Machine Learning

Abdelhak Mahmoudi
abdelhak.mahmoudi@um5.ac.ma

INPT - 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

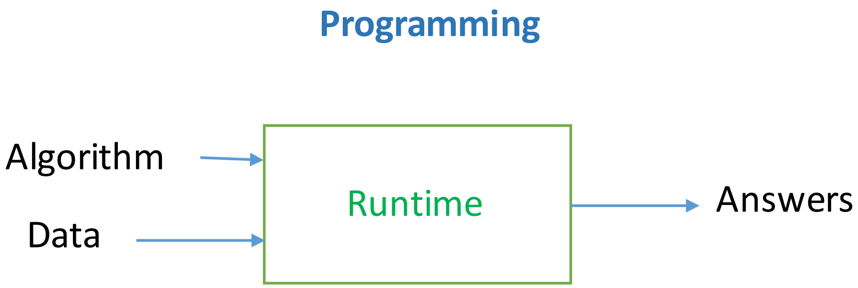
- The Big Picture of ML !
- Terminologies
- How can I Apply?
- How can I Learn?

The Big Picture!

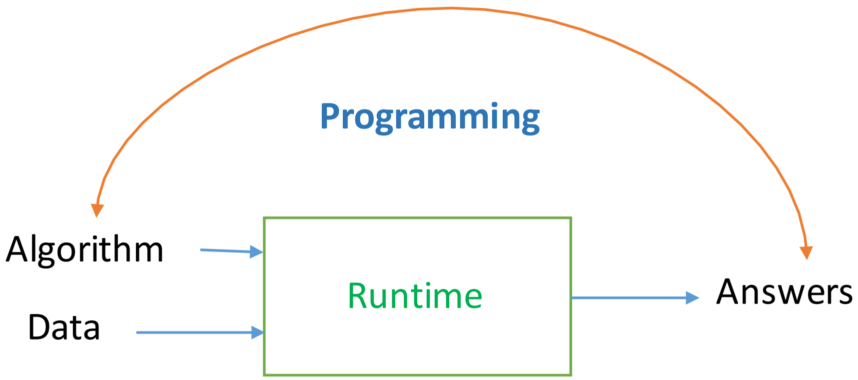
Forbes: "The Top 10 AI And Machine Learning Use Cases Everyone Should Know About"

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

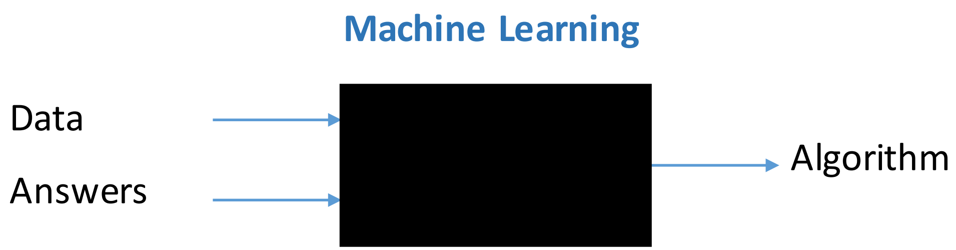
The Big Picture!



The Big Picture!

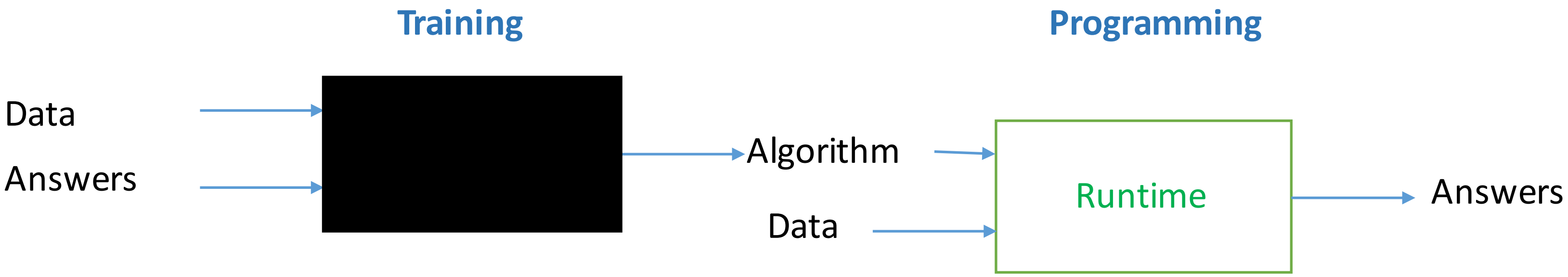


The Big Picture!



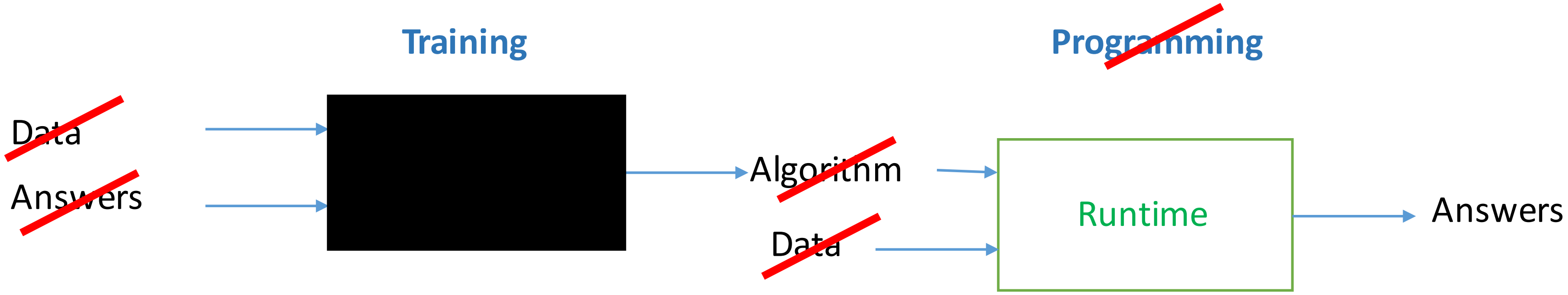
The Big Picture!

Machine Learning

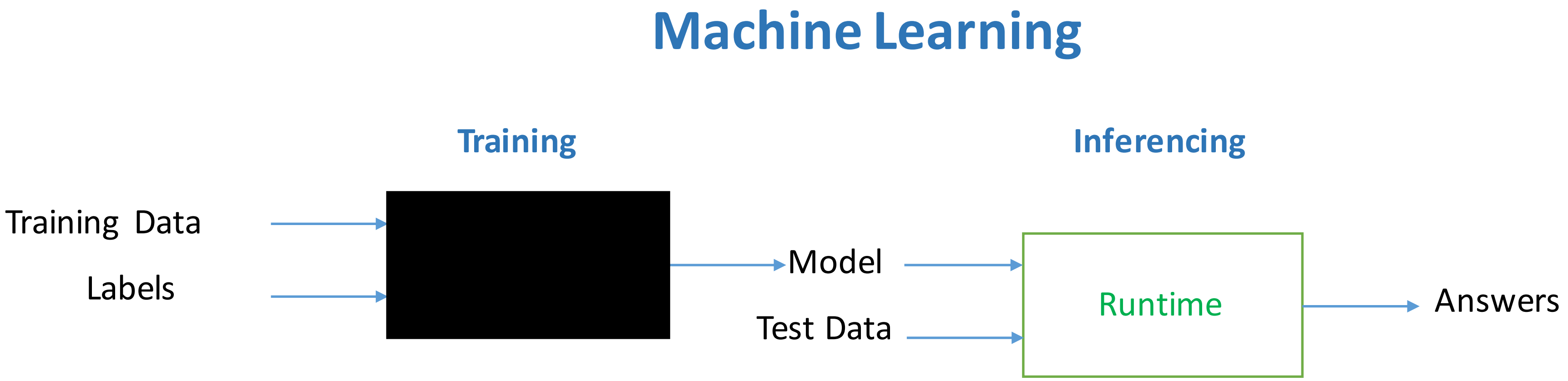


The Big Picture!

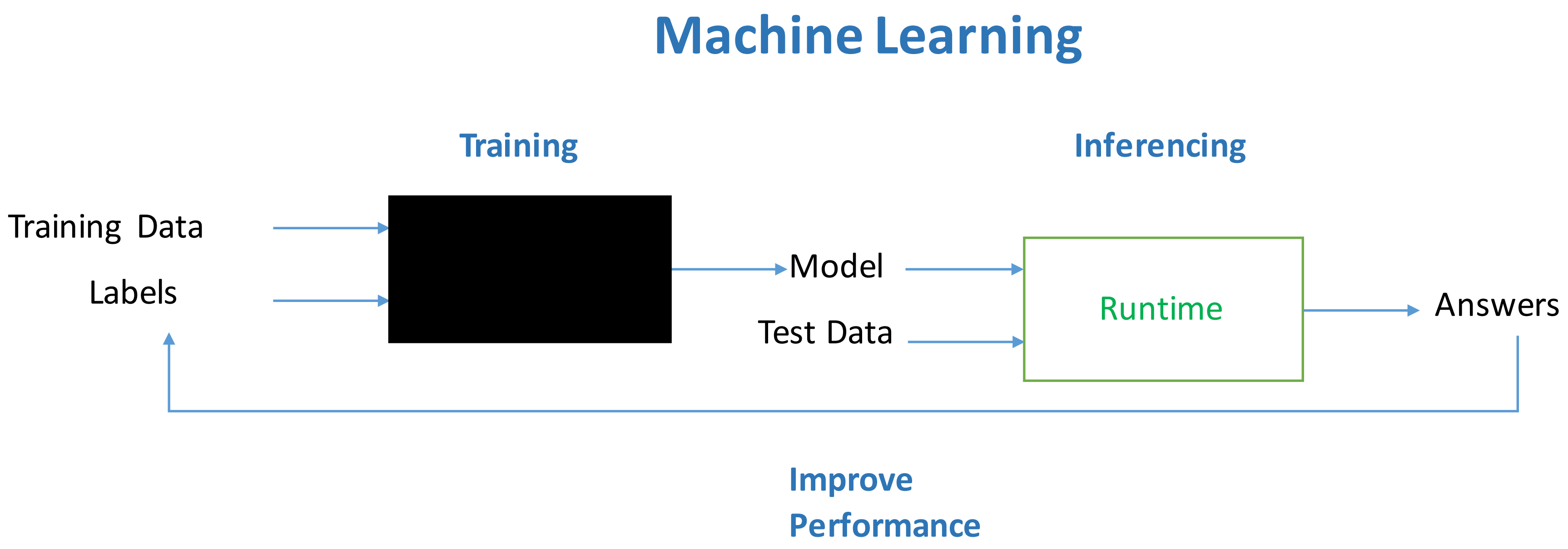
Machine Learning



The Big Picture!

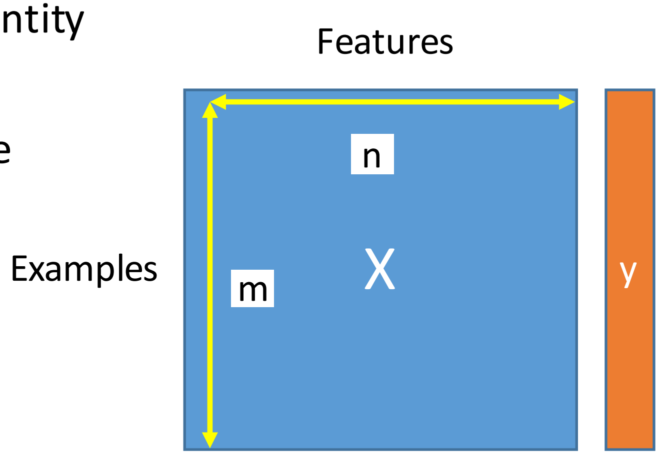


The Big Picture!



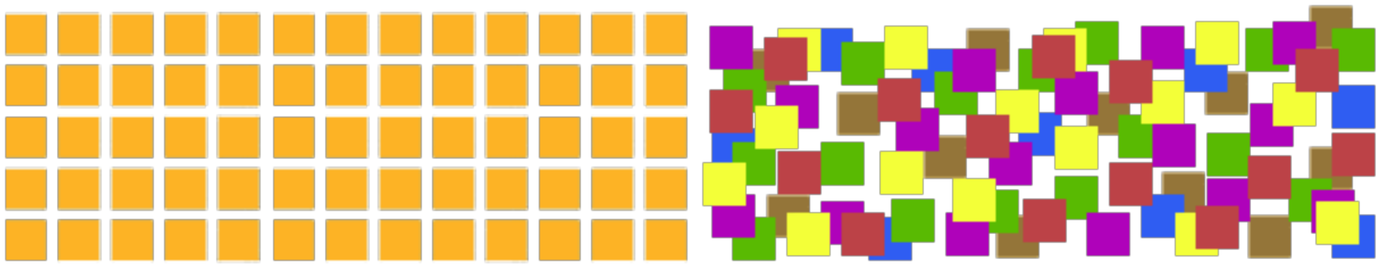
The Big Picture!

- 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!

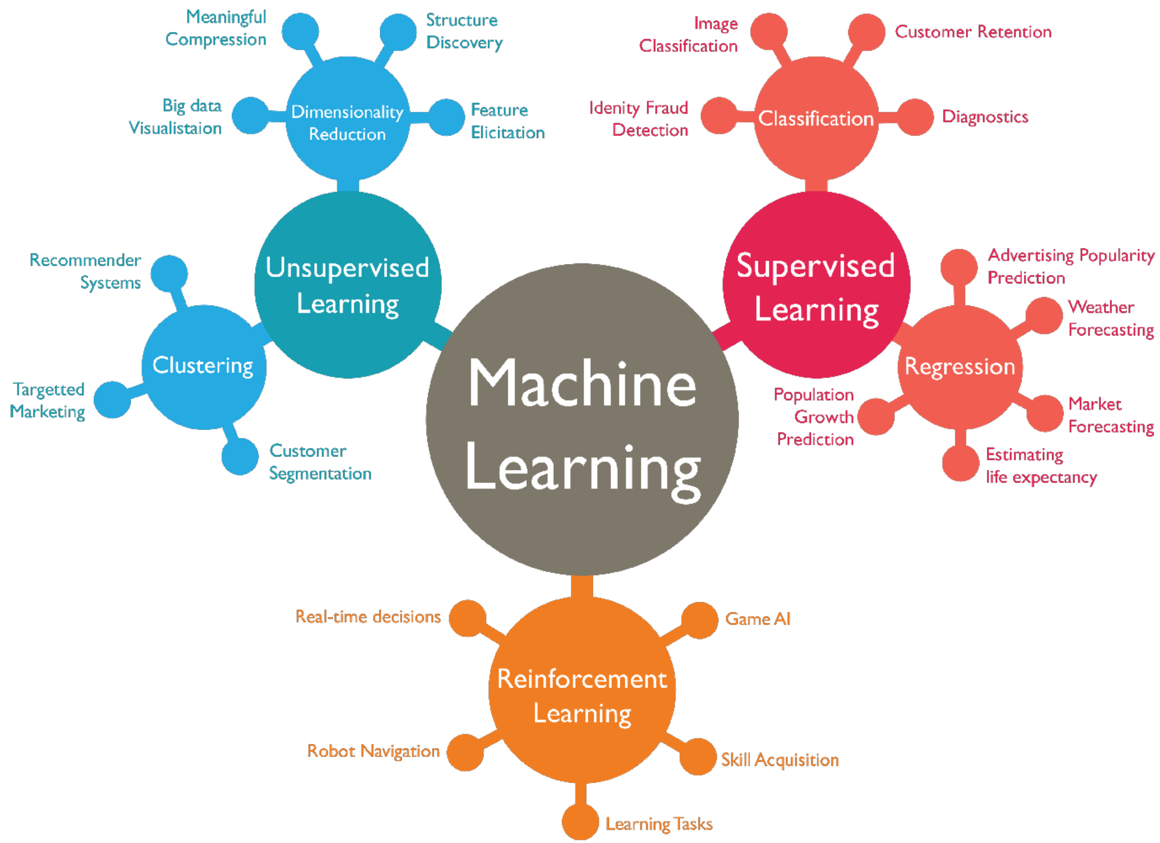
- Data
  - Structured
    - CSV, XML, JSON, XLSX, etc.
  - Unstructured
    - DOC, HTML, PDF, PNG, MP3, MP4, etc.



Text, Image, son

The Big Picture!

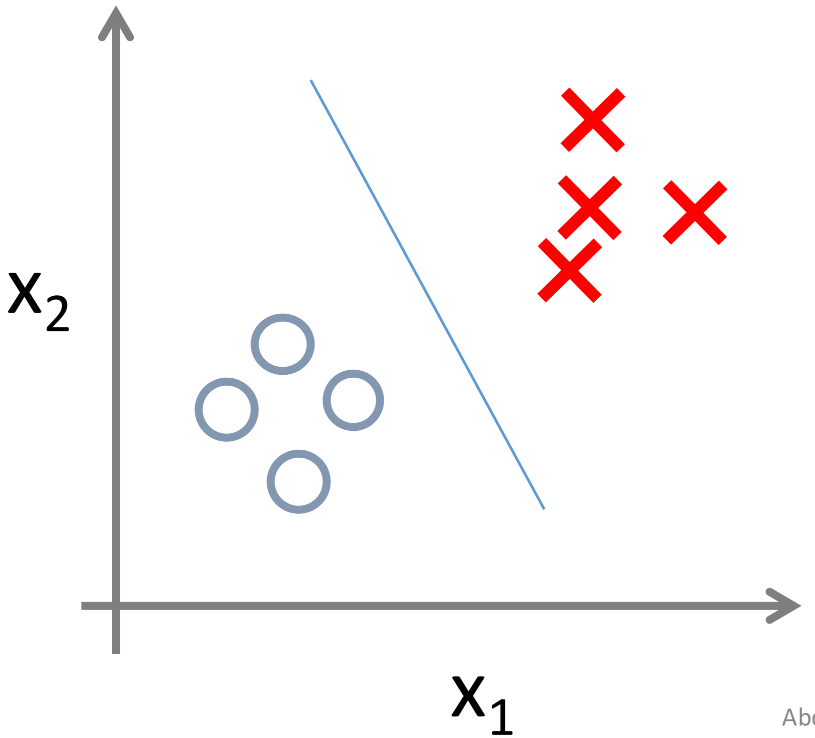
- Types of Learning
  - Supervised
    - Classification
    - Regression
  - Unsupervised
    - Dimensionality Reduction
    - Clustering
  - Semi-supervised
    - Little supervised data
  - Reinforcement



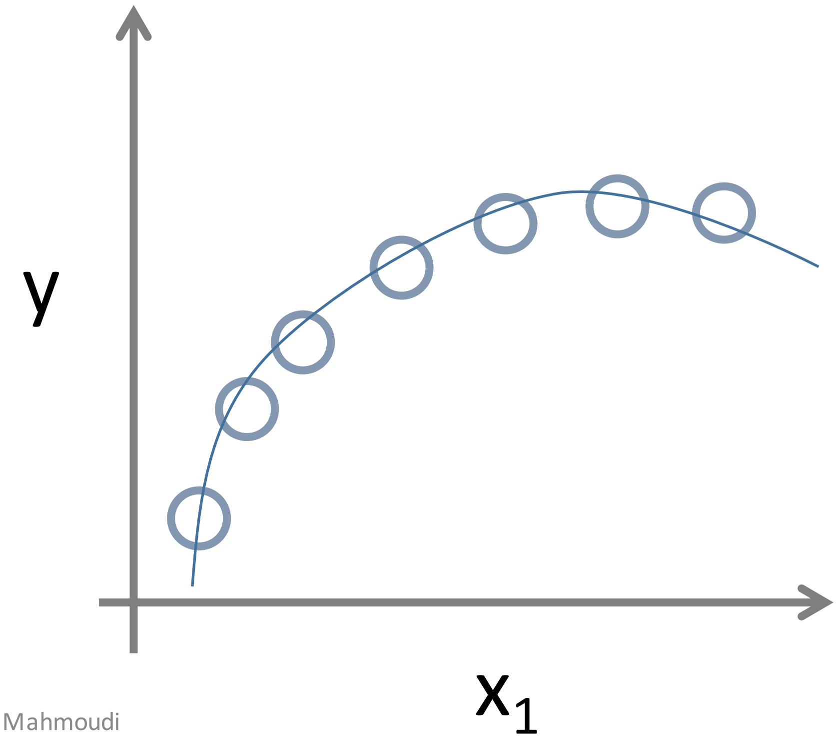
The Big Picture!

Supervised Learning

Classification (y is discrete)



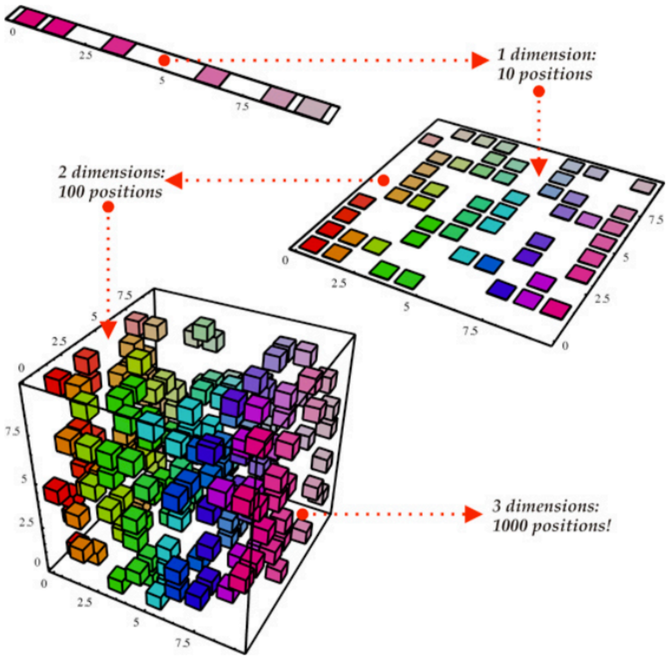
Regression (y is continuous)



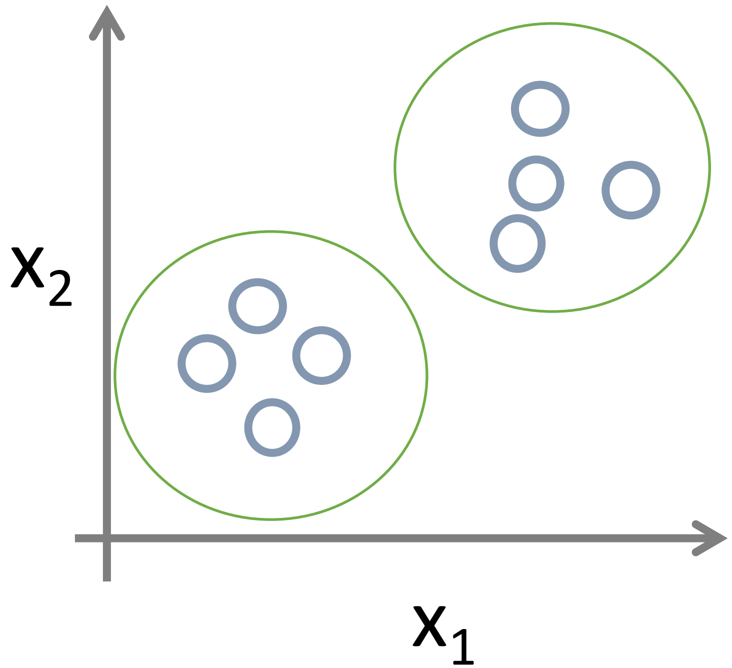
The Big Picture!

Unsupervised Learning (y absent)

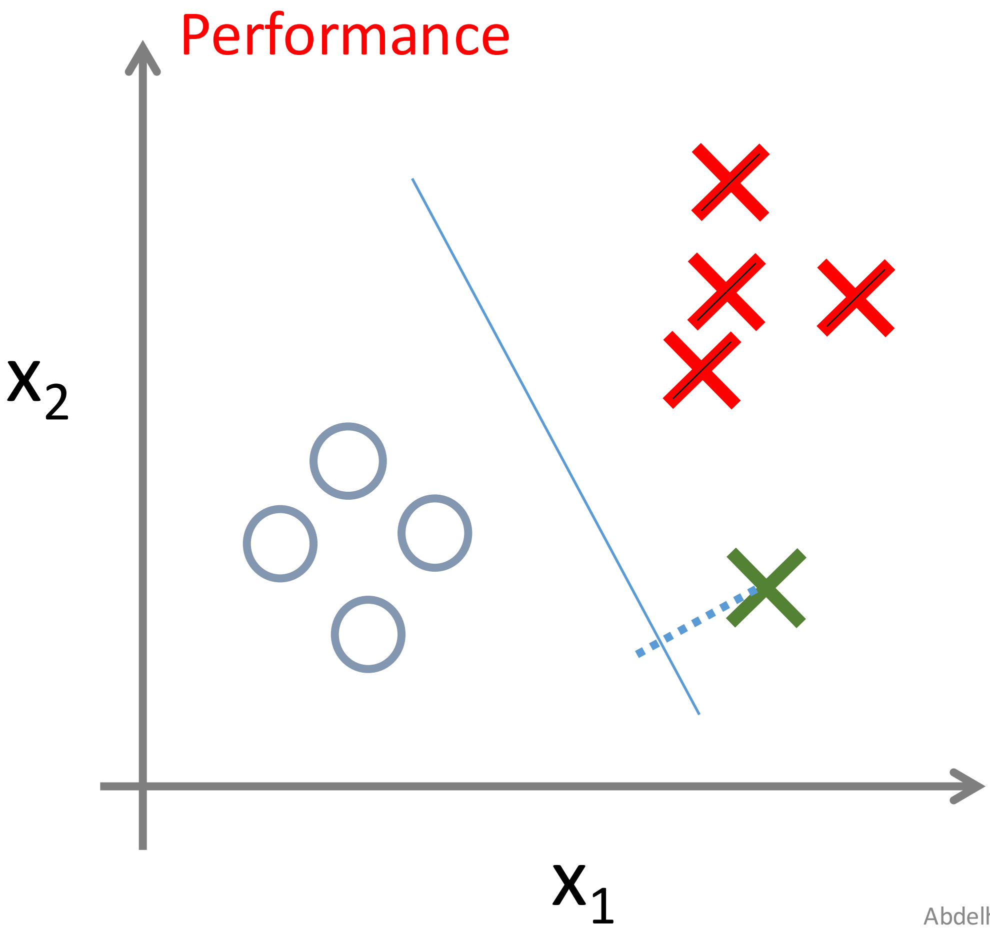
Dimensionality Reduction

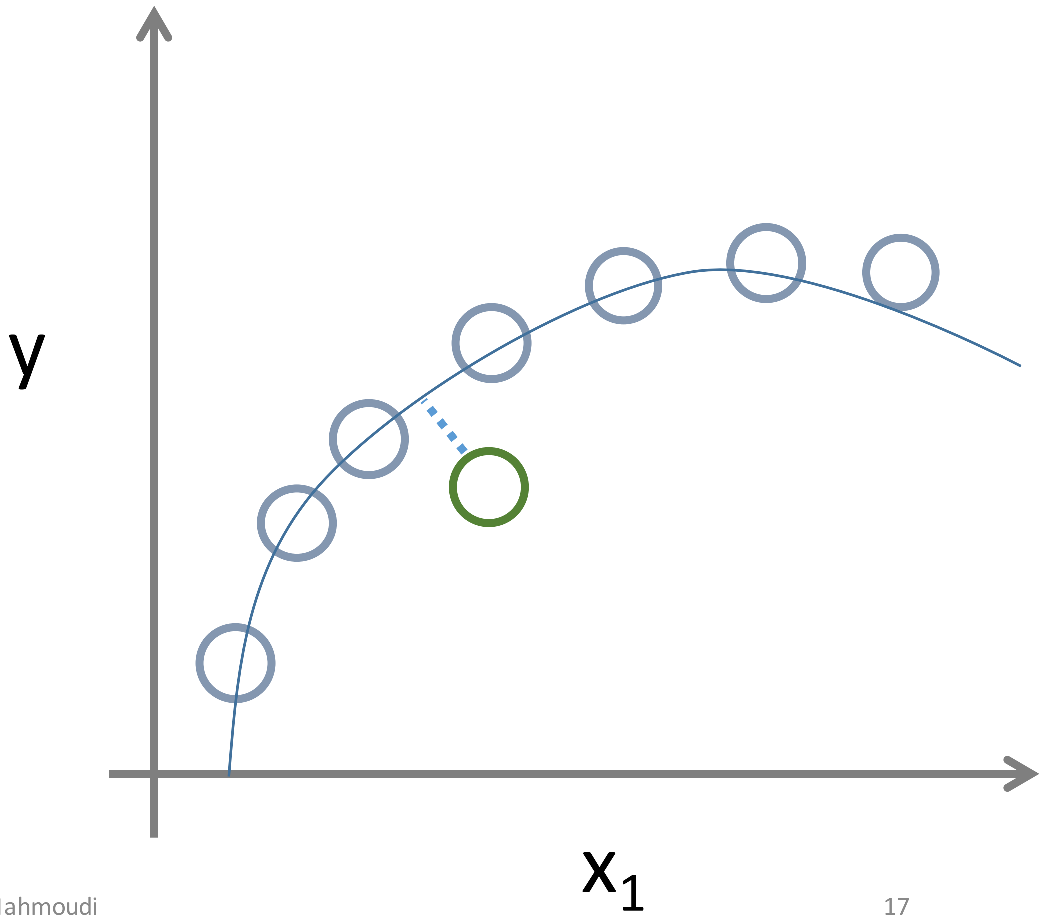


Clustering

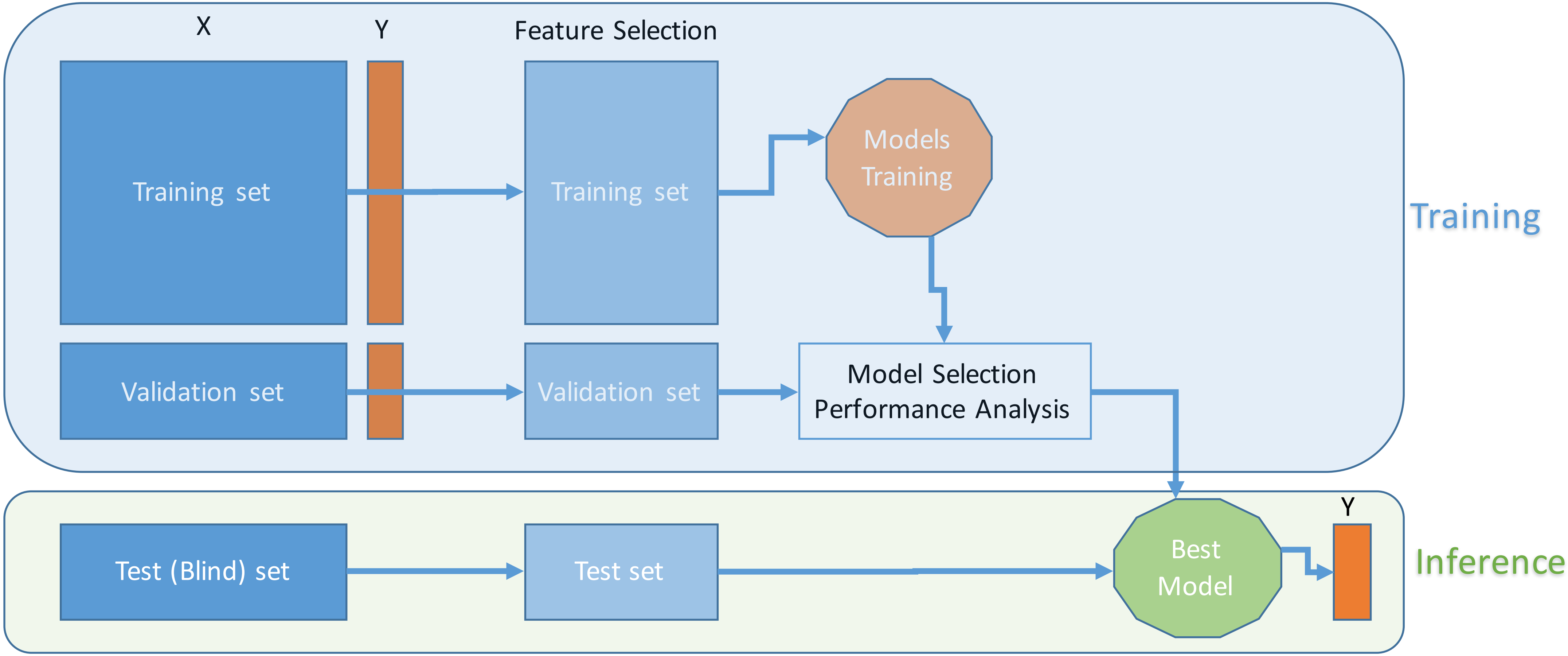


The Big Picture!





The Big Picture!



Terminologies

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

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,
- Book: “Machine Learning”. Tom M. Mitchell. 1997

Statistical Learning (**1968**)

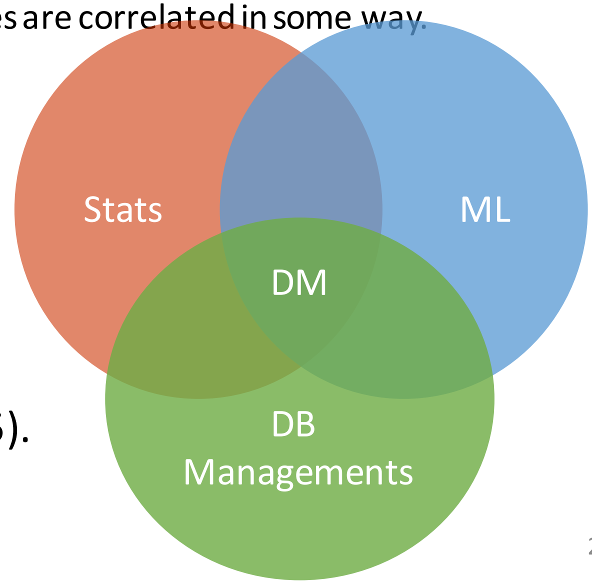
- VC Theory. “On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities". Vapnik, V. N.; Chervonenkis, A. Ya, 1968
- A subfield of Mathematics which deals with finding relationship between variables to predict an outcome
- Support Vector Machines (**1995**)
  - Much simpler, 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

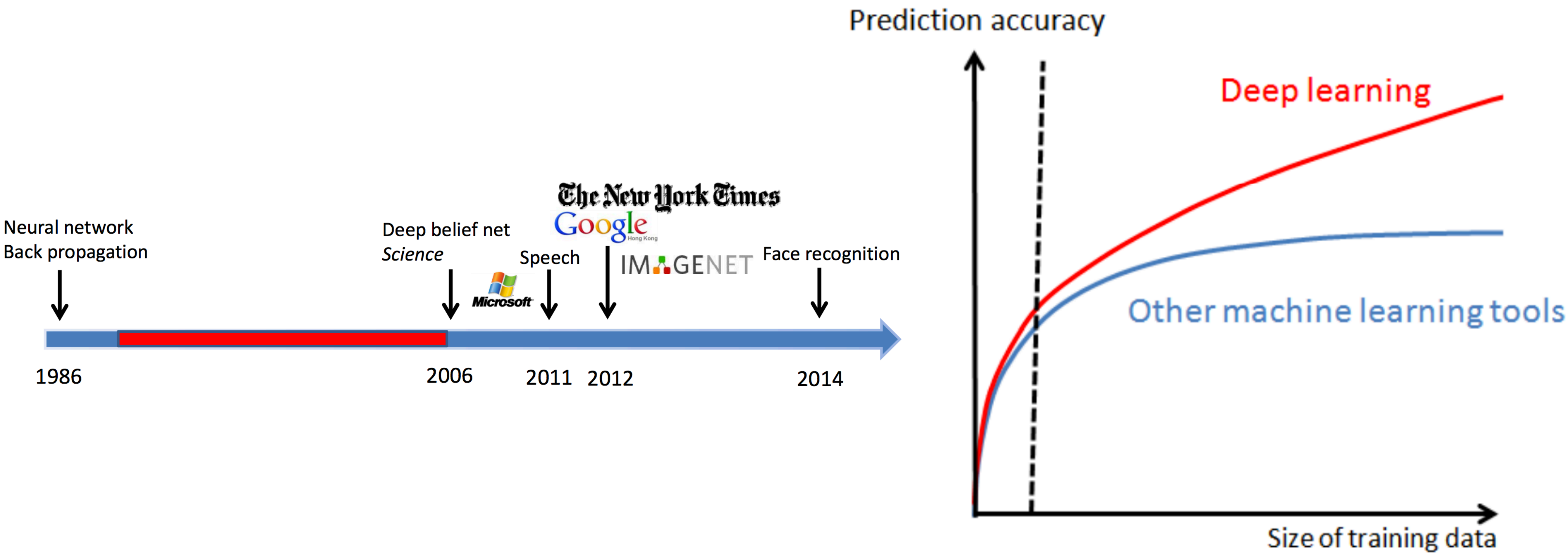
Definition : "The process of automatically discovering useful information in large repositories".

- 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

Deep Learning



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?

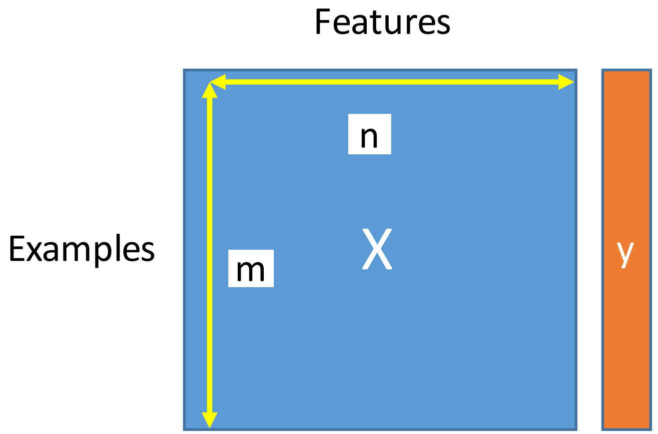
- Start small projects and use Framworks
  - Scikit-learn, TensorFlow, Keras, Pytorch, Caffe, Microsoft Cognitive Toolkit (CNTK), MXNet, Spark MLlib, etc.
- Challenge your self
  - Find data: Web, UCI Machine Learning Repo
  - Go for competitions: Kaggle, DrivenData, Zindi
- Github
  - Find codes
  -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

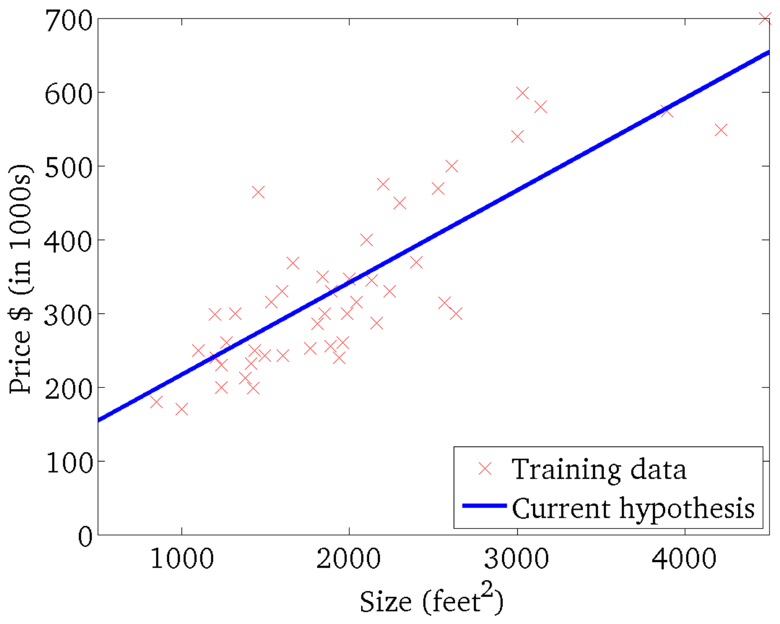
Supervised Learning

- 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 $y = w_0 + w_1 x$
- The best line is the line with minimum loss $\mathrm{L}(w)$
- Solved using Normal Equations
  - $W = (X^T X)^{-1} X^T y$
  - But not for big X !
- Find W iteratively using gradient descent



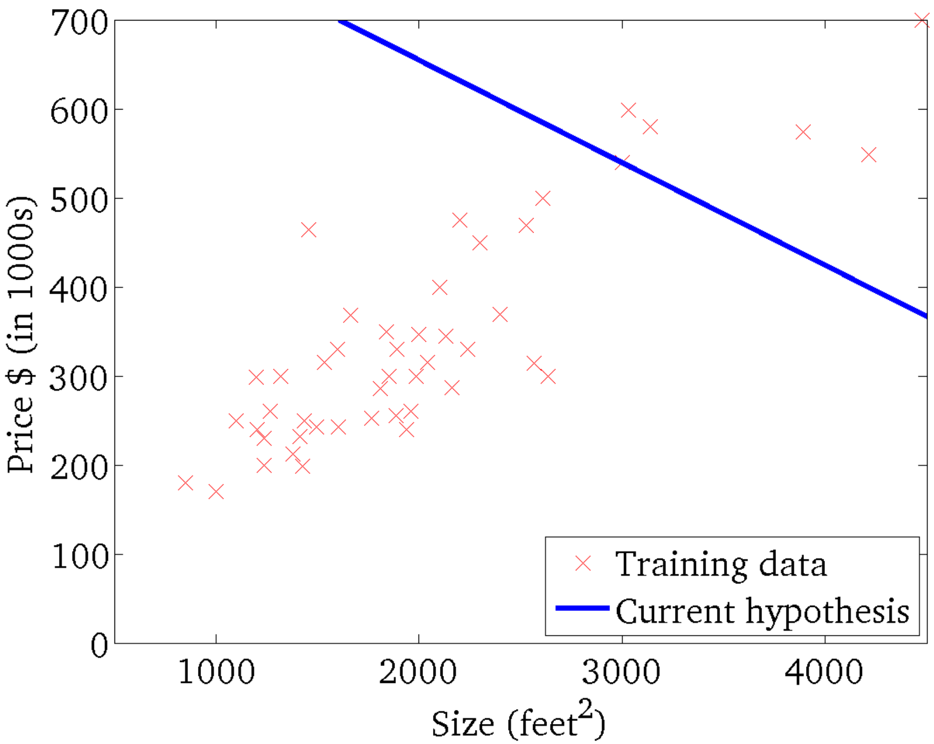


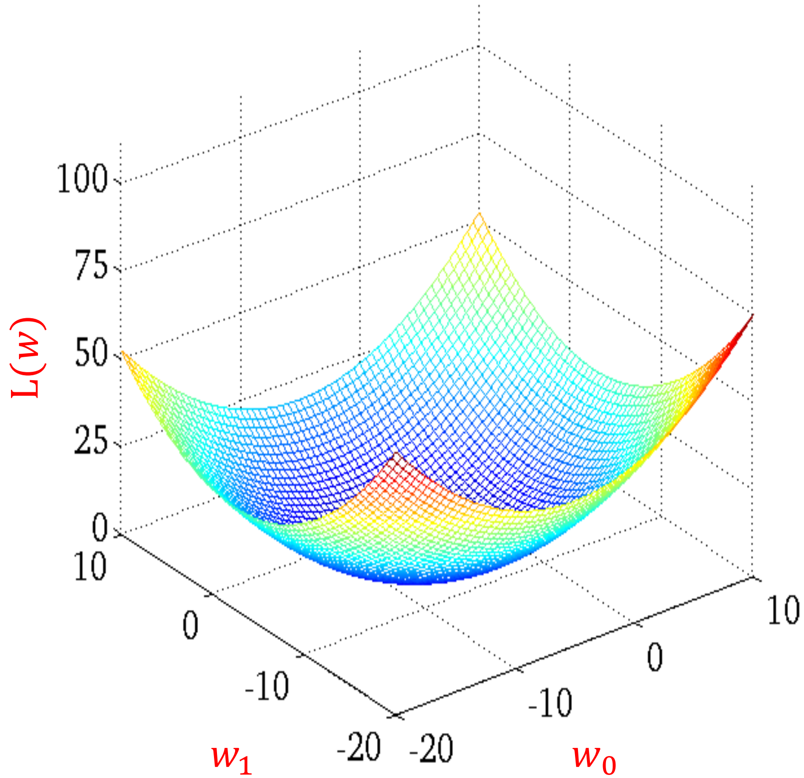
Gradient Descent

(Batch) GD

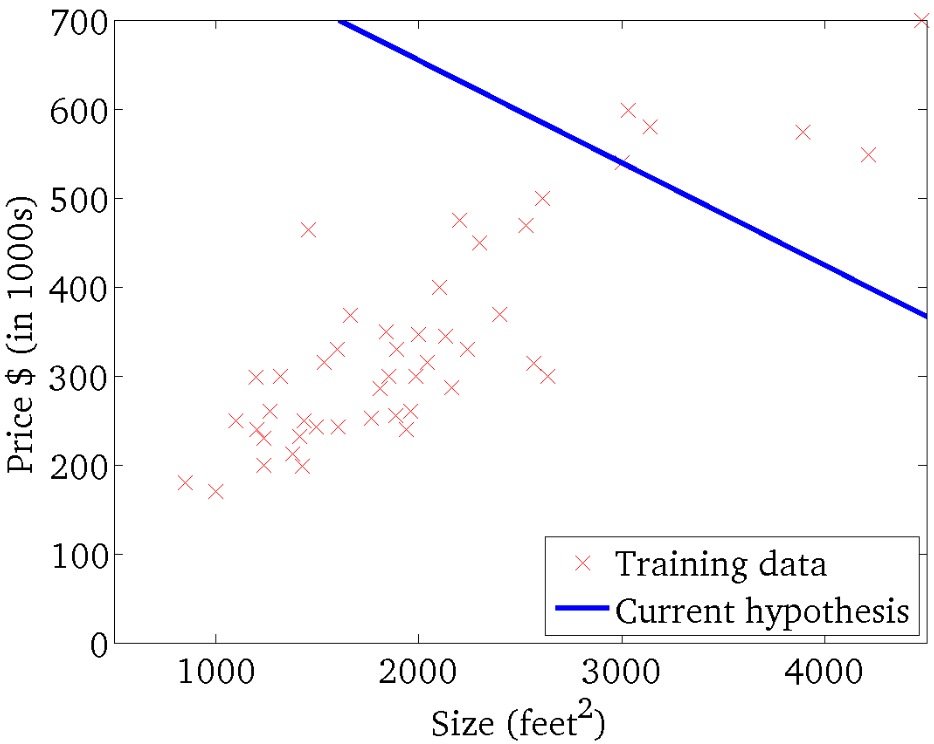
```
X = data_input
Y = data_output
W = initialize_parameters()
for it in range(num_iterations):
    Yhat = h(X, W)
    L = loss(Yhat, Y)
    dW = gradient(L(W))
    W = W - α dW
```

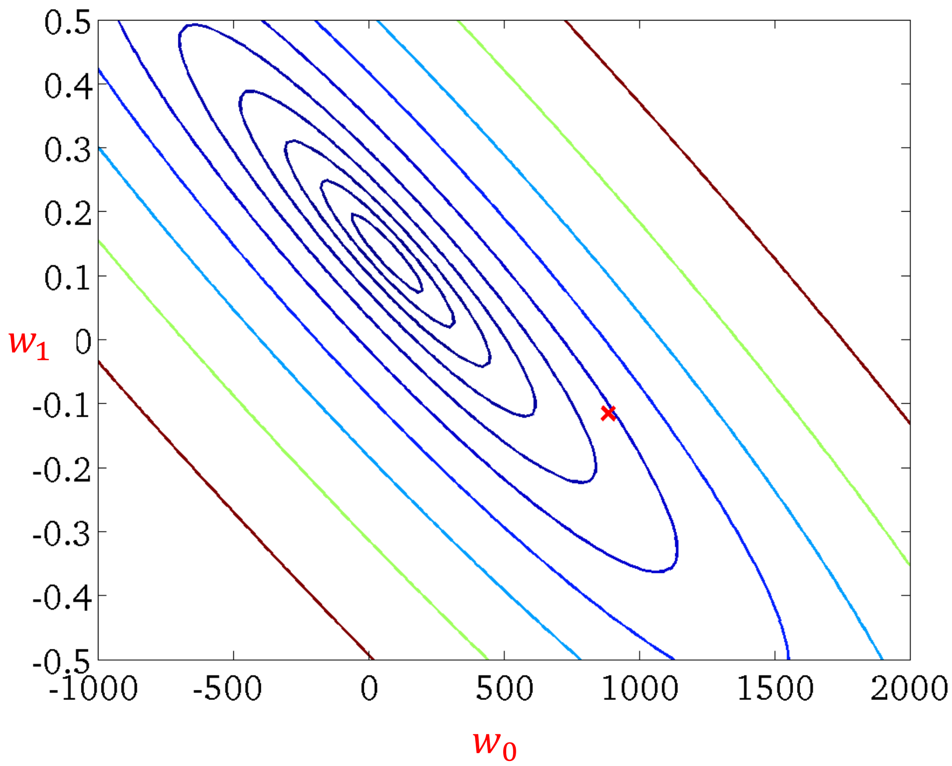
Linear Regression



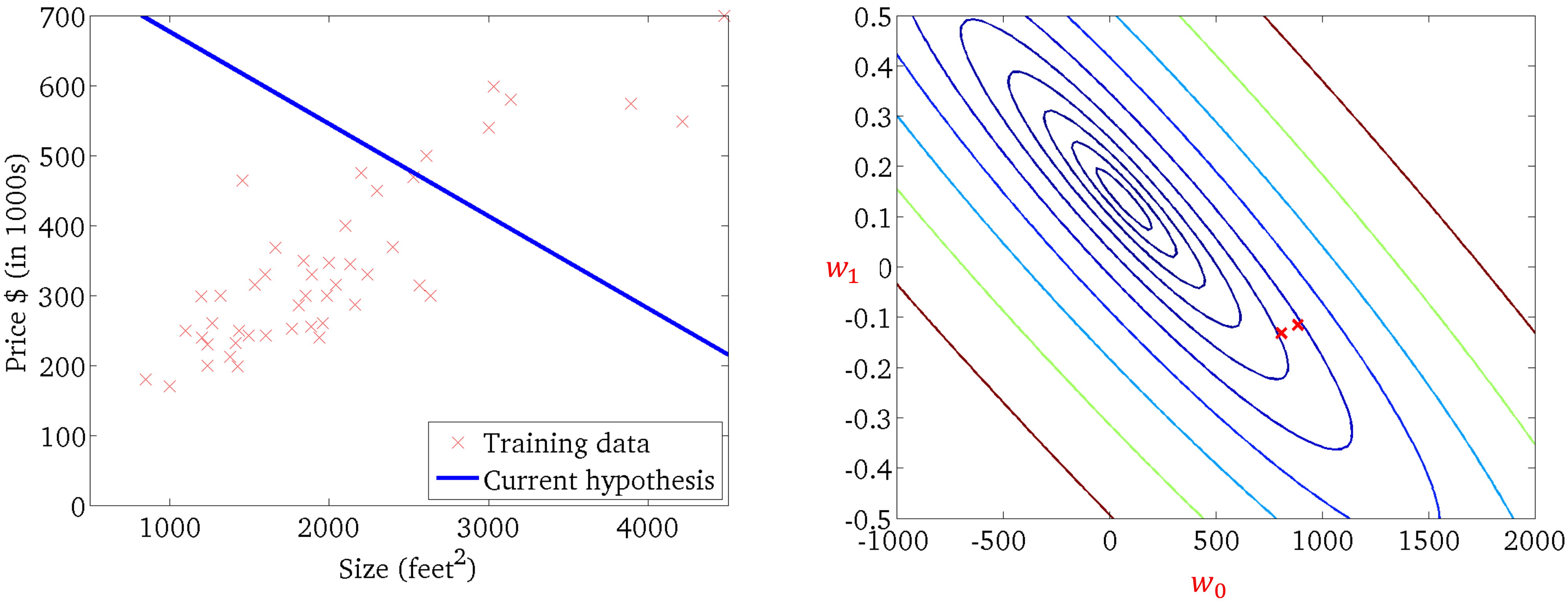


Linear Regression

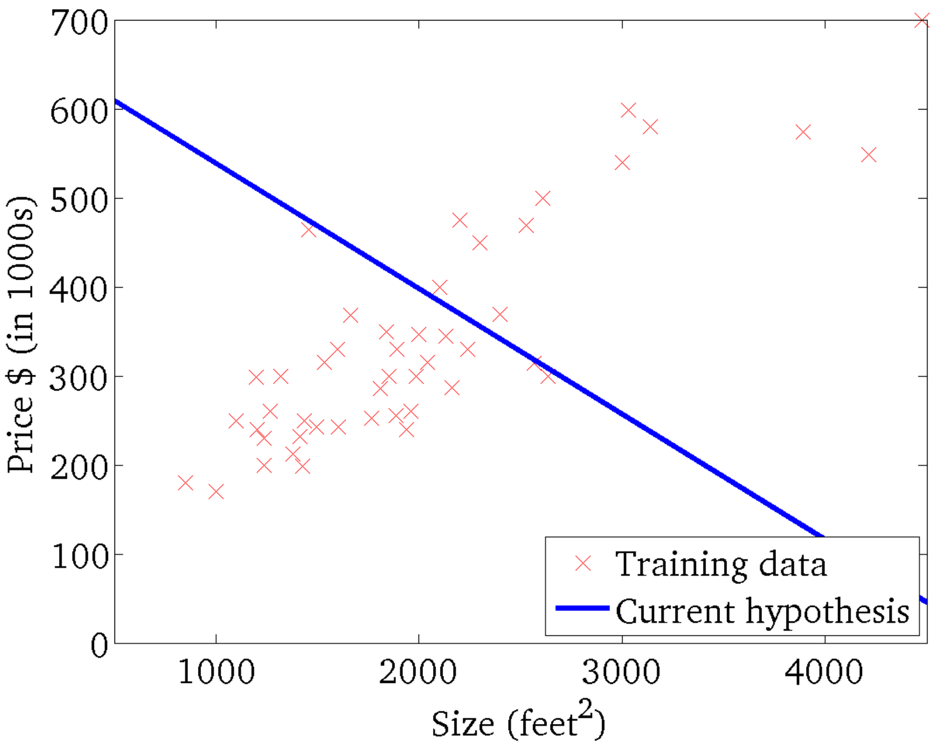


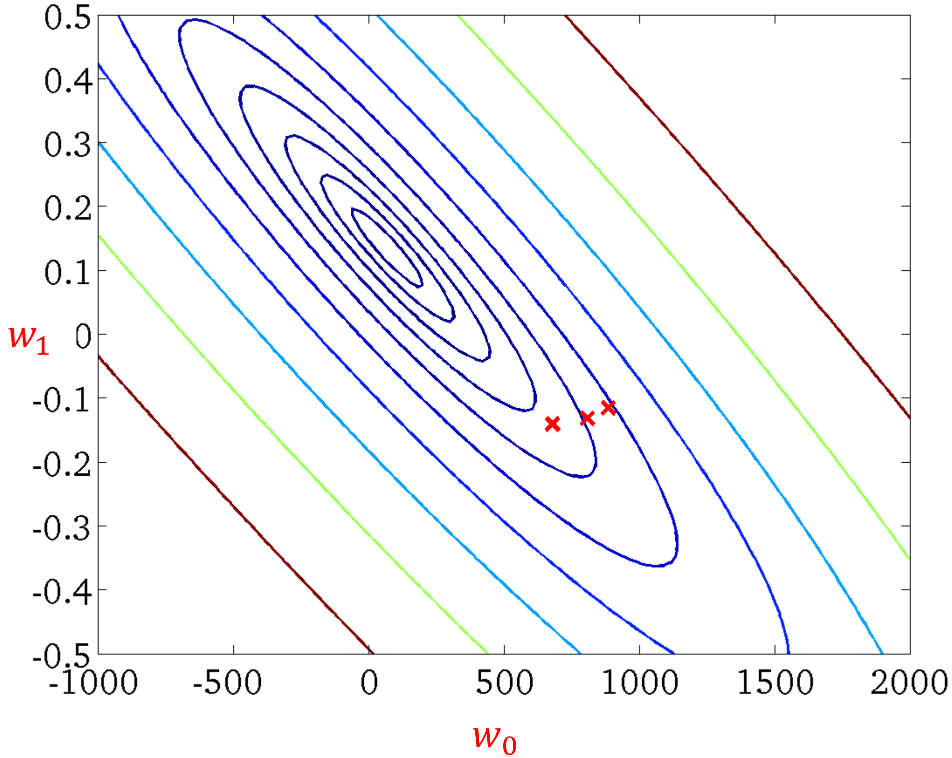


Linear Regression

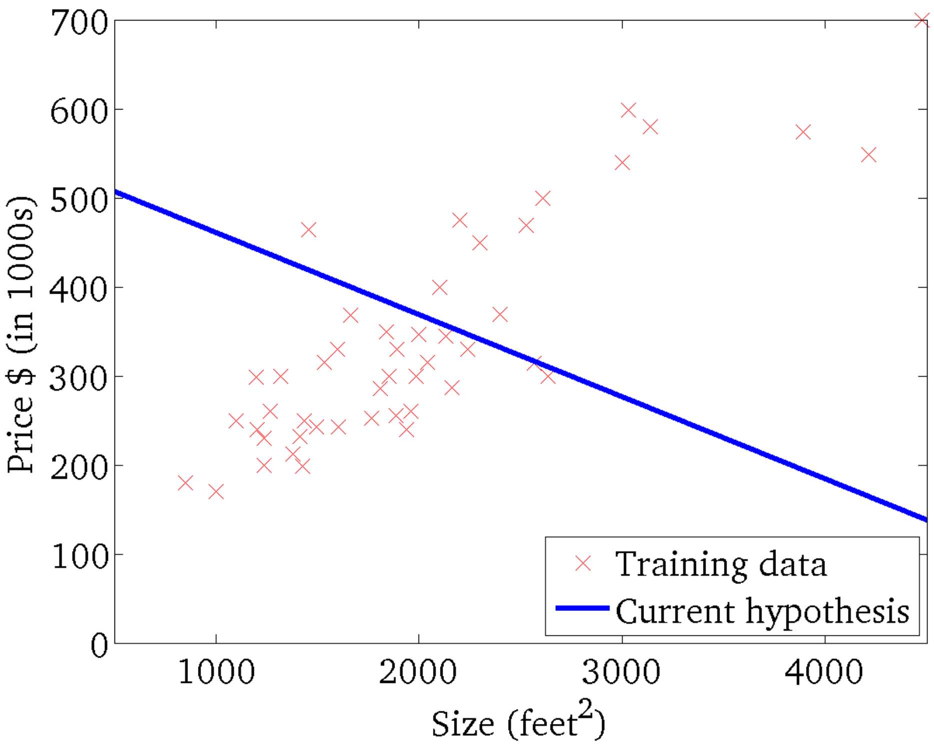


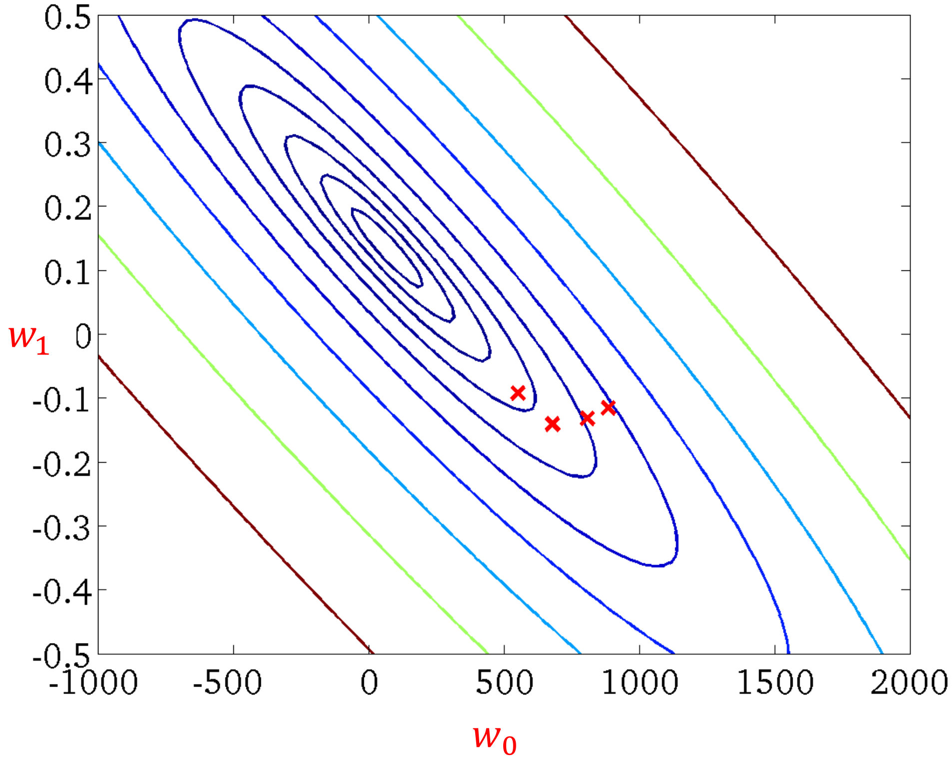
Linear Regression



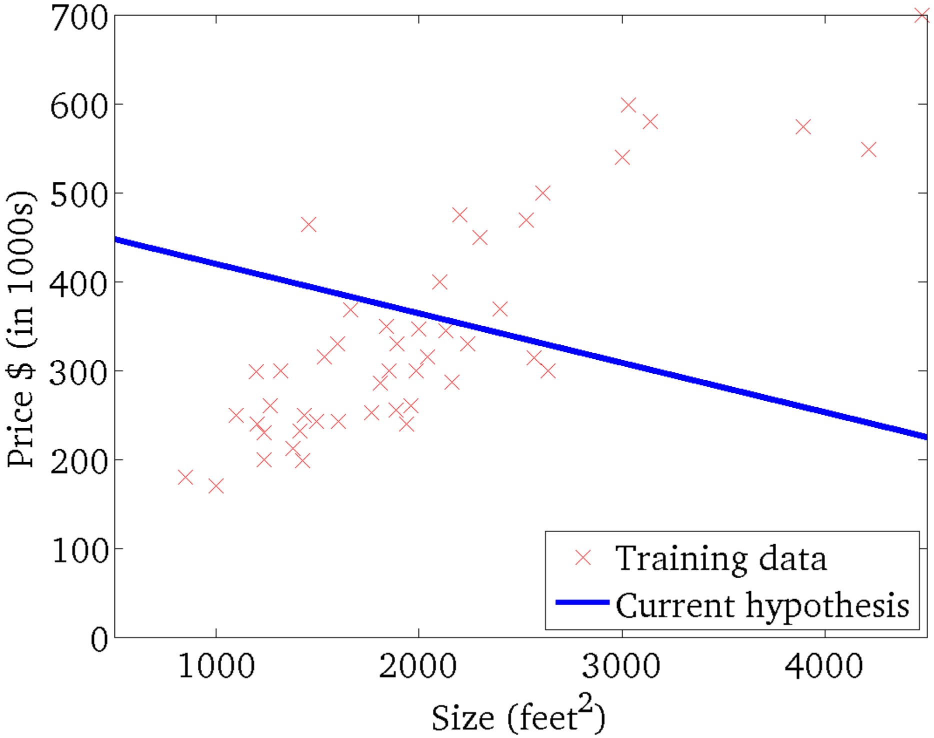


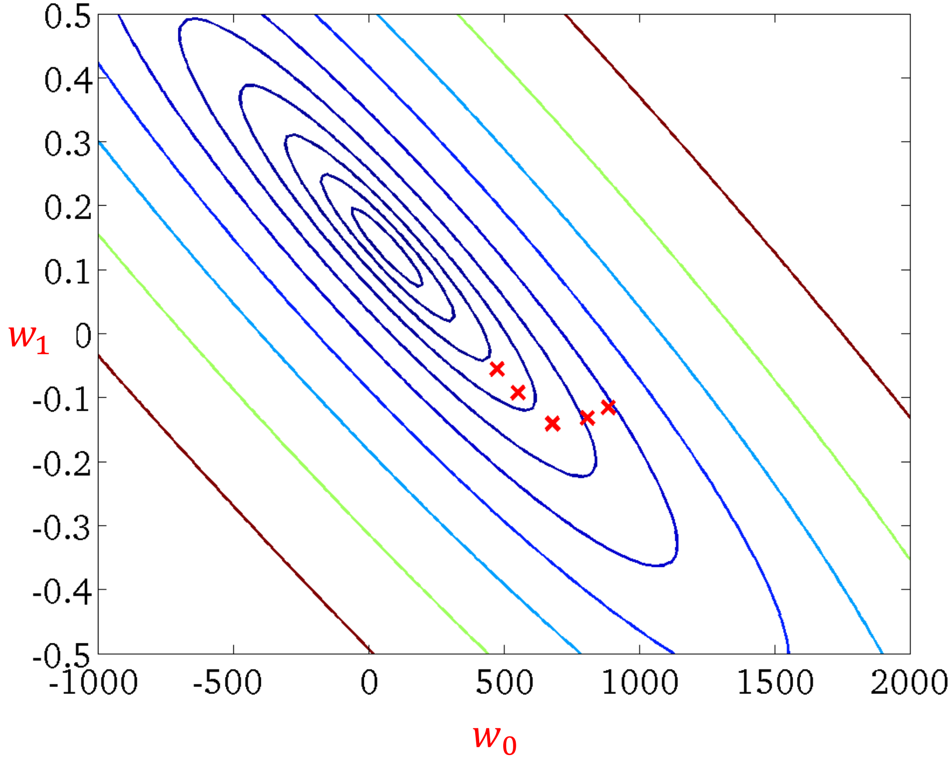
Linear Regression



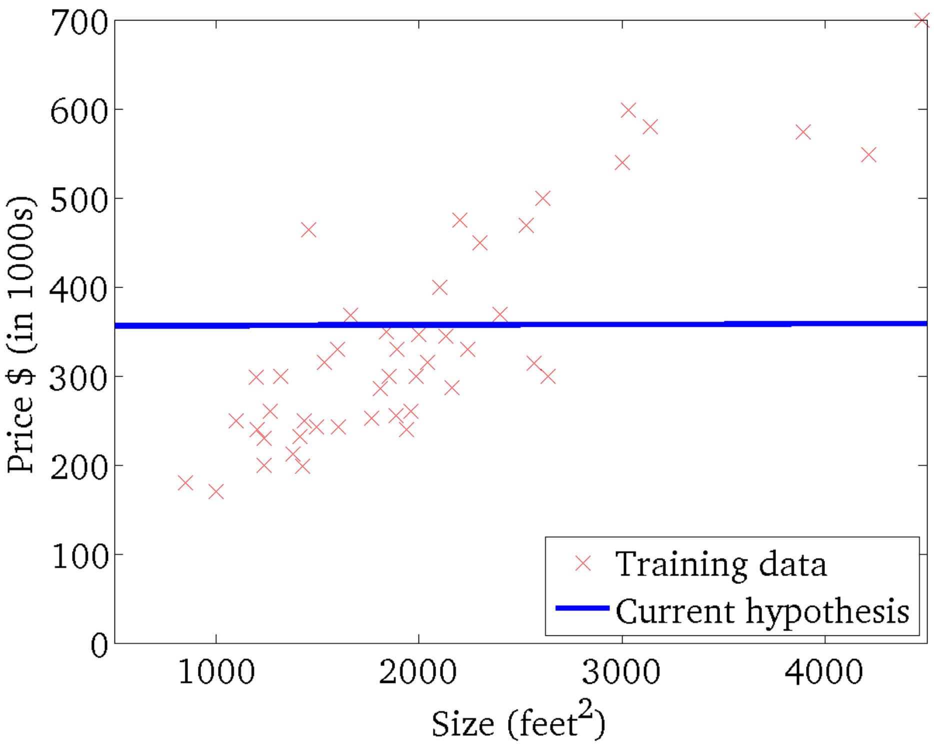


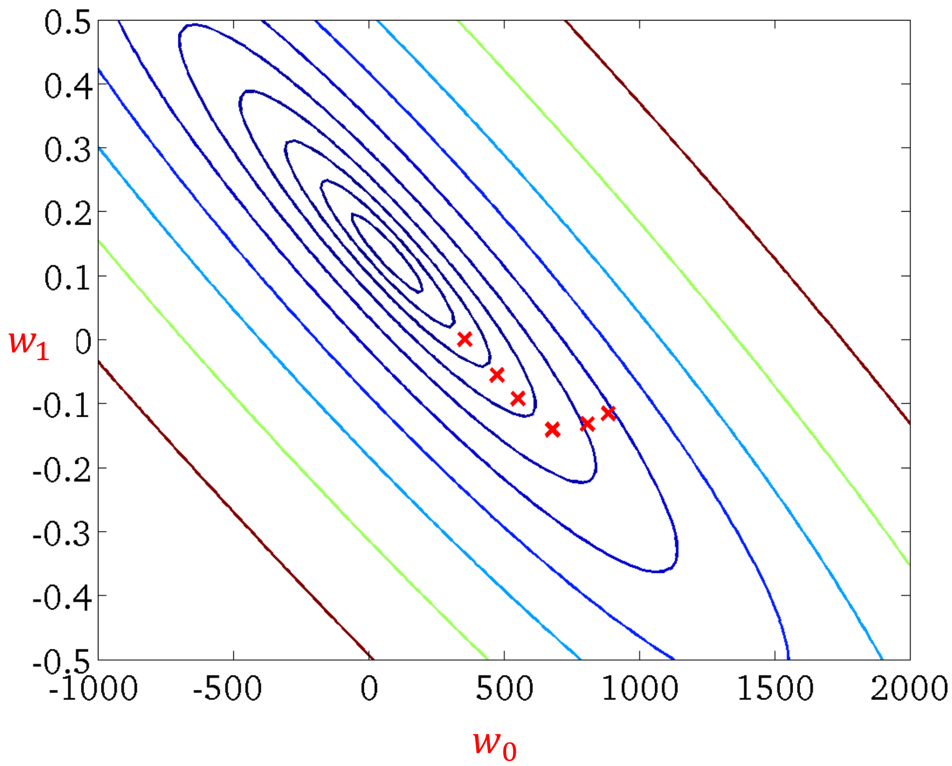
Linear Regression



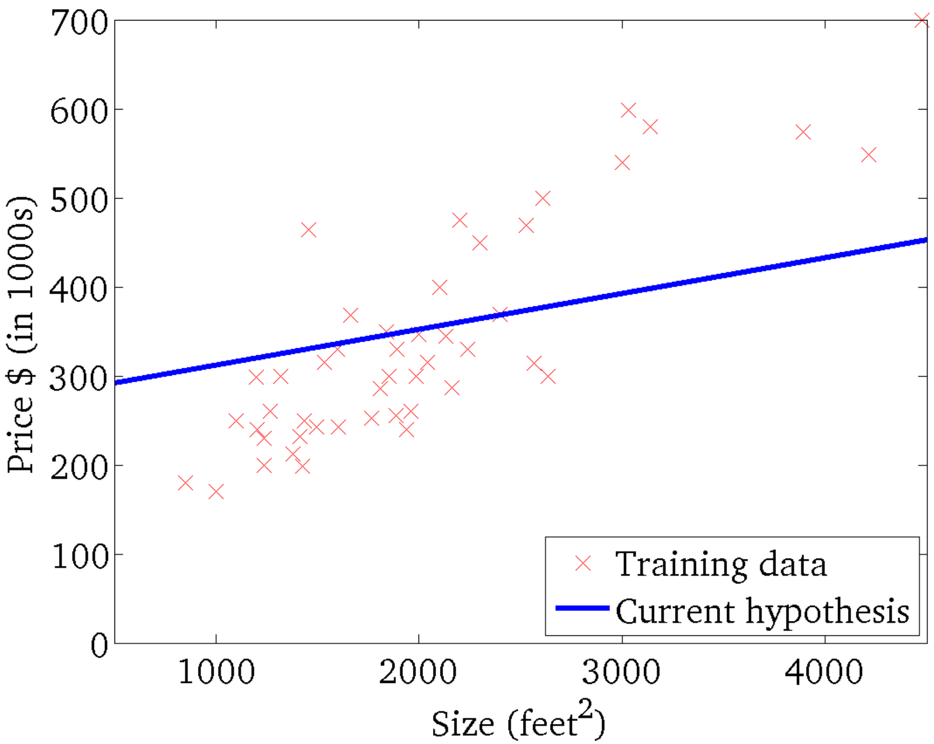


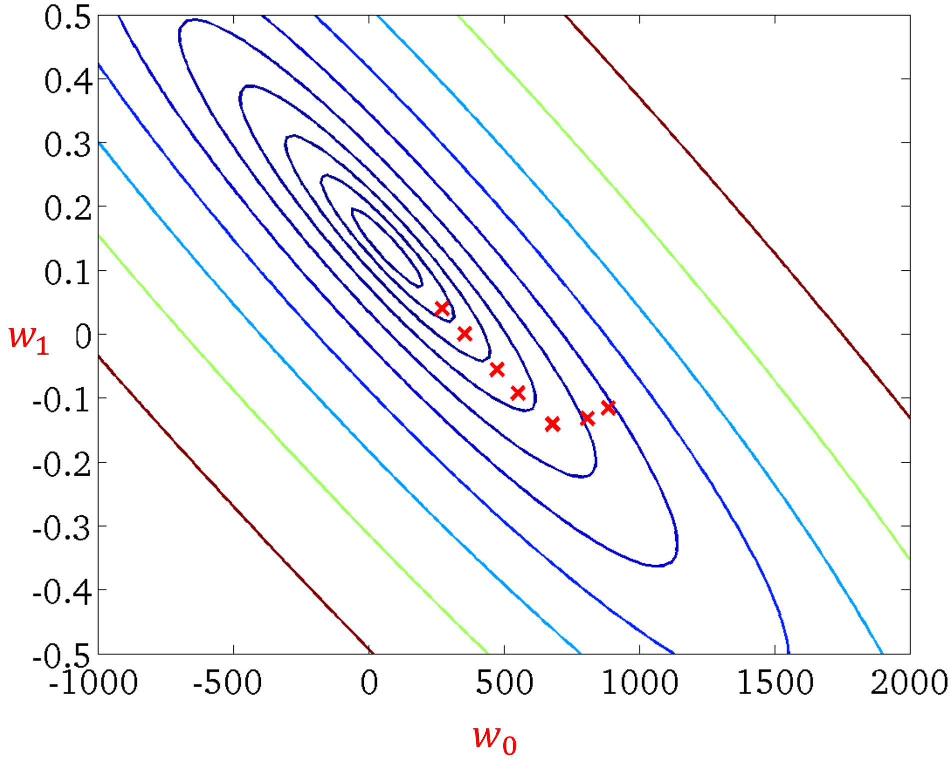
Linear Regression



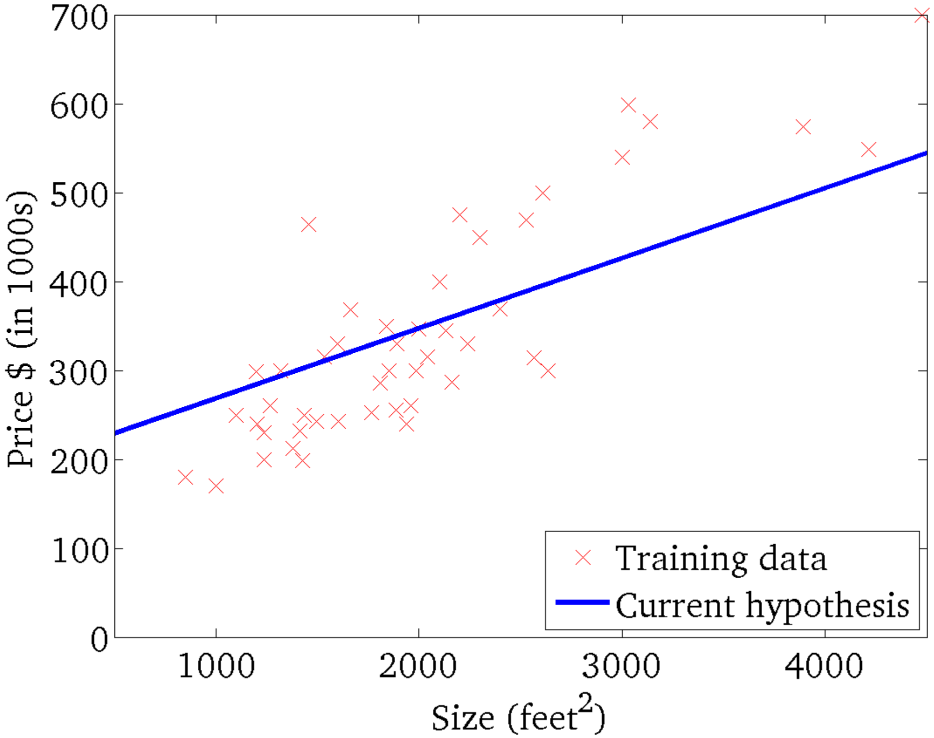


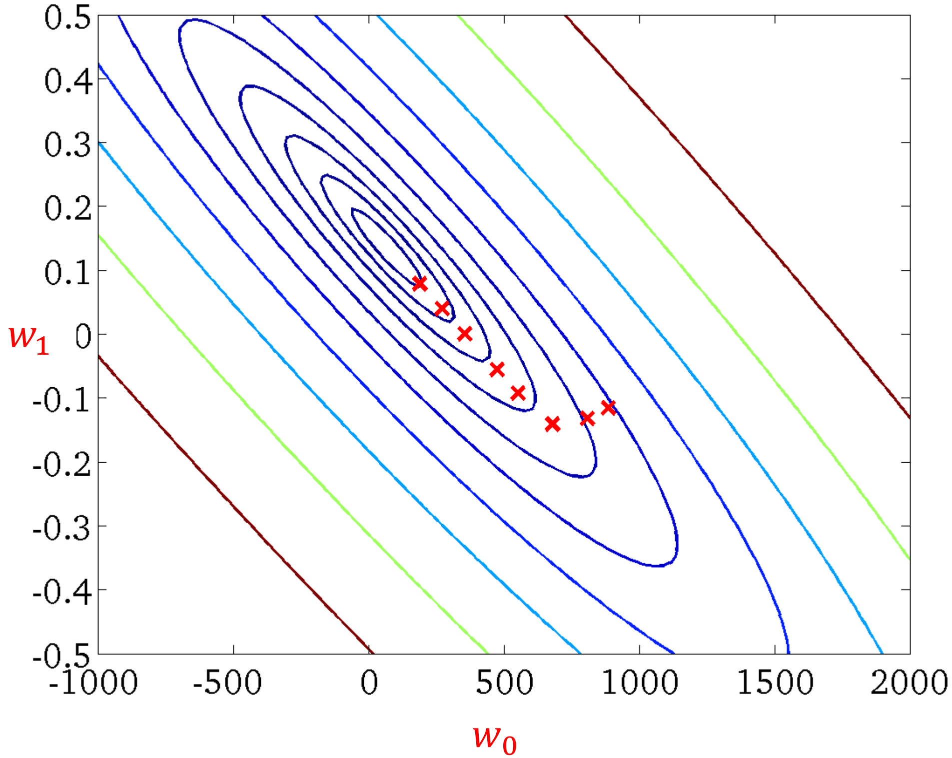
Linear Regression



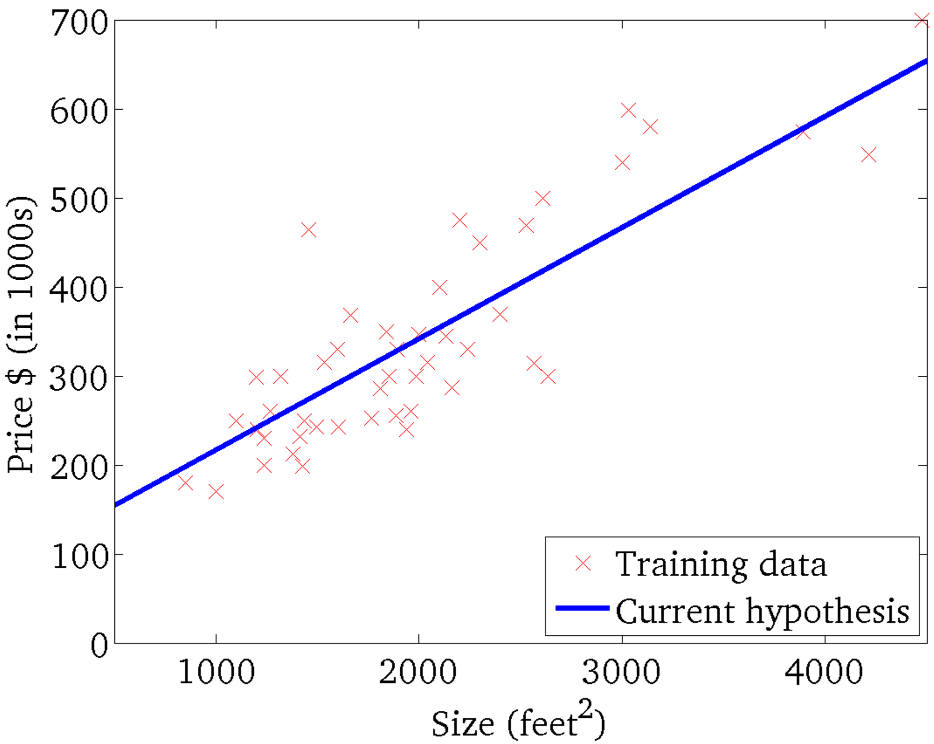


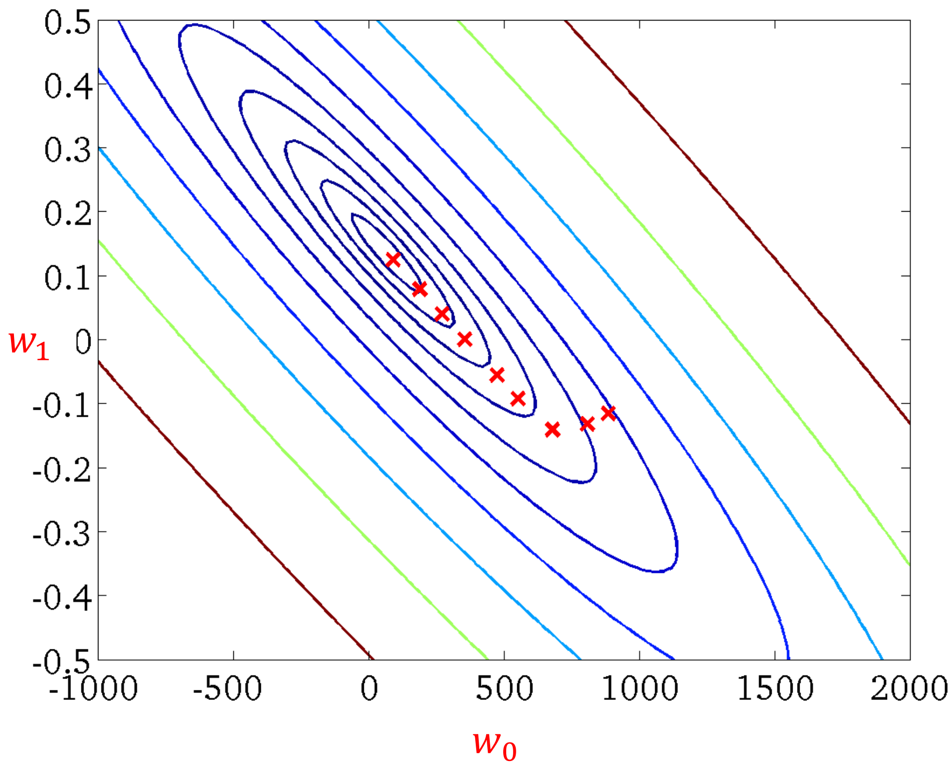
Linear Regression





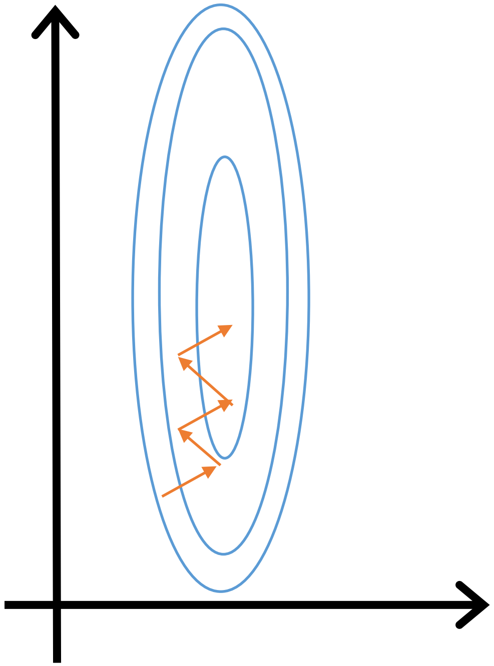
Linear Regression





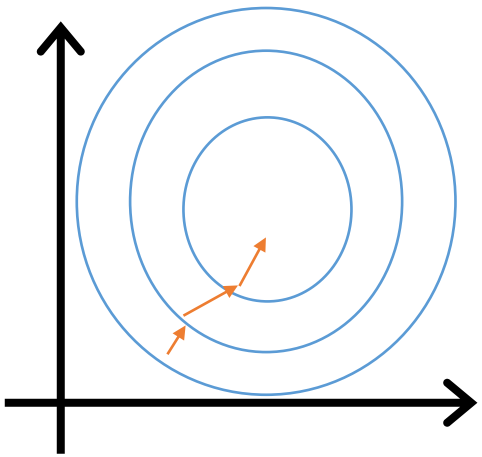
Feature Scaling

Problem: features are not on a similar scale

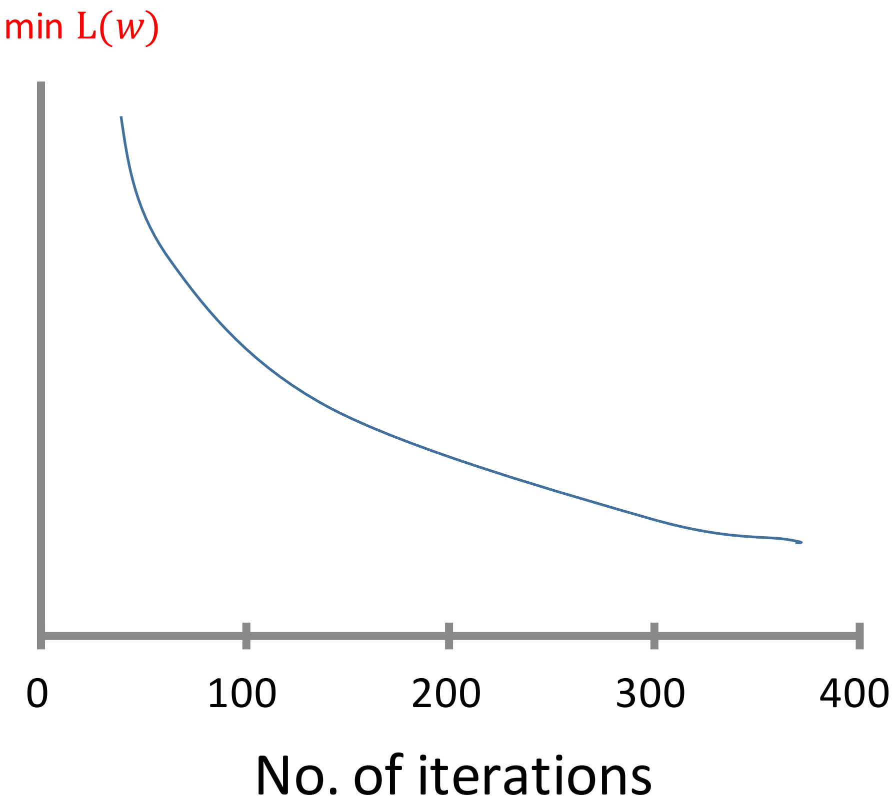


Solution: Mean Normalization

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

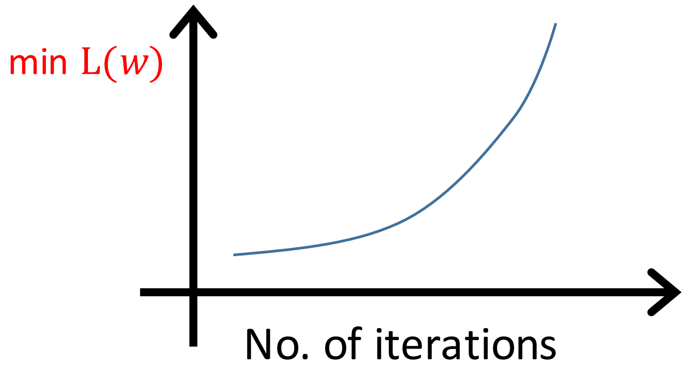


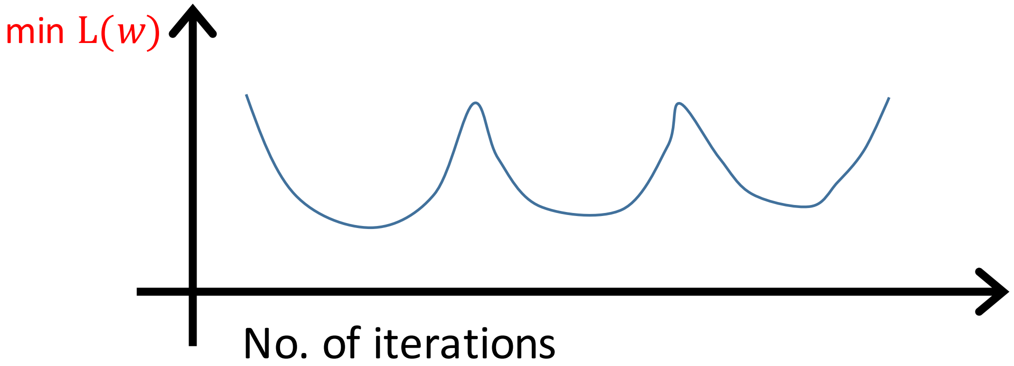
Gradient Descent: Debugging



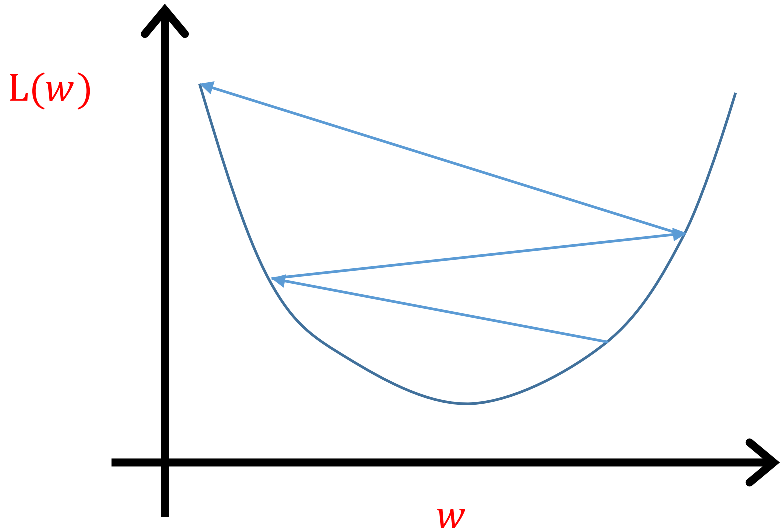
- How to make sure gradient descent is working correctly?
- How to choose learning rate
- Solution: Declare convergence if $\mathrm{L}(w)$ decreases by less than 10^{-3} in one iteration.

Gradient Descent: Debugging





Gradient descent not working. Use smaller α.



- For sufficiently small α, $\mathrm{L}(w)$ should decrease on every iteration.
- But if α is too small, gradient descent can be slow to converge.

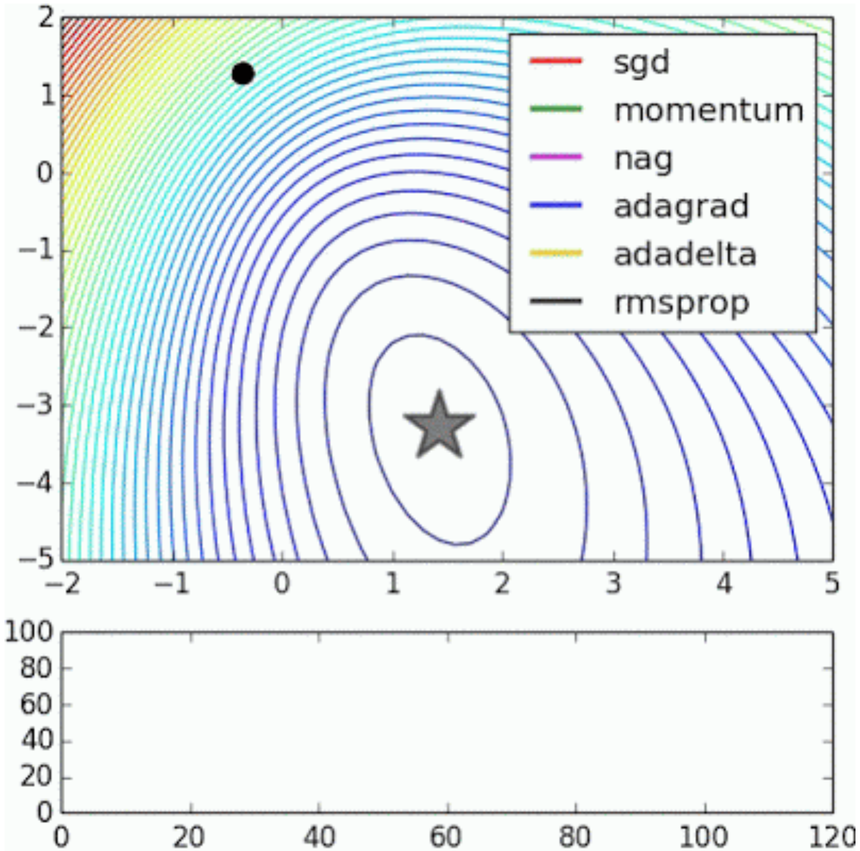
Gradient Descent: Debugging

- If α is too small: slow convergence.
- If α is too large: $\mathrm{L}(w)$ may not decrease on every iteration; may not converge.

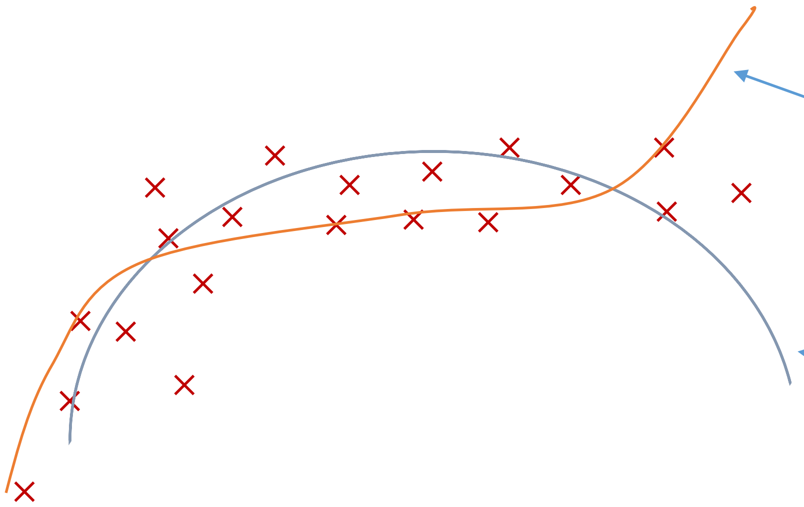
To choose α, try

$$\ldots, 0.001, \qquad , 0.01, \qquad , 0.1, \qquad , 1, \ldots$$

Other Optimization Methods



Polynomial Regression



$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$

Good but go up !

$y = w_0 + w_1 x + w_2 x^2$

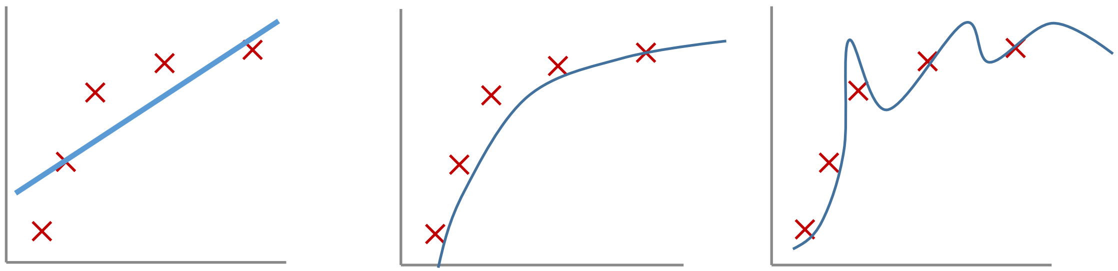
Good but go down !

Overfitting vs. Underfitting

Underfitting
High Bias
Low variance

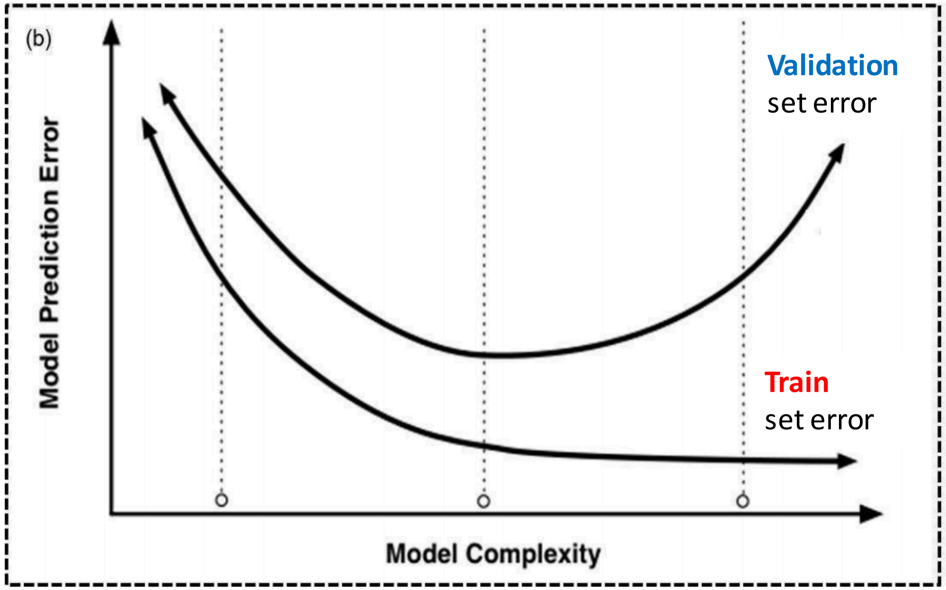
Trade-off

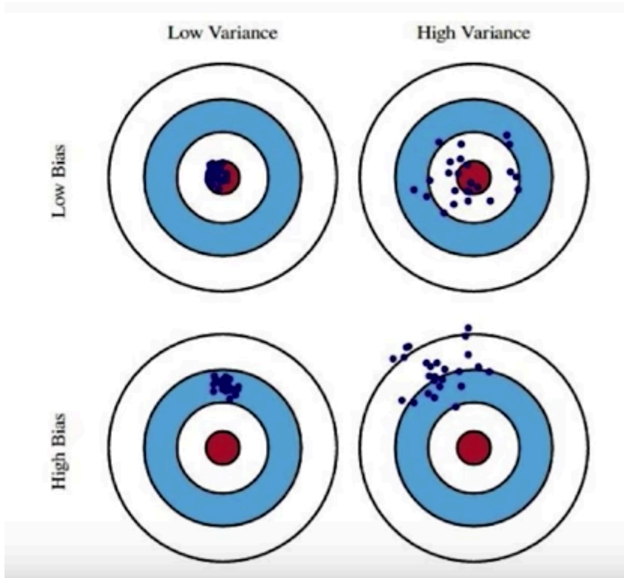
Overfitting
Low Bias
High Variance



Bias-Variance Tradeoff

Expected error (Human or Bayes optimal): **0%**	**Train** set error	1%	15%	15%	0.5%
	Validation set error	11%	16%	30%	1%
		High variance	High bias	High bias High variance	Low bias Low variance

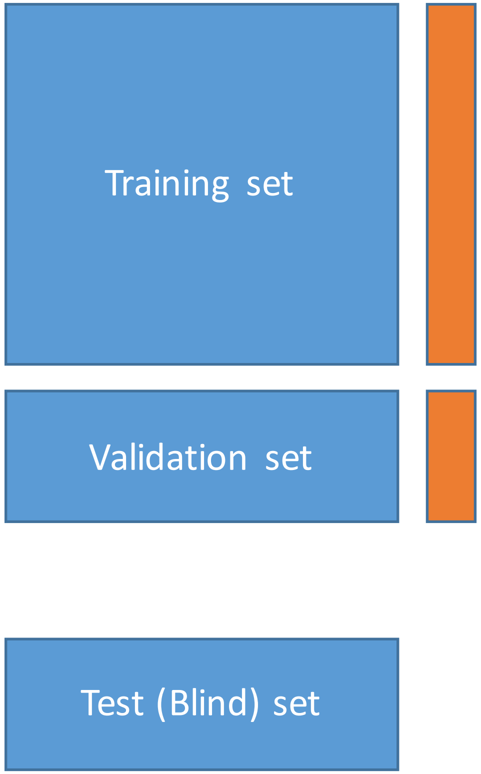


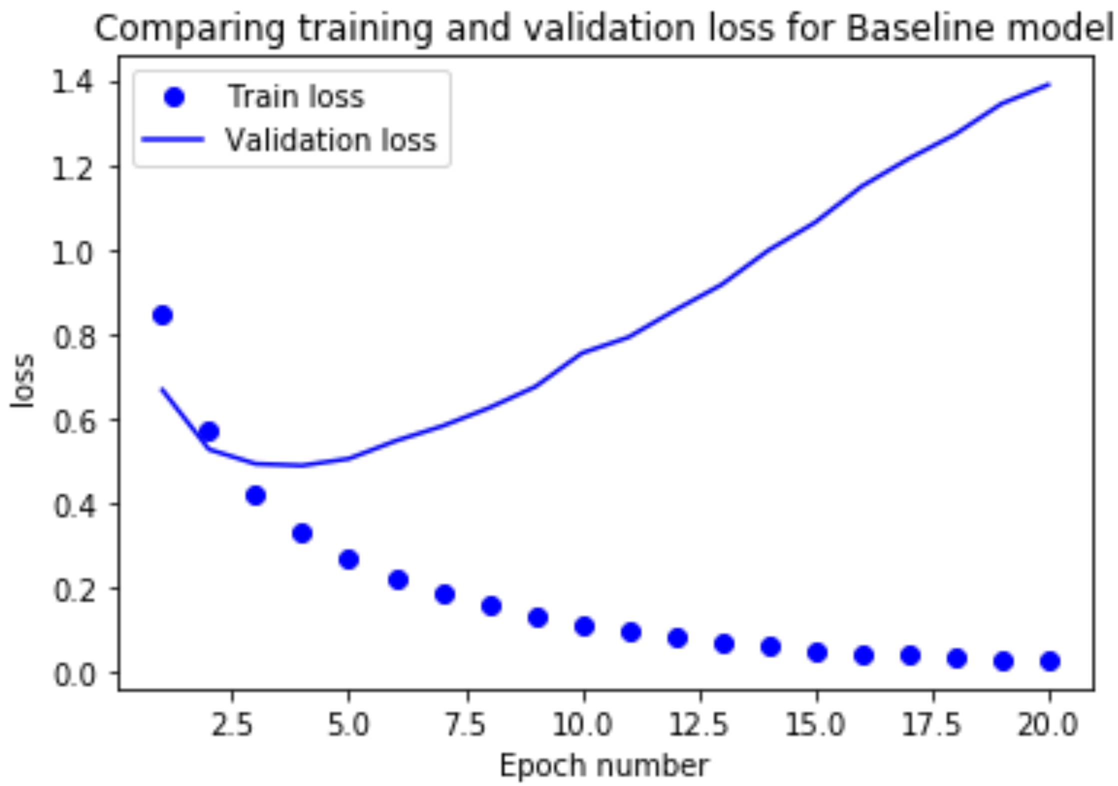


Address Overfitting

- Detect Overfitting
  - Performance analysis (Cross-Validation)
- Avoid Overfitting
  - Fewer features (Feature Selection, Dimensionality Reduction)
  - Constraint the model (Regularization : minimum loss $\mathrm{L}(w) + \lambda\, ww^T$)
  - Model Selection (Tune hyper-parameters using 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)
  - Derivability !

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

Feature Selection

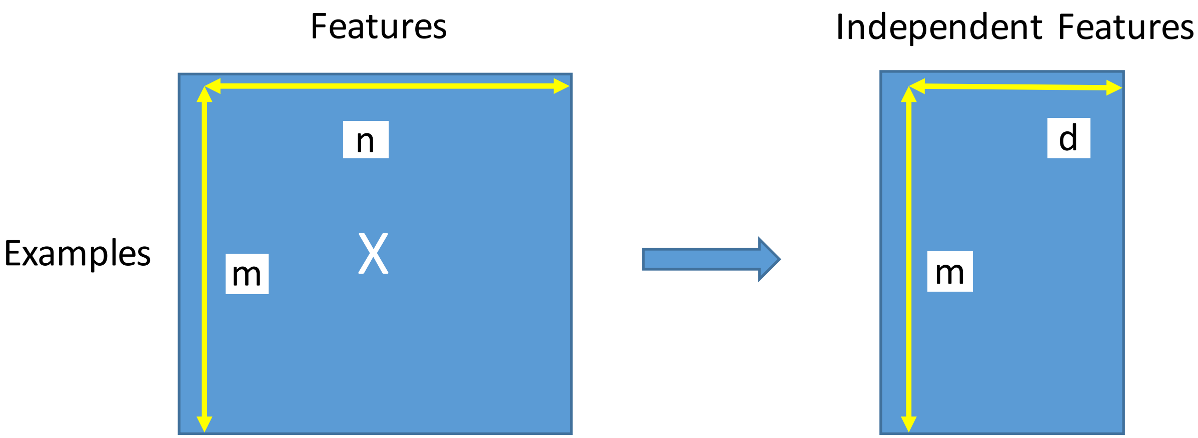
- Best Subset Selection
  Fit a separate least squares regression for each possible combination of the n features: 2^n possibilities!

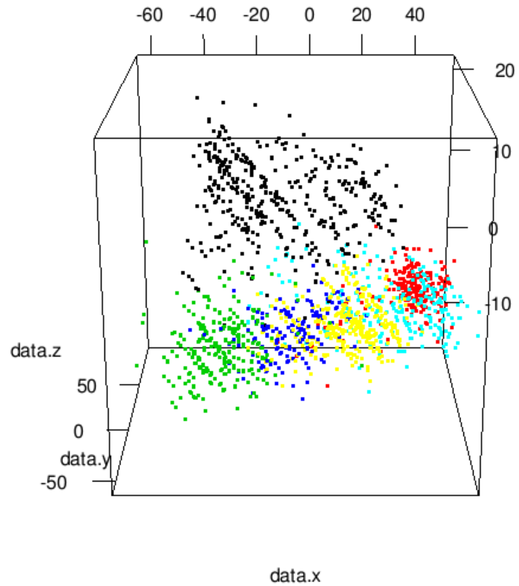
- Forward Stepwise Selection
  Begins with a model containing no feature, and then adds the feature that gives the greatest improvement (smallest cost) to the model, one-at-a-time.

- Backward Stepwise Selection
  Begins with a model containing all feature, and then removes the feature that gives the smallest improvement (highest cost) to the model, one-at-a-time.

Dimensionality Reduction

- Reducing or extracting features





Regularization

- See regularization as a penalty against complexity. Increasing the regularization strength penalizes "large" W

- The goal is to prevent the model from picking up "peculiarities," "noise," or "imagines a pattern where there is none.”

Regularization: Ridge Regression (L_2 norm)

Linear Regression

$$\hat{y} = h_w(x) = w_0 + w_1 x_1 + w_2 x_2$$

if λ is set to be extremely large, then w_j have to be very small.

→Algorithm results in underfitting

→Gradient Descent will fail to converge

$$\underset{w}{\text{minimize}} \quad L(y, \hat{y})$$

L_2 norm

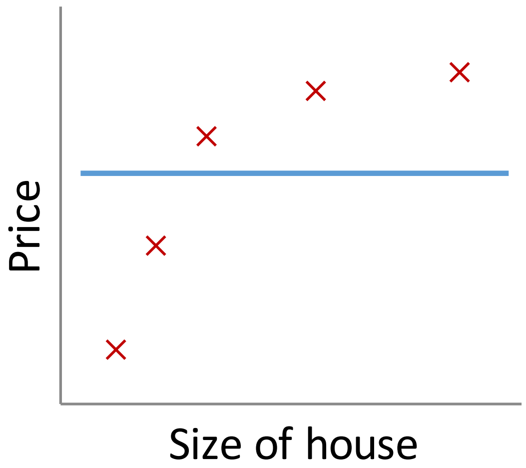
$$\underset{w}{\text{minimize}} \; L(y, \hat{y}) + \lambda \boxed{\sum_{j=1}^{n} w_j^2}$$

Do not regularize for j=0

Training $\quad w_0 = 1, w_1 = 2, w_2 = 0.01$

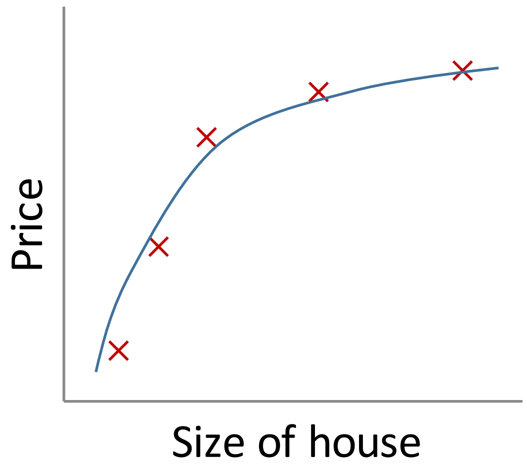
Test $\quad w_0 = 1, w_1 = 2, w_2 = 0$

Regularization: Ridge Regression (L$_2$ norm)



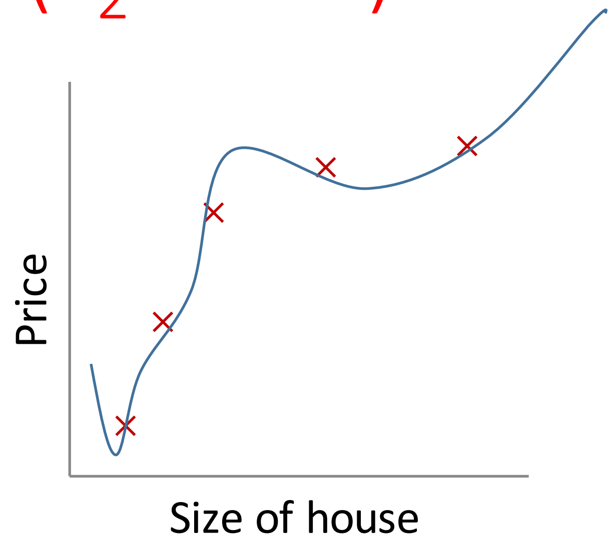
$$y = w_0 + \cancel{w_1}x + \cancel{w_2}x^2 + \cancel{w_3}x^3$$

Underfitting



$$y = w_0 + w_1x + w_2x^2 + \cancel{w_3}x^3$$

Tradeoff



$$y = w_0 + w_1x + w_2x^2 + w_3x^3$$

Overfitting

Regularization: LASSO Regression (L_1 norm)

Linear Regression

$$\hat{y} = h_w(x) = w_0 + w_1 x_1 + w_2 x_2$$

- LASSO: Least Absolute Shrinkage and Selection Operator
- LASSO is not differentiable for every value of w, but performs best feature selection

$$\underset{w}{\text{minimize}} \quad L(y, \hat{y})$$

L_1 norm

$$\underset{w}{\text{minimize}} \quad L(y, \hat{y}) + \lambda \boxed{\sum_{j=1}^{n} |w_j|}$$

Do not regularize for j=0

Training $\quad w_0 = 1, w_1 = 2, w_2 = 0$

Test $\quad w_0 = 1, w_1 = 2, w_2 = 0$

Model Selection

- Hyper-Parameters Tuning
  - λ : regularization hyper-parameter
  - d: degree of polynomial
  - Etc.
- Grid Search
- Randomized search