➔ stronger predictor.
  - Create 3 trees with **depth 1**.

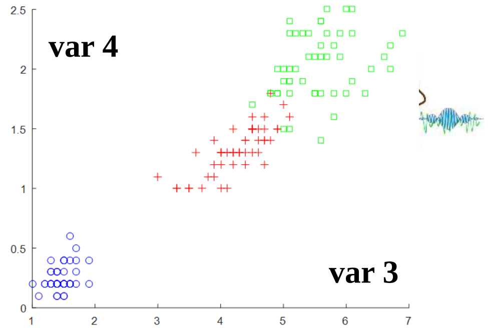


AdaBoosting– *Adaptive Boosting*

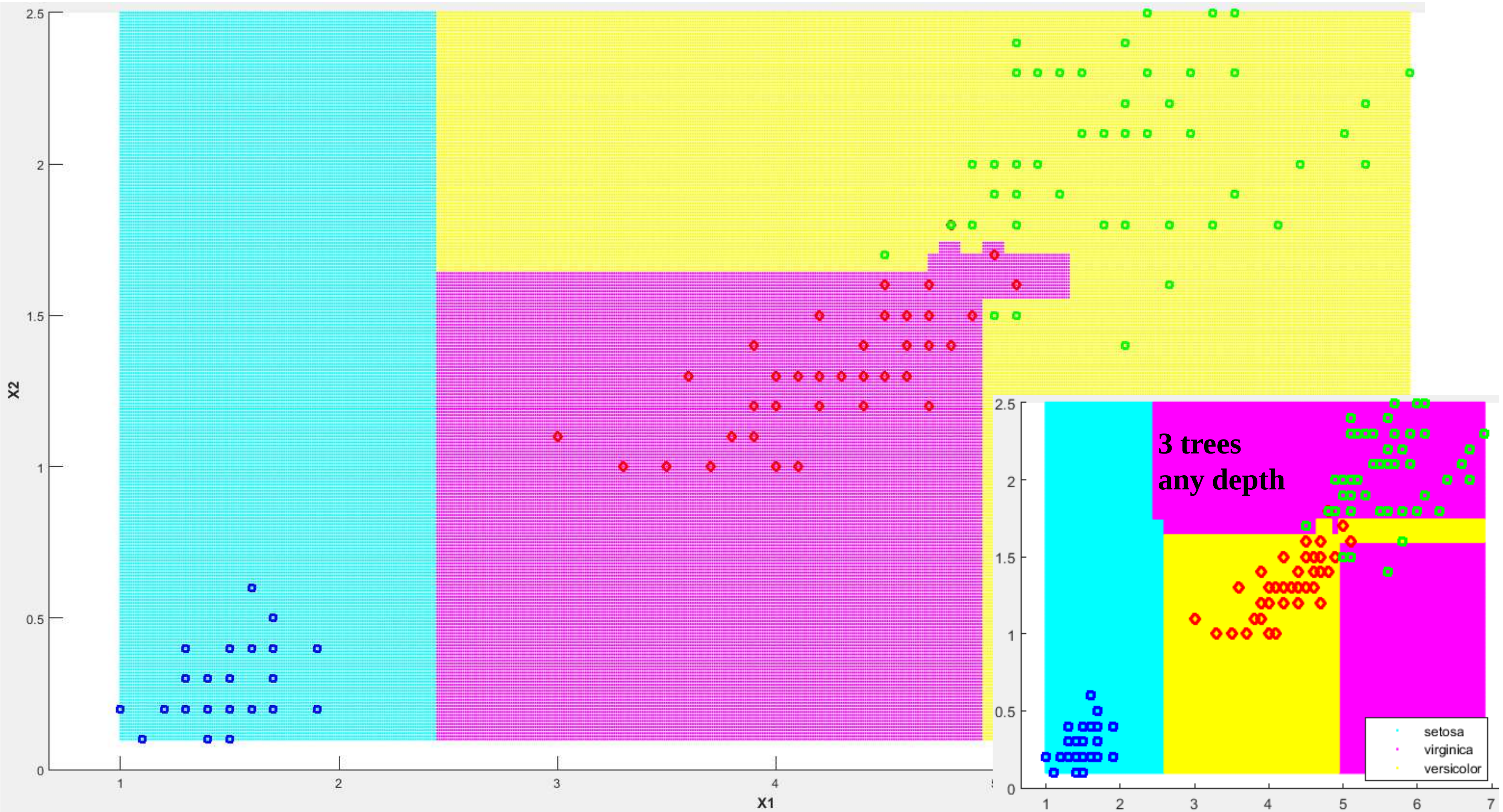
- AdaBoosting for a simple dataset.
  - **Iteratively** create 3 trees with depth 1.
  - Later models give more weights on previously misclassified data to try to predict them correctly.



Bagging, Iris Multiple Classes, 100 Trees (any depth)




```
load fisheriris
Mdl_All = fitensemble(meas, species, 'Bag', 100, 'Tree', 'Type', 'Classification')
flower = predict(Mdl,mean(meas))
```



Project

Process

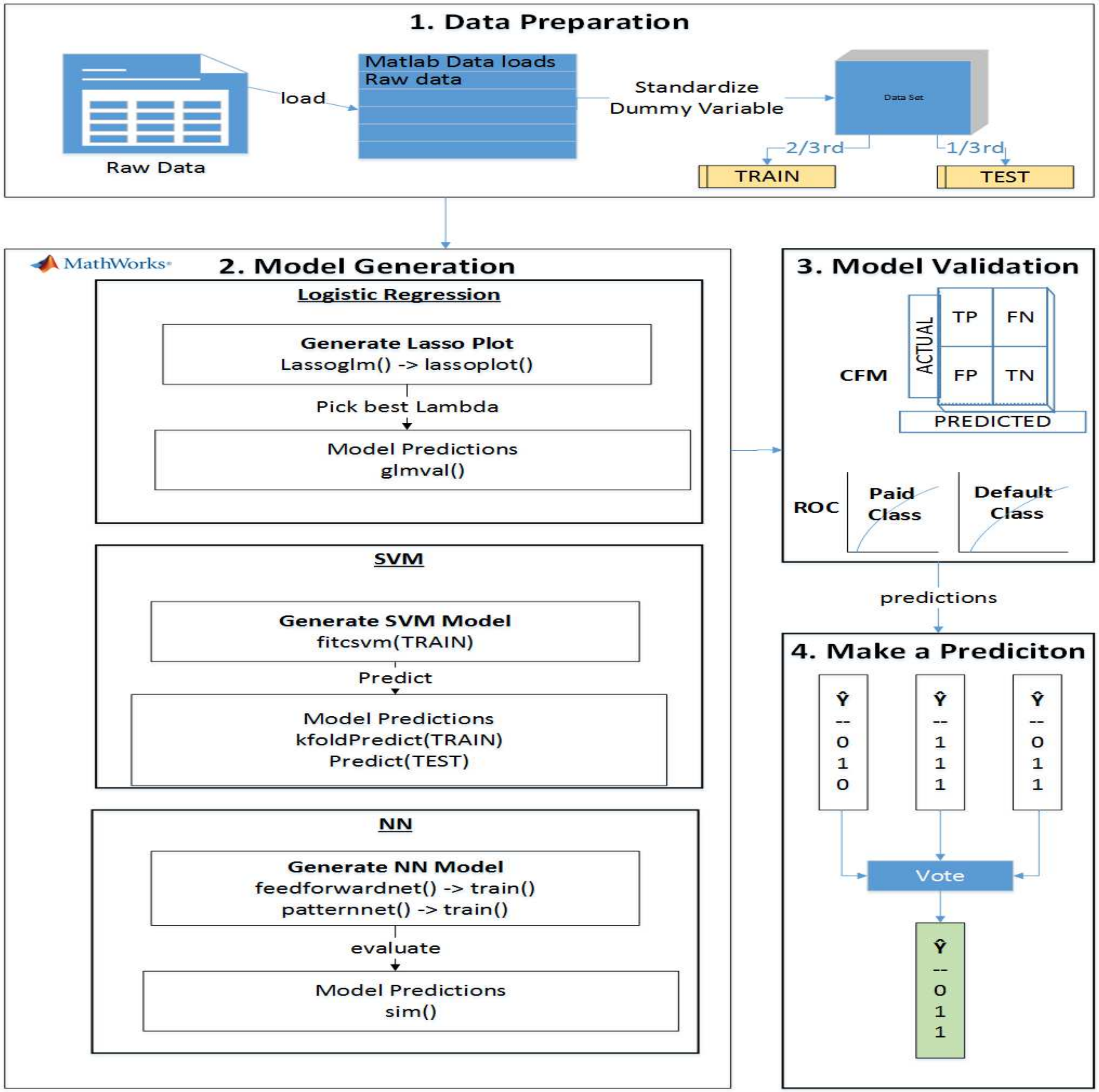
Flow

Final joint vote

v.s.

each method along

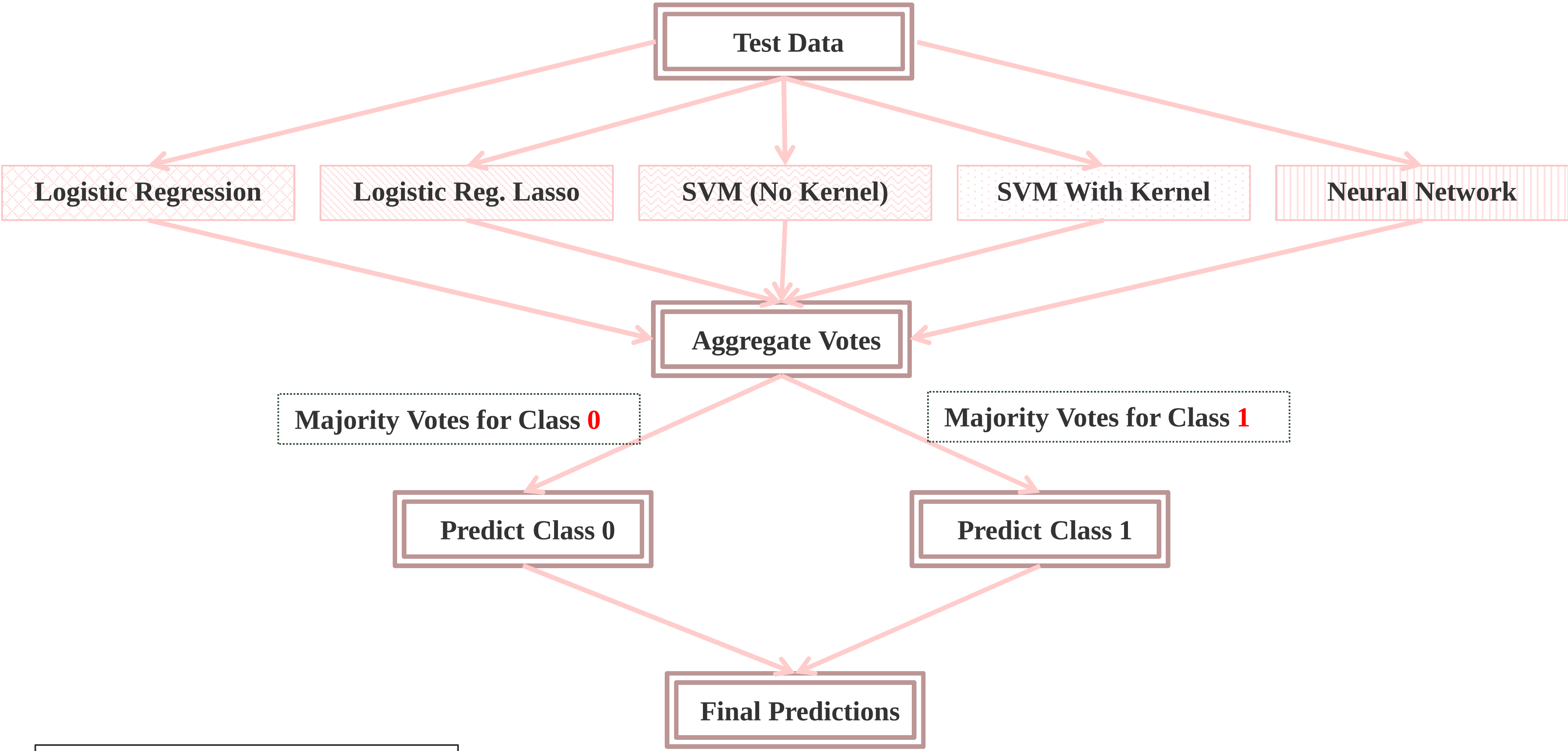
(**NOT** just compare performance of different methods)



2017 Fall

Radhika Bodapatla
Bhanu Pittampally
David Young
Kyle Thompson
Kehinde Kalejaiye

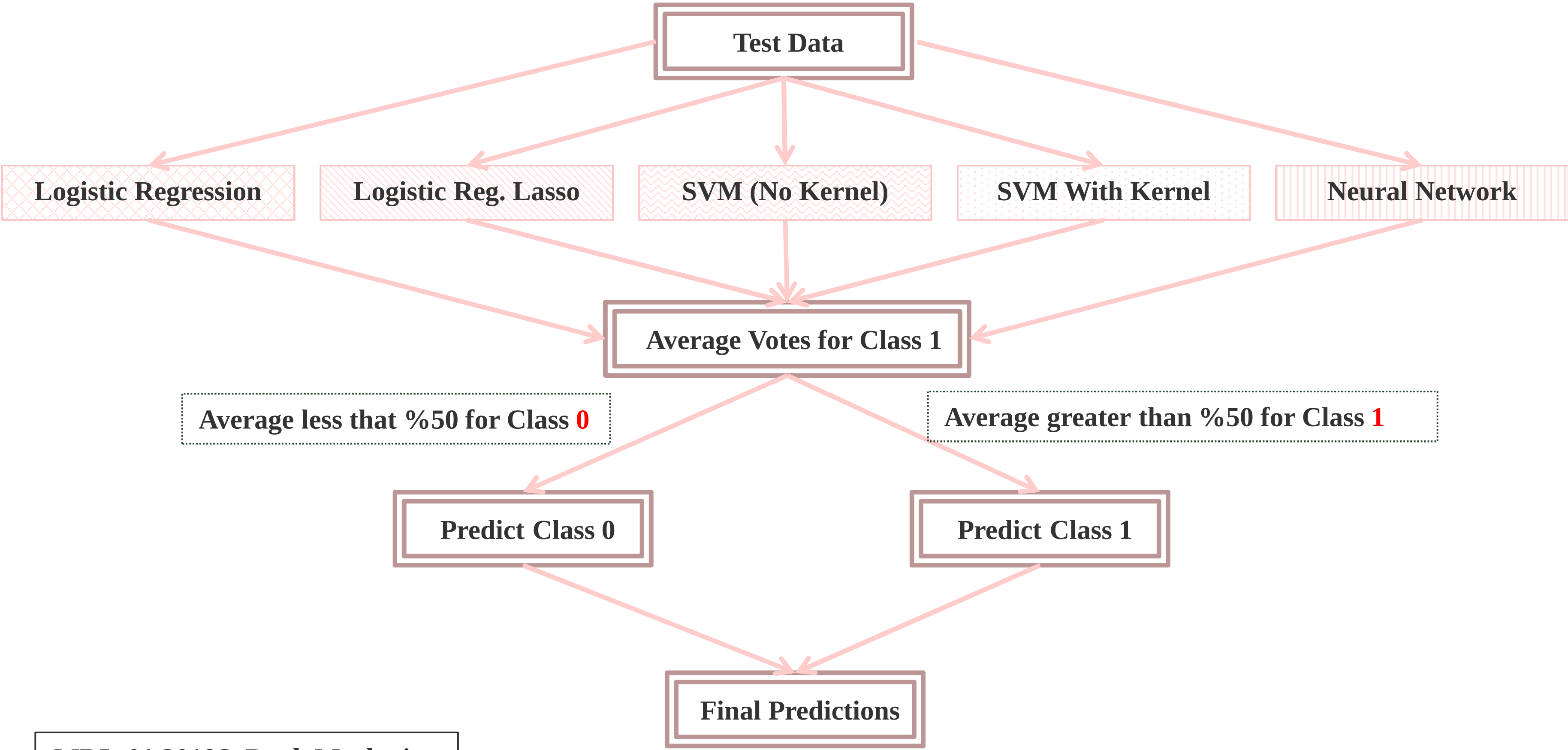
Ensemble Hard Voting



MDL-01 2018S, Bank Marketing

Terrence White
Ronald E Twite
Leela Sowjanya Chippada
Ahmad K Lubnani
Mowlid Abdillahi
Nathan Adams

Ensemble Soft Voting



MDL-01 2018S, Bank Marketing

Terrence White
Ronald E Twite
Leela Sowjanya Chippada
Ahmad K Lubnani
Mowlid Abdillahi
Nathan Adams

Outline

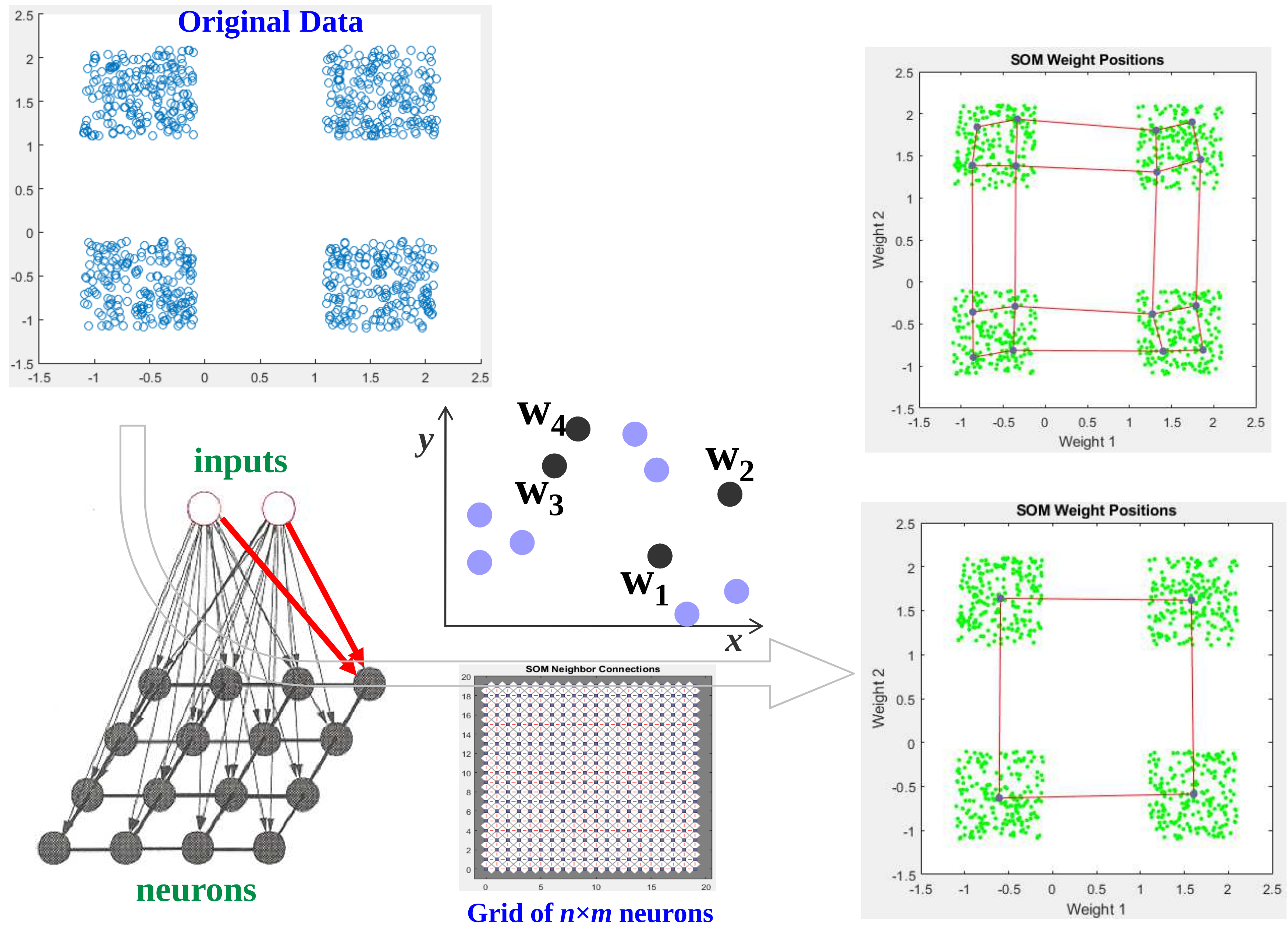
- Proliferation of Data (Structured and Unstructured) / Internet of Things.
  - Just manually query all the data? Descriptive Analytics?

- Machine Learning Tools.

- Major topics in ML.
  - Linear Regression, Logistic Regression, **Linear Discriminant Analysis (LDA)**.
  - Gradient Descent, Regularization & Overfitting.
  - Support Vector Machine (SVM) and Kernel Methods.
  - Ensemble, Boosting / Bagging, Quality Evaluation
  - Self-Organized Map (SOM), **PCA, ICA, SVD, MDS**.
  - Collaborative Filtering, Matrix Factorization, (Recommender System), Gaussian Mixture Model.

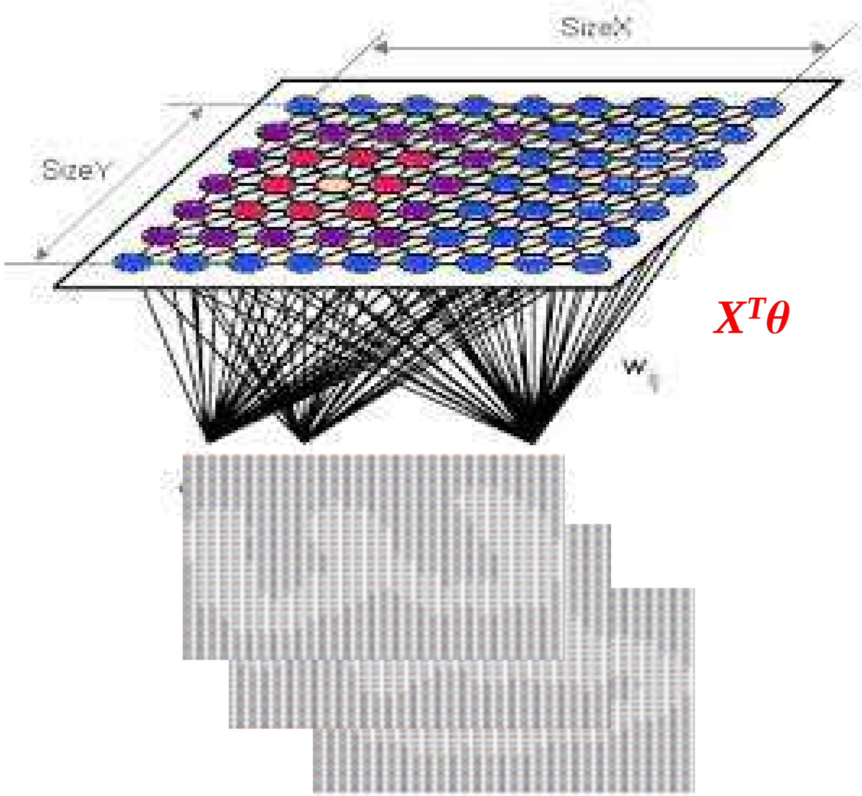
- Machine Learning Computing Platform.

Unsupervised Neural Network, Self Organized Map (SOM)



<u>**Unsupervised**</u> Neural Network for Digits Recognition

- Self Organized Map (**SOM**)
  - 1934 handwritten digits from 0 to 9 collected from a total of 30 people.
  - Each digit sample has been normalized to 32x32 binary image
  - See also at http://www.codeproject.com/Articles/793537/Self-organizing-Map-Implementation

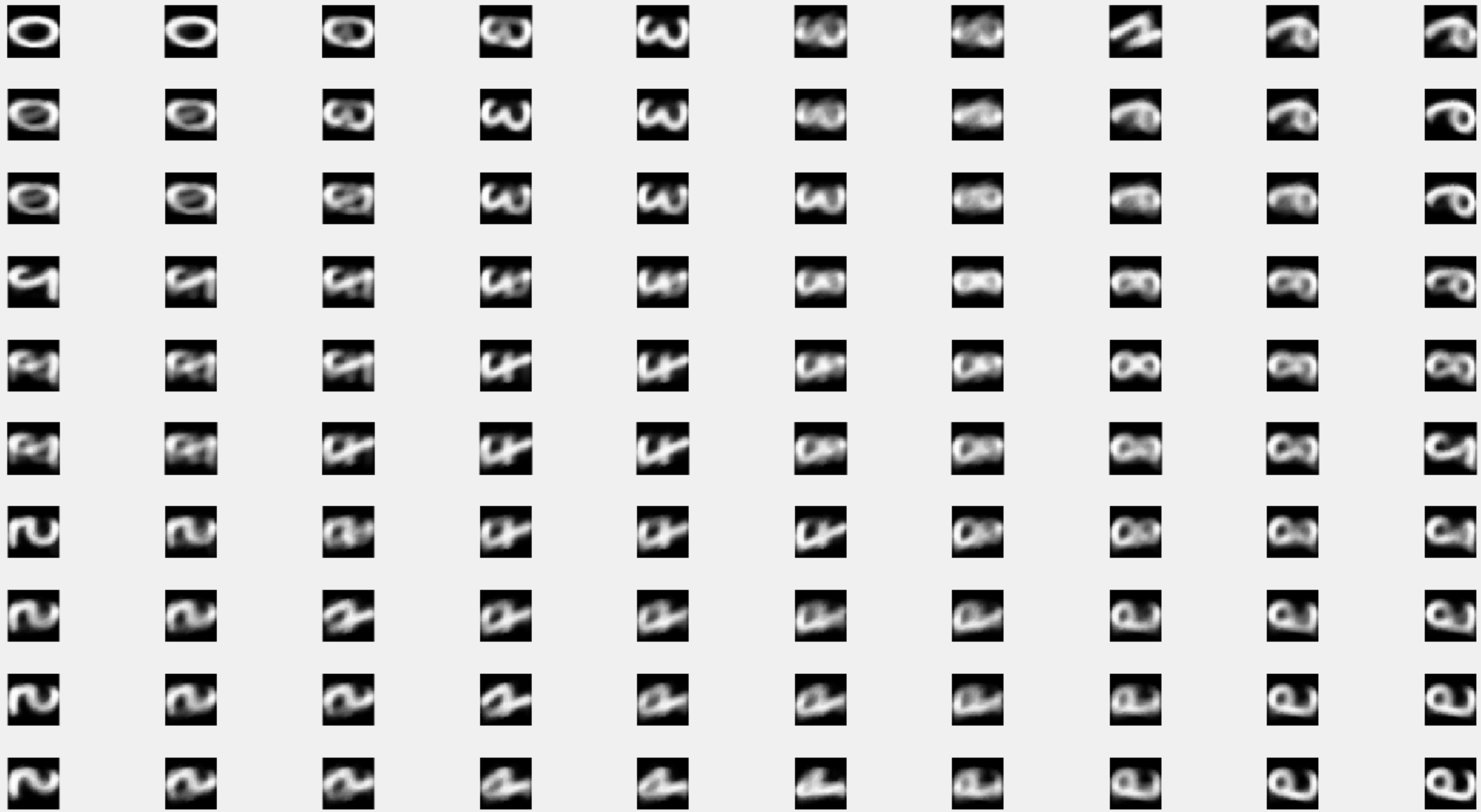


$X^T\theta$

How Does The Trained SOM Look Like?

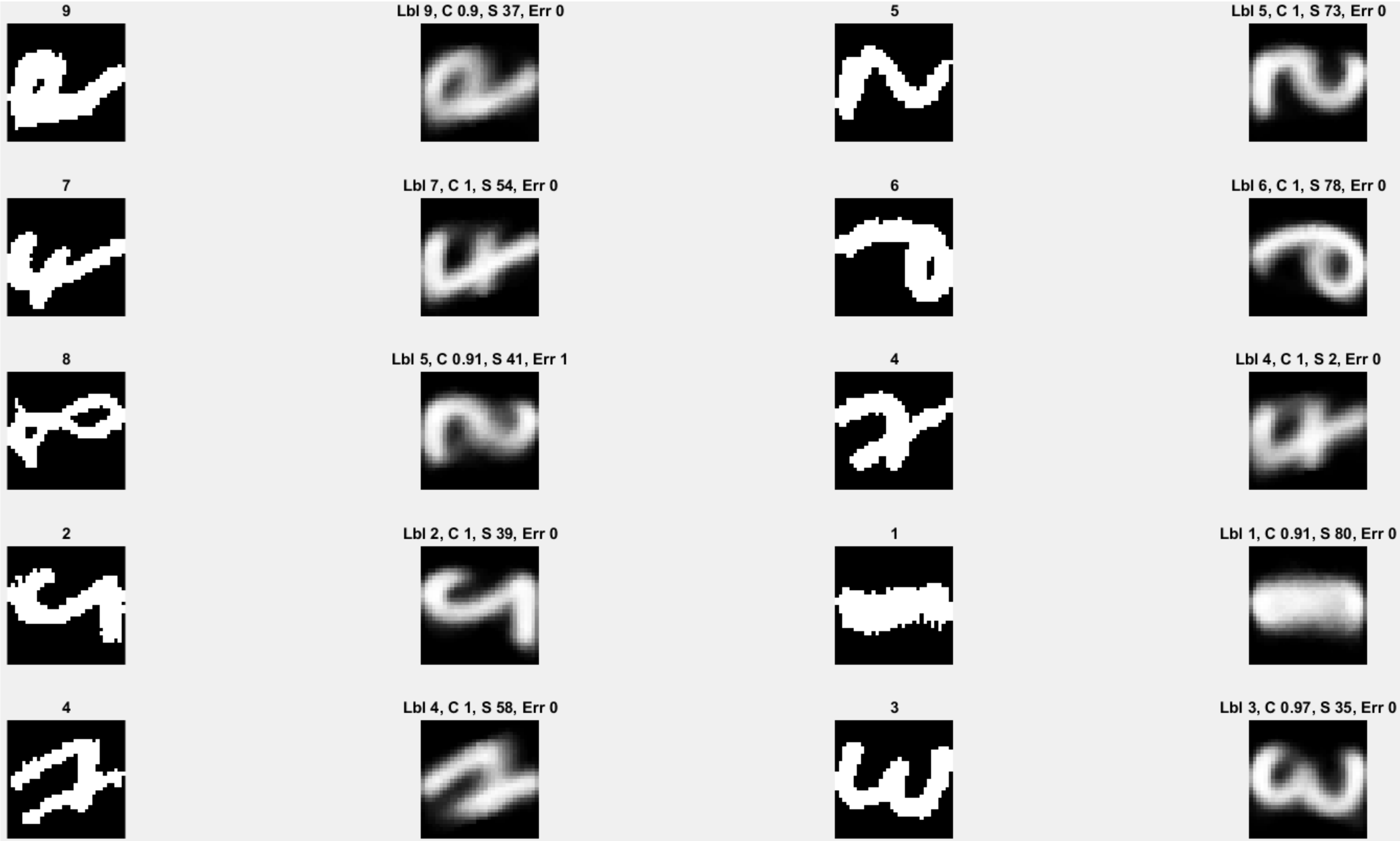
```
IW_Array = net.IW{1};
figure,
for i = 1 : 100,
   img=IW_Array(i,:);
   subplot(10, 10, i),
   imshow(reshape(img, [32 32]));
end
```

- Input weights on 10×10 SOM neurons (with 50 epochs).
  - Each neuron was trained by1024 (32×32) input pixels from all trainings.



Use SOM Clustering Results for Classification

- In this test, accuracy = 90%.



Distracted Driver Detection

- Classify driver’s behavior into one of the 10 classes.

c0: safe driving

c1: texting - right

c2: talking on the phone - right

c3: texting - left

c4: talking on the phone - left

c5: operating the radio

c6: drinking

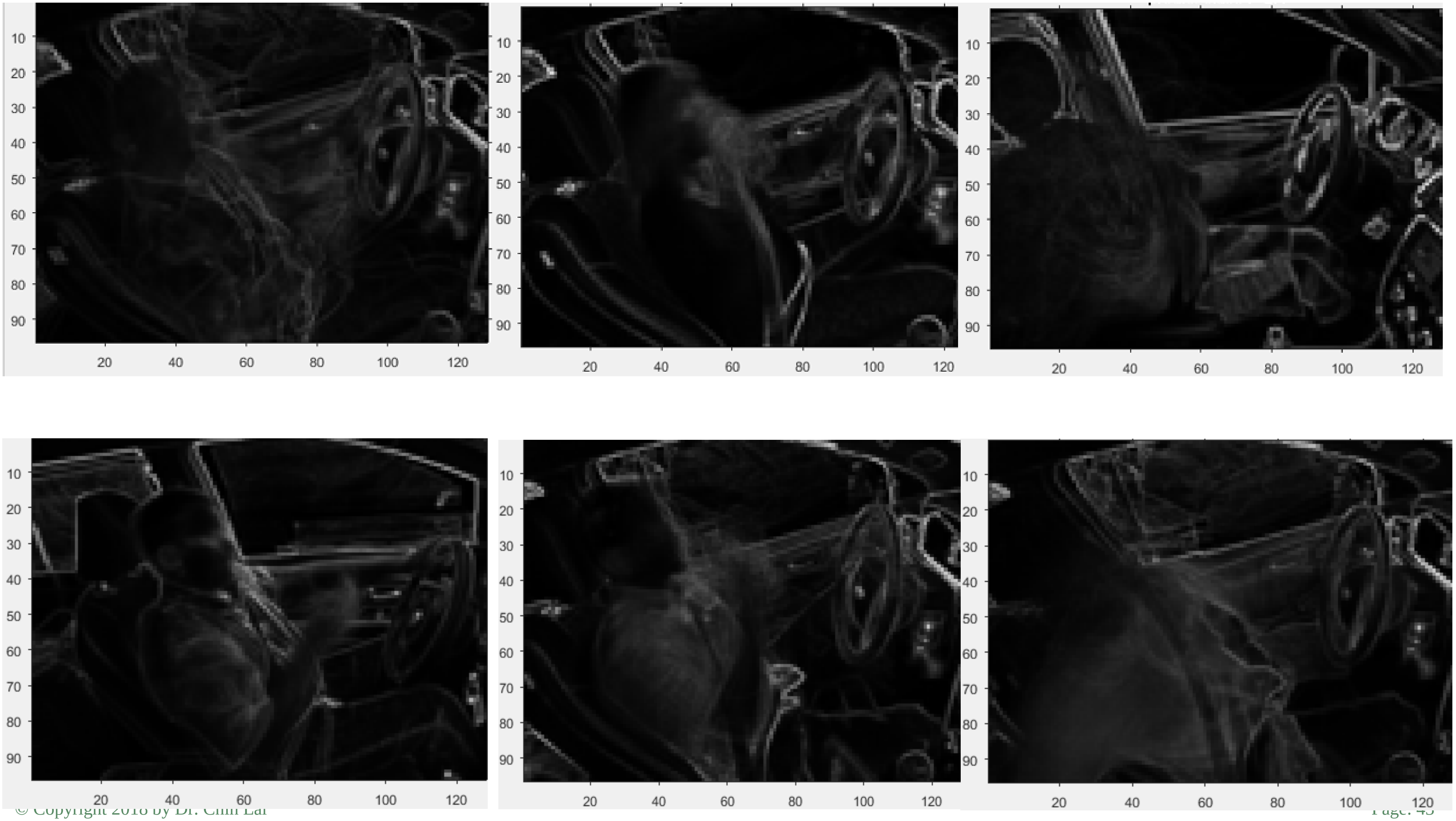
c7: reaching behind

c8: hair and makeup

c9: talking to passenger

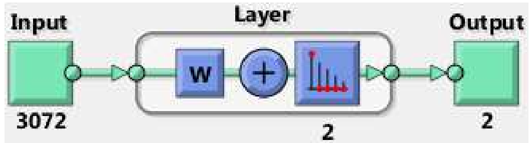
Past– Find Patterns of Similar Driving Behavior

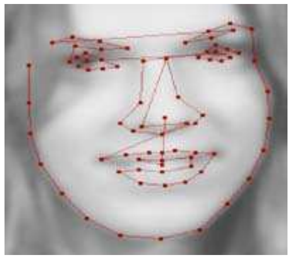
- SOM (Self-Organizing Map).
- ***k*NN (*k* nearest neibhgor)**



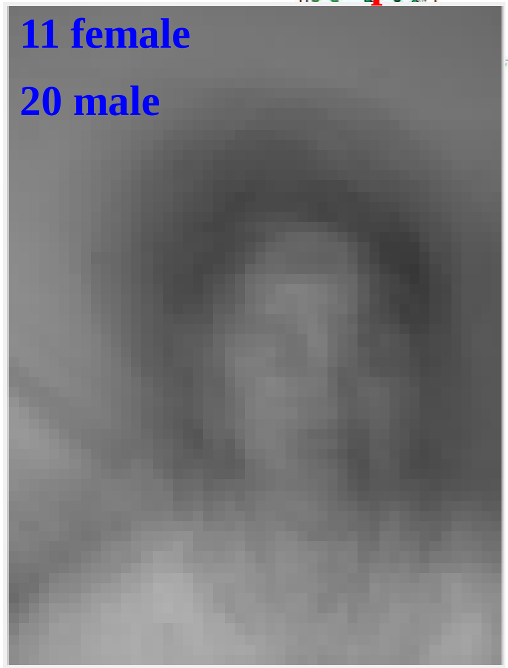
Gender Clustering

- MUCT Face DB www.milbo.org/muct
- Sample 36 female, 40 male.
  - Input 3072 gray-scale pixels.
  - Output SOM 2×1.
    - Average face? of different gender?

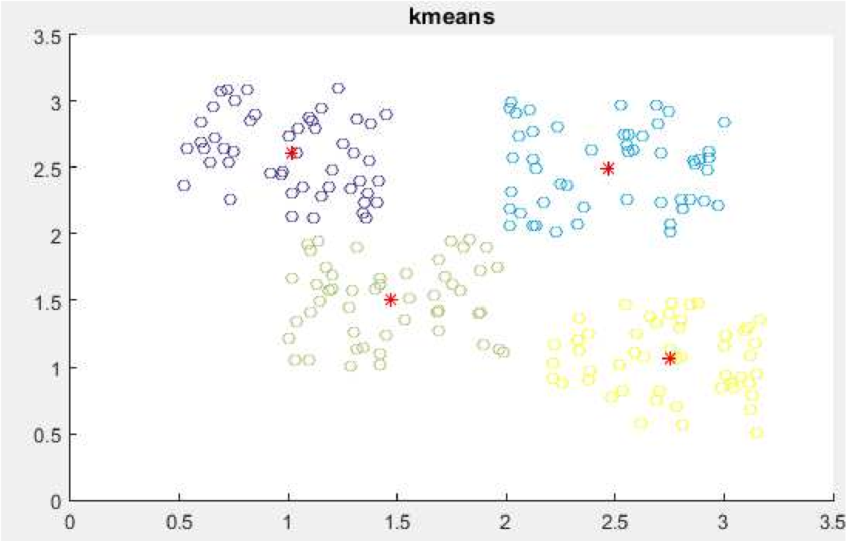


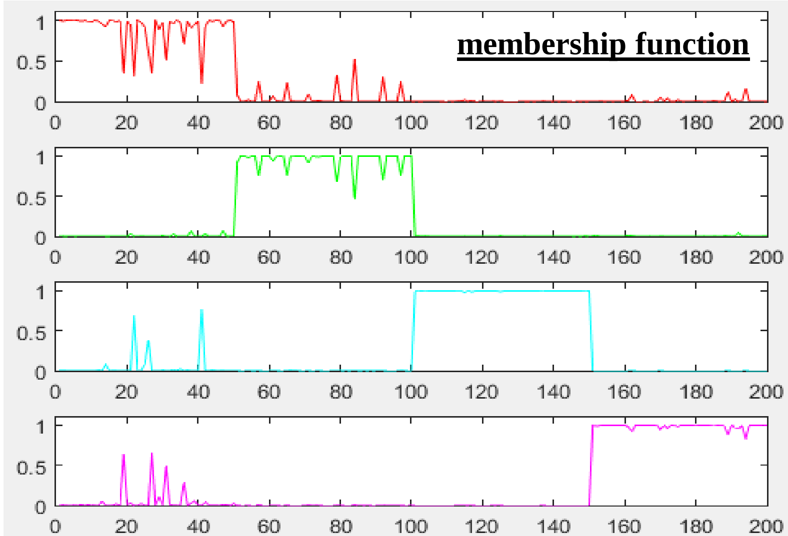


2×1 SOM output

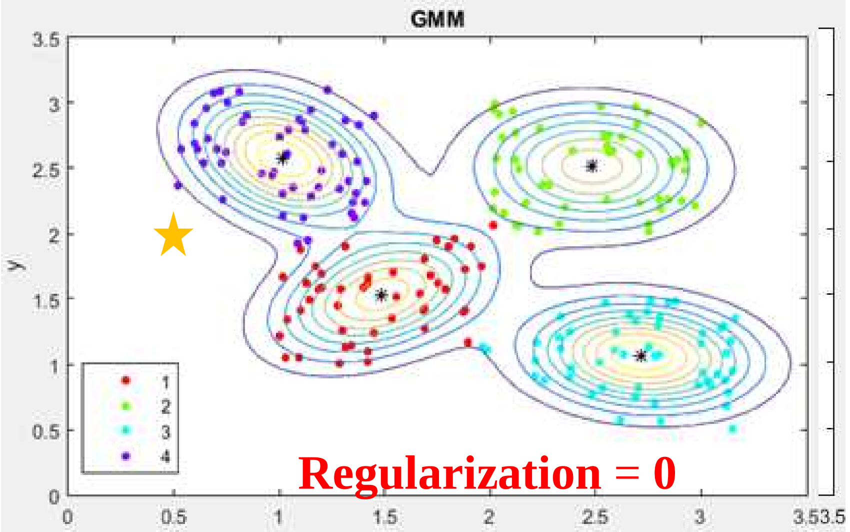


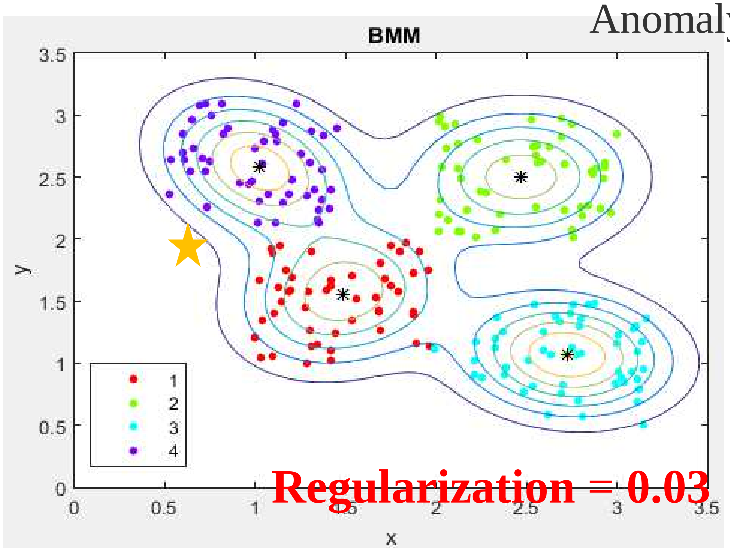
Gaussian Mixture Model (GMM) vs. *k*-means





GMM for
Anomaly Detection





Outline

- Proliferation of Data (Structured and Unstructured) / Internet of Things.
  - Just manually query all the data? Descriptive Analytics?

- Machine Learning Tools.

- Major topics in ML.
  - Linear Regression, Logistic Regression, **Linear Discriminant Analysis (LDA)**.
  - Gradient Descent, Regularization & Overfitting.
  - Support Vector Machine (SVM) and Kernel Methods.
  - Ensemble, Boosting / Bagging, Quality Evaluation
  - Self-Organized Map (SOM), **PCA, ICA, SVD, MDS**.
  - Collaborative Filtering, Matrix Factorization, (Recommender System), Gaussian Mixture Model.

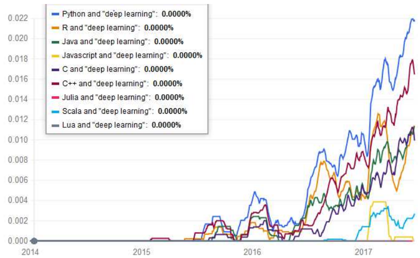
- Machine Learning Computing Platform.

Some Popular ML+DL Tools

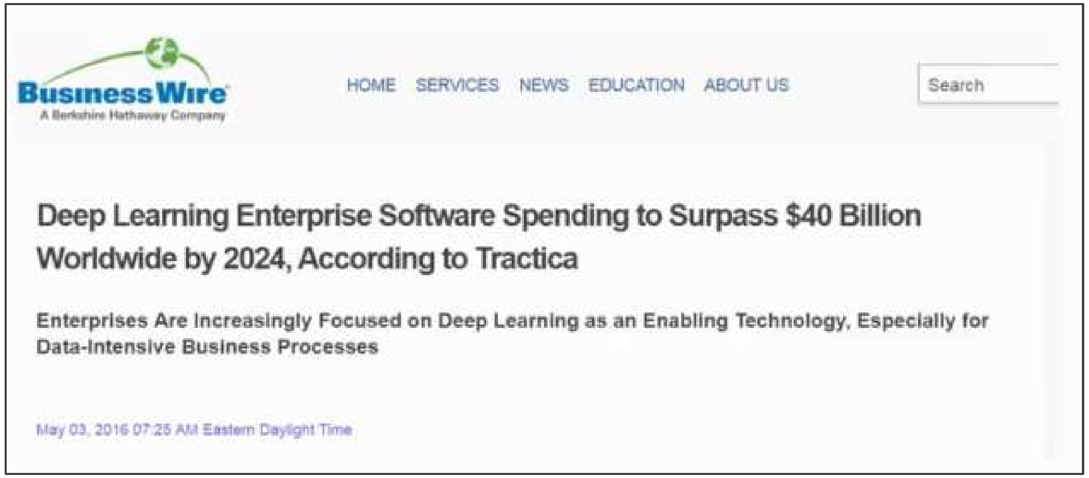
GPU??	Tool				
	Python	R	Spark	Matlab	TensorFlow
License	Open source	Open source	Open source	Proprietary	Open source
Distributed	No	No	Yes	No **YES**	No
Visualization	Yes	Yes	No	Yes	No
Neural nets	Yes	Yes	Multilayer perceptron classifier	Yes	Yes
Supported languages	Python	R	Scala, Java, Python, and R	Matlab	Python and C++
Variety of machine-learning models	High	High	Medium	High	Low
Suitability as a general-purpose tool	High	Medium	Medium	High	Low
Maturity	High	Very high	Medium ??	Very high	Low

[IEEE Computing Edge April 2017, pp.12]

- http://www.kdnuggets.com/2017/01/most-popular-language-machine-learning-data-science.html



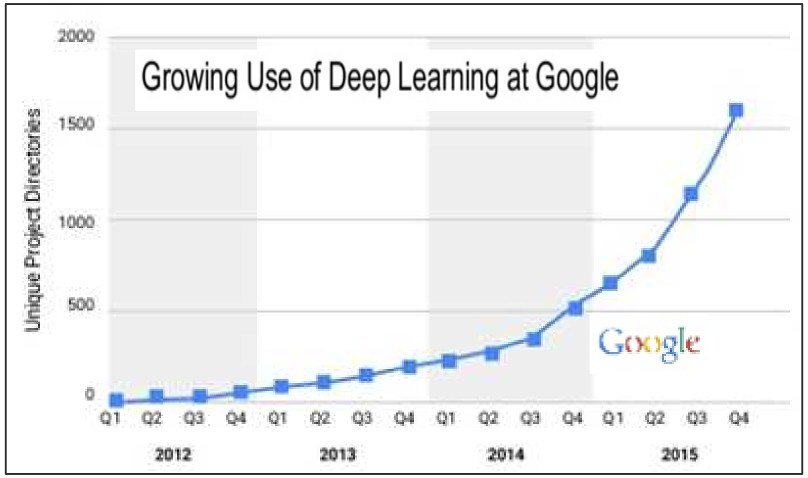
Deep Learning Investment Is On the Rise


BusinessWire
A Berkshire Hathaway Company
HOME SERVICES NEWS EDUCATION ABOUT US
Search

Deep Learning Enterprise Software Spending to Surpass $40 Billion Worldwide by 2024, According to Tractica

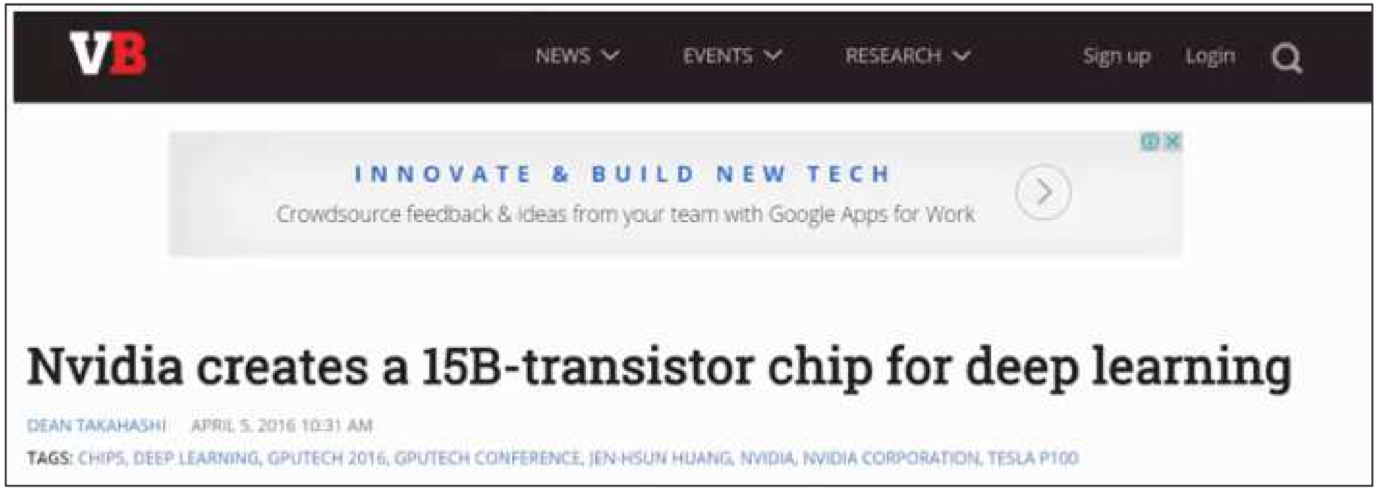
Enterprises Are Increasingly Focused on Deep Learning as an Enabling Technology, Especially for Data-Intensive Business Processes

May 03, 2016 07:25 AM Eastern Daylight Time



Renewed success of deep NN is largely attributable to advances in hardware.

The use of GPUs instead of CPUs can bring training times down <u>from months to mere days</u>.


VB
NEWS EVENTS RESEARCH Sign up Login

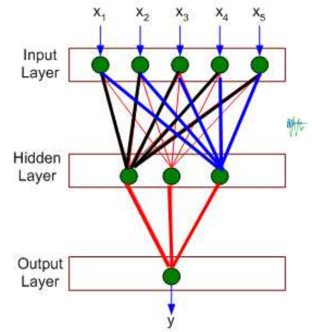
INNOVATE & BUILD NEW TECH
Crowdsource feedback & ideas from your team with Google Apps for Work

Nvidia creates a 15B-transistor chip for deep learning

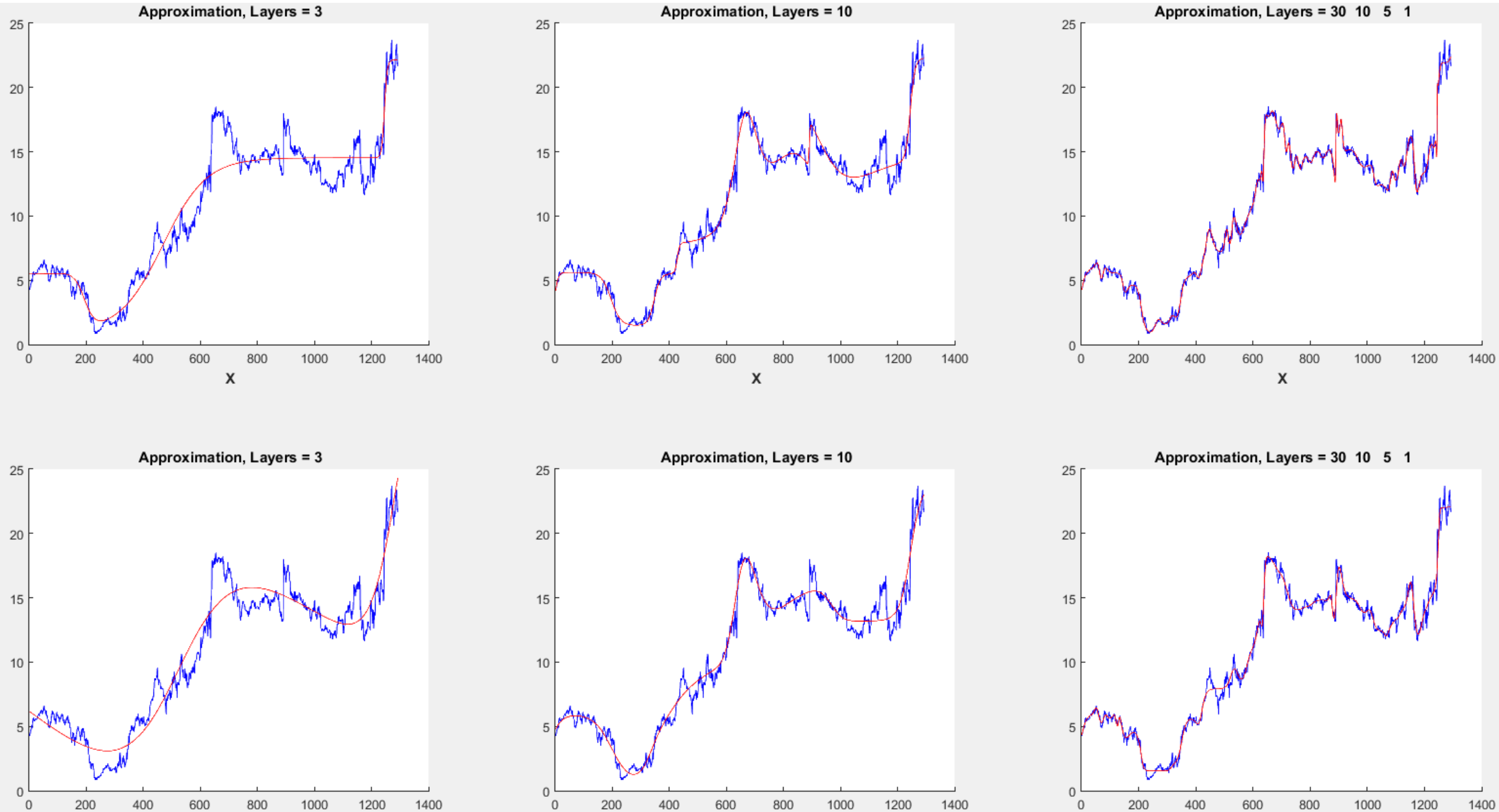
DEAN TAKAHASHI APRIL 5, 2016 10:31 AM

TAGS: CHIPS, DEEP LEARNING, GPUTECH 2016, GPUTECH CONFERENCE, JEN-HSUN HUANG, NVIDIA, NVIDIA CORPORATION, TESLA P100

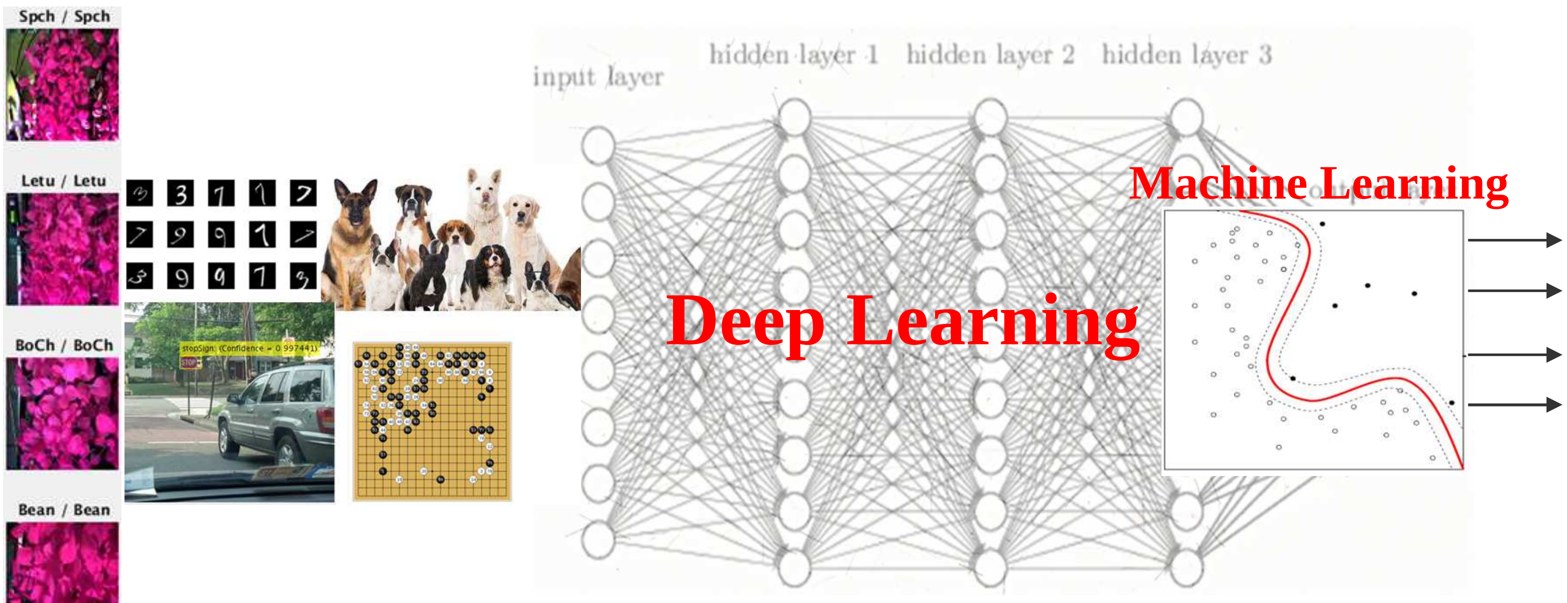
Modeling Stock Market Curve

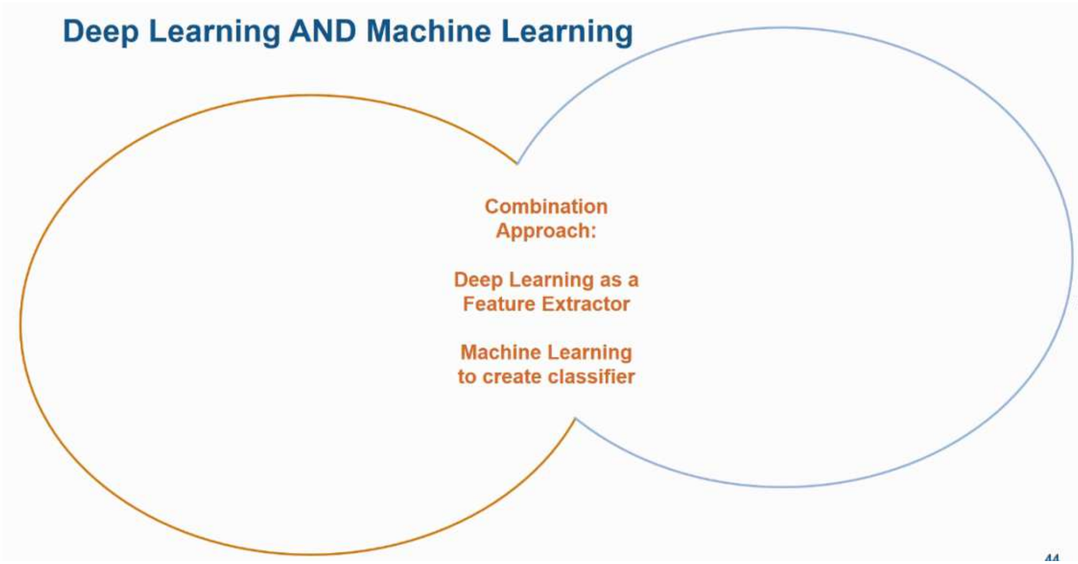


- Closely follow the original curve.



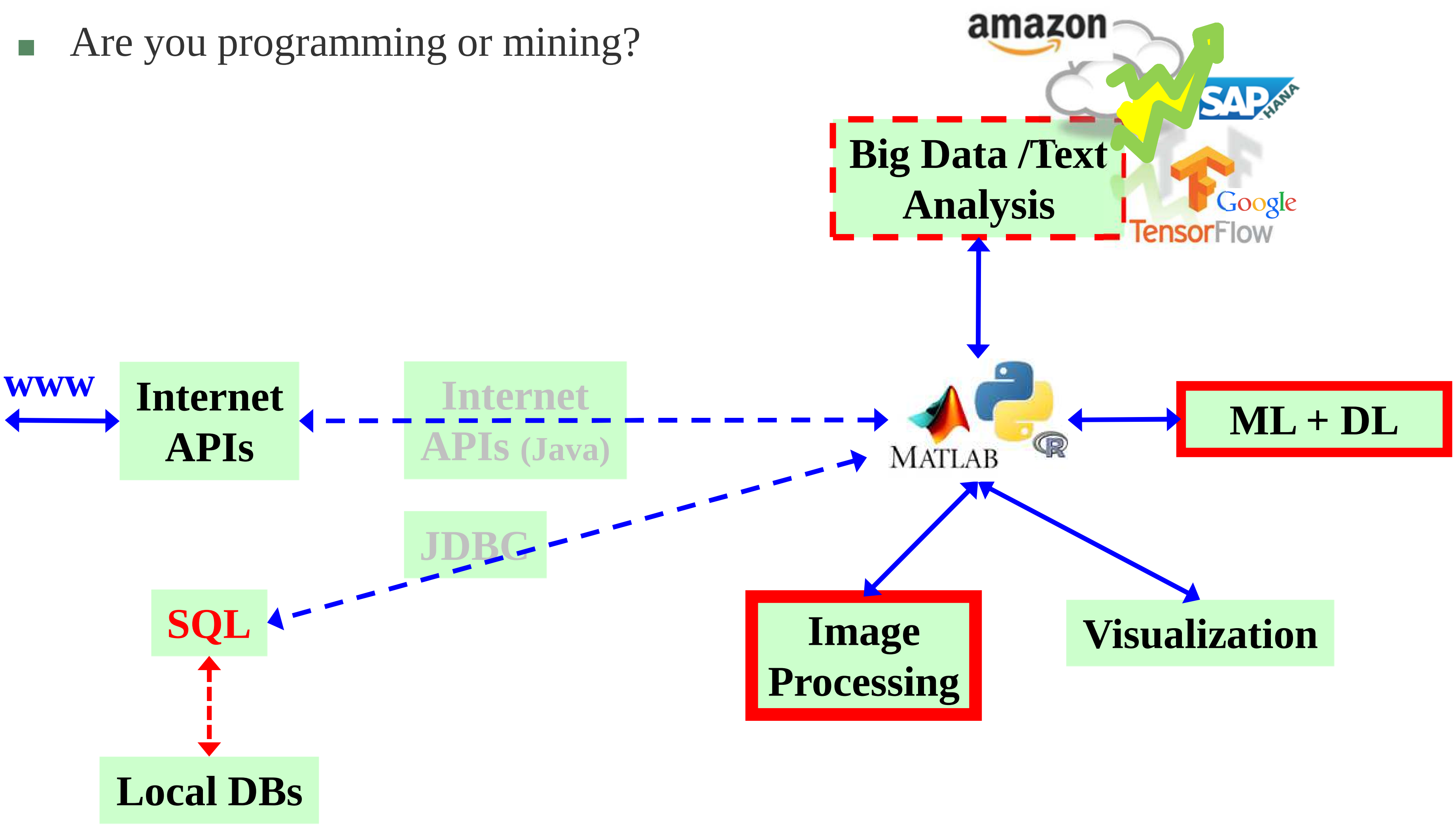
Relationships between ML and DL





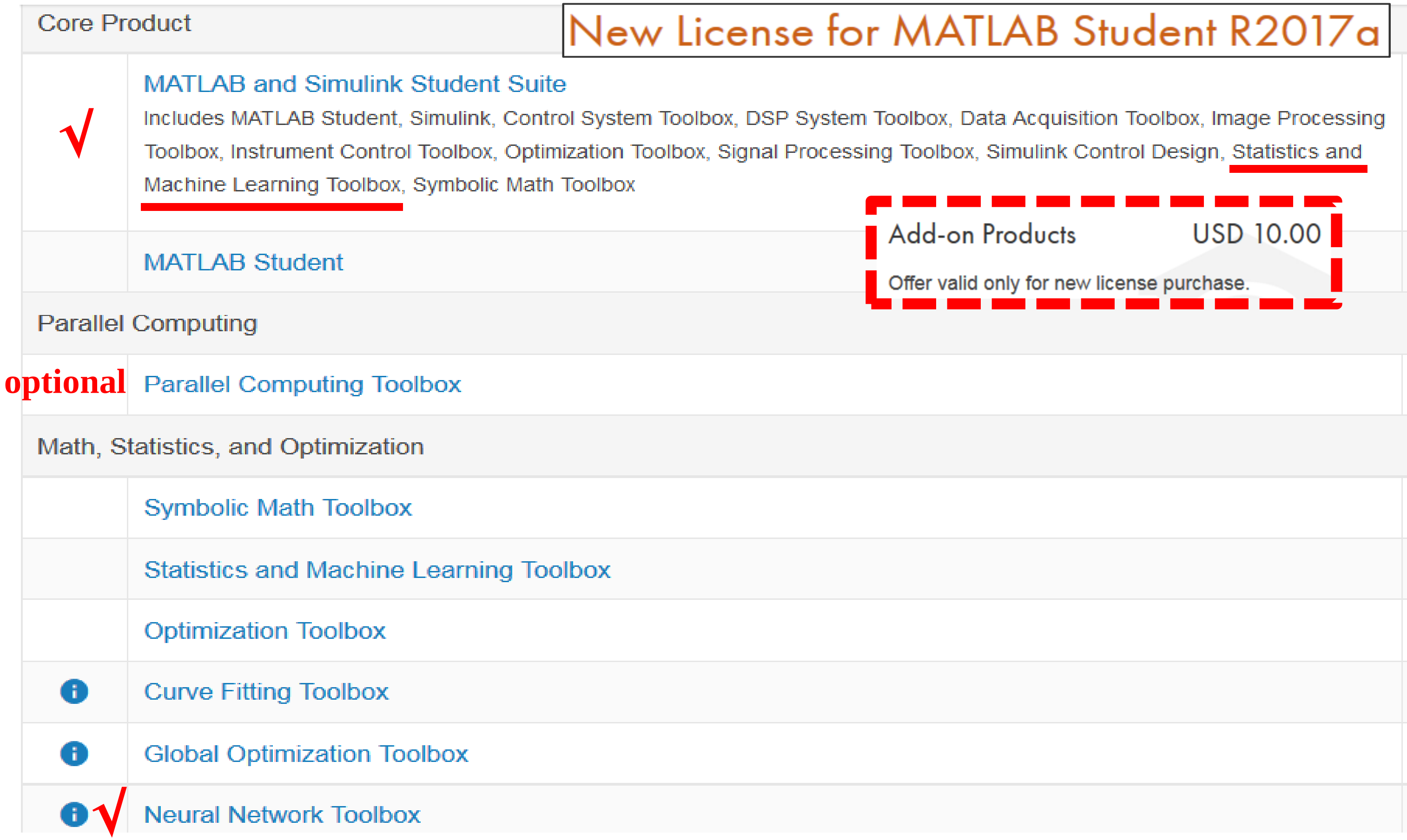
How Things Work Together???

- Languages / Systems / Tools Architecture
- Are you programming or mining?

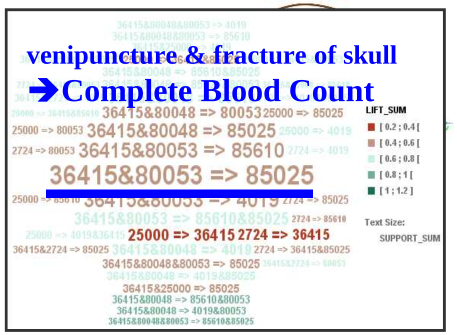


What is Matlab

- MATLAB® is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in script notation.
- https://www.mathworks.com/store/link/products/student/new?s_iid=htb_buy_gtwy_cta3

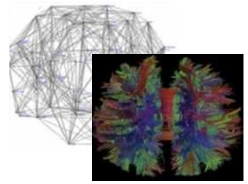


Past Projects by Dr. Chih Lai



- Healthcare analytics
  - 200-million patient records, Frequent medical sequences.
  - Fraud claims. Rank doctors on re-admissions.

- Predicting index codes for medical journal papers.
  - 450-million codes from 4-million papers.



- Find frequent patterns from brain connectivity graphs.
  - 500 patients w/ 8-billion fMRI links = 4-trillion links. ***Amazon Research Grant***.

- Predicting wind power component failure and their health status.
  - 500^+ wind power generators, 3-year sensor data, 7^+terabyte. ***Research Grant***

- Predicting vegetation growth, identifying weeds in farm field.
  - Use aerial images, soil types & ground elevation info.

- Parkinson Patient Evaluation with Medtronic and U of Minnesota.
  - 2 U.S. patents. ***Medtronic Research Award***.