Machine Learning

Abdelhak Mahmoudi
abdelhak.mahmoudi@um5.ac.ma

Direction Générale des Impôts

February 4st, 2020

Content

1. The Big Picture
2. Supervised Learning
    - Linear Regression, Logistic Regression, Support Vector Machines, Trees, Random Forests, Boosting, Artificial Neural Networks
3. Unsupervised Learning
    - Principal Component Analysis, K-means, Mean Shift

The Big Picture

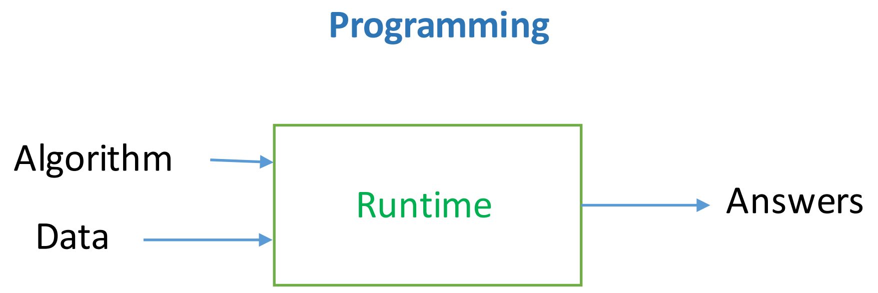
- The Big Picture of ML !
- Terminologies
- How can I Apply?
- How can I Learn?

The Big Picture!

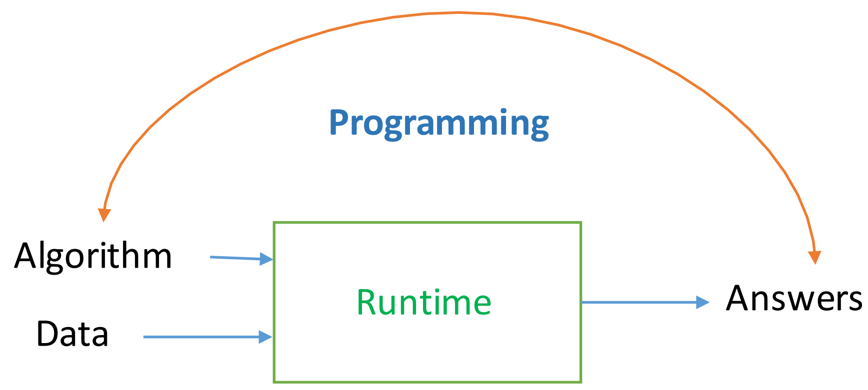
Forbes: "The Top 10 AI And Machine Learning Use Cases Everyone Should Know About"

1. Data Security,
2. Personal Security,
3. Financial Trading,
4. Healthcare,
5. Marketing personalization,
6. Fraud Detection,
7. Recommendations,
8. Online Search,
9. Natural Language Processing (NLP),
10. Smart Cars

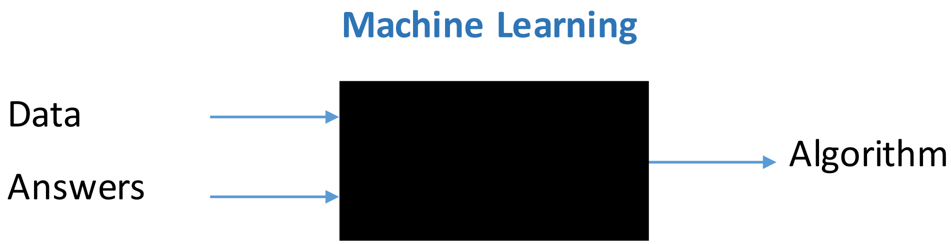
The Big Picture!



The Big Picture!

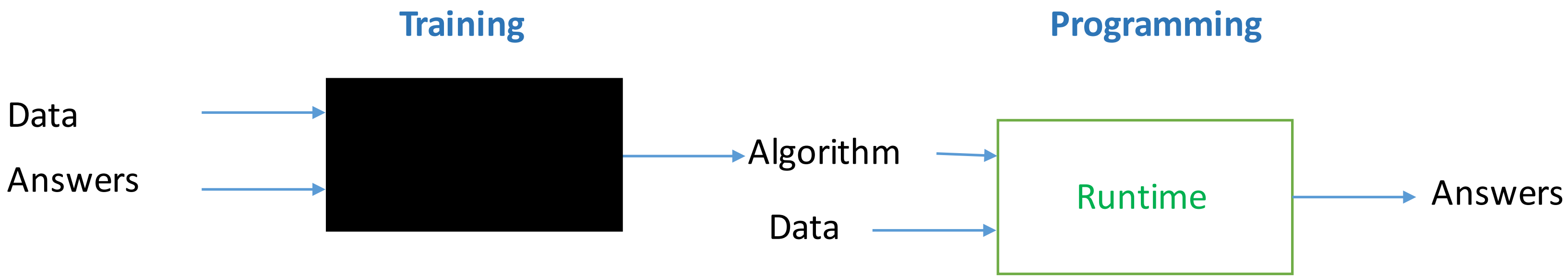


The Big Picture!



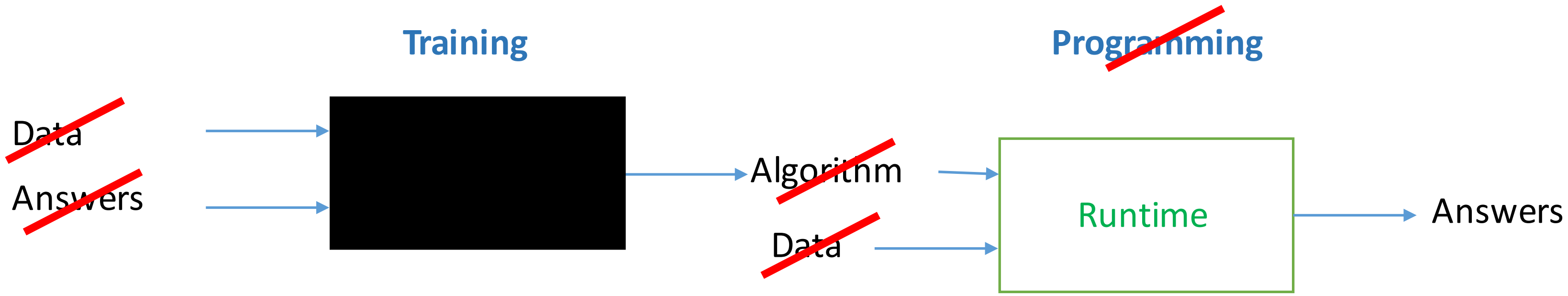
The Big Picture!

Machine Learning



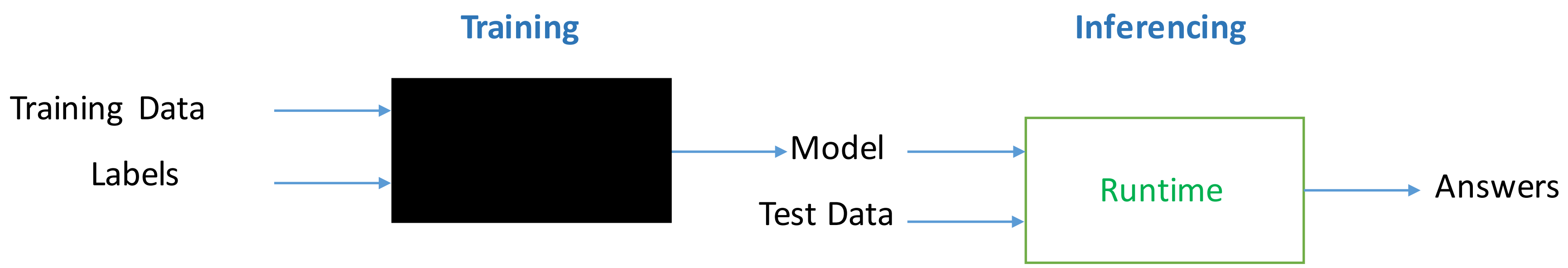
The Big Picture!

Machine Learning



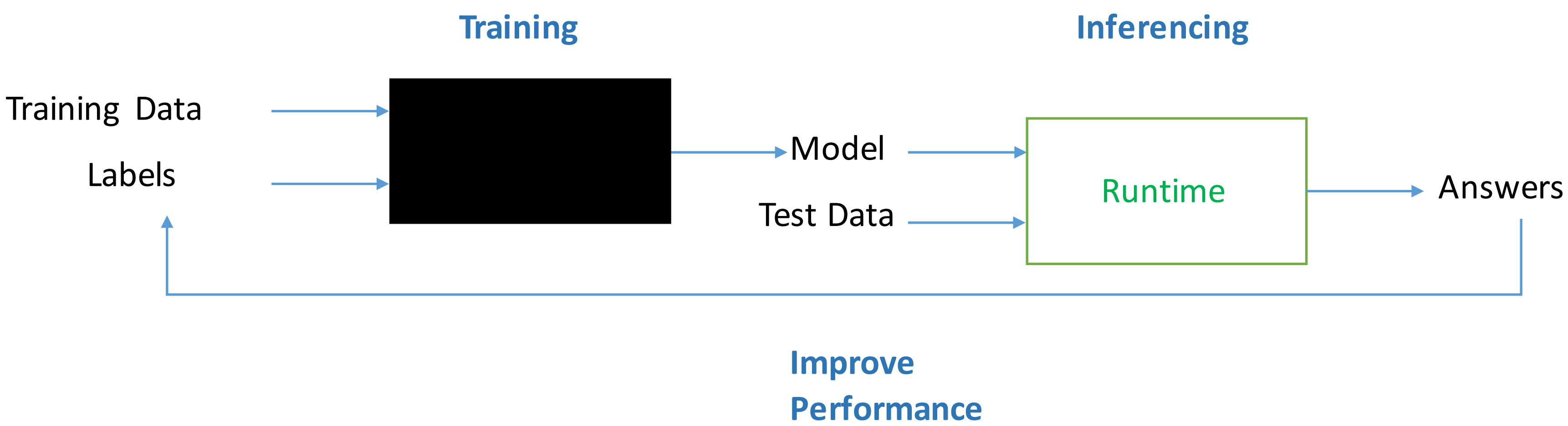
The Big Picture!

Machine Learning



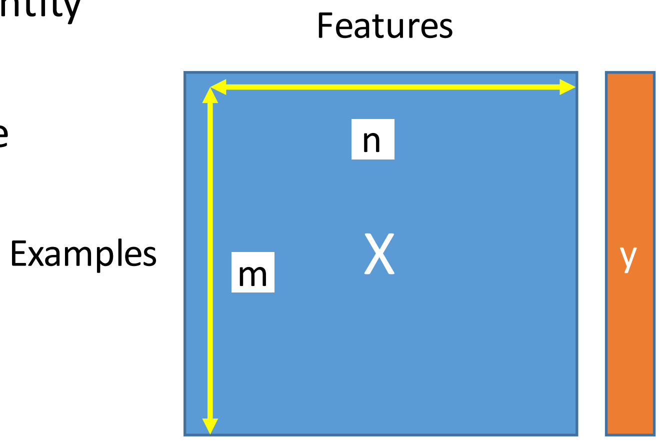
The Big Picture!

Machine Learning



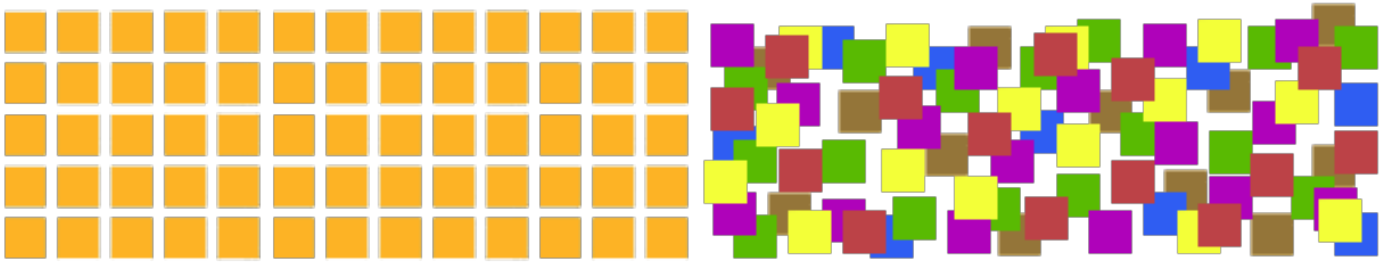
The Big Picture!

- Data
  - Example $x^{(i)}$
    - Row/Instance/Input/Observation/Record/Point/Sample/Entity
  - Feature $x^{(i)}_j$
    - Columns/Variable/Predictor/Characteristic/Field/Attribute
    - Quantitative (numeric, continue)
    - Qualitative (textual, category)
  - Dimension, Visualization
    - m Examples: i = 1..m
    - n Features: j = 1..n
  - Output : $y_i = x^{(i)}_k$ (k in 1..n)
    - target/class/output
    - For each example (0/1)



The Big Picture!

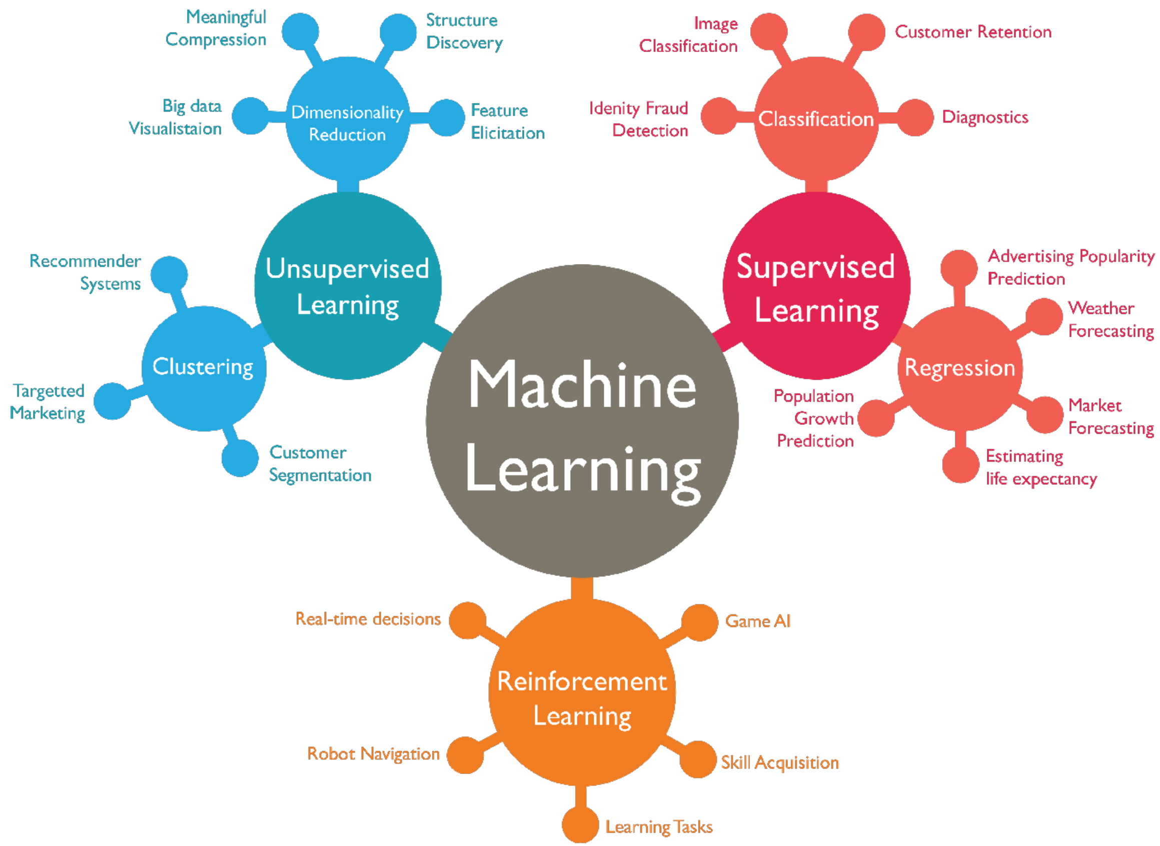
- Data
  - Structured
    - CSV, XML, JSON, XLSX, etc.
  - Unstructured
    - DOC, HTML, PDF, PNG, MP3, MP4, etc.



Text, Image, son

The Big Picture!

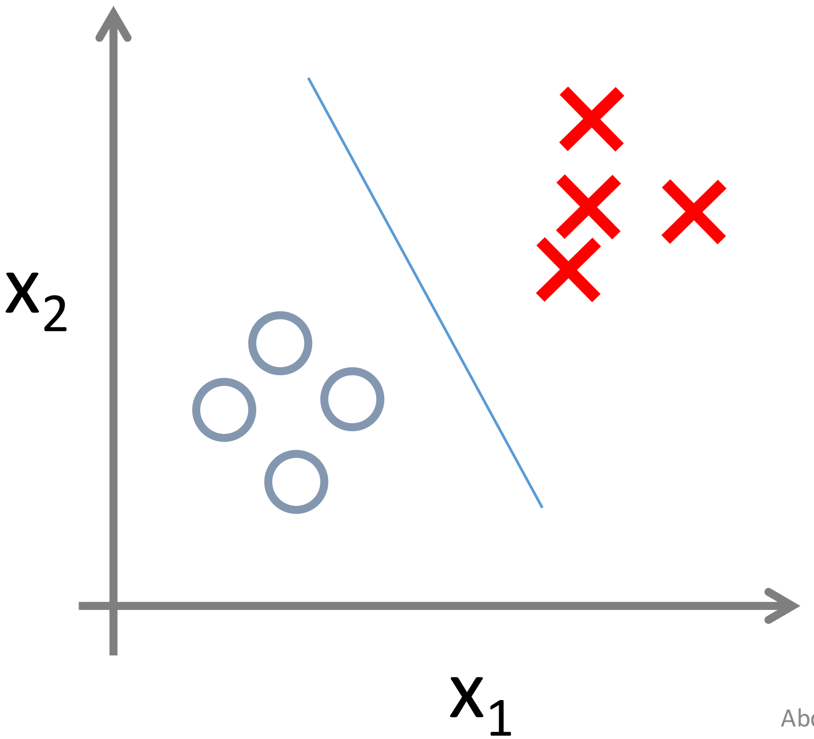
- Types of Learning
  - Supervised
    - Classification
    - Regression
  - Unsupervised
    - Dimensionality Reduction
    - Clustering
  - Semi-supervised
    - Little supervised data
  - Reinforcement



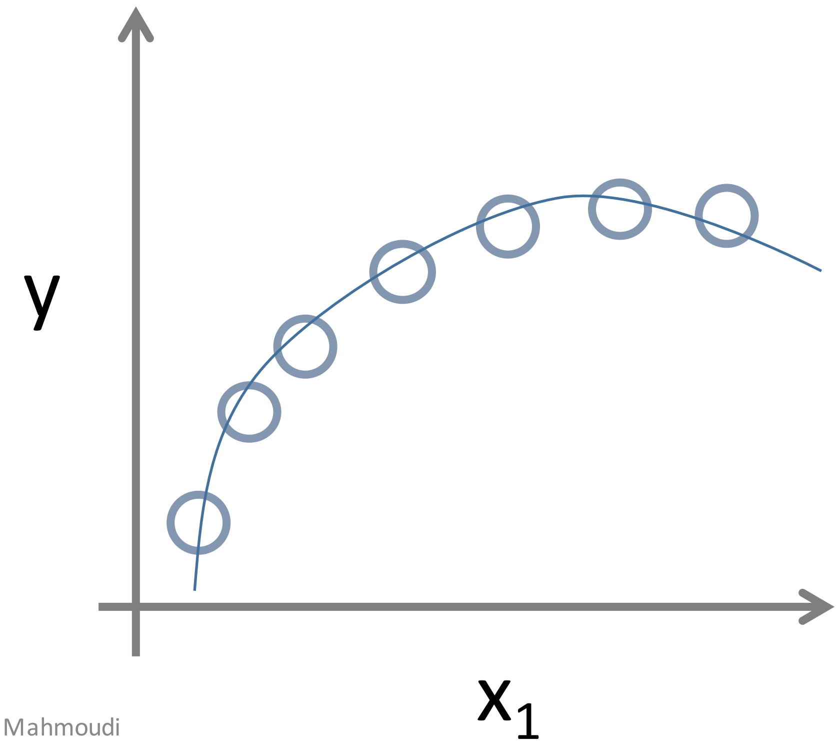
The Big Picture!

Supervised Learning

Classification (y is discrete)



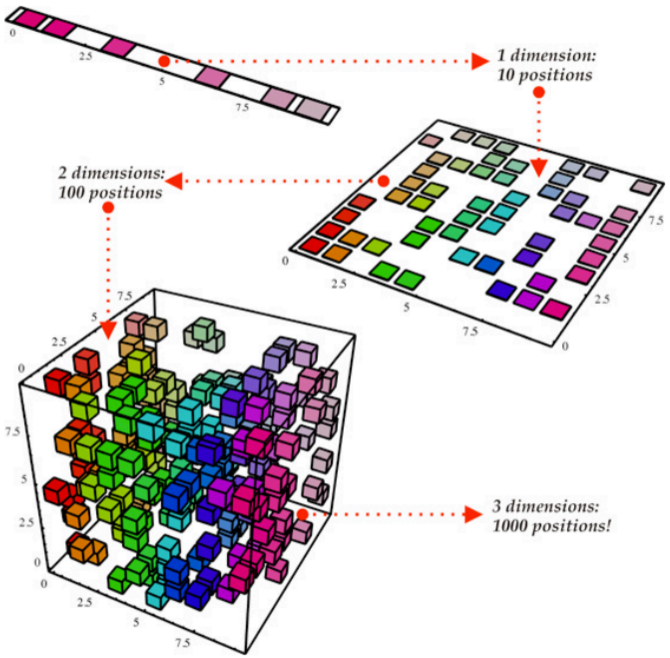
Regression (y is continuous)



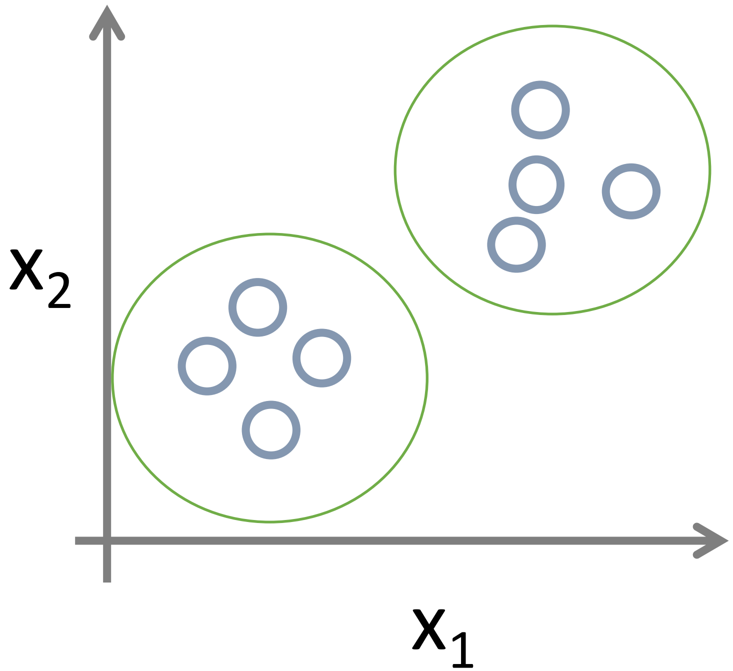
The Big Picture!

Unsupervised Learning (y absent)

Dimensionality Reduction

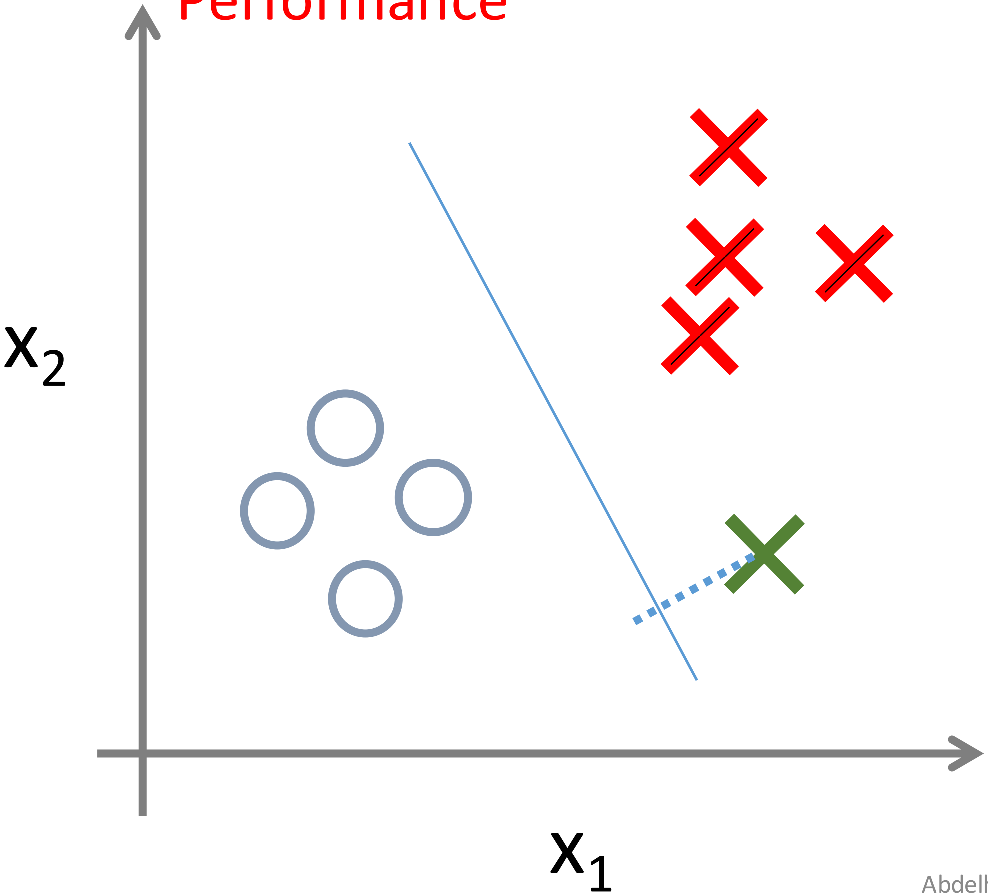


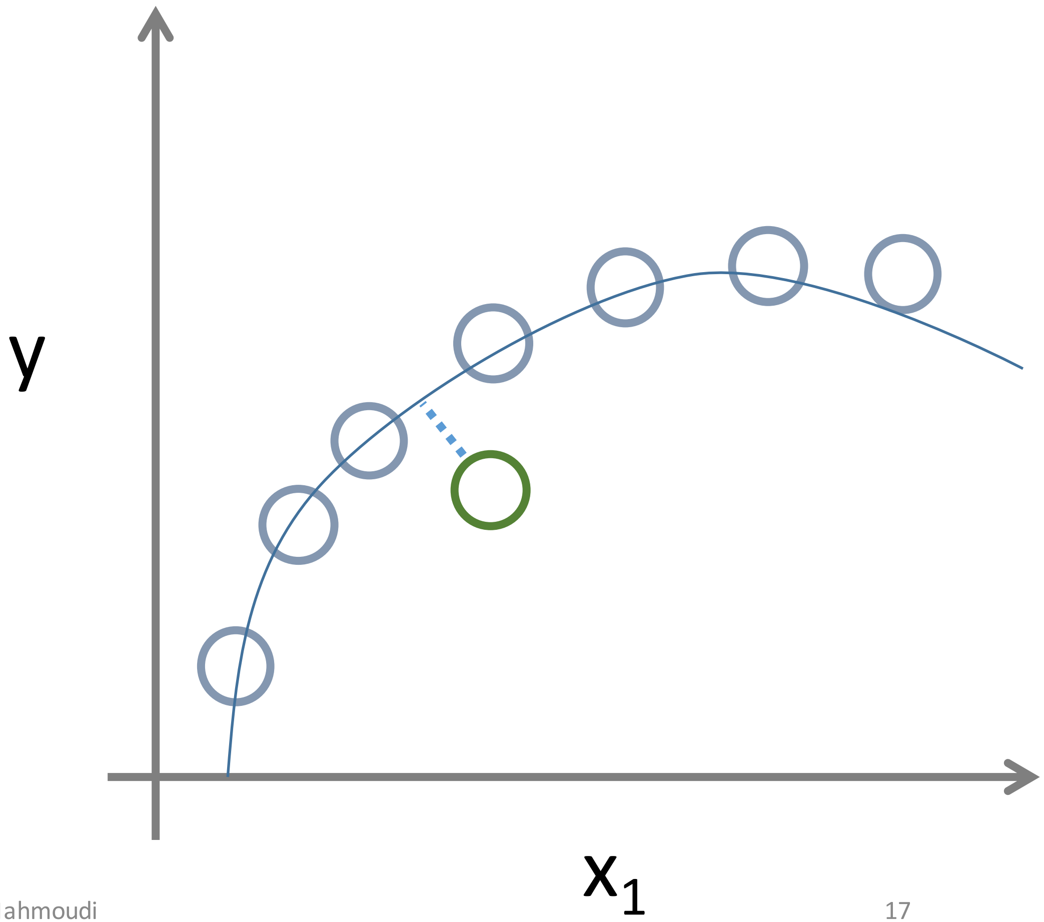
Clustering



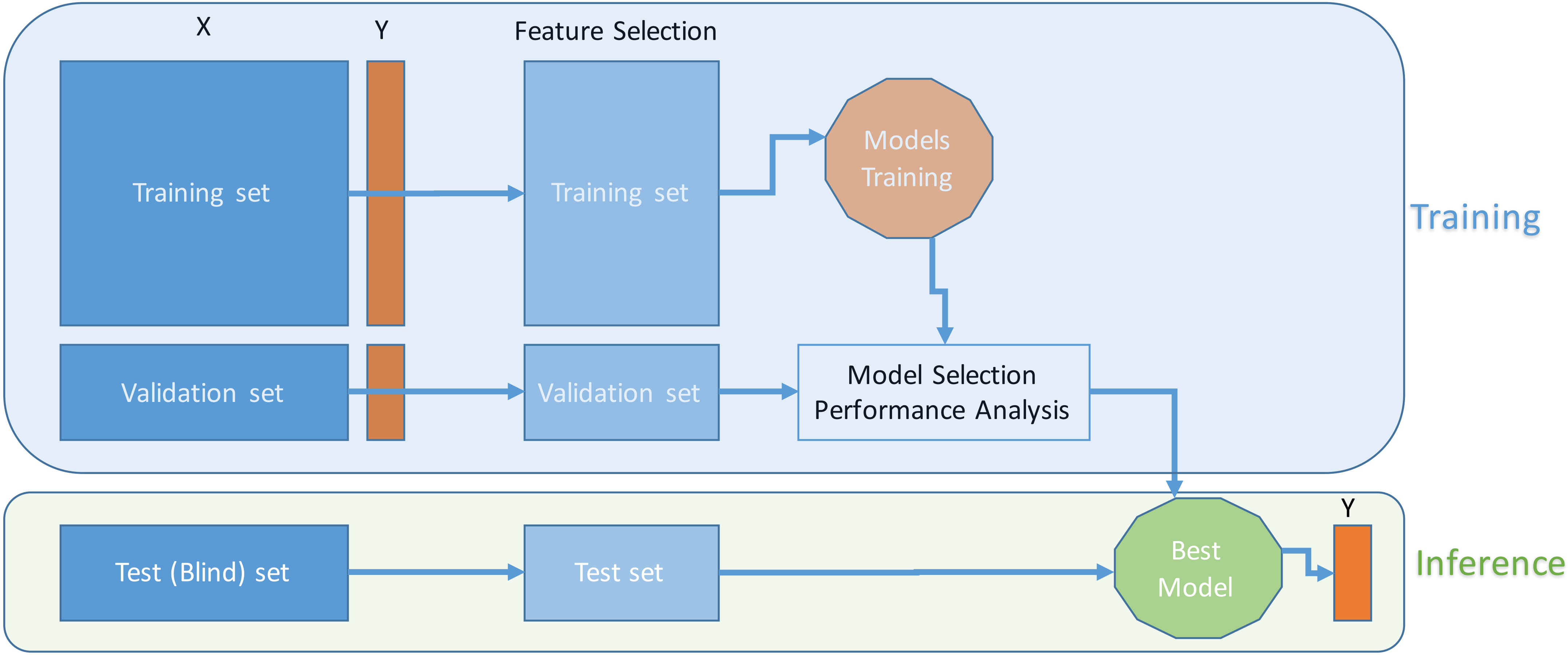
The Big Picture!

Performance

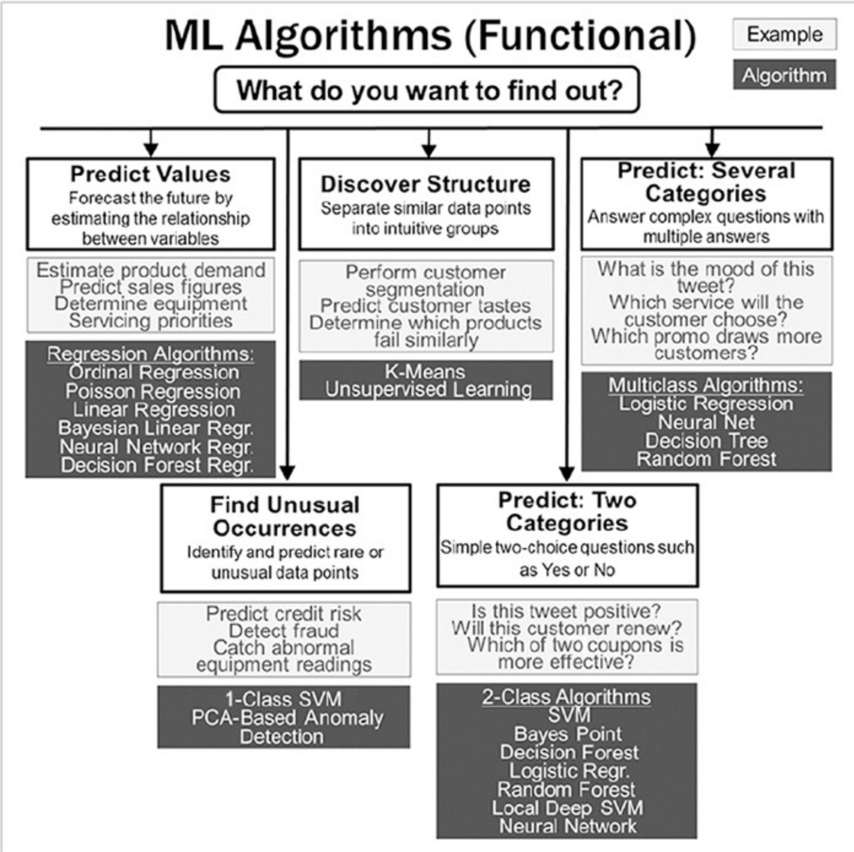


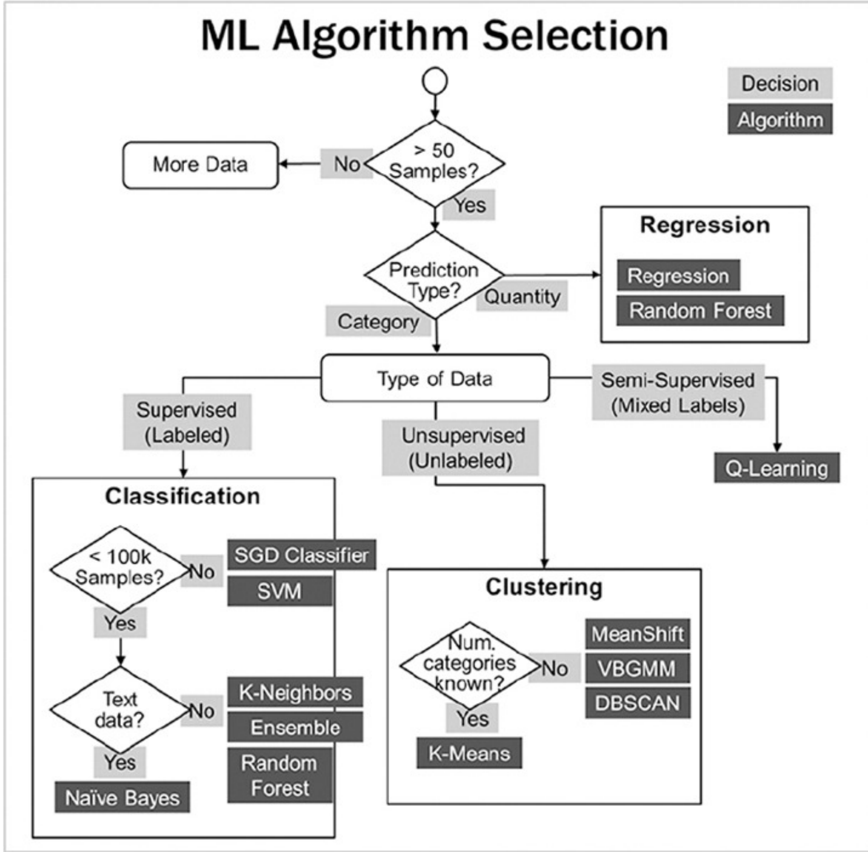


The Big Picture!



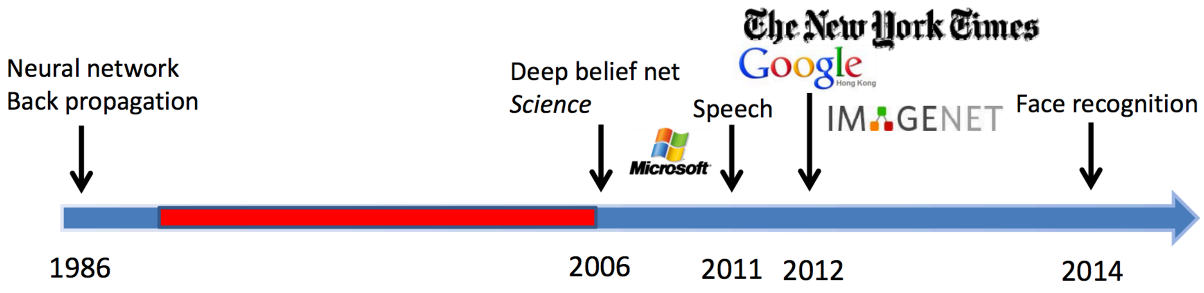
The Big Picture!

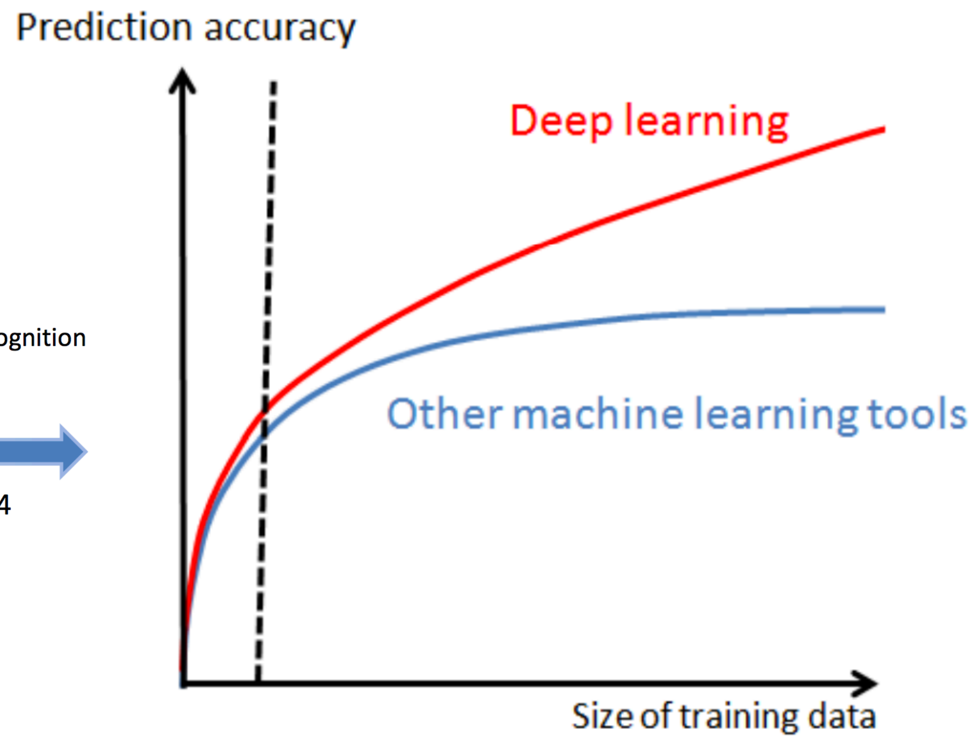




The Big Picture!

- Why Deep Learning?





Terminologies

- Artificial Intelligence
- Machine Learning, Deep Learning
- Statistical Learning
- Data Mining

Artificial Intelligence (1943)

- “The first work that is now generally recognized as AI was McCulloch and Pitts' 1943 formal design for Turing-complete "artificial neurons". Wikipedia
- Intelligent Machines mimics Natural Intelligence (NI)
- Natural Intelligence (General Intelligence)
  - Reasoning, Problem solving,
  - Knowledge representation, Learning,
  - Planning, Perception, Motion and manipulation, Natural Language
  - Etc.

Machine Learning (**1959**)

- "Arthur Samuel, an American pioneer in the field of computer gaming and artificial intelligence, coined the term "Machine Learning" in 1959 while at IBM". Wikipedia
- A subfield of Computer Science and Artificial Intelligence which deals with building systems that can learn from data, instead of explicitly programmed instructions.
- Artificial Neural Networks (**1975**)
  - Begin in 1943, stagnated in 1969, relaunched in 1975 by the Backpropagation algorithm,
  - Deep Learning (**2006)**
    - Much powerful in the Age of Big data and distributed processing
- Book: "Machine Learning". Tom M. Mitchell. 1997

Statistical Learning (**1968**)

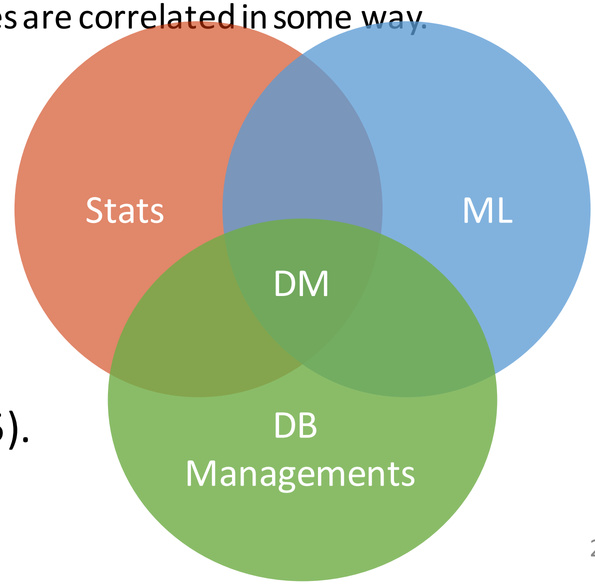
- VC Theory. “On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities". Vapnik, V. N.; Chervonenkis, A. Ya, 1968
- A subfield of Mathematics which deals with finding relationship between variables to predict an outcome
- Support Vector Machines (**1995**)
  - Much simpler, overtook ANN, Vapnik V. N.
- Book
  - “An introduction to statistical learning with applications in R” (1st Edition 2013). Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani.

Data Mining (1990)

Appeared in the database and financial community to recognize customer and products trends

Definition : “The process of automatically discovering useful information in large repositories”.

- Automatically
  - Stats: correlation between 2 variables, what is the problem?
  - DM: parallel correlation between 1000 variables, send and email if two variables are correlated in some way.
- Discovering useful information
  - Stats: answer a specific question
  - DM: look for any specific reason
- Large Repositories
  - Stats: Collect data to answer a specific question
  - DM: Collect all, you don’t know the reason yet!



Book: Introduction to Data Mining (2nd edition 2018, 1st Edition in 2005).
Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar

How can I Learn?

- Math
  - Statistics, Probabilistic Graphical Models, Algebra, Optimization
- Programming Languages
  - Python, R,
- Books
  - Ian Goodfellow et al. “Deep Learning”. 2016
  - Aurélien Géron. “Hands on ML with sklearn”. 2017
  - Gareth James et al., “An introduction to statistical learning with R”. 2013
  - Tom M. Mitchell. “Machine Learning”. 1997
  - Etc.

How can I Learn?

- MOOCs
  - Coursera.org, Udemy.com, ocw.mit.edu, etc.
- StackOverflow
- Research Papers
  - Read and rewrite algorithms from scratch
- Follow People:
  - Androw Ng, Yann LeCun, Jeff Hinton, Sebastian Thrun, etc.

How can I Apply?

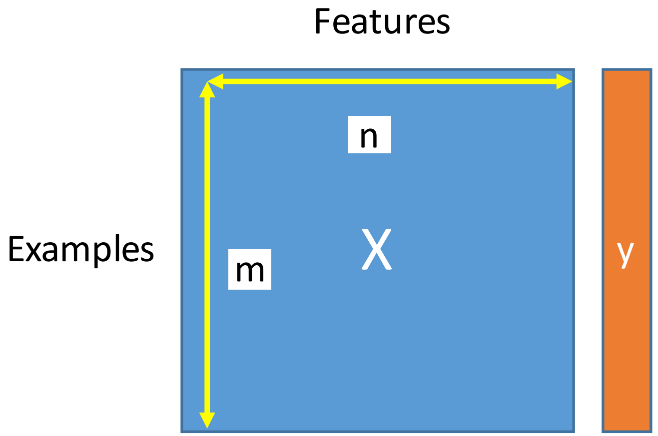
- Start small projects and use Framworks
  - Scikit-learn, TensorFlow, Keras, Pytorch, Caffe, Microsoft Cognitive Toolkit (CNTK), MXNet, Spark MLlib, etc.
- Challenge your self
  - Find data: Web, UCI Machine Learning Repo
  - Go for competitions: Kaggle, DrivenData, Zindi
- Github
  - Find codes
  - Share your code
- Softwares (for non-pro !)
  - Knime, IBM SPSS Modeler

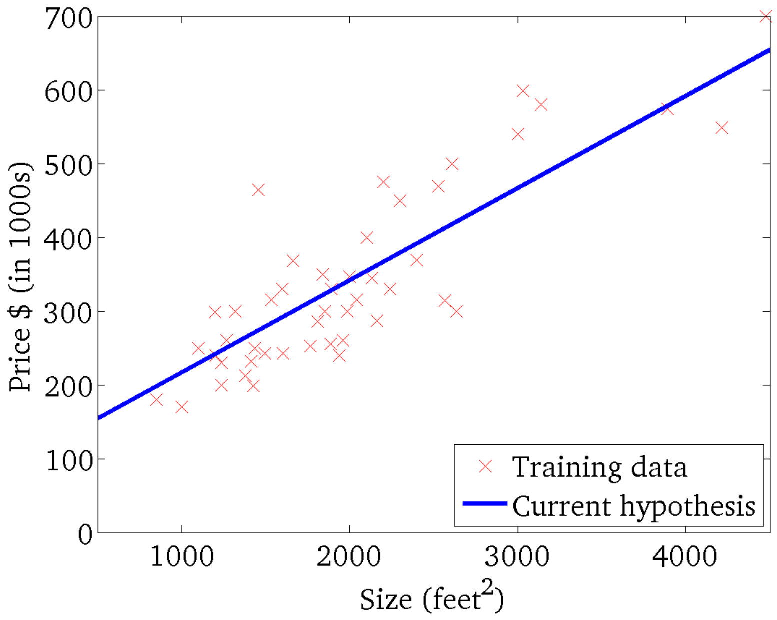
Supervised Learning

- Linear Regression
- Logistic Regression
- Support Vector Machines
- Trees (Decision and Regression)
- Random Forests
- Boosting
- Artificial Neural Networks

Linear Regression

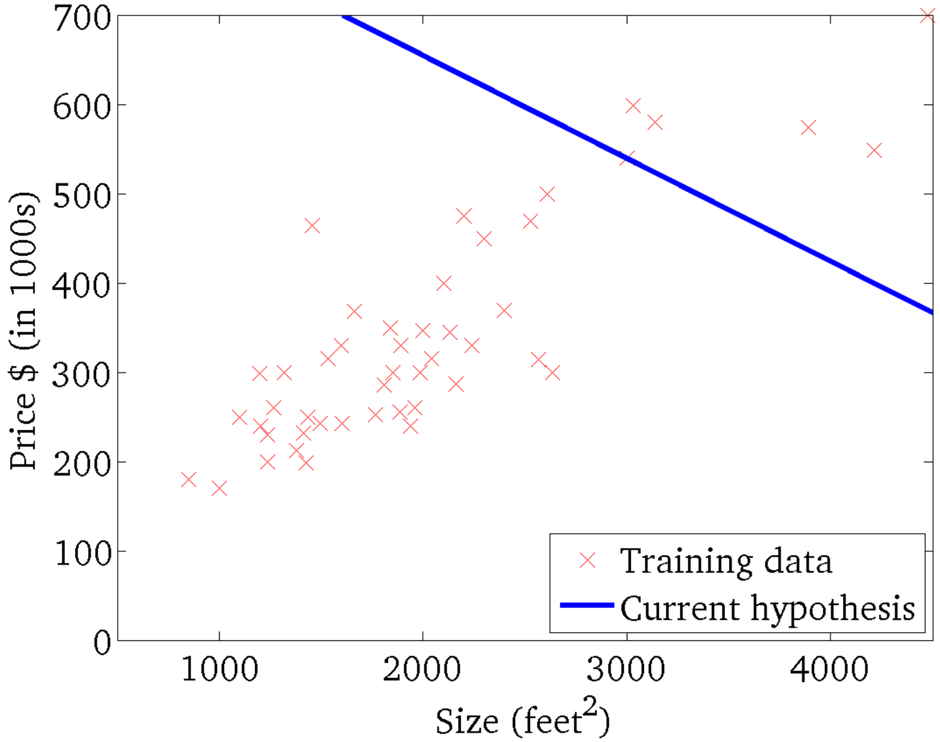
- The output y is continuous
- Fit X with a line $y = \theta_0 + \theta_1 x$
- The best line is the line with minimum loss $\mathrm{L}(\theta)$
- Solved with gradient descent





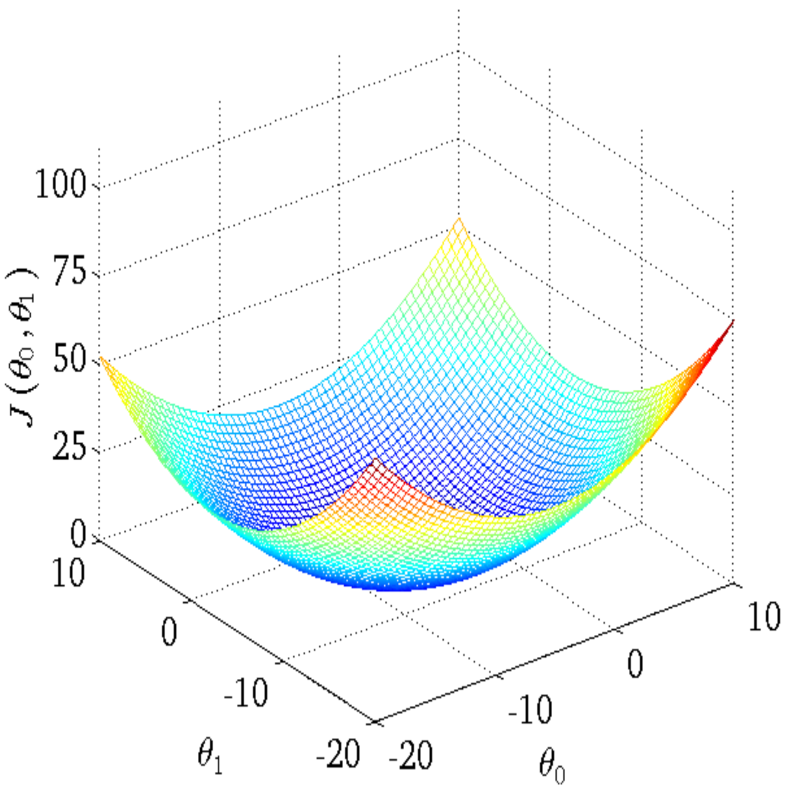
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



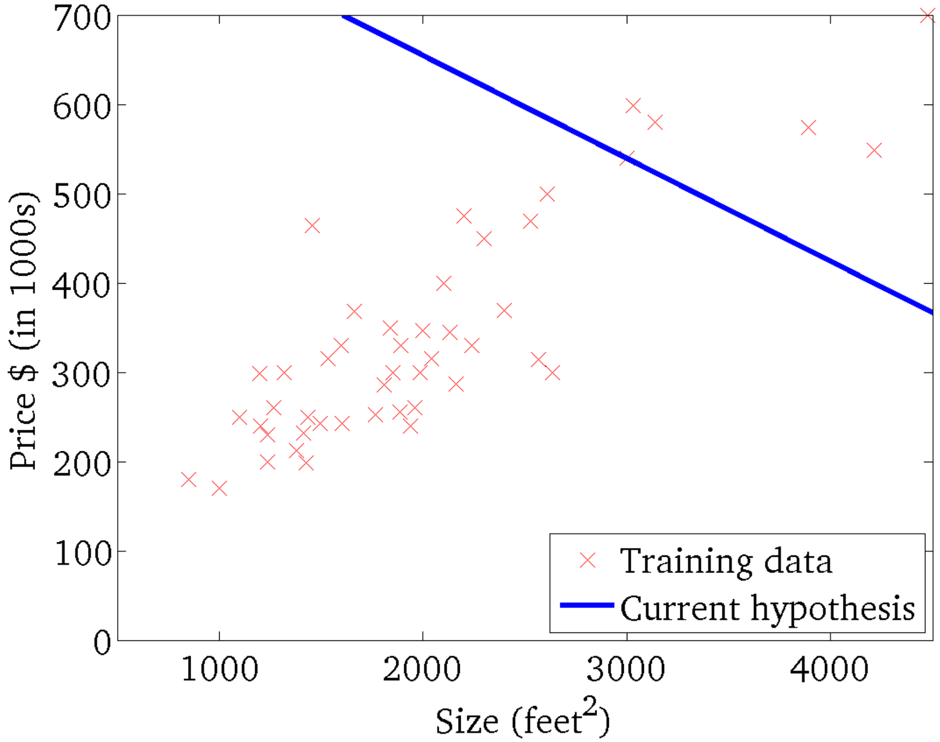
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



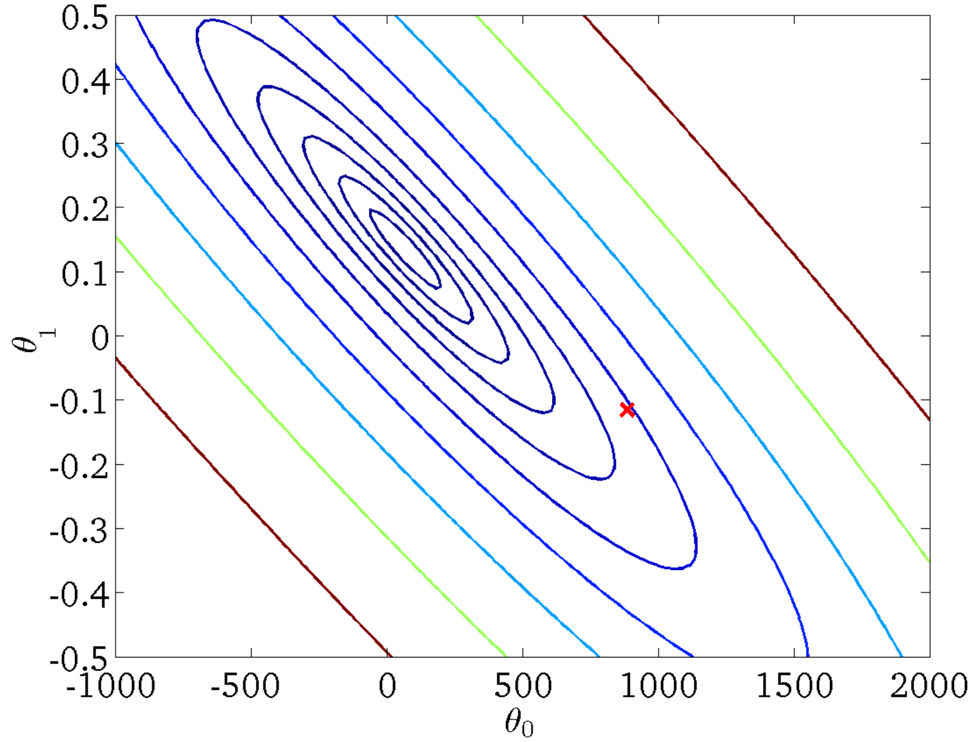
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



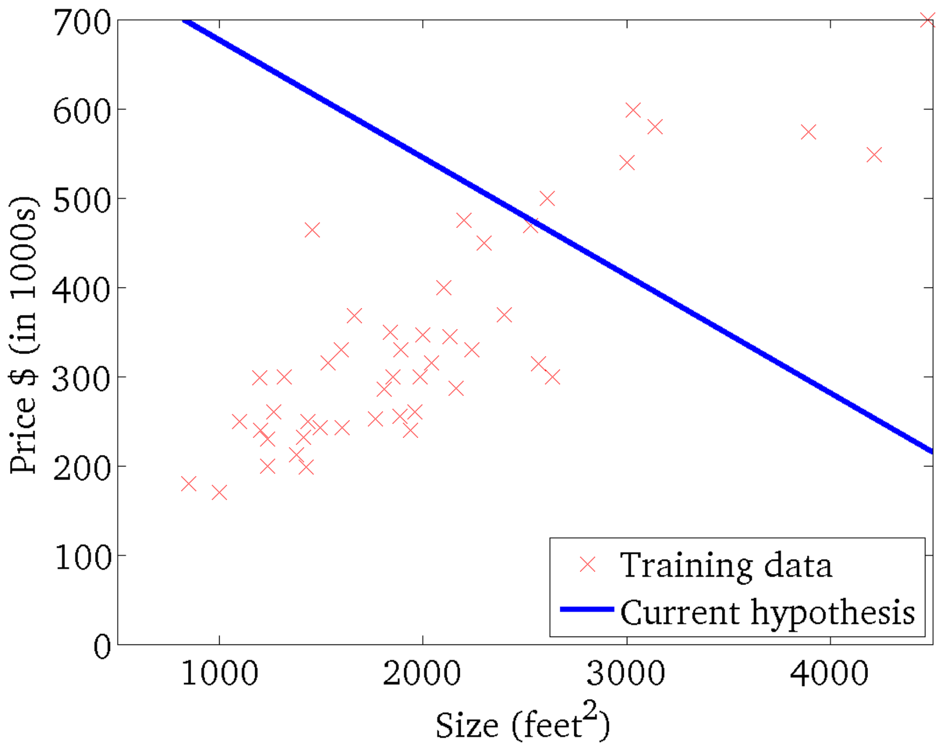
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



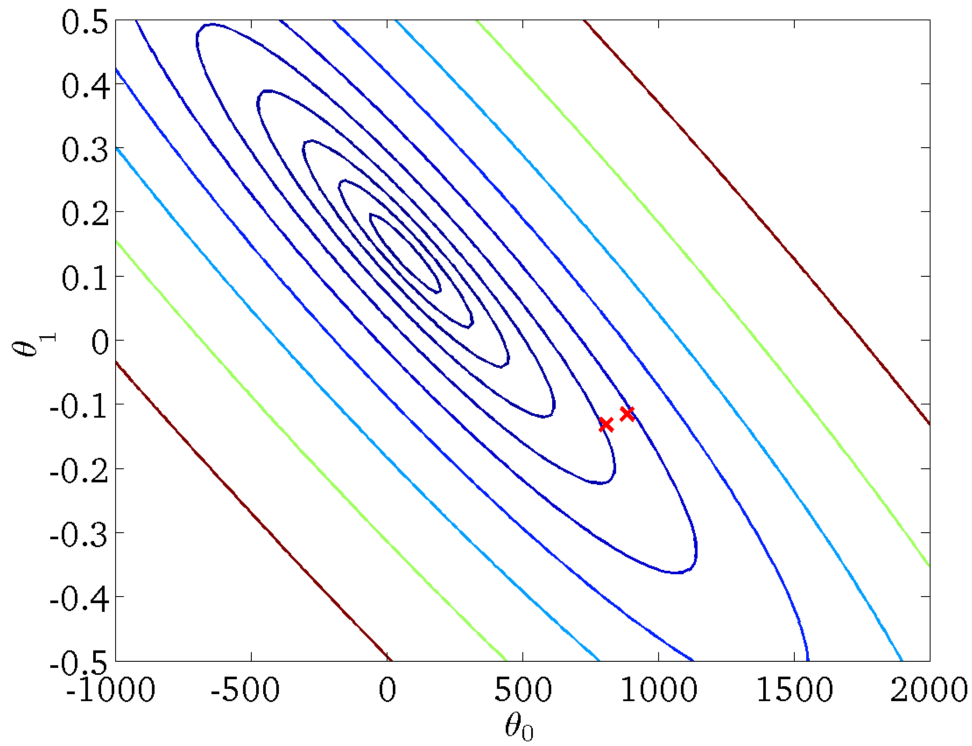
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



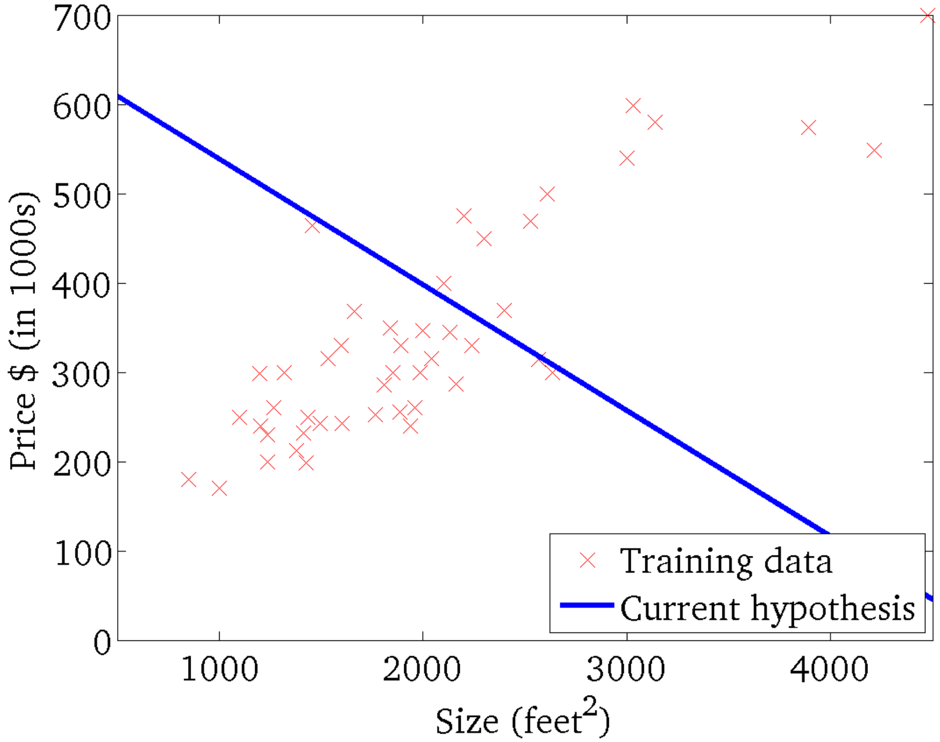
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



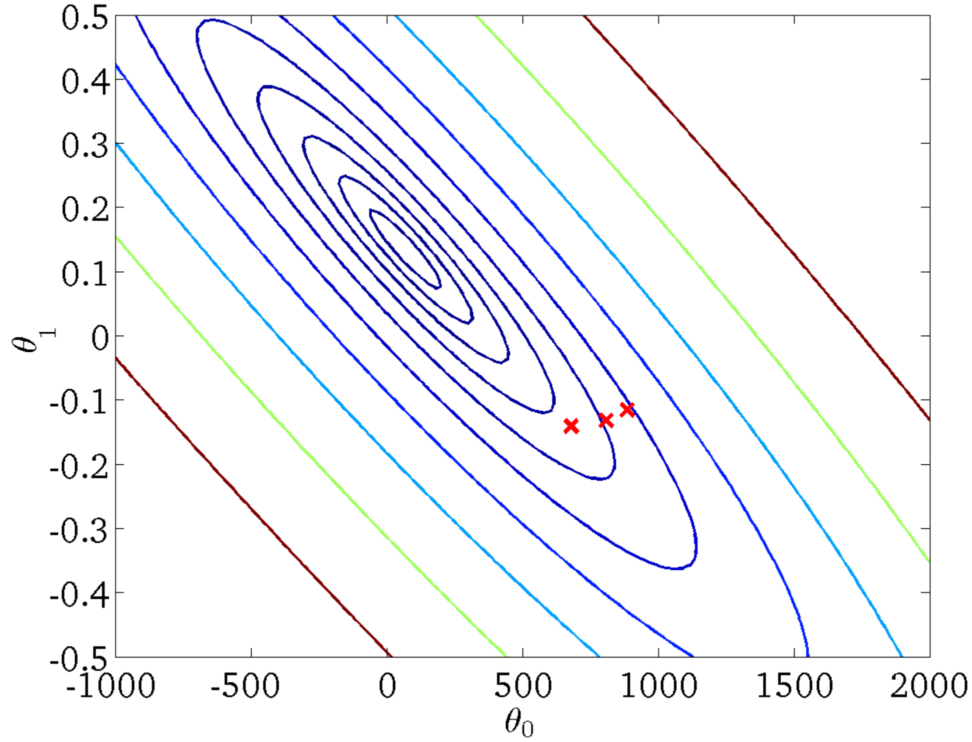
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



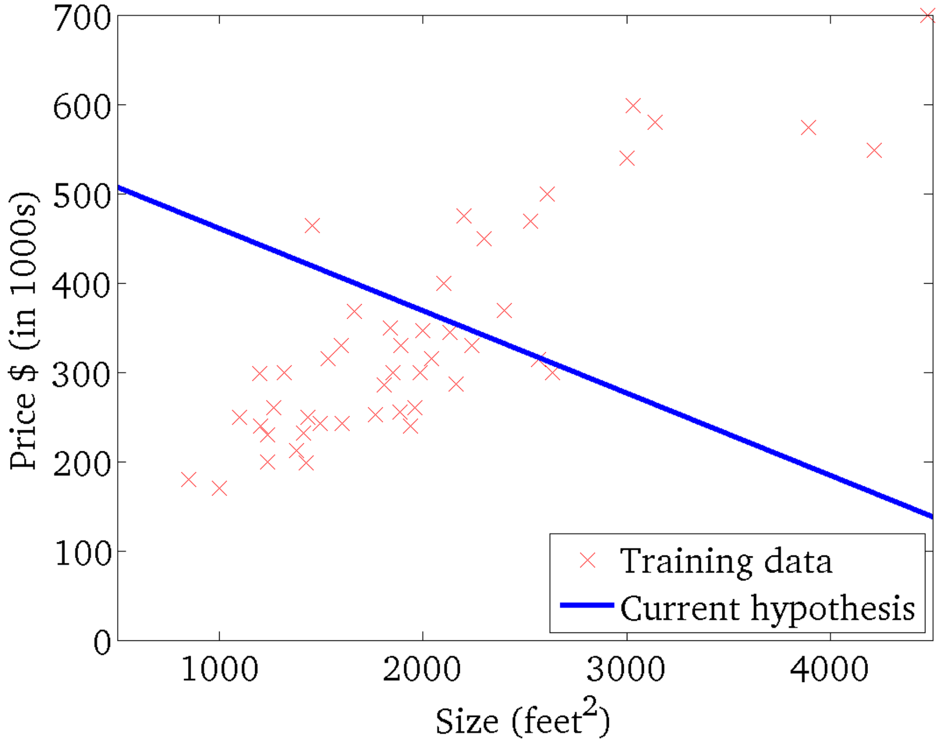
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



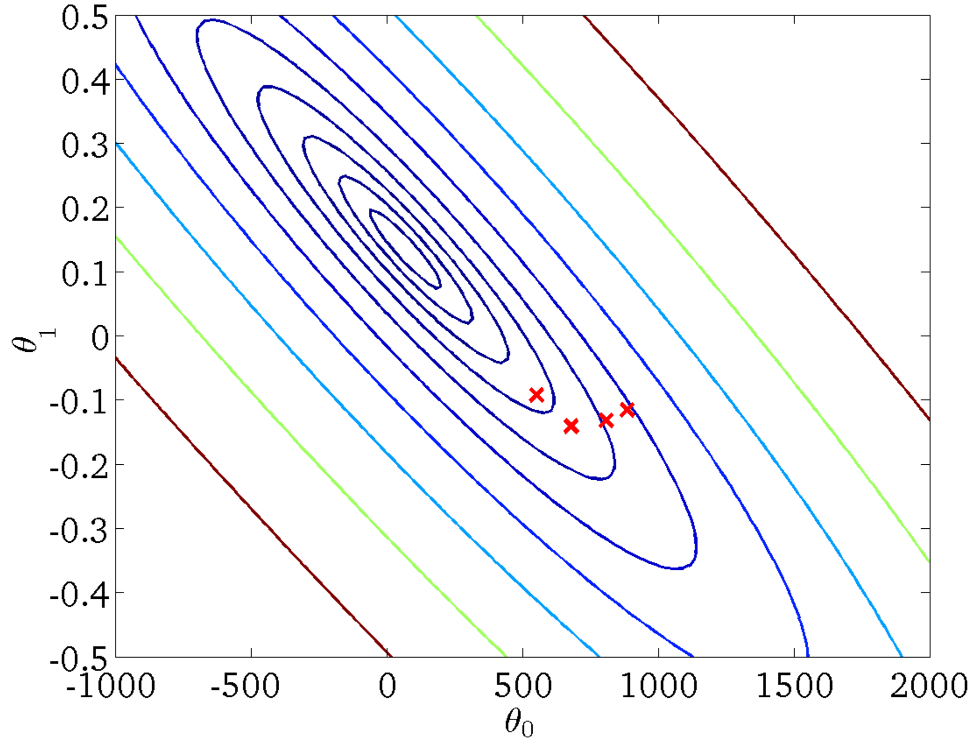
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



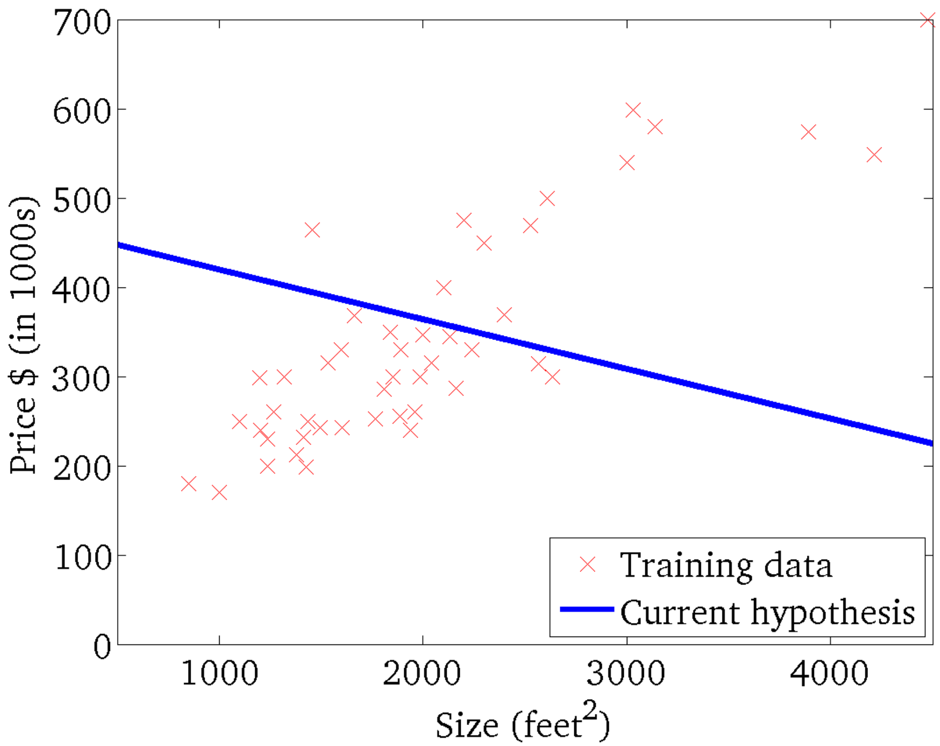
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



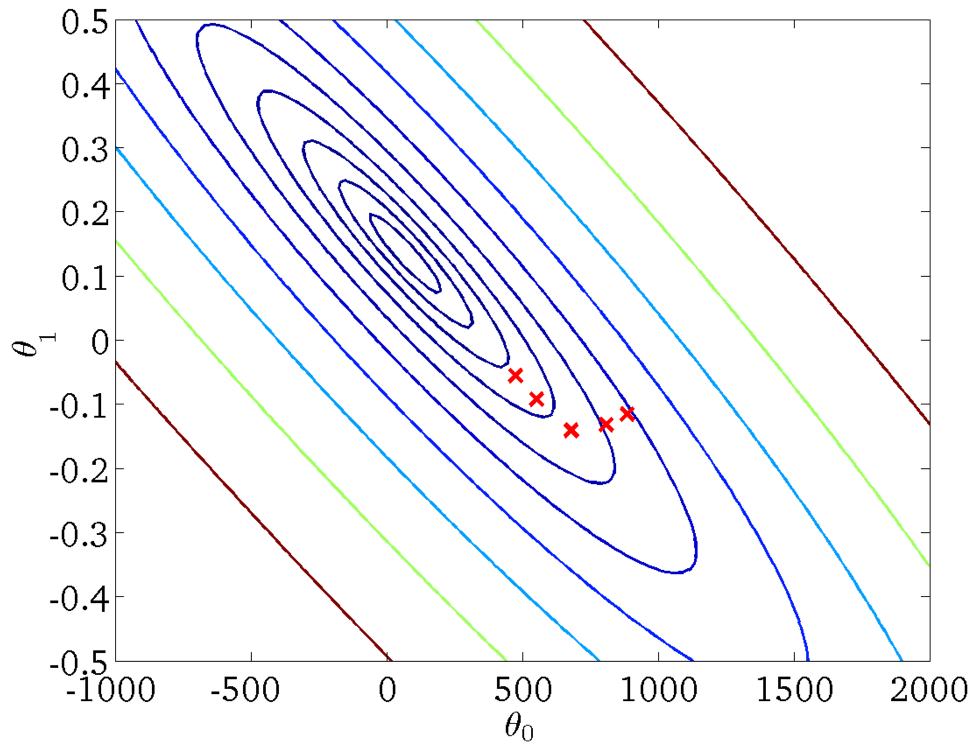
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



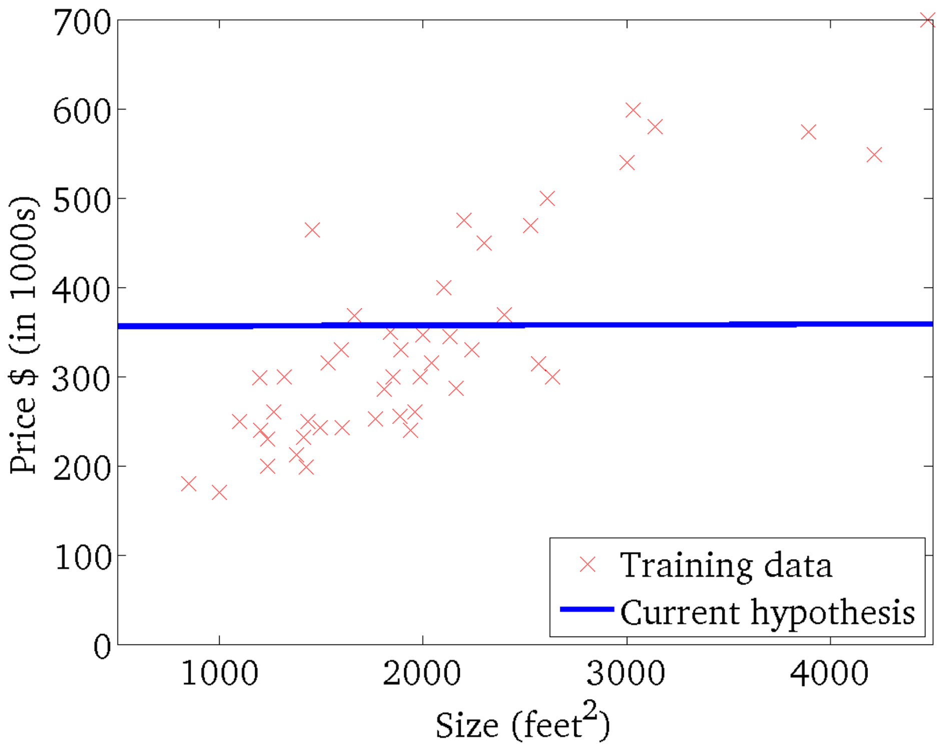
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



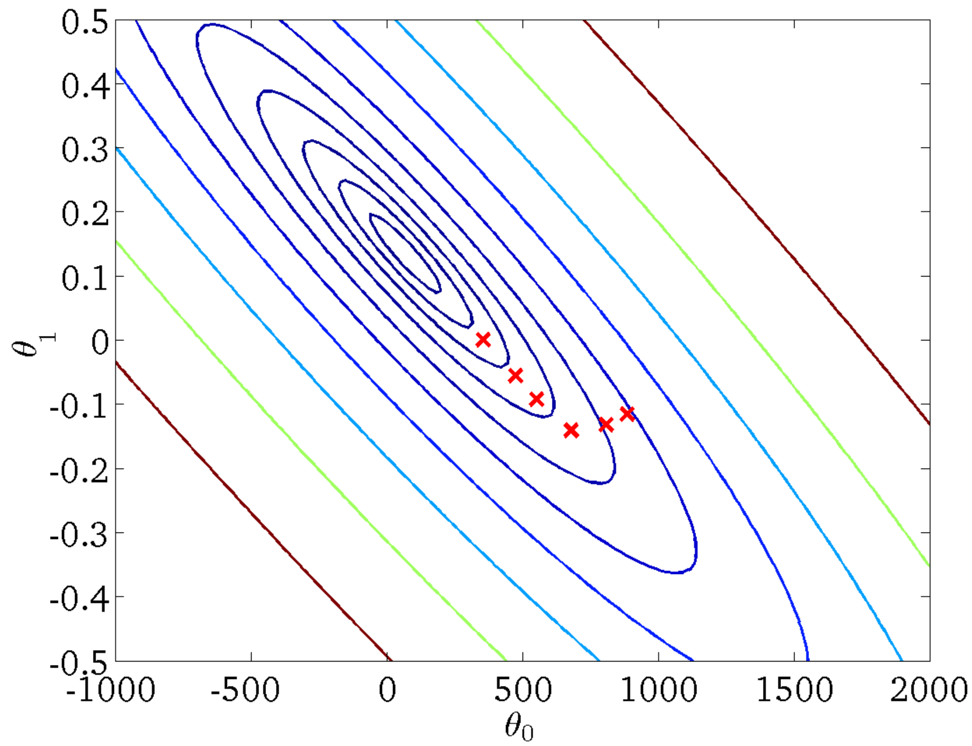
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



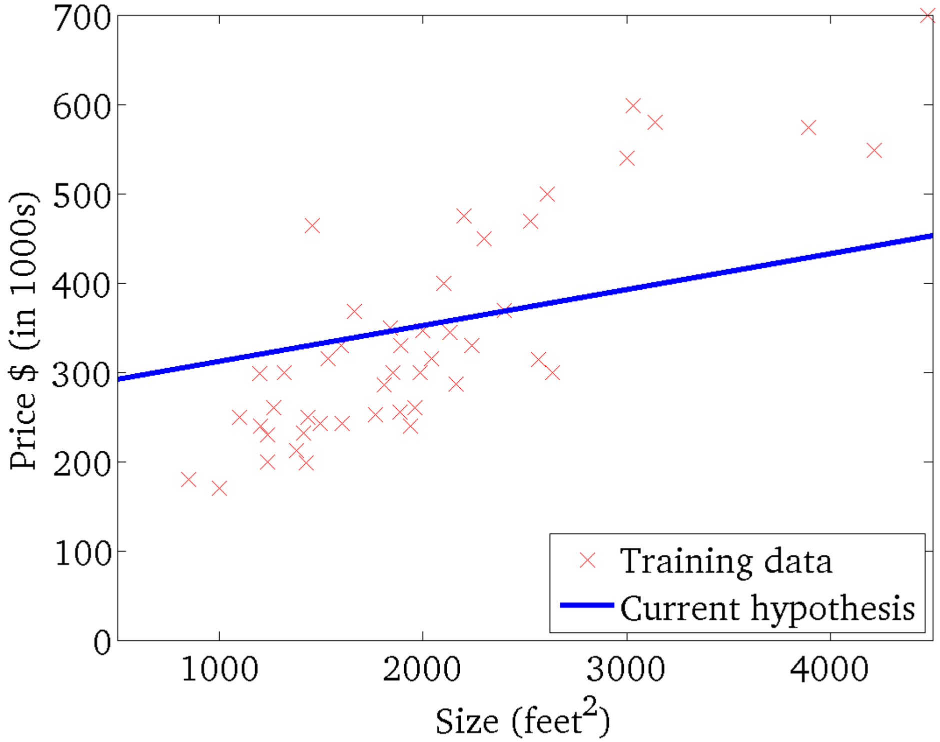
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



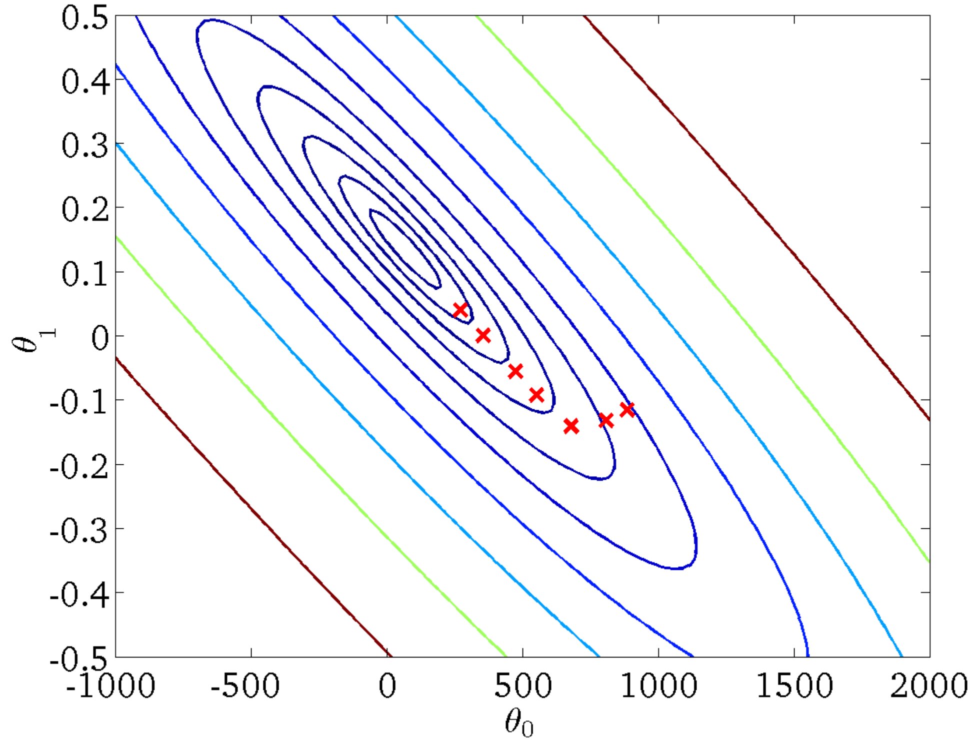
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



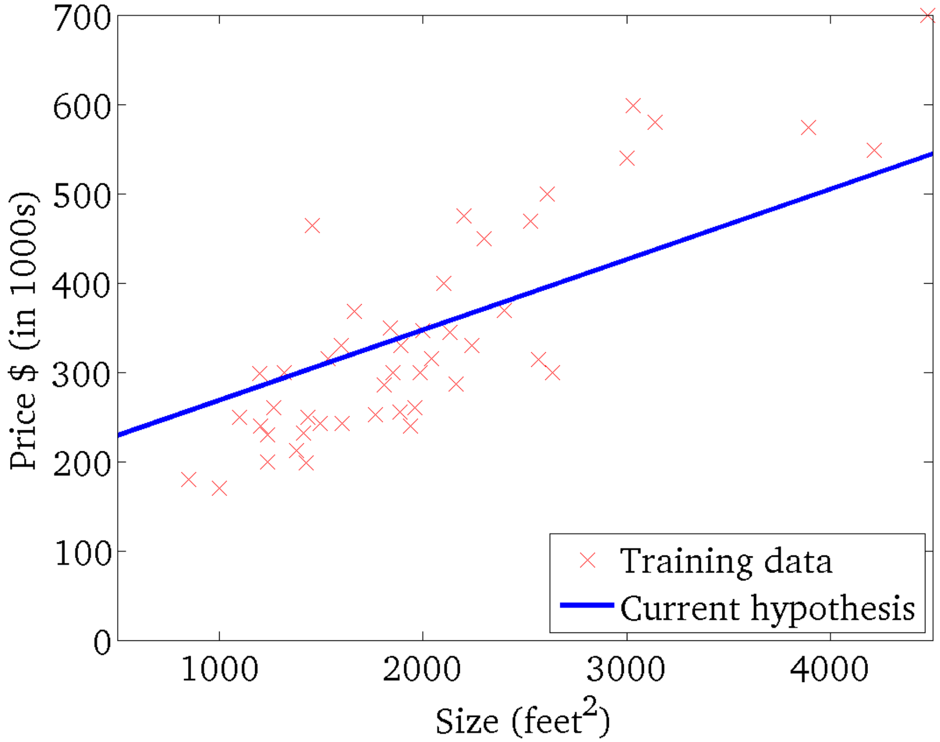
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



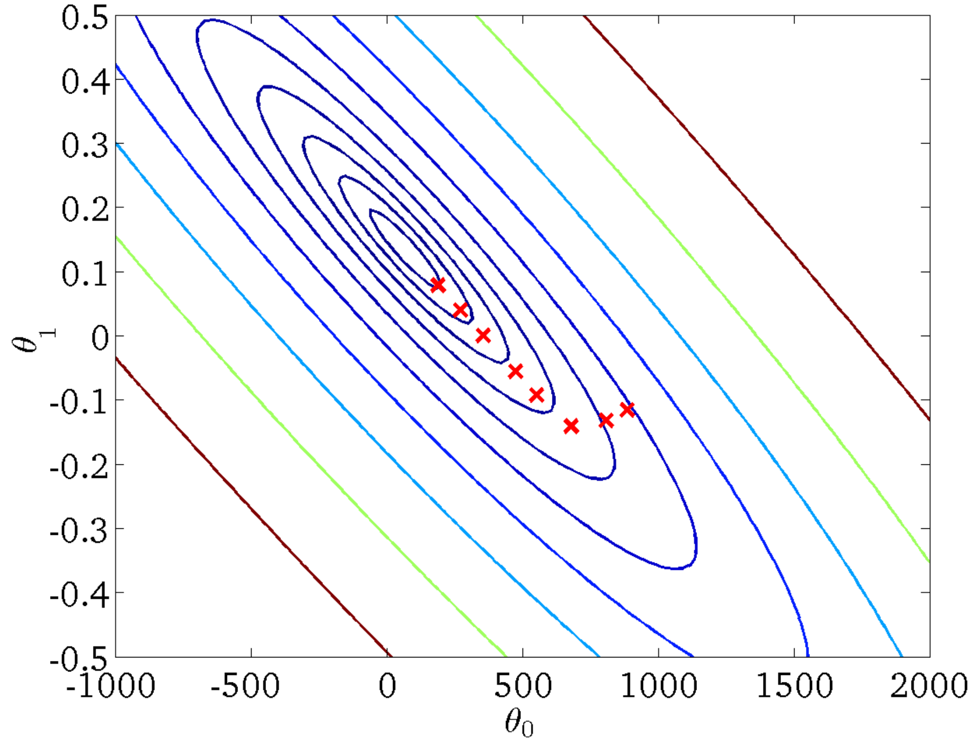
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



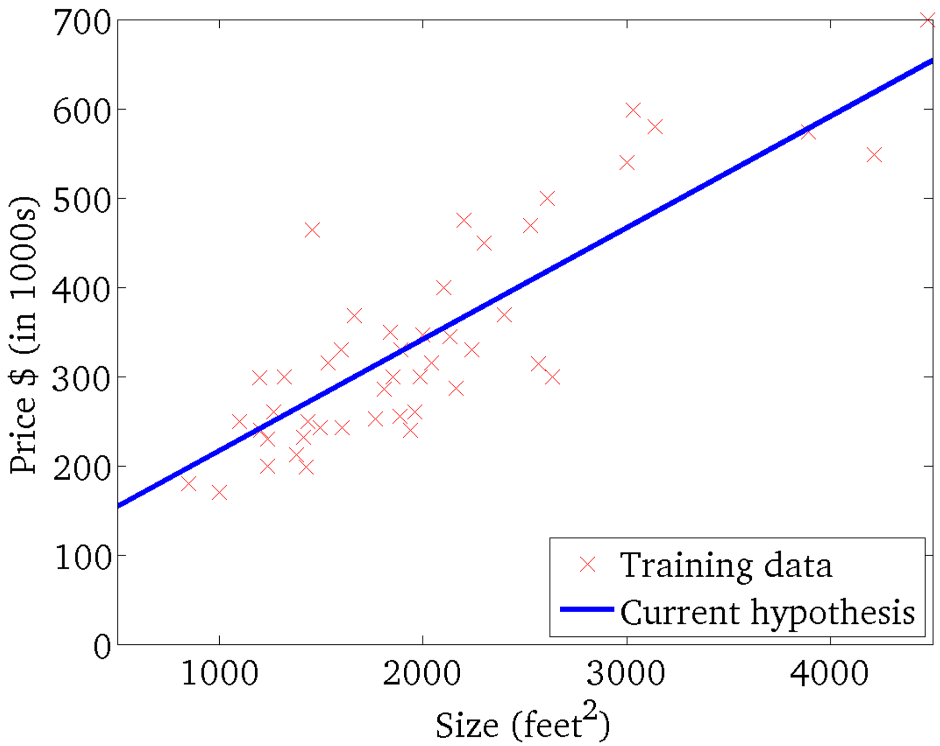
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



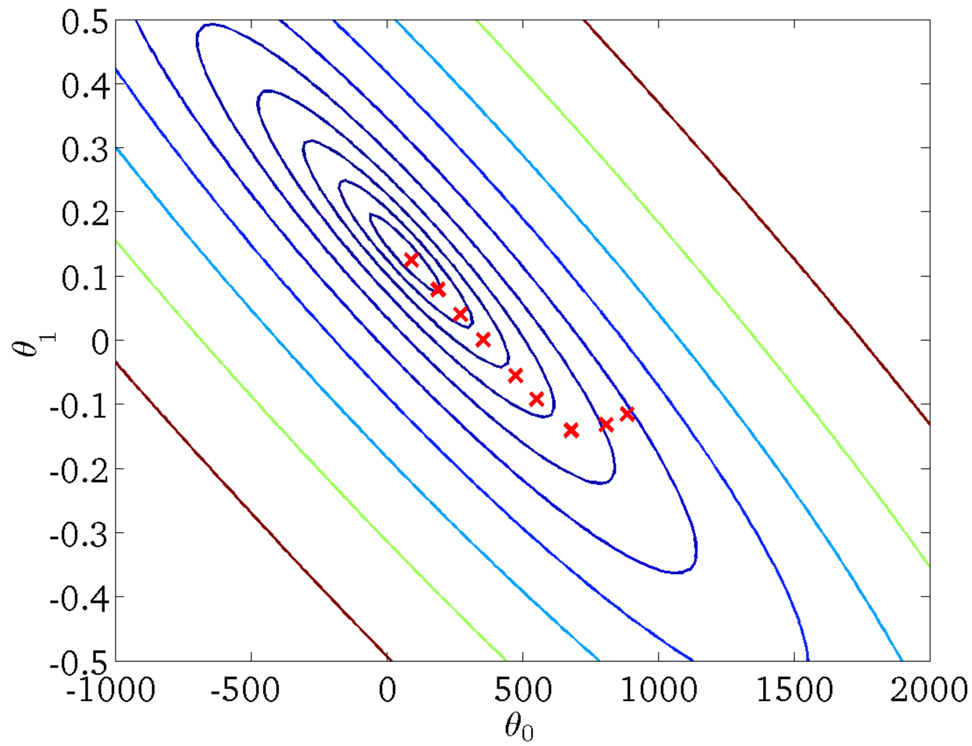
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



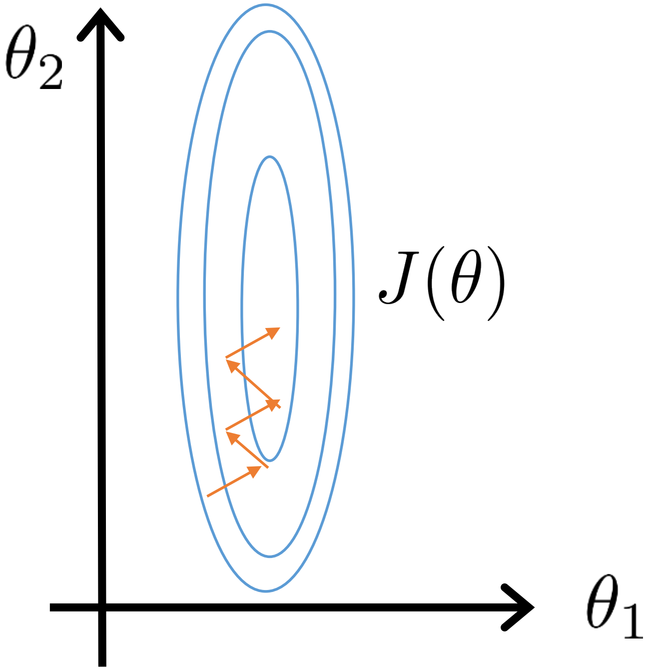
$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



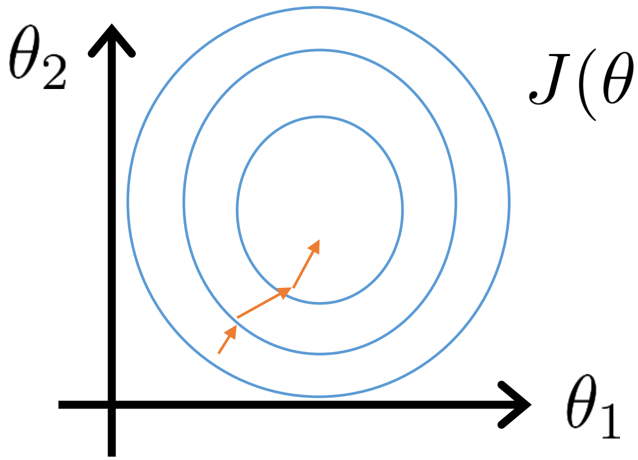
Feature Scaling

Problem: features are not on a similar scale

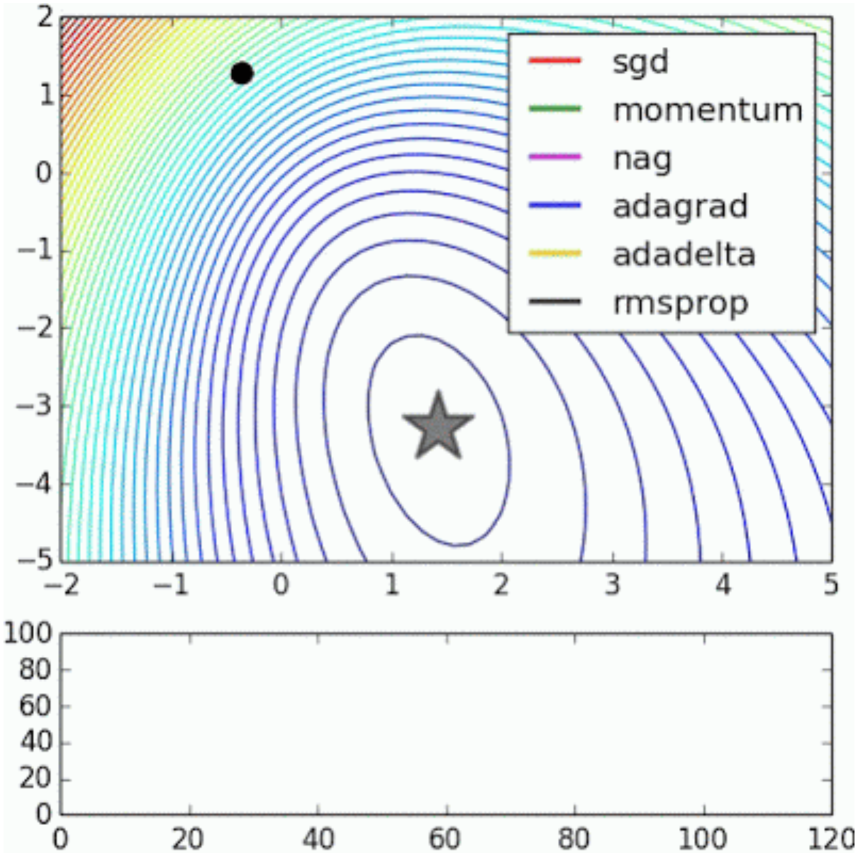


Solution: Mean Normalization

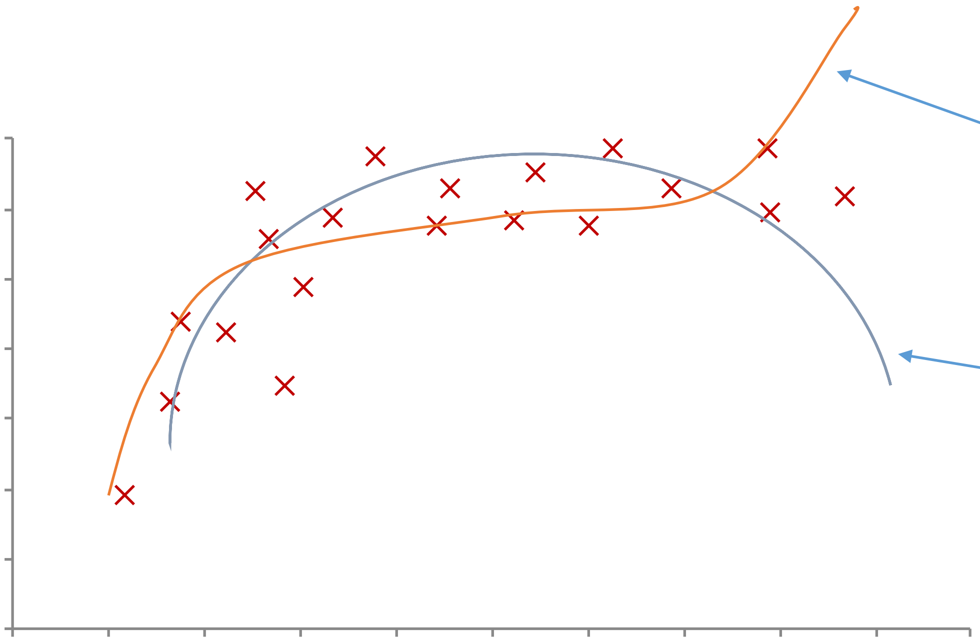
$$\frac{x_j - \mu_j}{\sigma_j} \qquad -1 \leq x_j \leq 1$$



Other Optimization Methods



Polynomial Regression



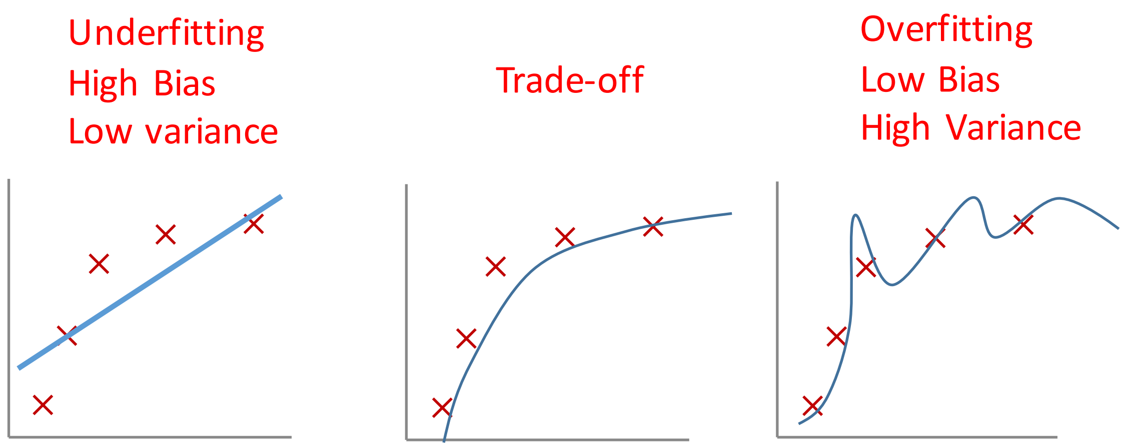
$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

Good but go up !

$$\theta_0 + \theta_1 x + \theta_2 x^2$$

Good but go down !

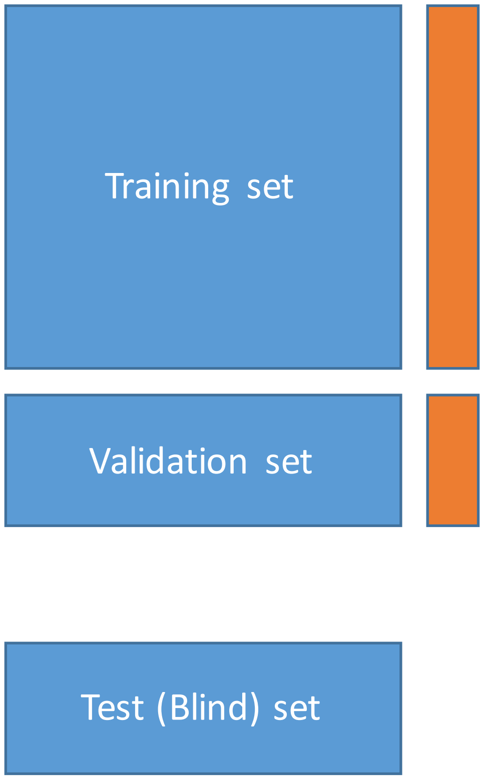
Overfitting vs. Underfitting

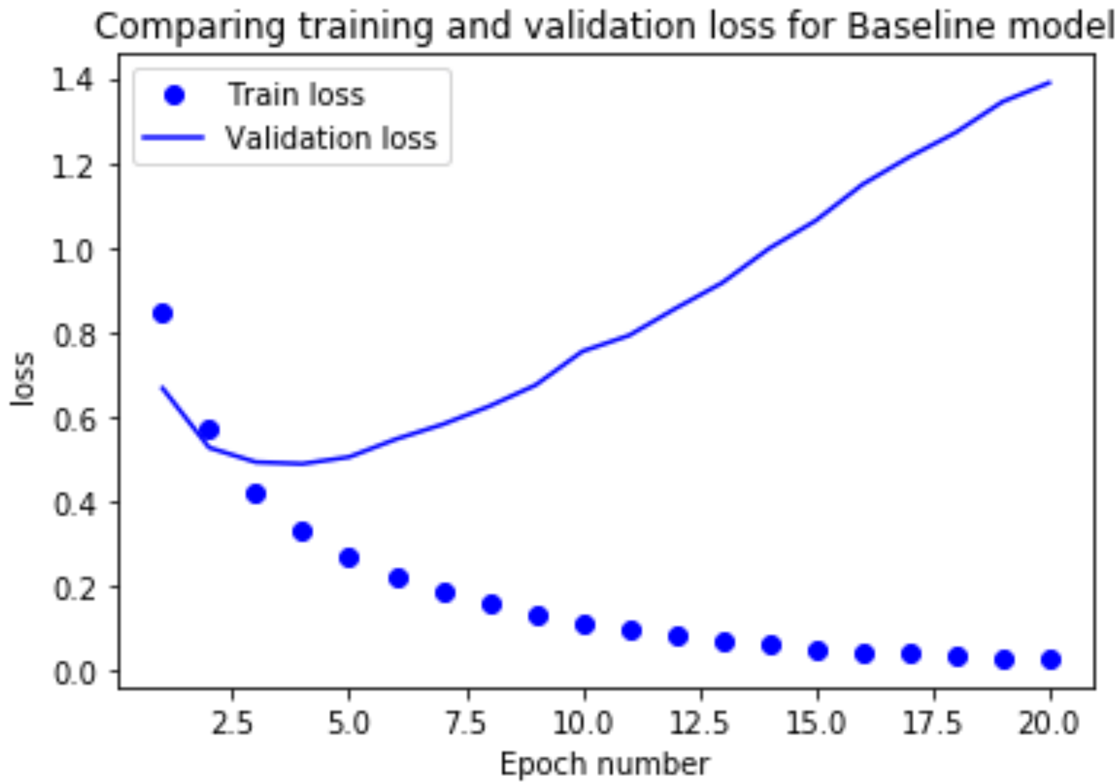


Address Overfitting

- Detect Overfitting
  - Performance analysis (Cross-Validation)
- Avoid Overfitting
  - Fewer parameters (Feature Selection)
  - Constraint the model (Regularization : minimum loss $\mathrm{L}(\theta) + \lambda\, \theta\theta^{T}$)
  - Tune hyper-parameters (Grid Search)

Performance Analysis





Performance Measures

- Measure of distance between predictions $\hat{y} = \mathrm{h}(\mathrm{x})$ and targets y

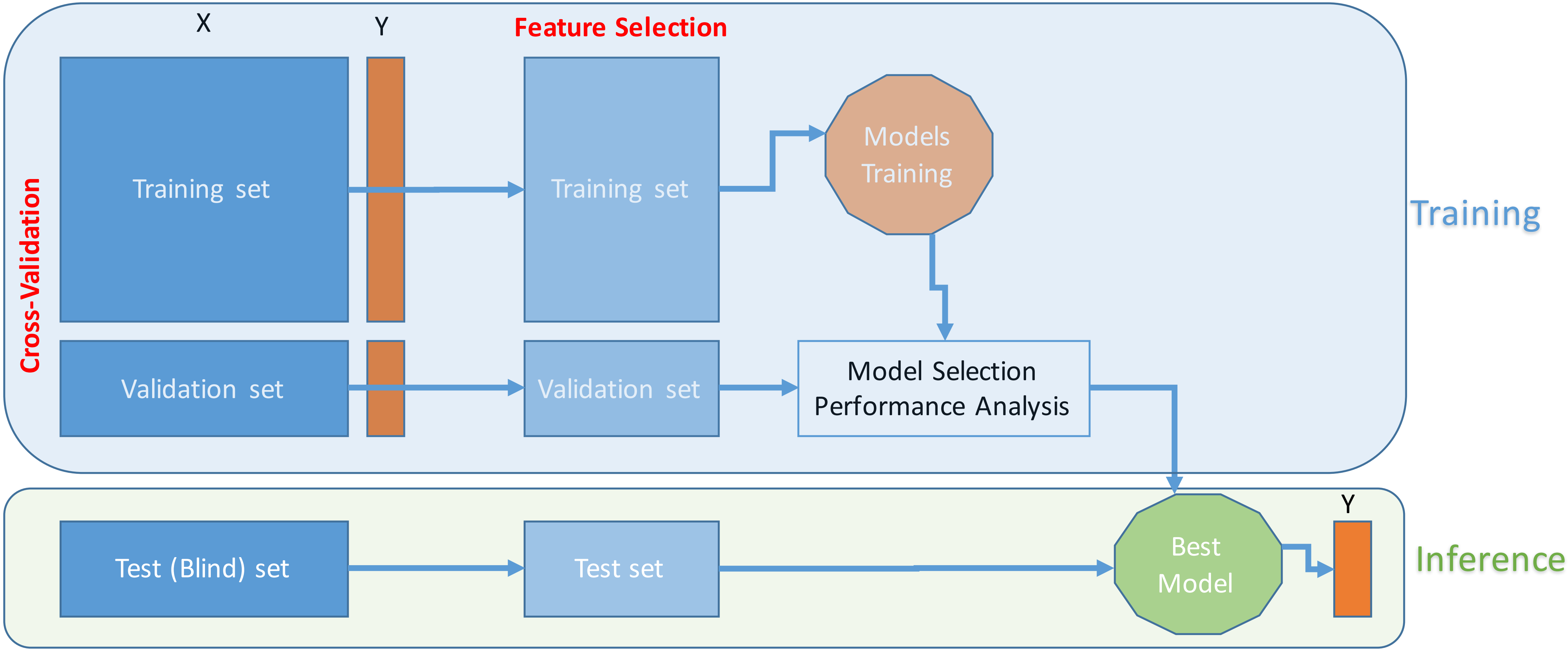
- L2 norm: Root Mean Square Error (RMSE)
  - Sensitive to outliers

$$\mathrm{RMSE}(\mathbf{X}, h) = \sqrt{\frac{1}{m} \sum_{i=1}^{m} \left(h\left(\mathbf{x}^{(i)}\right) - y^{(i)} \right)^2}$$

- L1 norm: Mean Absolute Error (MSE)

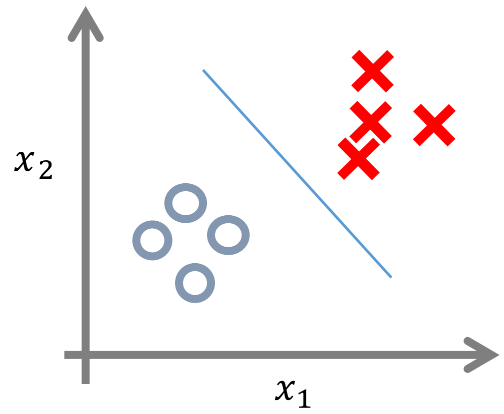
$$\mathrm{MAE}(\mathbf{X}, h) = \frac{1}{m} \sum_{i=1}^{m} \left| h\left(\mathbf{x}^{(i)}\right) - y^{(i)} \right|$$

Address Overfitting

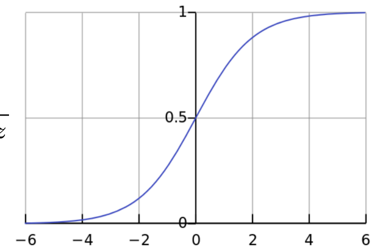


Logistic Regression

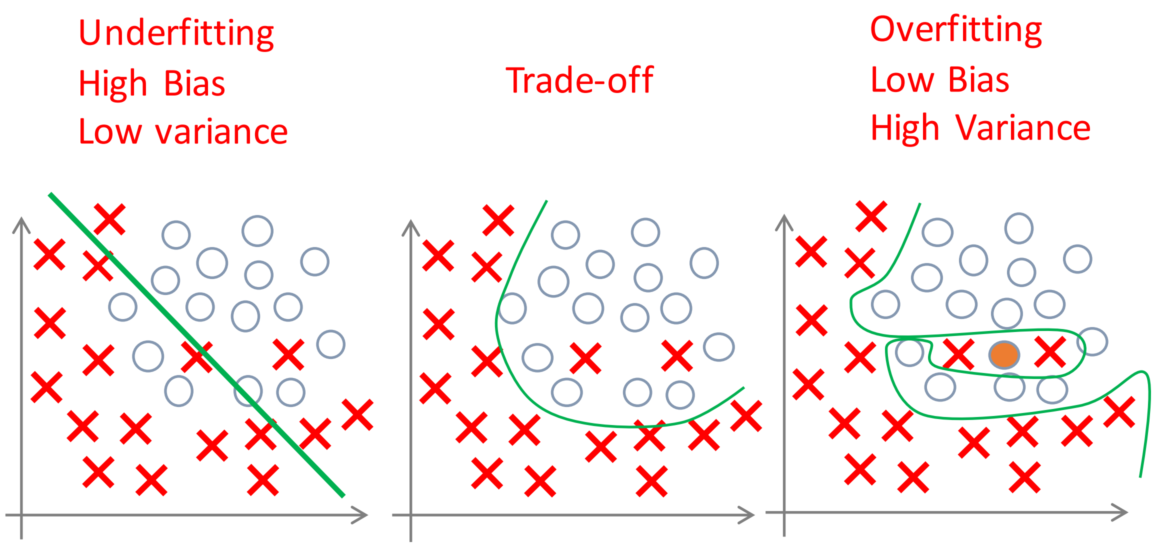
- The output y is discrete
- Classify X with a line $y = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$
- The best line is the one with minimum loss $\mathrm{L}(\theta)$
- Solved with gradient descent



$$g(z) = \frac{1}{1 + e^{-z}}$$



Overfitting vs. Underfitting



Performance Measures

- Confusion Matrix
- Accuracy
- Precision/Recall
- F1 score
- ROC curve
  - Receiver Operating Characteristic
  - Sensitivity versus (1 – Specificity)
- AUC
  - Area Under ROC curve

		Actual	
		Positives	Negatives
Predicted	Positives	TP	FP
	Negatives	FN	TN

Precision = $\frac{TP}{TP+FP}$ Recall = $\frac{TP}{TP+FN}$

Sensitivity = Recall Specificity= $\frac{TN}{FP+TN}$

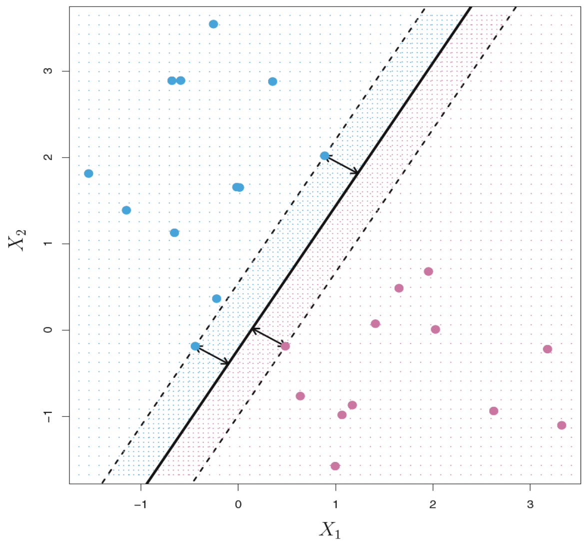
F1 = $\frac{2PR}{P+R}$

Linear and Logistic Regression

- Hyper-Parameters Tuning
  - λ : regularization hyper-parameter
  - d: degree of polynomial

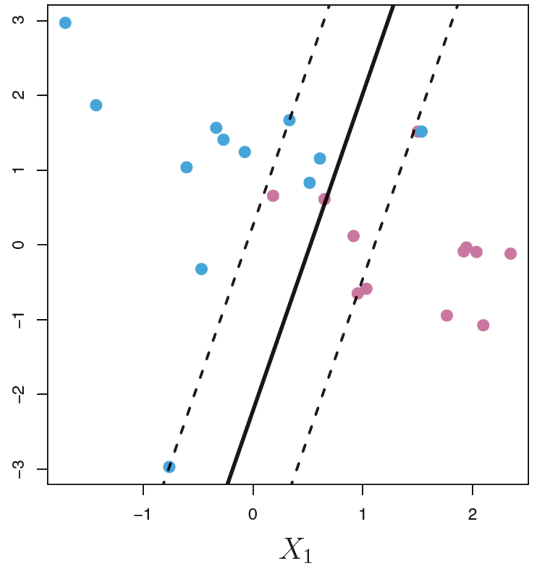
Support Vector Machines

Maximum Margin Classifier



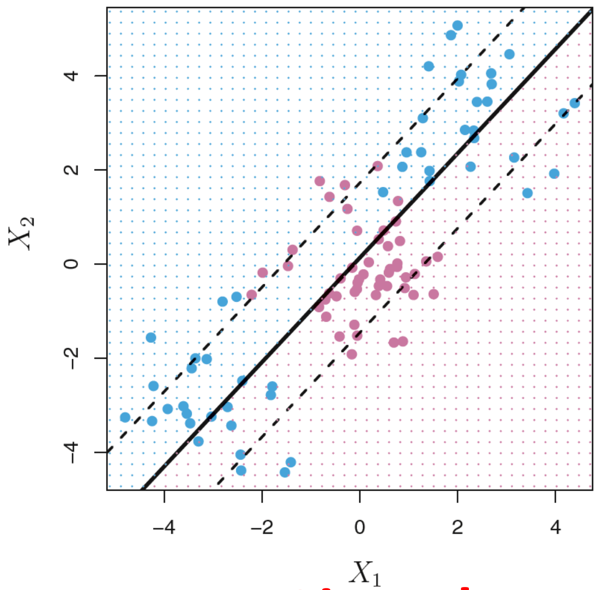
Linearly separable

Soft Margin Classifier



Slightly Linearly separable

Support Vector Machines

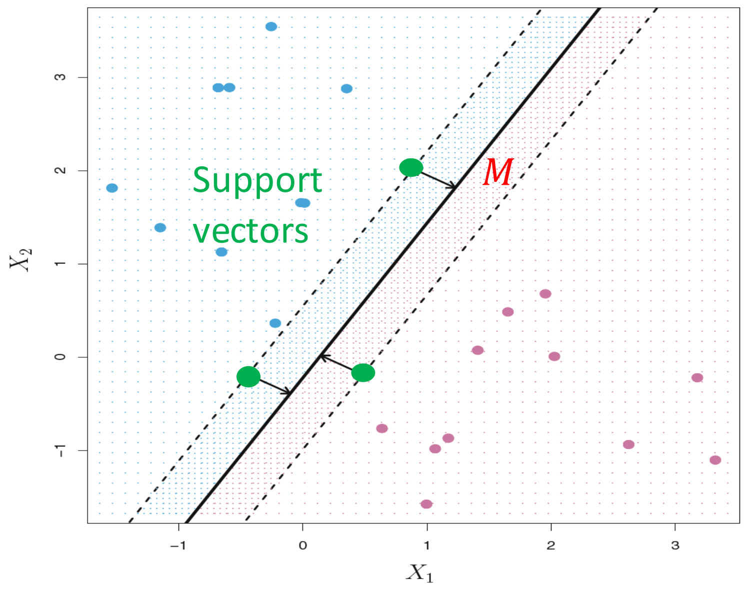


Non Linearly separable

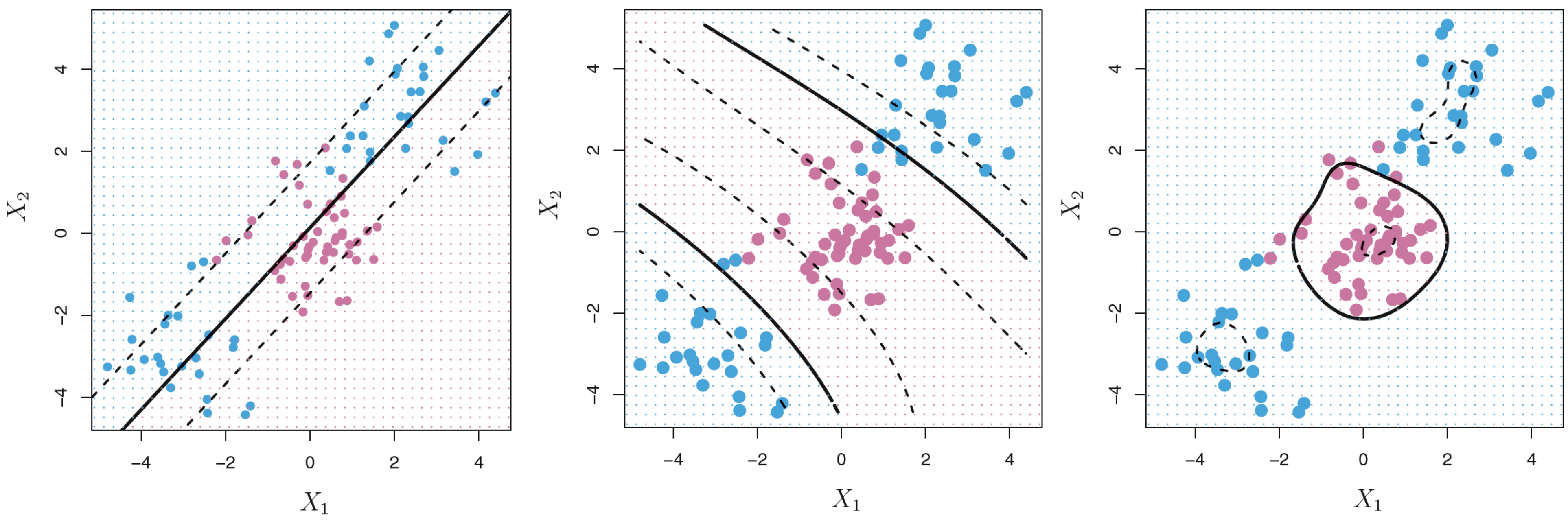
Support Vector Machines

- Support vectors defines the hyperplane
- The non-support vectors have no impact

- Classification
  - Fit the widest possible street between the classes



Support Vector Machines



Linear Kernel

Polynomial Kernel
d=3

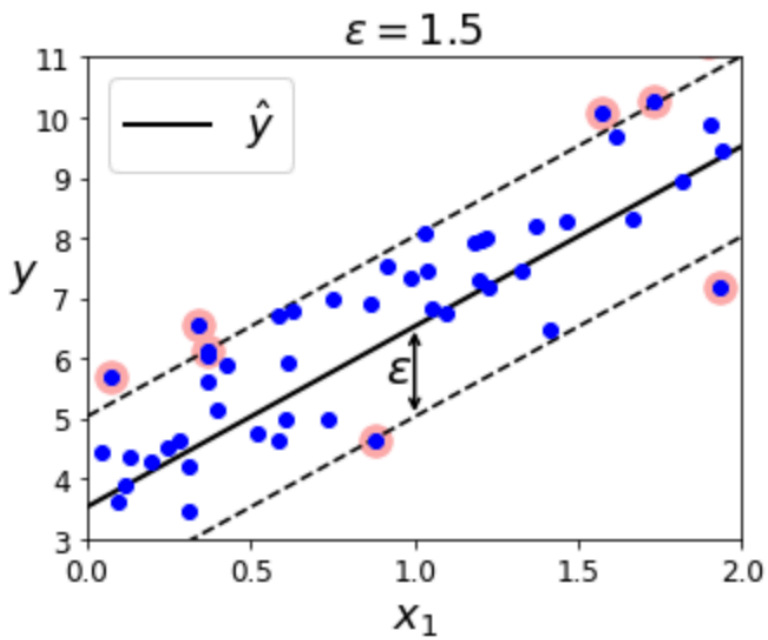
Radial Kernel
γ =0.1

Support Vector Machines

- Non linearly separable data become separable in higher space!
- So, first go to higher feature space $x -> \varphi(x)$
- To solve SVM, you have to compute the Kernel $K(u, v) = \varphi(u)^T \varphi(v)$
  - But: very costly !!!
- Kernel Trick: If you chose φ carefully, you end up getting K, without calculating the very costly dot product $\varphi(u)^T \varphi(v)$.
- Kernels
  - Linear Kernel $K(u,v) = u^T v$,
  - Polynomial Kernel: $K(u, v) = (c + u^T v)^d$,
  - Radial Basis Function (RBF) Kernel (Gaussian Kernel) : $K(u, v) = \exp(-\gamma \|u - v\|^2)$,
  - Etc.

Support Vector Machines

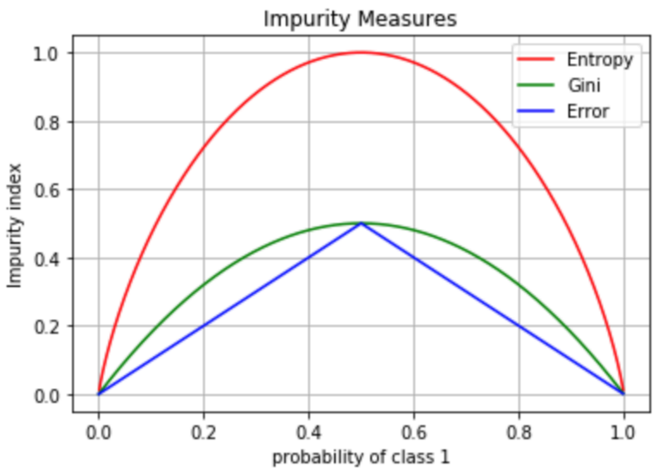
- Regression
    - Fit as many points as possible on the street while limiting margin violations.
    - The width of the street is controlled by a hyper-parameter ϵ



Support Vector Machines

- Hyper-Parameters Tuning
  - C, d: polynomial Kernel
  - γ: RBF kernel
  - ϵ: for regression
  - Etc.

Classification And Regression Trees

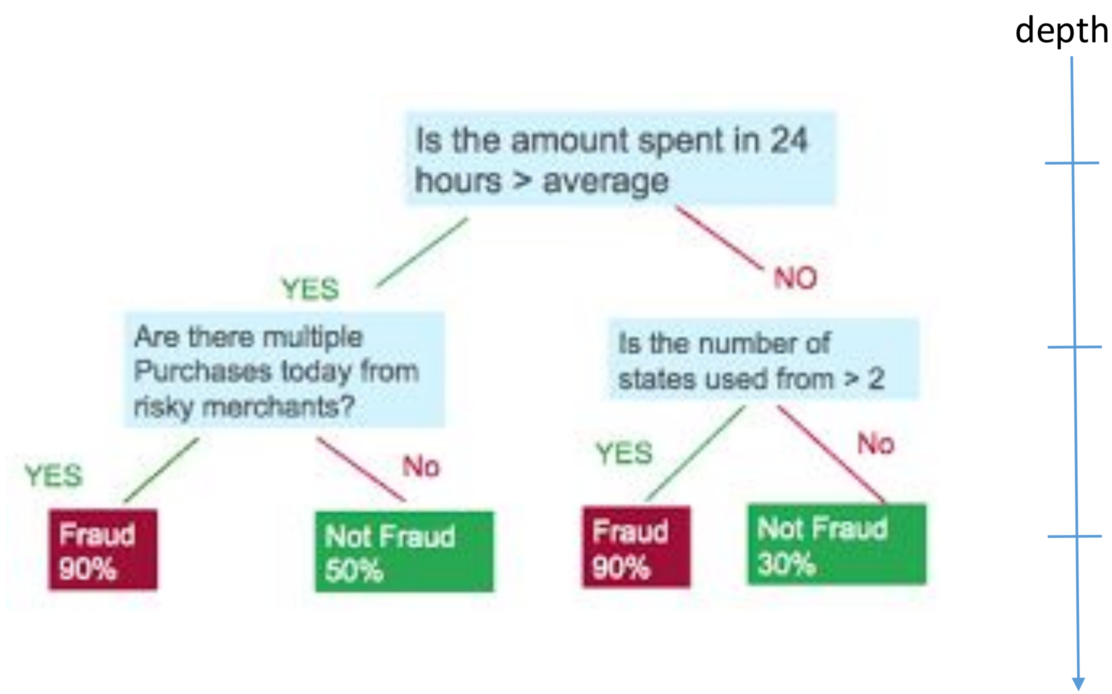


Classification

- Entropy: $H(S) = -\sum_{c} p_c \log_2(p_c)$
- Gini Index: $G(S) = 1 - \sum_{c} p_c^2$
- Class Error: $\mathrm{E}(S) = 1 - \max_{c}(p_c)$

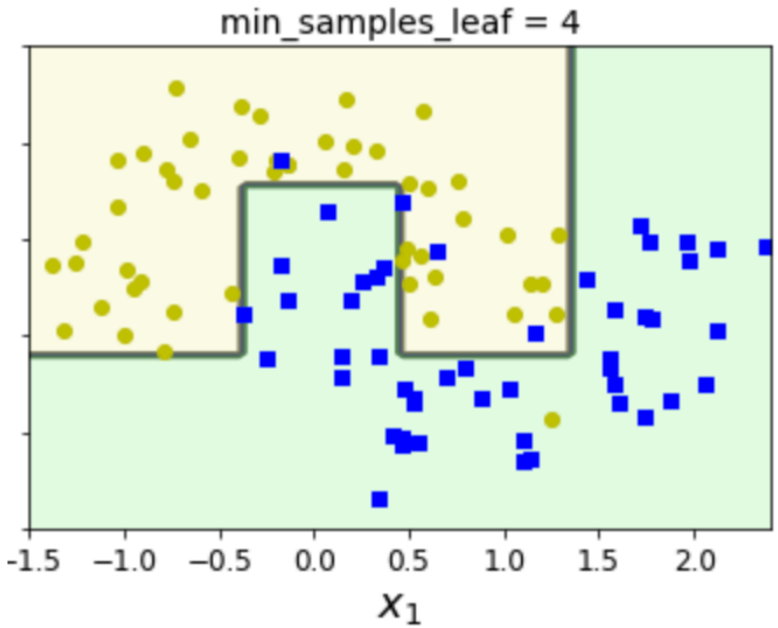
Regression

- RSS: $\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$

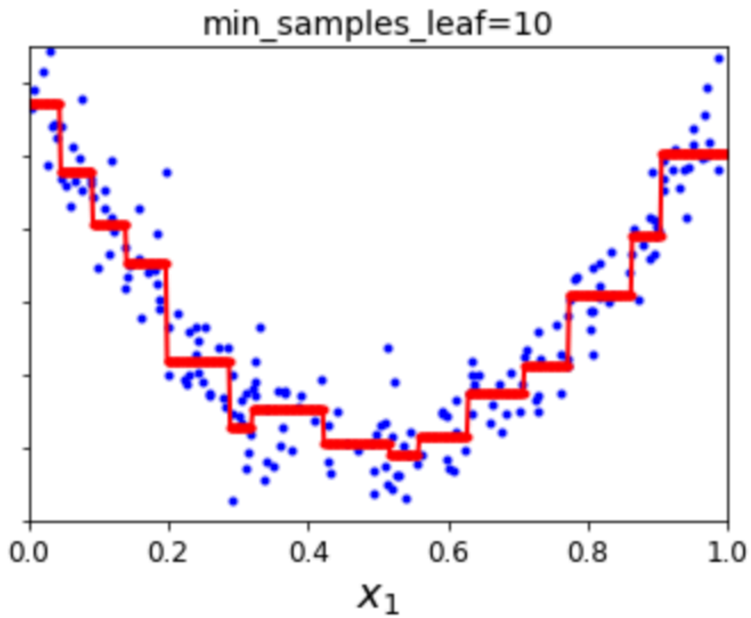


Classification And Regression Trees

Classification (Decision) Trees



Regression Trees

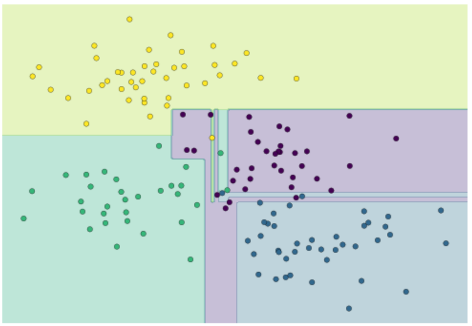


Classification And Regression Trees

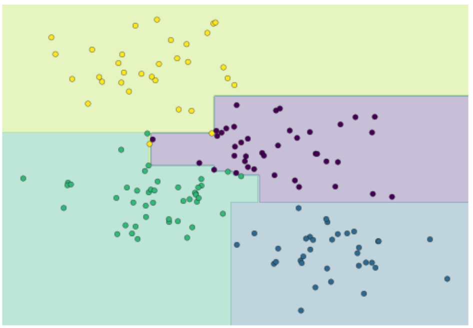
- Advantages
  - Easy to interpret
  - Deals with non linearity
  - Handle qualitative features without the need to create fictive ones (one hot vector)
  - Provide most important features (in terms of information gain)

- Disadvantages
  - Leads to overfitting (high variance): little change in little number of examples affect the whole tree.

DT on Data



DT on half of Data

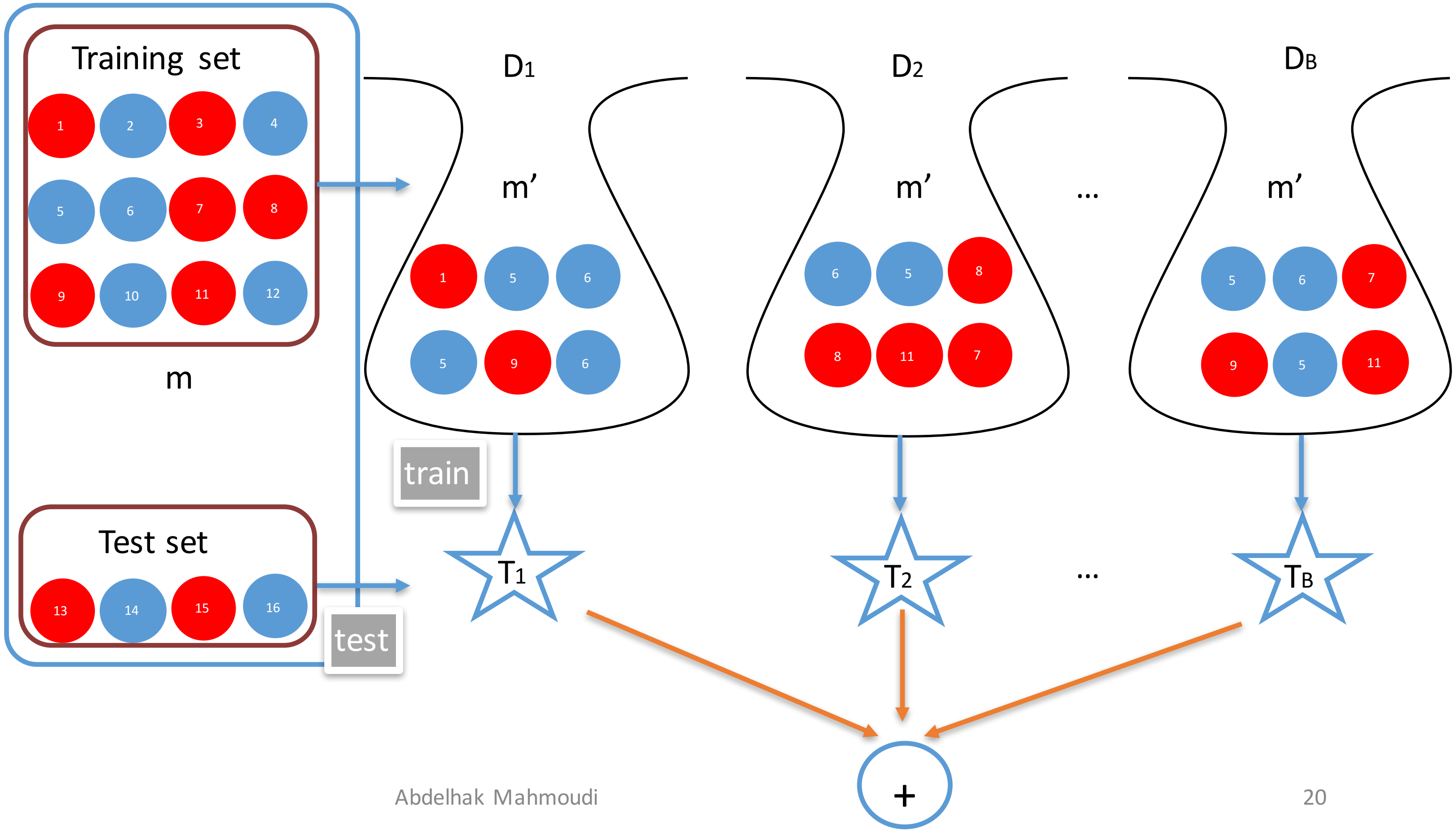


Random Forest

Bagging
Bootstrap Aggregating

Training
Pick m' examples with replacement and train B trees

Testing
Regression: mean errors of all the B trees
Classification: vote

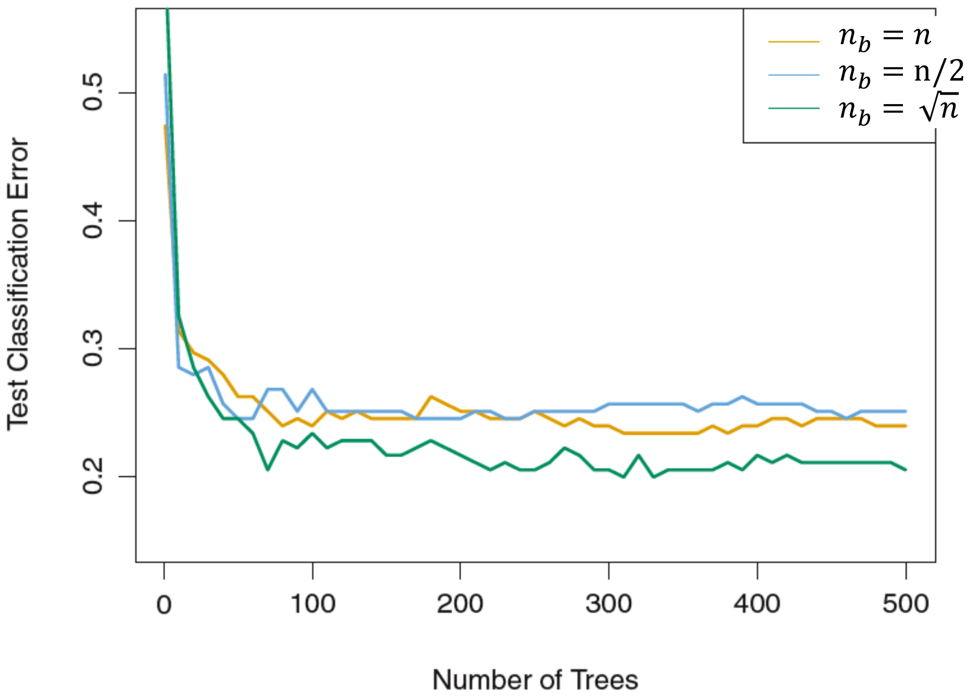


Random Forest

- **Problem:** Bagged trees will look quite similar to each other, so averaging them will not led to much reduction of variance!

- **Solution:** Random Forest constructs multiple trees where each tree uses n_b random features from the n initial features (generally $n_b = \sqrt{n}$)

- $n_b = n$-> Bagging case



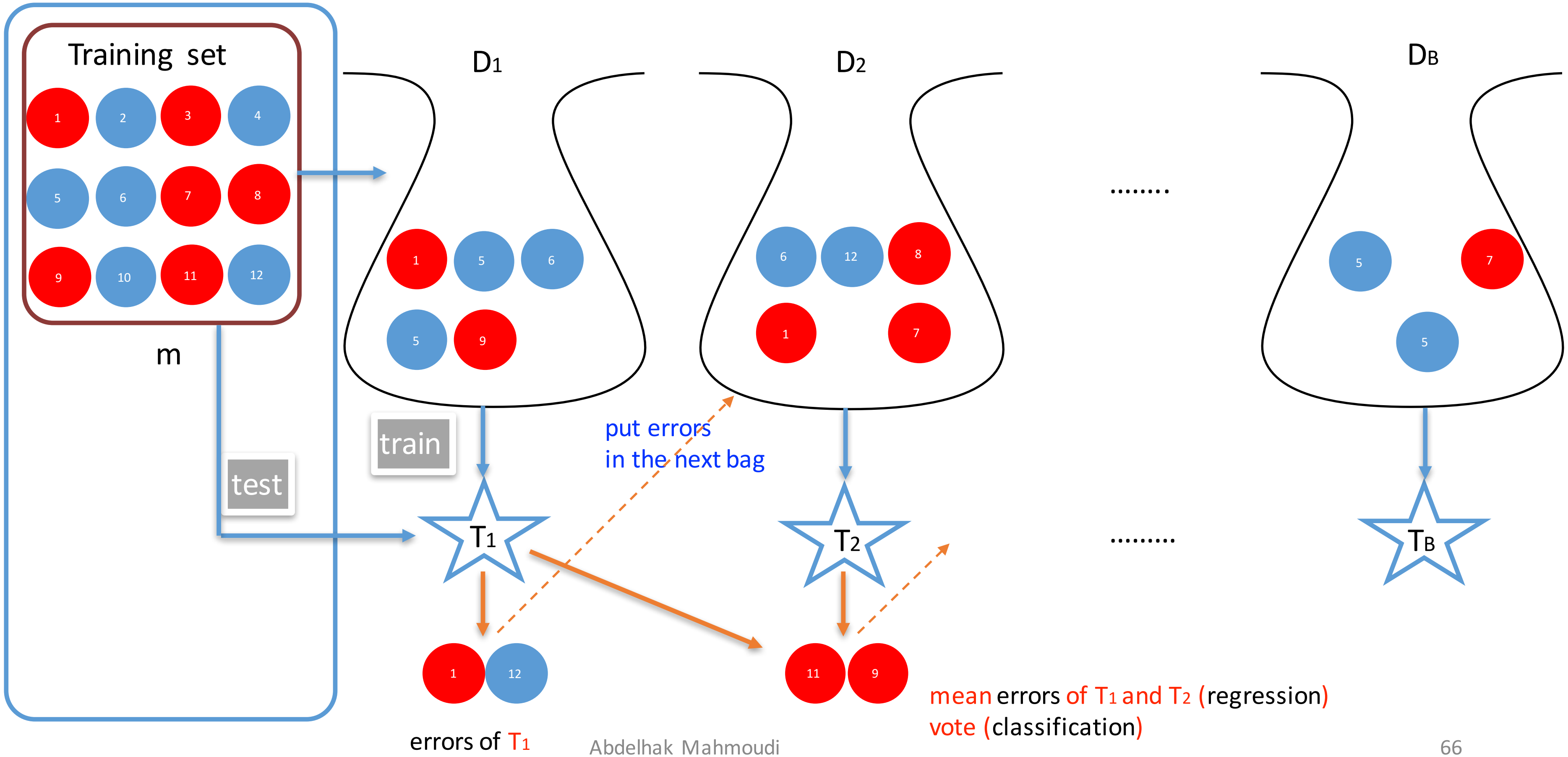
Random Forest

- Both training and prediction are very fast, because of the simplicity of the underlying decision trees.
- Tasks can be straightforwardly parallelized, because the individual trees are entirely independent entities.
- The multiple trees allow for a probabilistic classification: a majority vote among estimators gives an estimate of the probability
- RF is a Nonparametric model, extremely flexible, and can thus perform well on tasks that are under-fit by other models.

Random Forest

- Hyper-Parameters Tuning
  - d: Depth of the trees
  - B: number of Bags

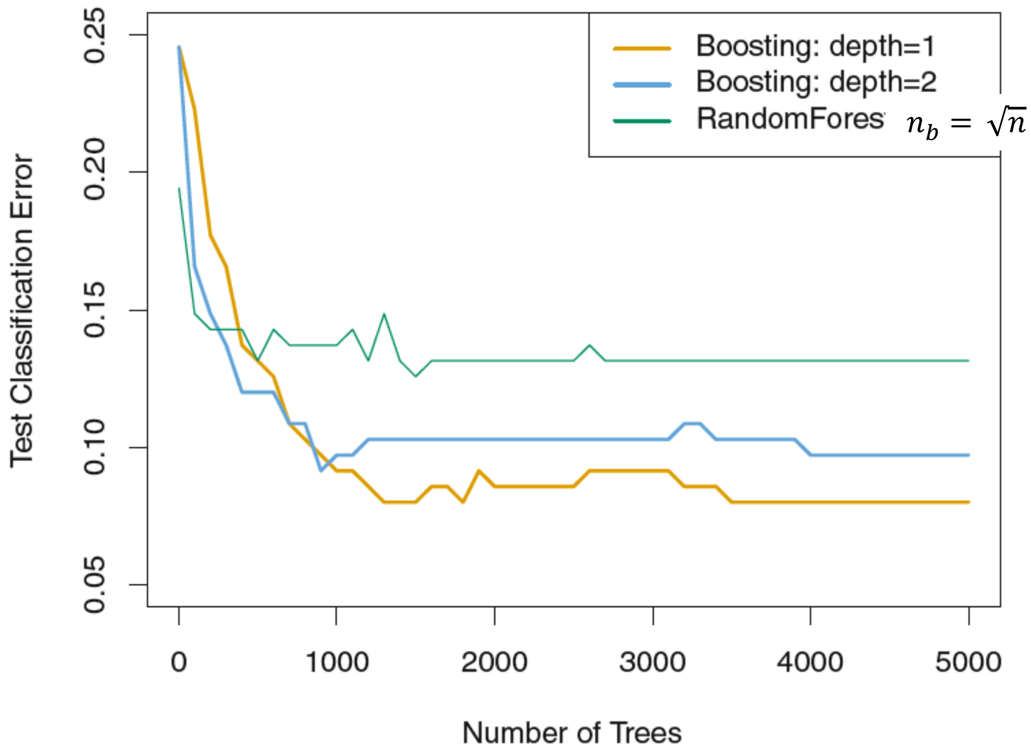
Boosting



Training set
1
2
3
4
5
6
7
8
9
10
11
12
m
test
train
D1
1
5
6
5
9
D2
6
12
8
1
7
....
DB
5
7
5
T1
T2
....
TB
put errors
in the next bag
1
12
errors of T1
11
9
mean errors of T1 and T2 (regression)
vote (classification)

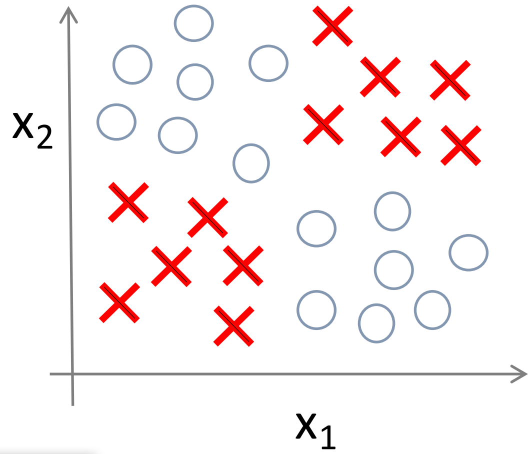
Boosting

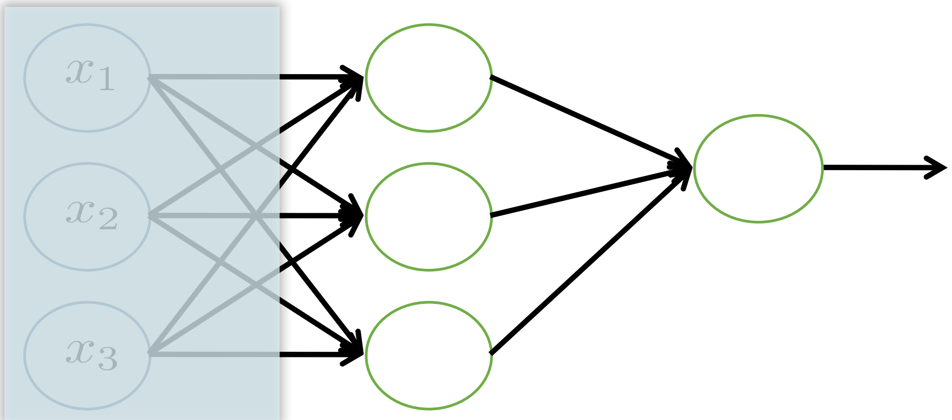
- Outperforms RF
- Smaller Trees (depth = 1) are sufficient because the growth of a particular tree takes into account preceding trees.
- Smaller trees can aid in interpretability.
- Boosting (Freund & Schapire 1990)
- Adaboost (Adaptive Boosting), 1996



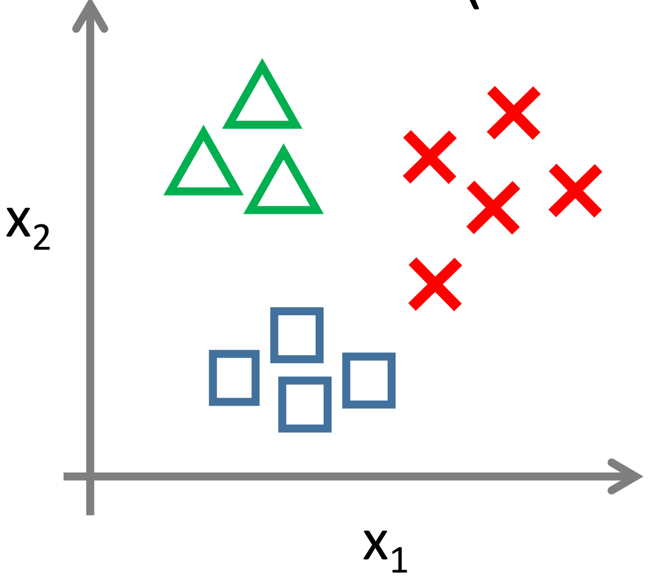
Artificial Neural Networks

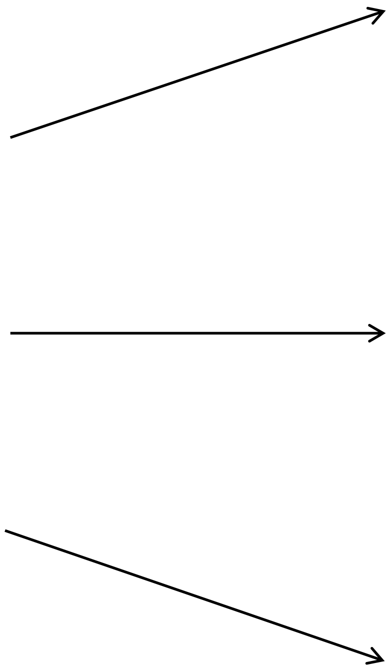
- Structured vs unstructured data
- Why ANN?
  - Learn Features by it self
  - Data non linearly separable
- Different types of Architectures
  - Convolutional Neural Networks (Vision)
  - Recurrent Neural Networks (Sequence)
  - Generative Adversarial Networks (Generate data)

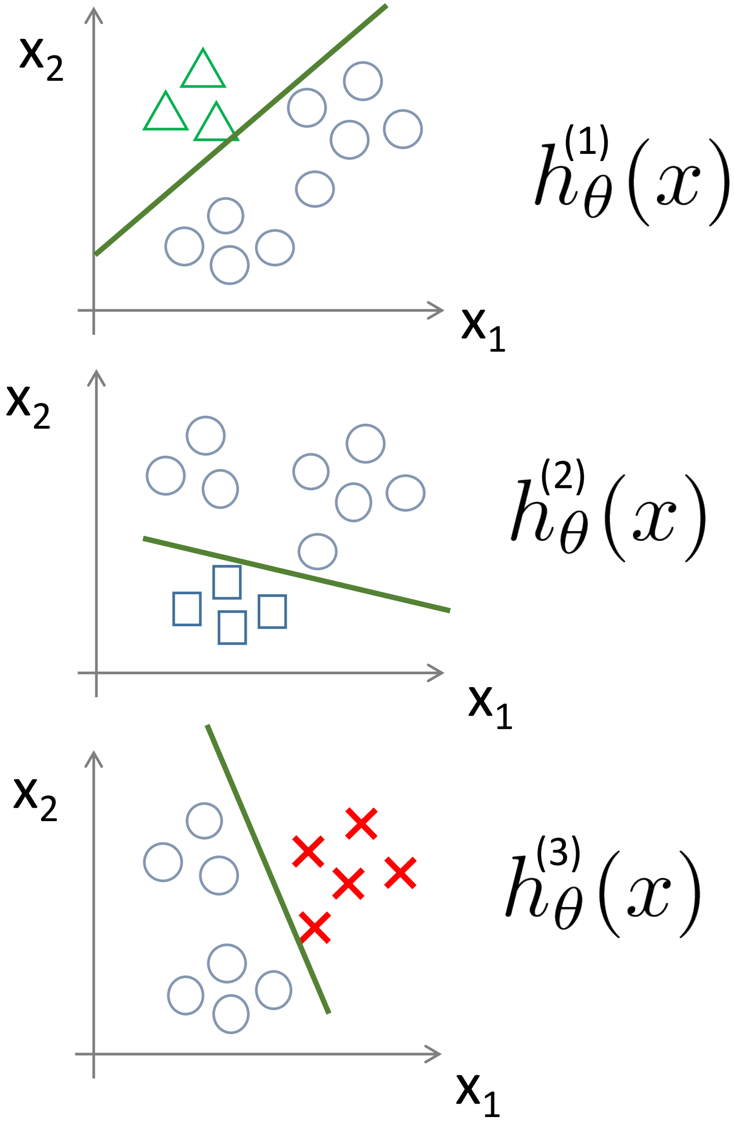




Multi-Class (N-classes)

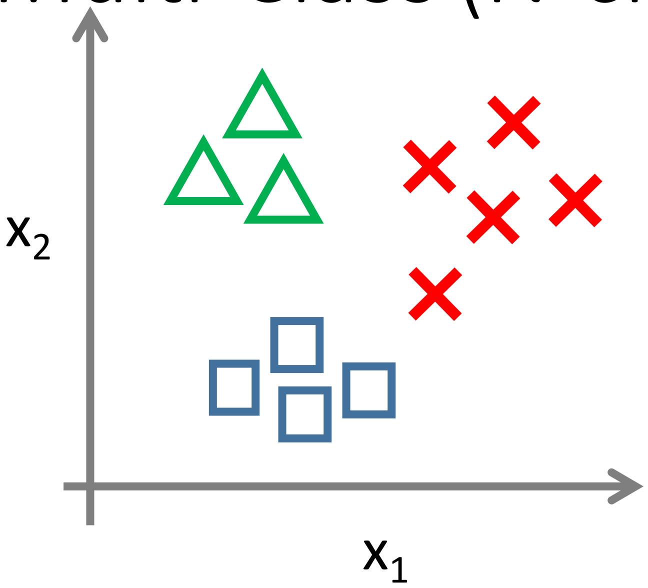


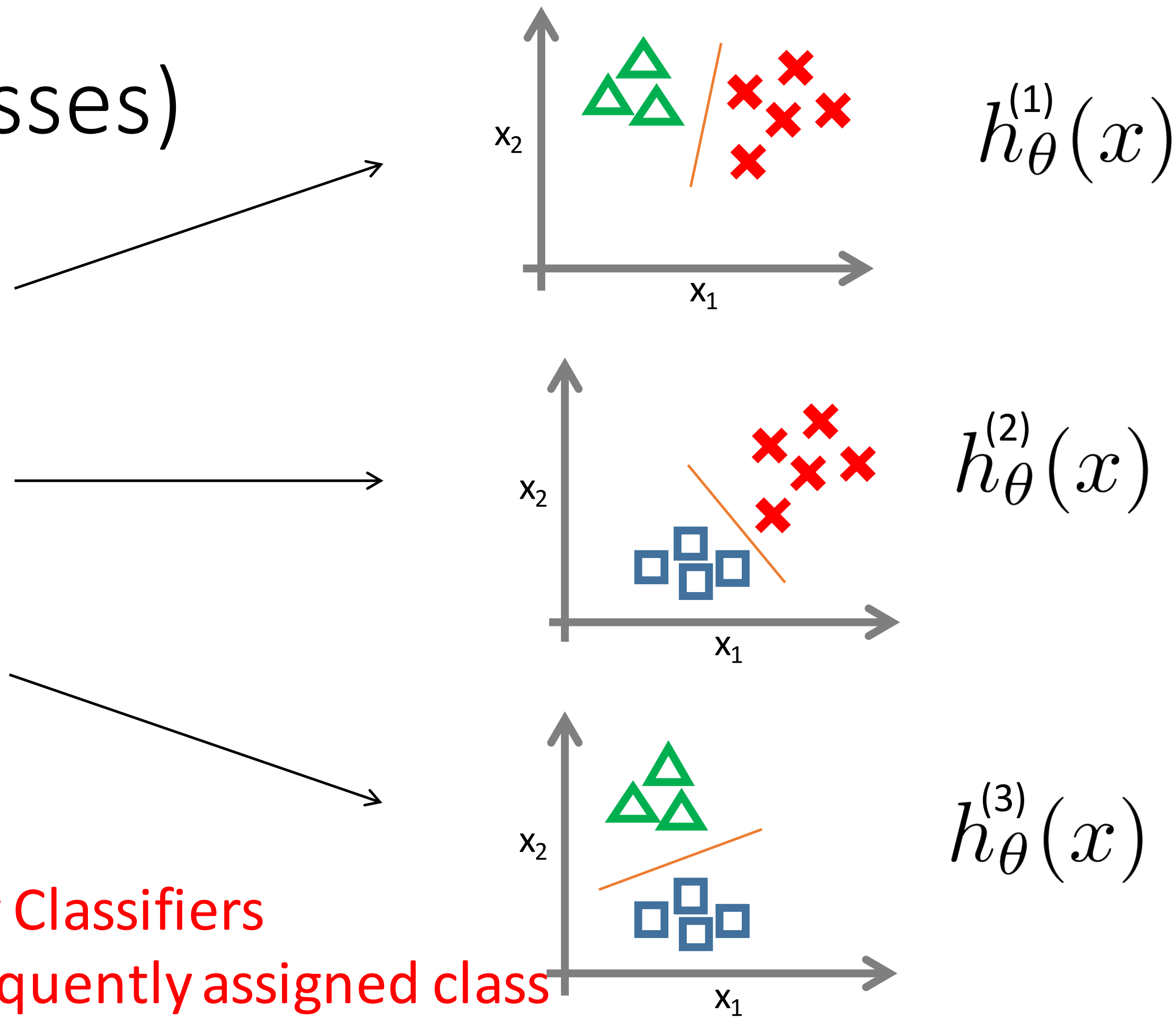




- One-vs-All (One-vs-Rest)
- Train **N** binary classifiers
- Classify to the class with higher $h_\theta(x)$

Multi-Class (N-classes)





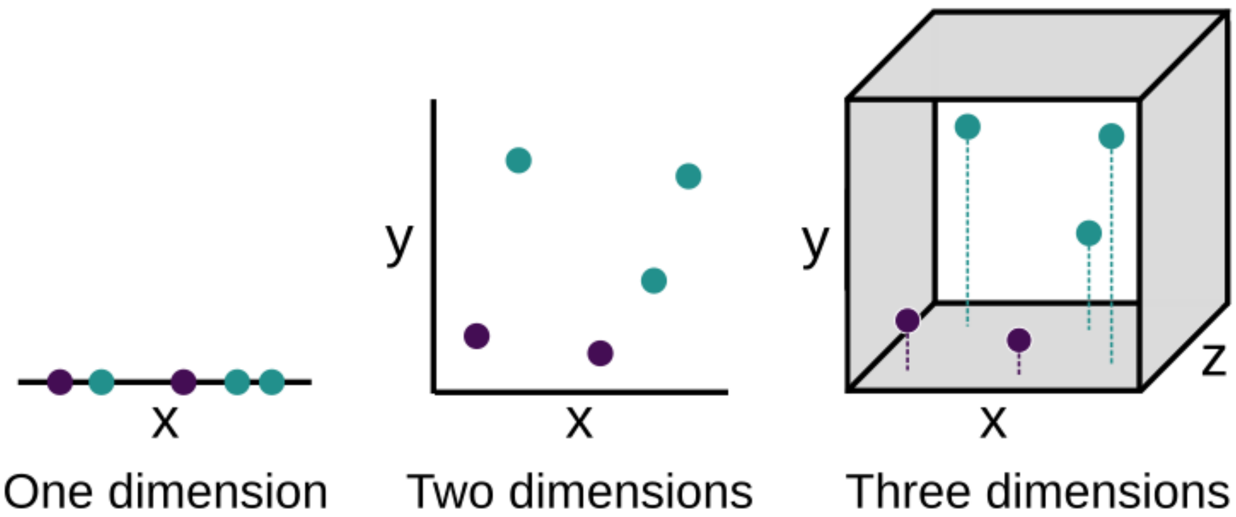
- One-vs-One
- Train **N–(N–1)/2** binary Classifiers
- Classify to the most frequently assigned class

Unsupervised Learning

- Principal Component Analysis (PCA)
- K-Means
- Mean-Shift

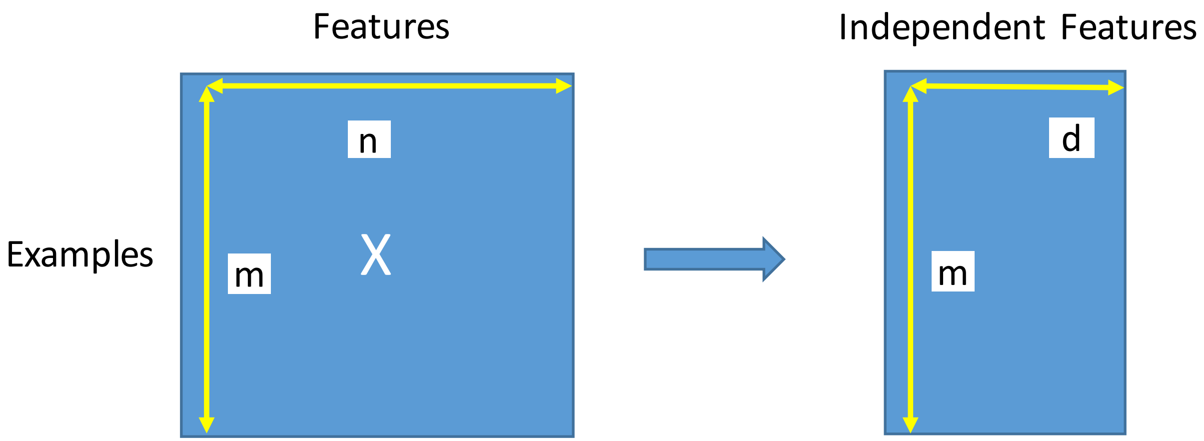
Principal Component Analysis

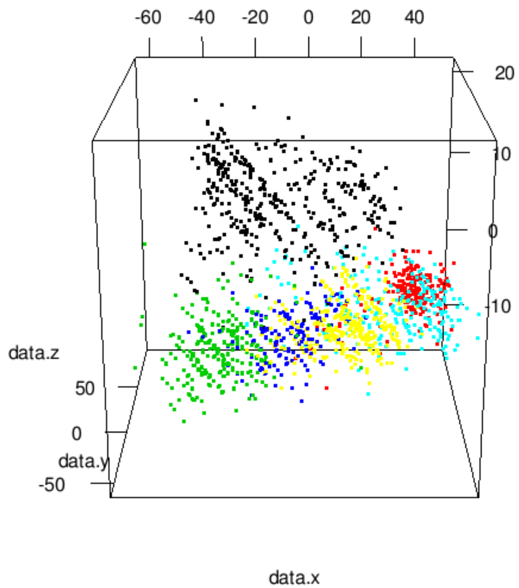
- Curse of dimensionality



Principal Component Analysis

- Reducing or extracting features
- Preserves the maximum of the data variance
- For Visualization (1D, 2,D, 3D)





Principal Component Analysis

- Singular Value Decomposition (SVD) (very costly)
  - Parallelization: Incremental PCA (fast), Randomized PCA (faster)
- PCA assumes that the dataset is centered around the origin
- How many dimensions to preserve?
  - Reduce dimensions that add up to a sufficiently large portion of the variance (e.g., 95%)
- Kernel PCA (kPCA): use the kernel trick like SVM
- In practice, use kPCA to transform the feature space, then perform classification or regression

Principal Component Analysis

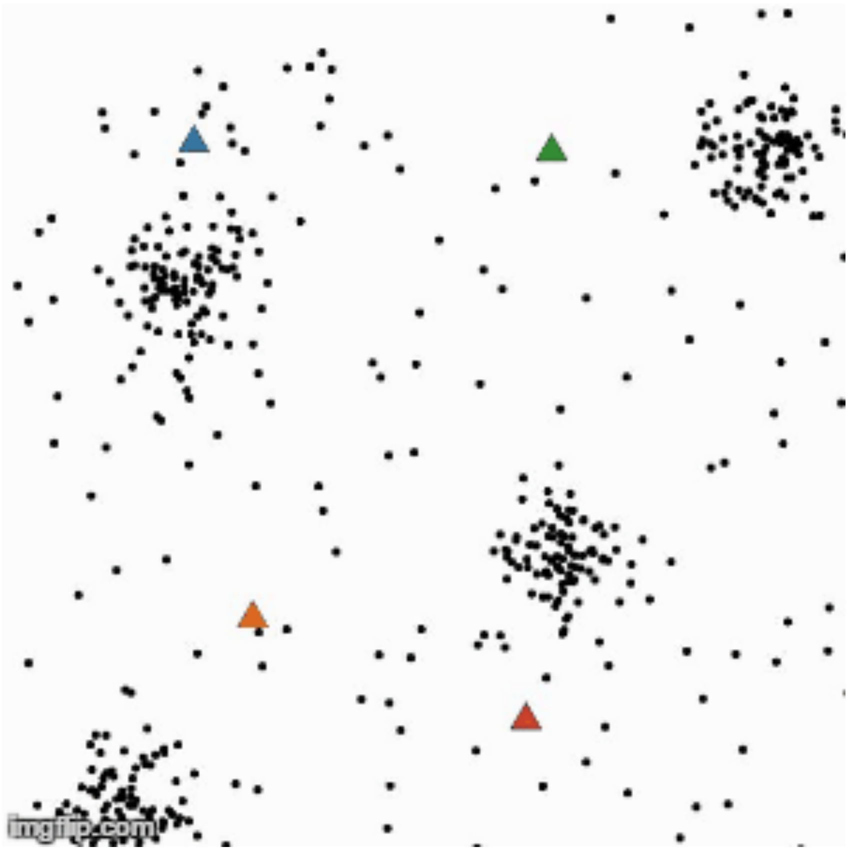
- Hyper-Parameters Tuning
  - d: polynomial Kernel
  - γ: RBF kernel
  - Number of best features
  - Etc.

Other Dimensionality Reduction Methods

- Locally Linear Embedding (LLE)
- Multidimensional Scaling (MDS)
- Isomap
- t-Distributed Stochastic Neighbor Embedding (t-SNE)
- Etc.

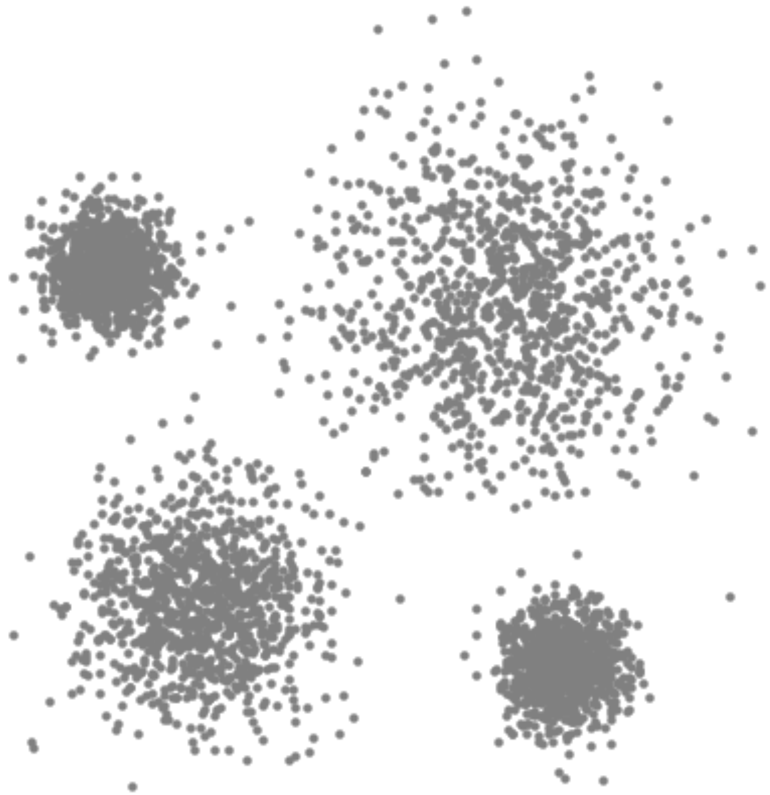
K-Means

- Specify the number of clusters



Mean Shift

- No need to specify the number of clusters



Other Clustering methods

- Expectation Maximization (EM)
- Hierarchical Clustering
- Affinity Propagation (AP)
- Etc.

The End ..