First Step in Machine Learning: Understanding Machine Learning

Babak Tourani
AOUG, June 2018

Agenda

- What is Machine Learning?
- Supervised Learning
  - Linear Regression
  - Logistic Regression
  - Neural Networks
- Unsupervised Learning
  - Clustering
- ML in RDBMS example
- Available Tools
- Resources

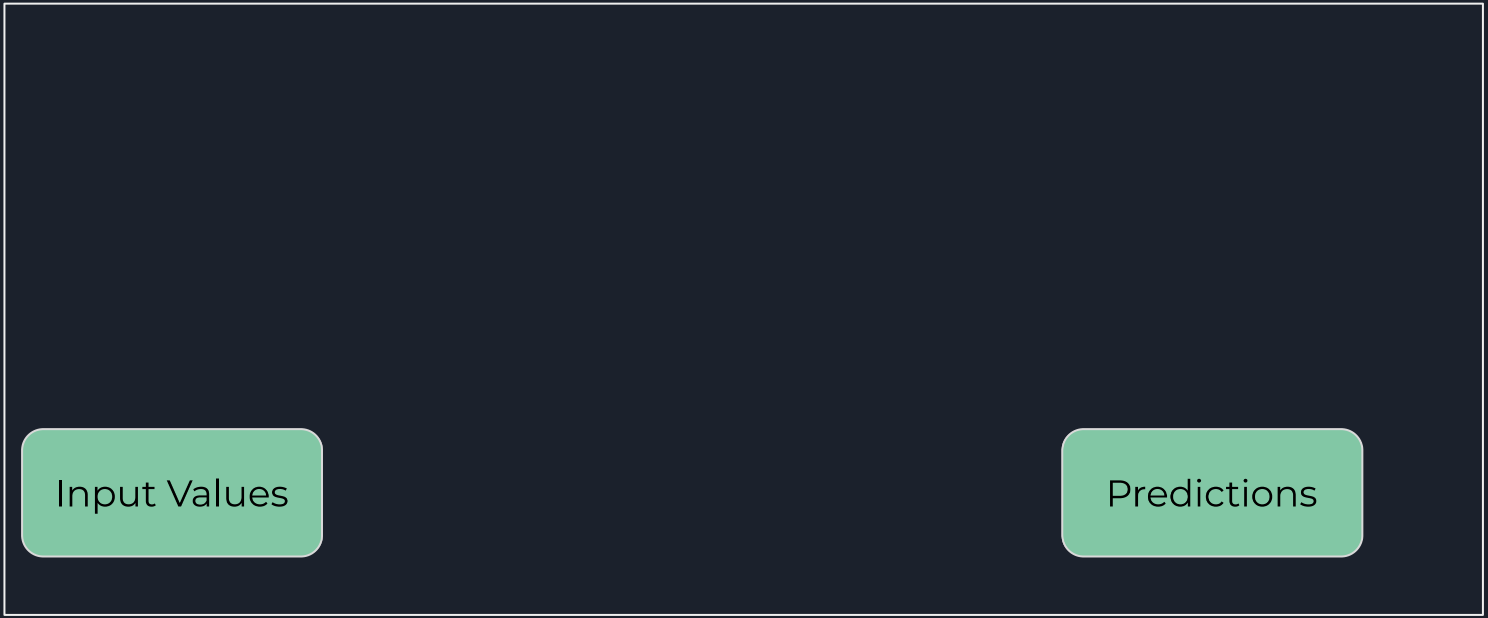
Who Am I?

- Oracle Developer/DBA
- Started with Oracle 8i, PSP
- Asst coach of Iran’s Basketball team
- Radio/TV presenter & producer,
  BBC World Service
- Ontologies, Graph, Time-Series DBs
- NOT a Machine Learning expert!

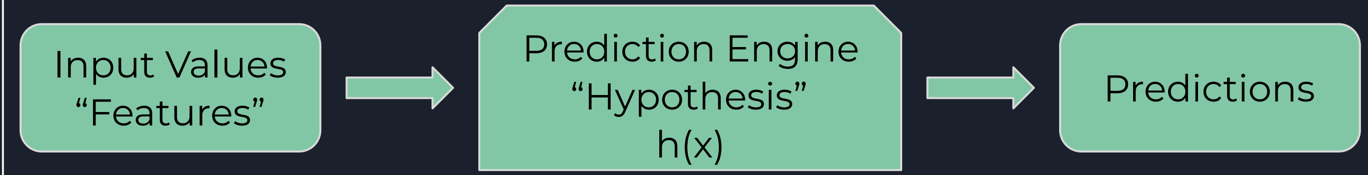
What Machine Learning Is Trying To Do?

Predictions

What Machine Learning Is Trying To Do?



What Machine Learning Is Trying To Do?



What Is a Hypothesis?

- Simply a mathematical function
- Mapping between inputs and predictions

Example: Predicting house prices based on house size

$$h(x) = \theta_0 + \theta_1 x_1$$

Hypothetical Values:

$$\theta_0 = 100, \theta_1 = 0.1$$

$$x_1 = 1400 \ (size \ of \ house \ in \ ft^2)$$

$$\implies \quad h(x) = 100 + 0.1 * 1400 = 240$$

Hypothesis/Model

- Hypothesis is a function:

$$h(x) = \theta_0 + \theta_1 x_1$$

- Model is an instance of hypothesis:

$$\theta_0 = 100, \theta_1 = 0.1$$

- Model is “an artefact created by the training process”
  * Amazon Machine Learning

- Model is portable and can be deployed on a different environment

Shape of the Models

What Machine Learning Is Trying To Do?

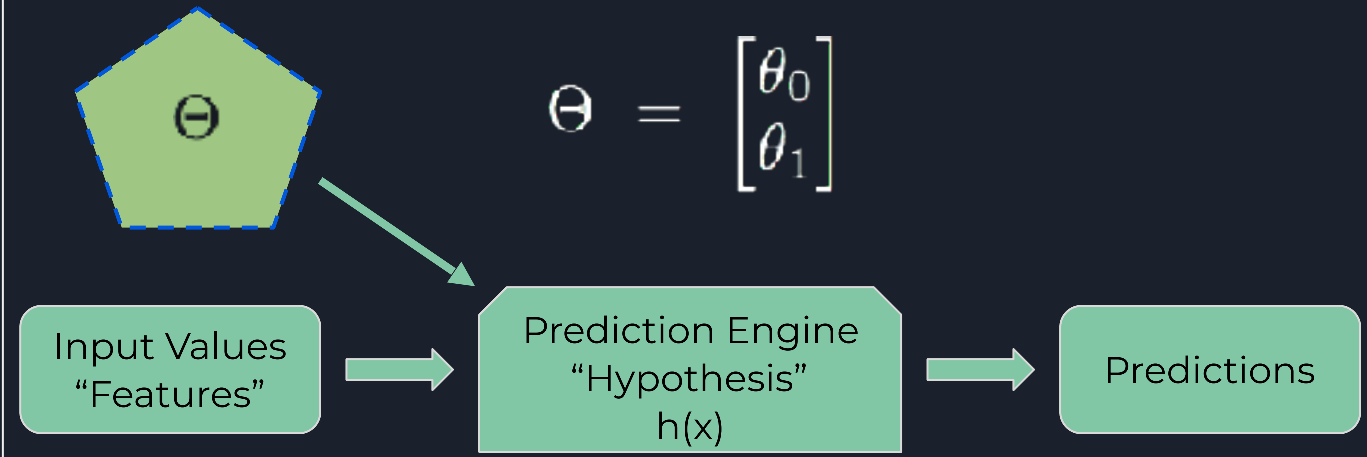
$$h(x) = \theta_0 + \theta_1 x_1$$

Input Values "Features" → Prediction Engine "Hypothesis" h(x) → Predictions

What Machine Learning Is Trying To Do?

$$h(x) = \theta_0 + \theta_1 x_1$$

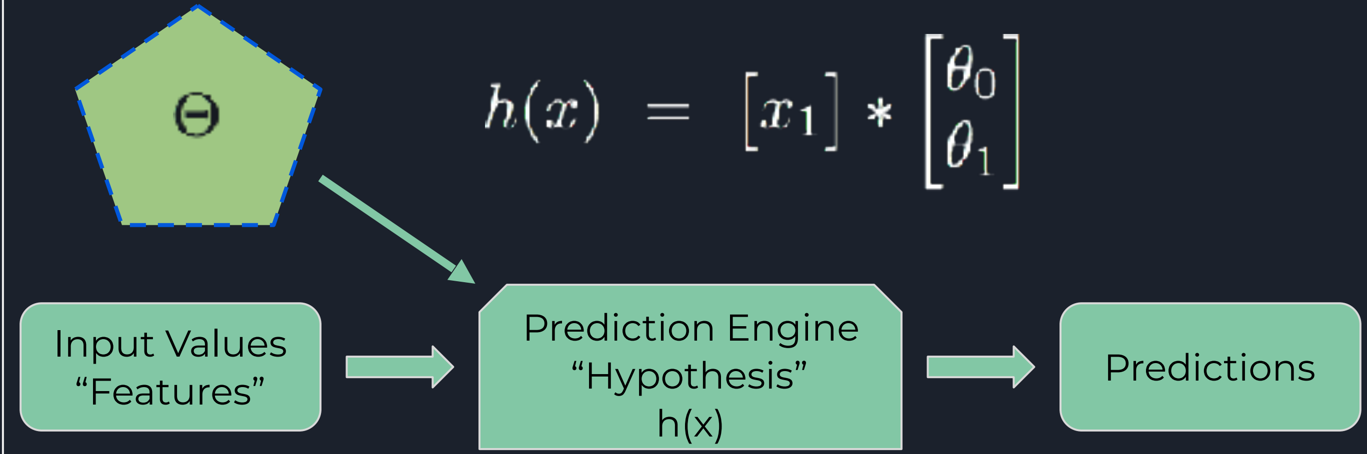
$$\Theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$



What Machine Learning Is Trying To Do?

$$h(x) = \theta_0 + \theta_1 x_1$$

$$h(x) = [x_1] * \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$

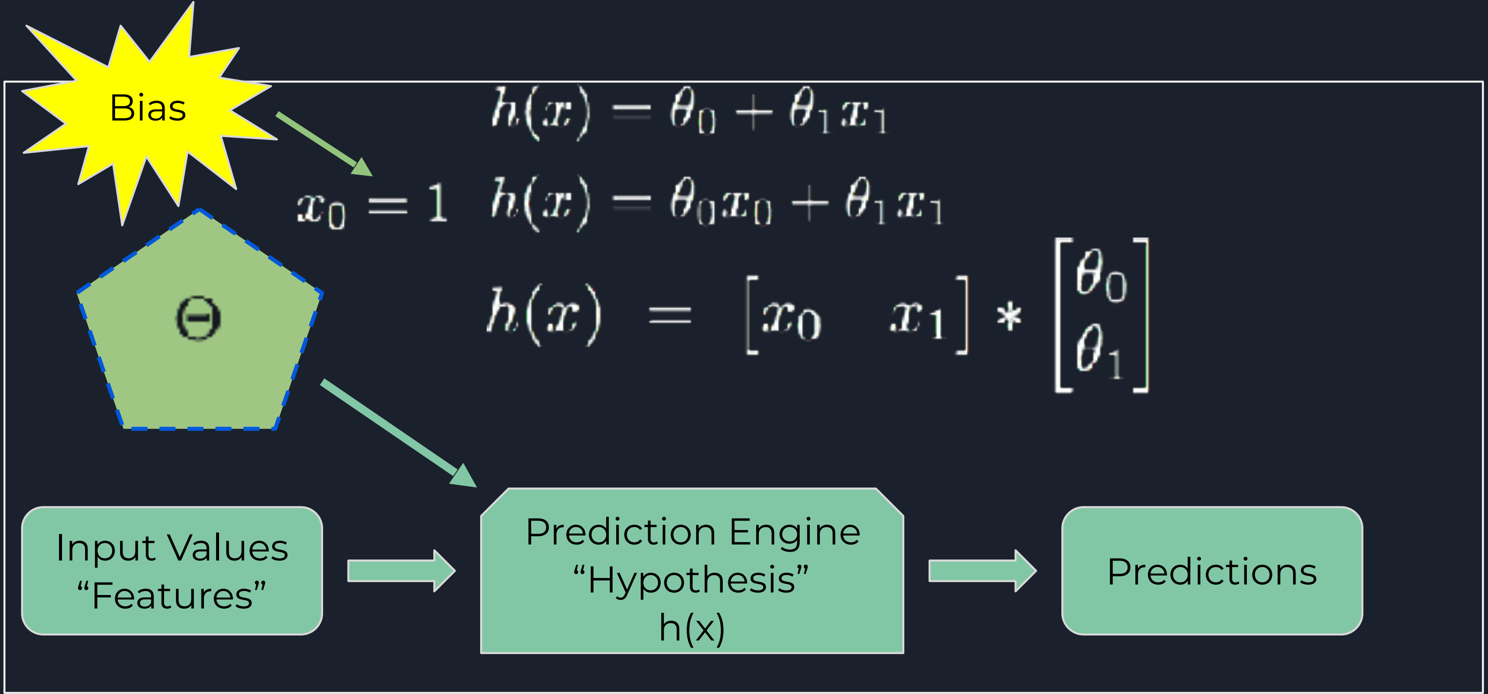


What Machine Learning Is Trying To Do?

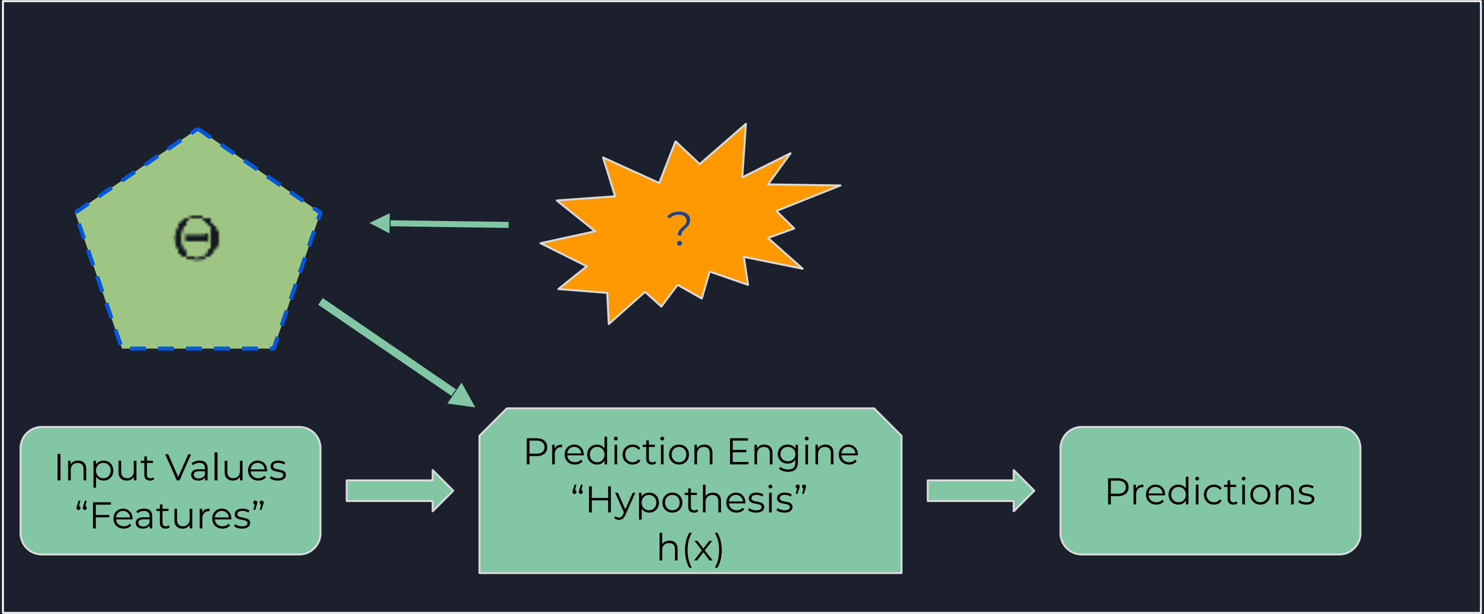
$$h(x) = \theta_0 + \theta_1 x_1$$

$$x_0 = 1 \quad h(x) = \theta_0 x_0 + \theta_1 x_1$$

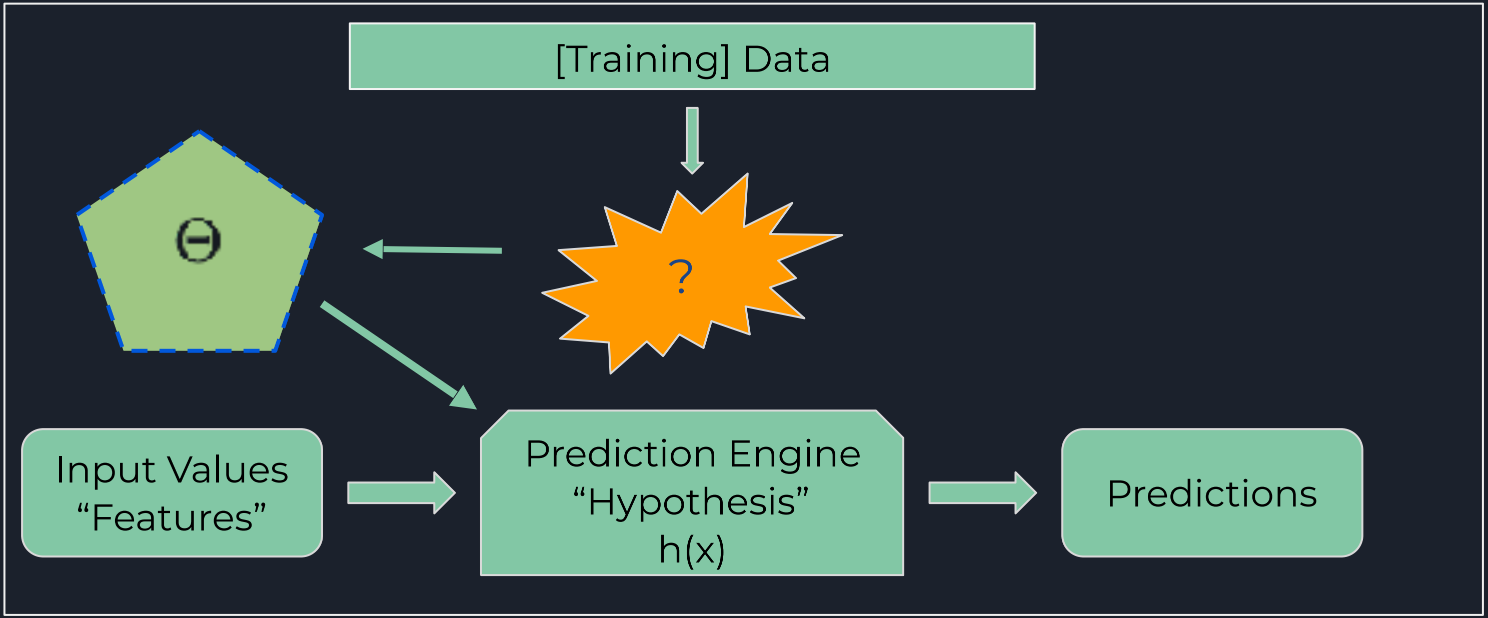
$$h(x) = \begin{bmatrix} x_0 & x_1 \end{bmatrix} * \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$$



What Machine Learning Is Trying To Do?

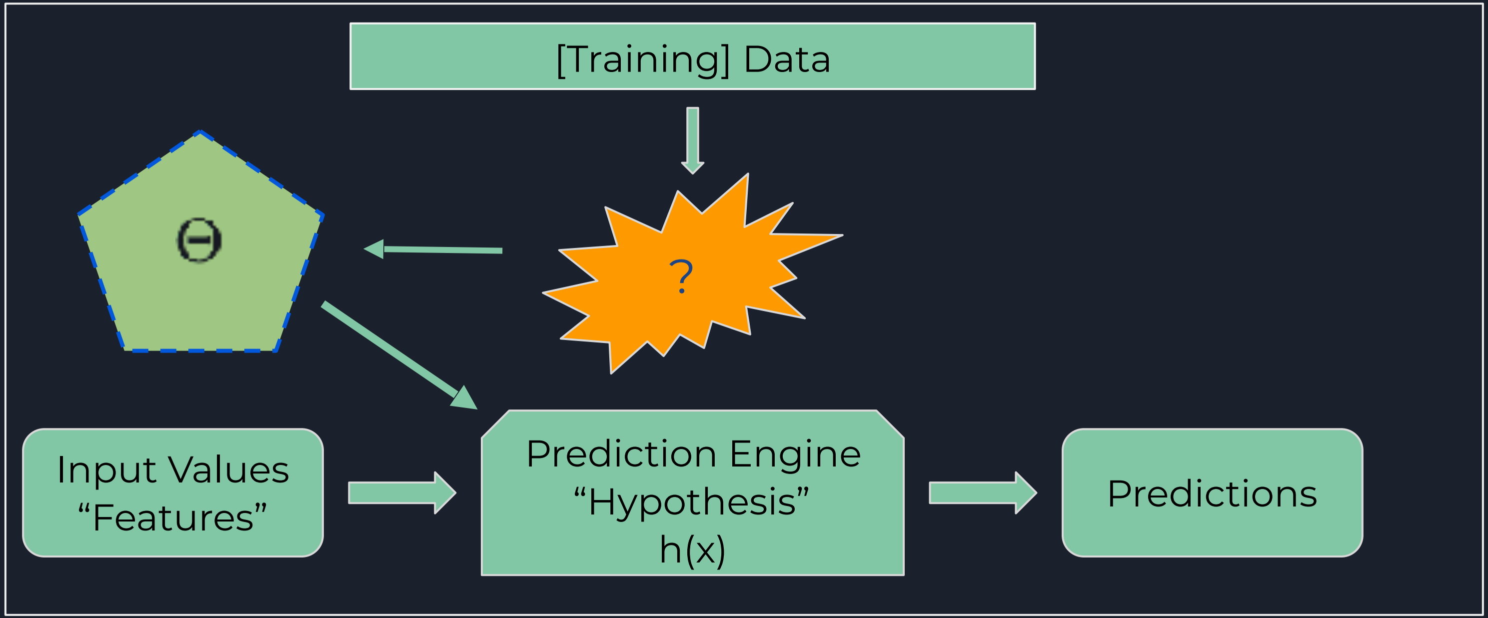


Supervised Learning

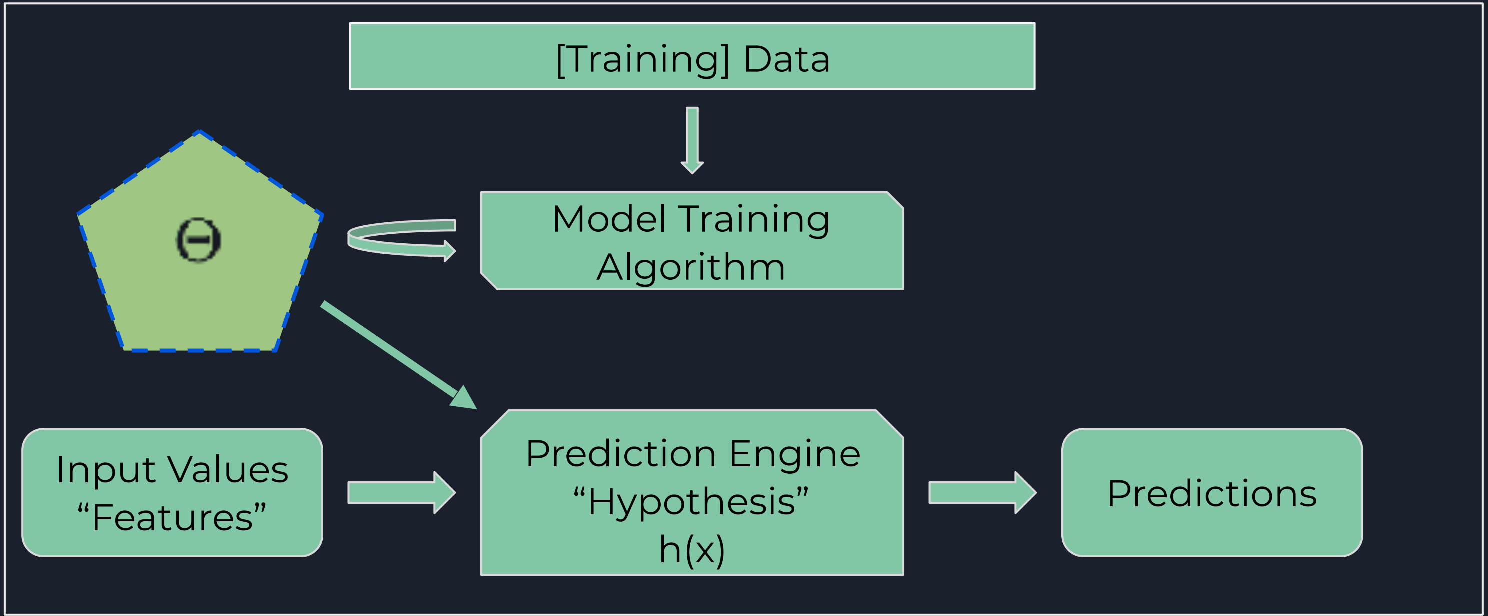


Machine Learning Genesis

What Machine Learning Is Trying To Do?



What Machine Learning Is Trying To Do?



“Train the Model!”

The “Hello World” Data Sample

- House prices in Portland, Oregon (<50 samples)

Size of The House (sq ft)	Price (USD)
2104	399900
1600	329900
1416	232000

* https://github.com/girishkuniyal/Predict-housing-prices-in-Portland

The “Hello World” Data Sample

- House prices in Portland, Oregon (<50 samples)

Size of The House (sq ft)	Price (USD)
2104	399900
1600	329900
1416	232000

X

Experience

Observation

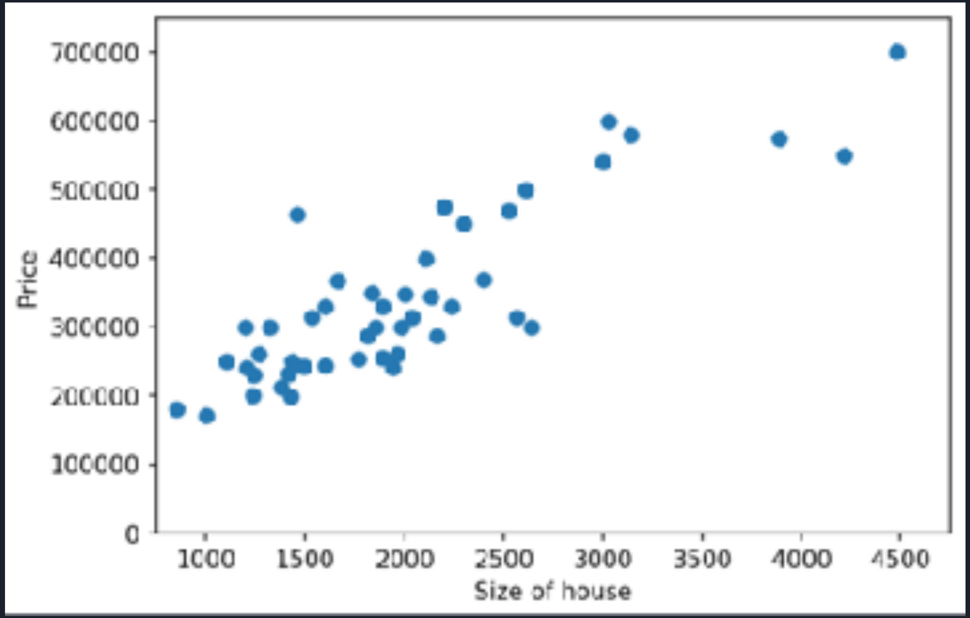
Feature

Label

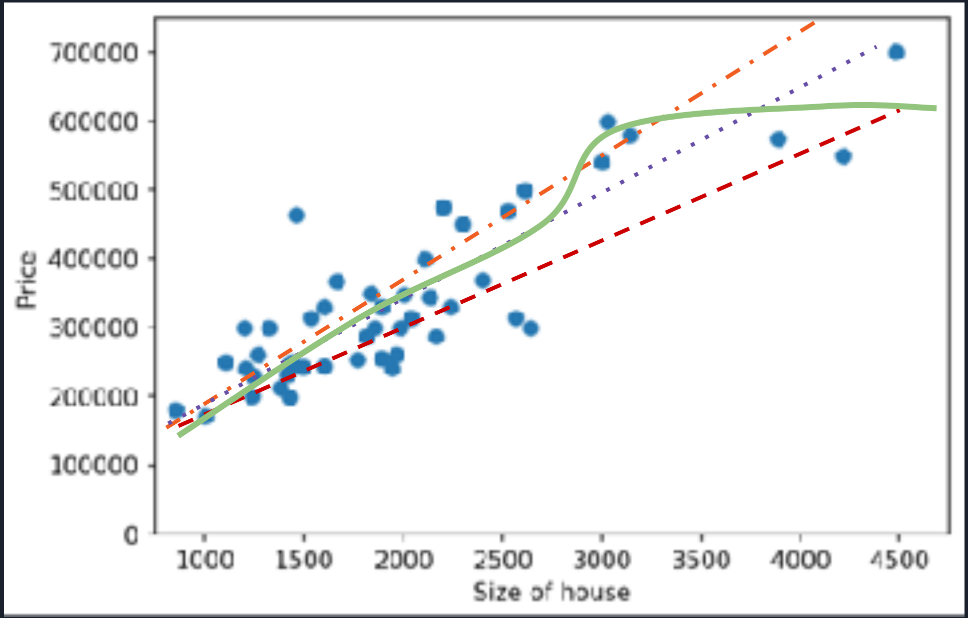
y

Target

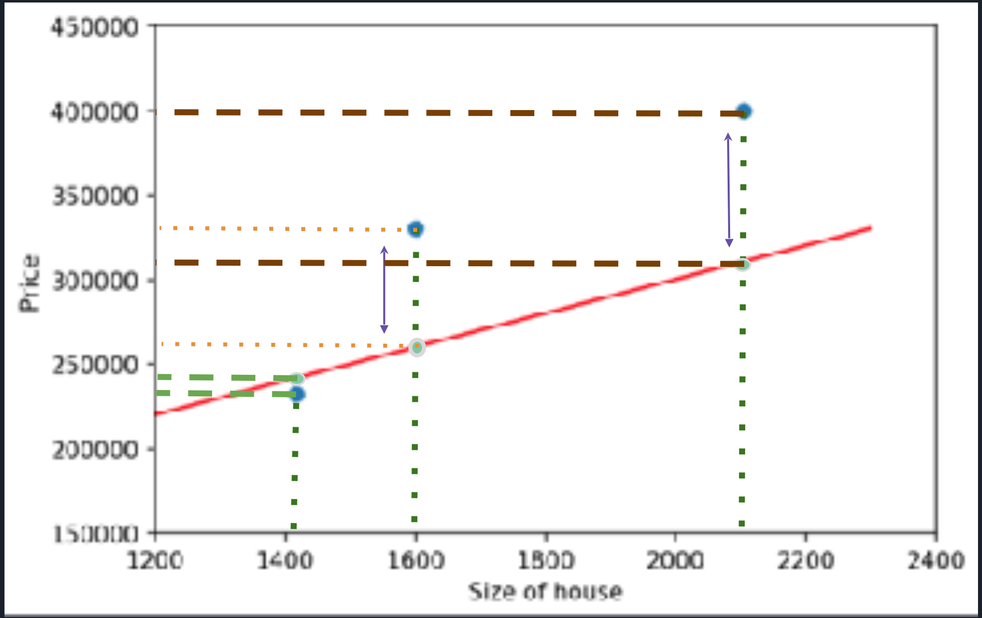
The “Hello World” Data Sample



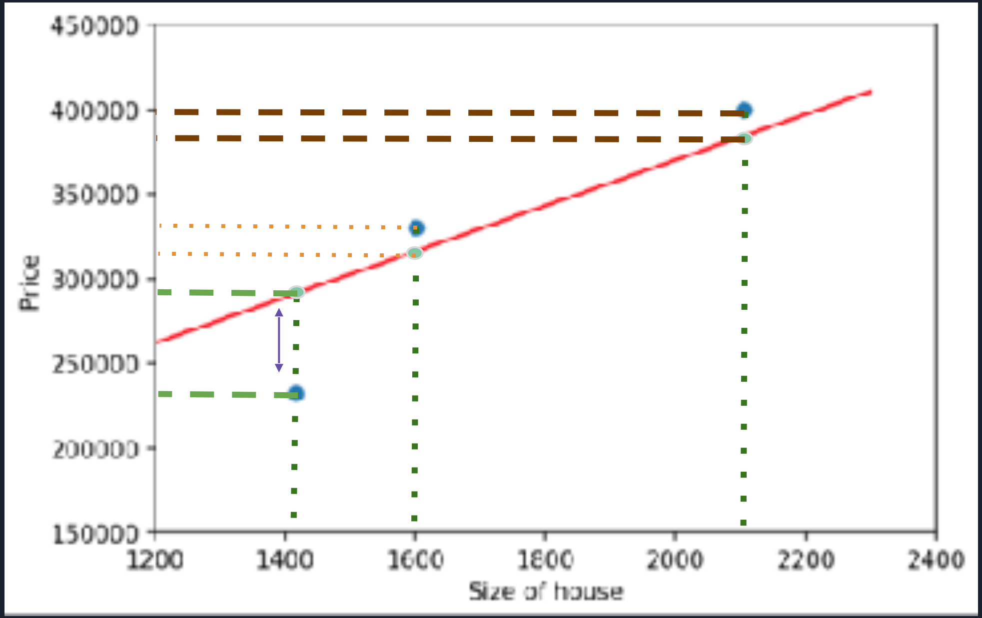
Which One Is the Best Model?



Best Fit = Lowest [Prediction] Error



Best Fit = Lowest [Prediction] Error



How to Find the “Best Fit?”

$$h(x) = \theta_0 + \theta_1 x_1$$

- Cost Function, a.k.a. Error Function
  Example: Mean Squared Error

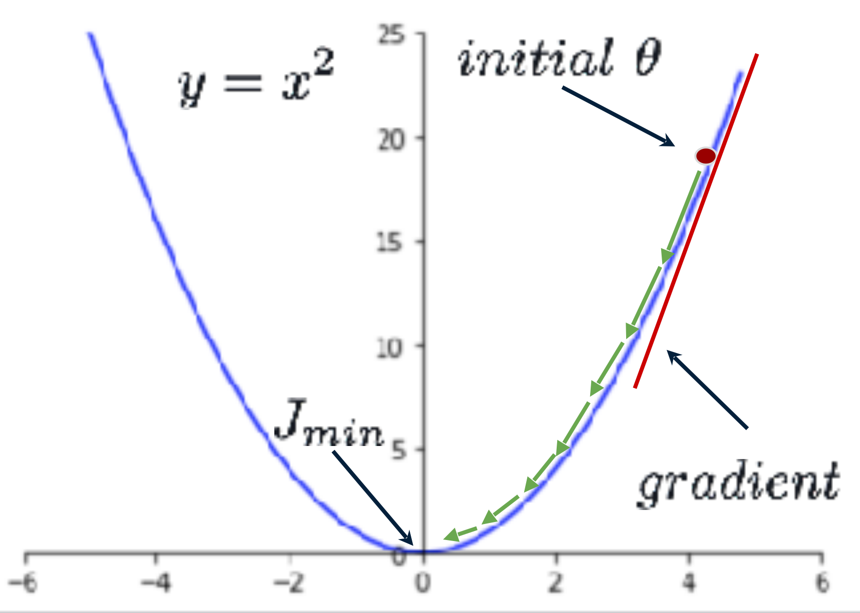
$$J(\Theta) = \frac{1}{2m} \sum_{i=1}^{m} (h(x) - y)^2$$

- Find Thetas which minimizes the cost function; i.e. leads to lowest error.
- Using Gradient Descent

* https://developers.google.com/machine-learning/crash-course/fitter/graph

Gradient Descent Algorithm

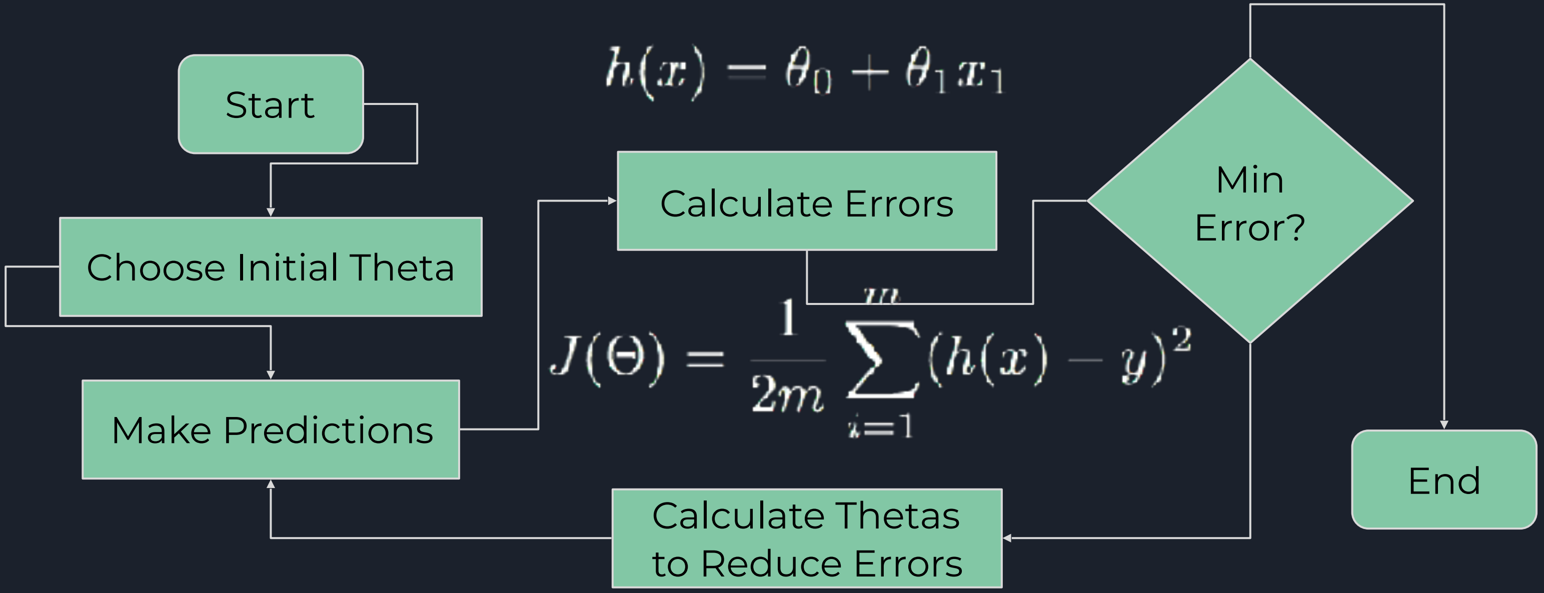
- Minimizes functions
- Iterative process
- It "adapts" as it gets closer
  To local minimum
- Size of the steps depends on
  "Learning Rate"



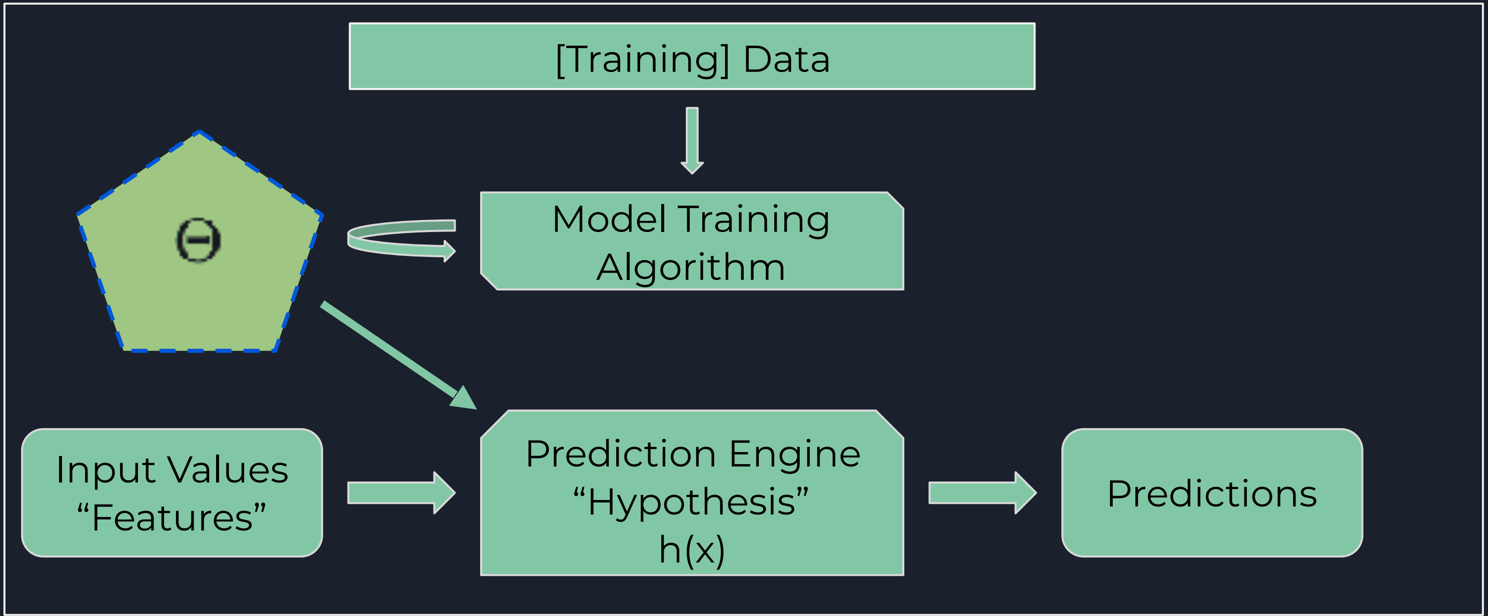
How it works:

- Multiple passes through data
- Calculates cost
- Calculates new Theta
  based on cost
- Convergence => Stops!

Training/Learning Process



What Machine Learning Is Trying To Do?



Performance

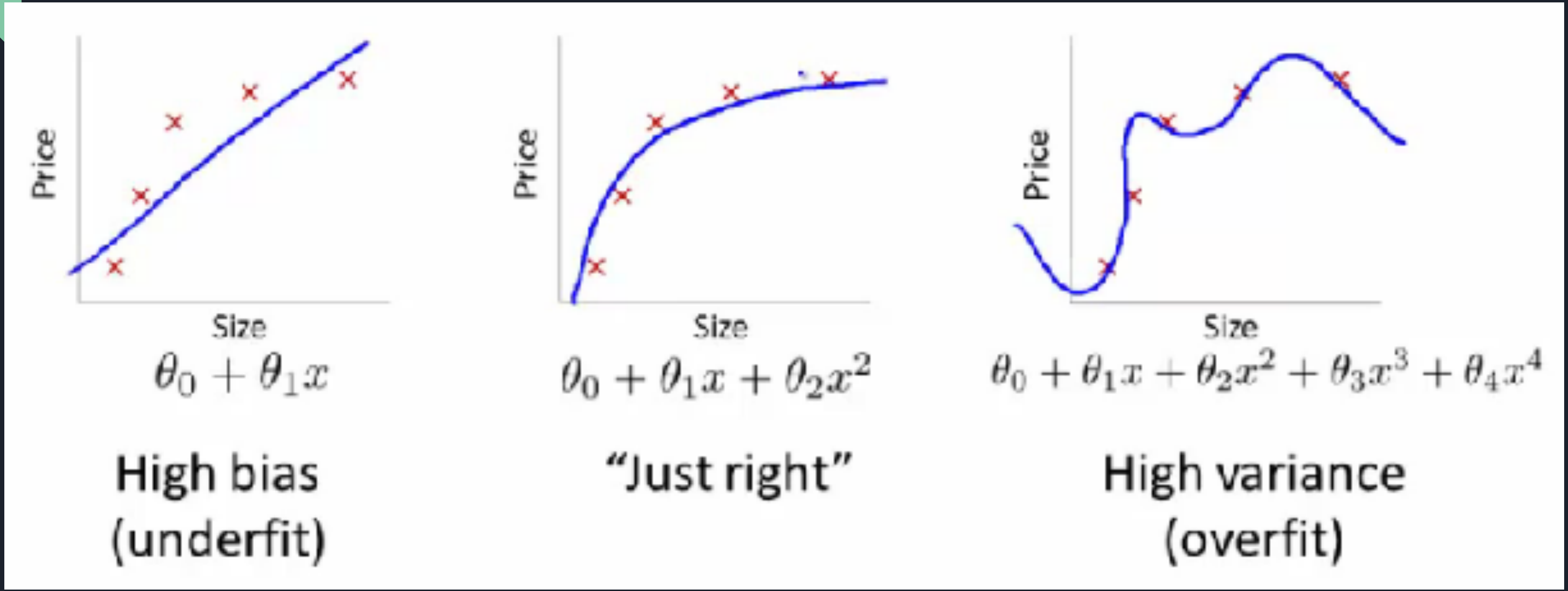
- Not about speed
- Not about resource consumption
- Nothing to do with scalability of process

- It’s about the model’s ability to do correct predictions.
  i.e. High performance models can provide better predictions.

Parameters vs. Hyperparameters

	Parameters	Hyperparameters
Valuation	Algorithm $(\theta_0,\ \theta_1,\ \theta_2, \ldots)$	Human (polynomial degree)
Optimization Method	Cost Function	Chosen According to Their Accuracy

Bias vs. Variance



* Image: Machine Learning, Stanford University via Coursera, Andrew Ng

Regression vs. Classification

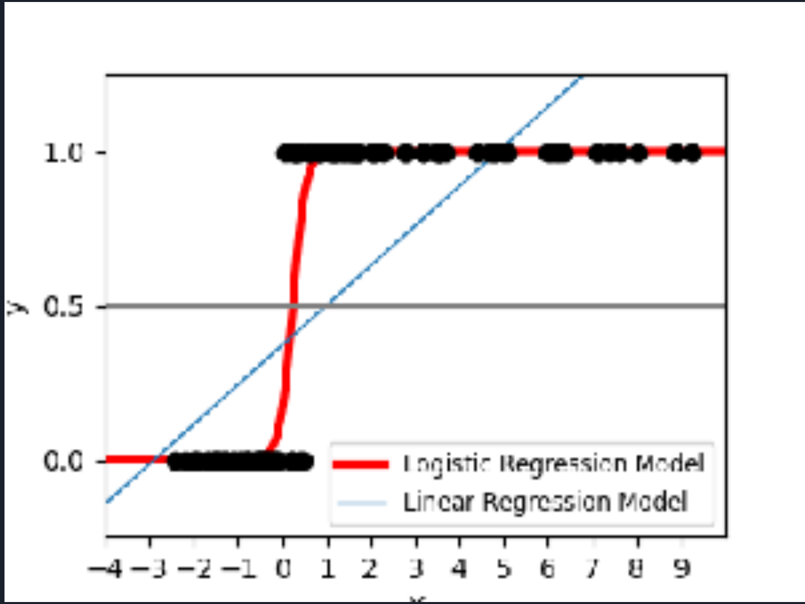
Regression:

- Continuous values (numbers)
  e.g. House price
  Poverty rate among teenage parents

Classification:

- Discrete values (classes)
  E.g. Emails: spam or not spam
  Tumour x-rays: malignant or benign
  Picture identification: cat or dog

Regression vs. Classification

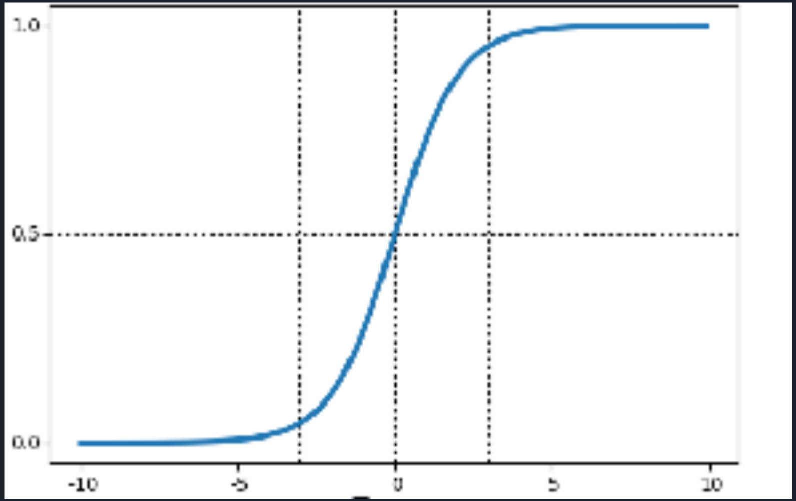


Sigmoid Function

A.k.a. Logistic Function

Output i.e. g(z):
Estimated probability that y = 1

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



e.g. If g(z) = 0.85
=> P(y=1 | z) = 85%

Logistic [AND] Regression?!

How to present “features” in the input to the function?

Using Linear Regression Hypothesis!

$$h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x2 = x\Theta$$

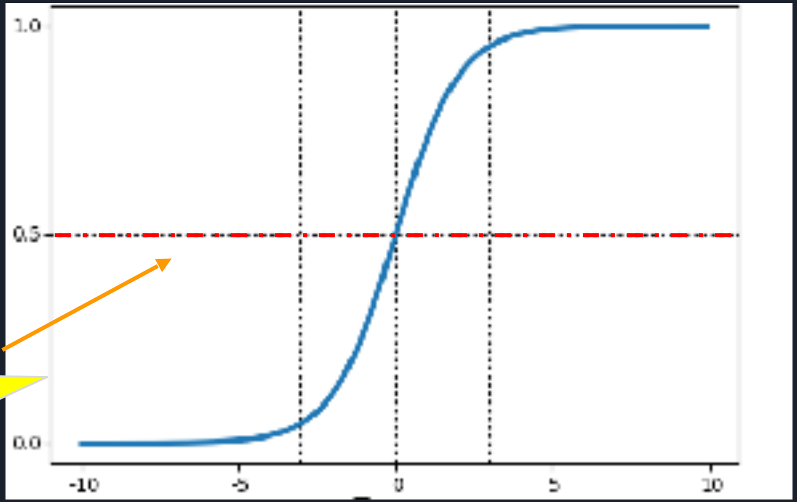
e.g.

$$x_1 \implies \text{tumor size}$$

$$x_2 \implies \text{growth rate}$$

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

Decision Boundary



Binary Classification

e.g. Decision Boundary: 67%

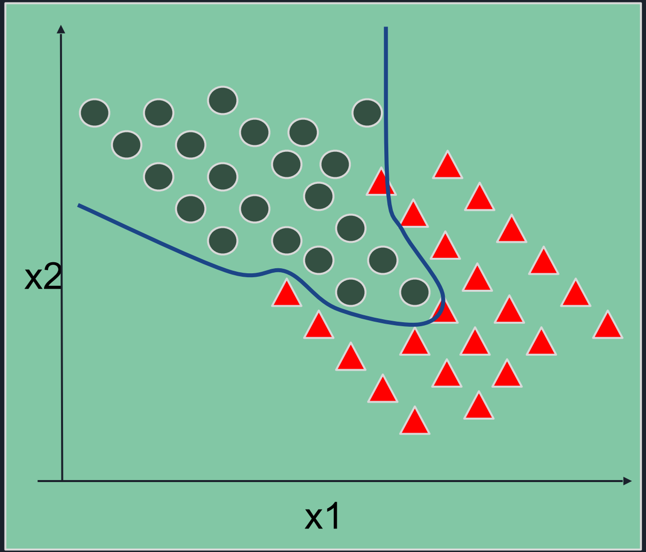
Age	Married	Incoming	Outgoing	Loans	Sigmoid	Eligible
25	0	2500	1400	0	80%	1
46	1	4300	2200	1	73%	1
37	1	3700	2900	1	30%	0
19	0	3400	1200	1	47%	0

Complicated Polynomials

- Modelling small number of features using polynomials is possible, but requires new features

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 + \theta_6 x_1^2 x_2 + \theta_7 x_1 x_2^2 + \theta_8 x_1^3 x_2^2 + \ldots$$

- => introduce hundreds of features which is computationally expensive and not easy to choose



What We Need Is...

An algorithm that can:

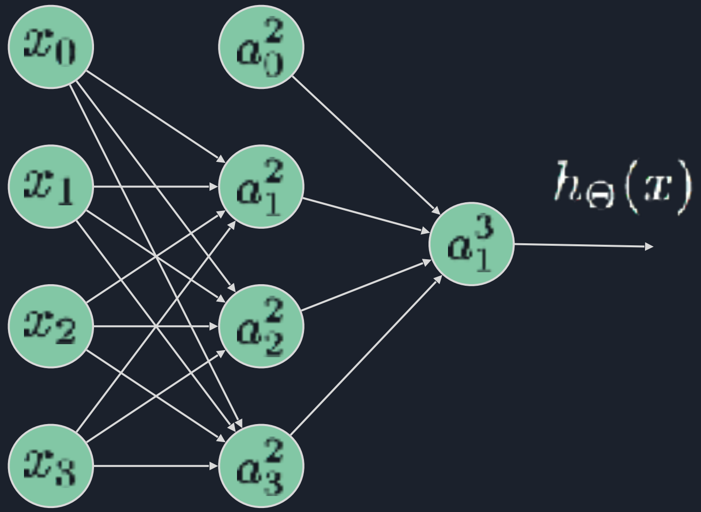
- mix and match features
- assess the impact of each combination

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \rightarrow [\quad] \rightarrow h_\Theta(x)$$

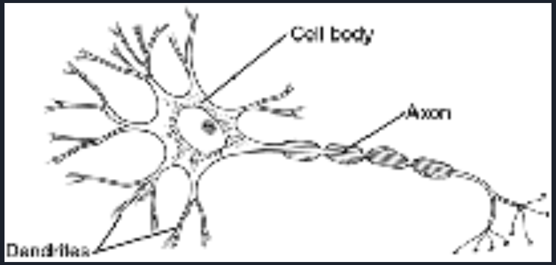
$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2$$
$$+\theta_4 x_2^2 + \theta_5 x_1 x_2 + \theta_6 x_1^2 x_2$$
$$+\theta_7 x_1 x_2^2 + \theta_8 x_1^3 x_2^2 + \ldots$$

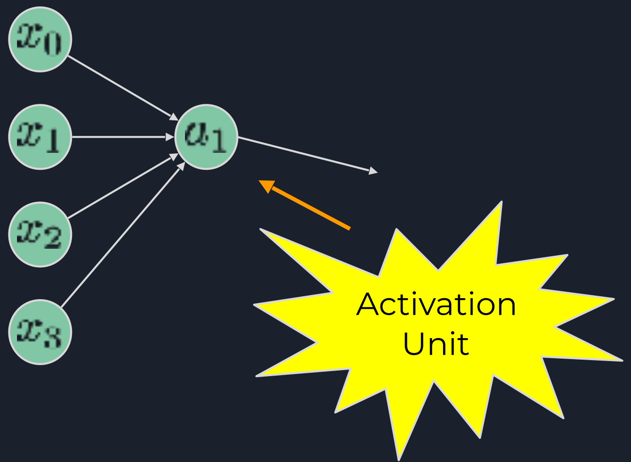
Neural Network

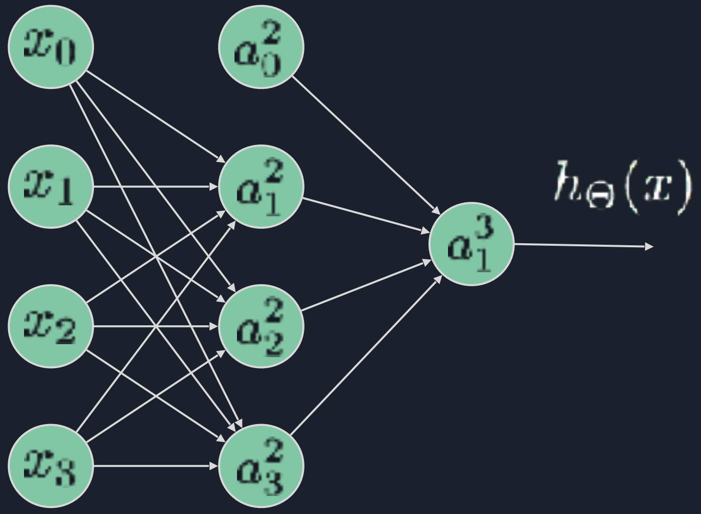
$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \rightarrow \begin{bmatrix} \quad \end{bmatrix} \rightarrow h_\Theta(x)$$



Neural Network - Network of Neurons

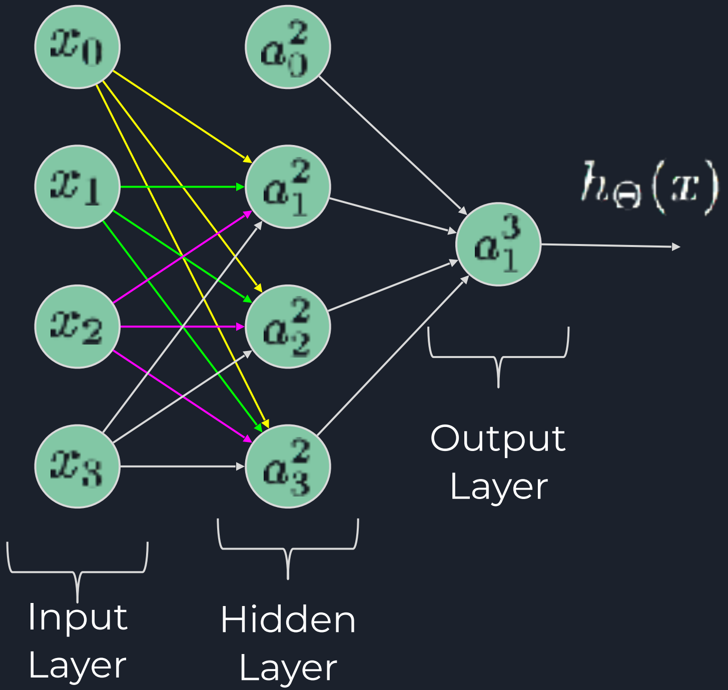






Neural Network

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \rightarrow [\quad] \rightarrow h_\Theta(x)$$



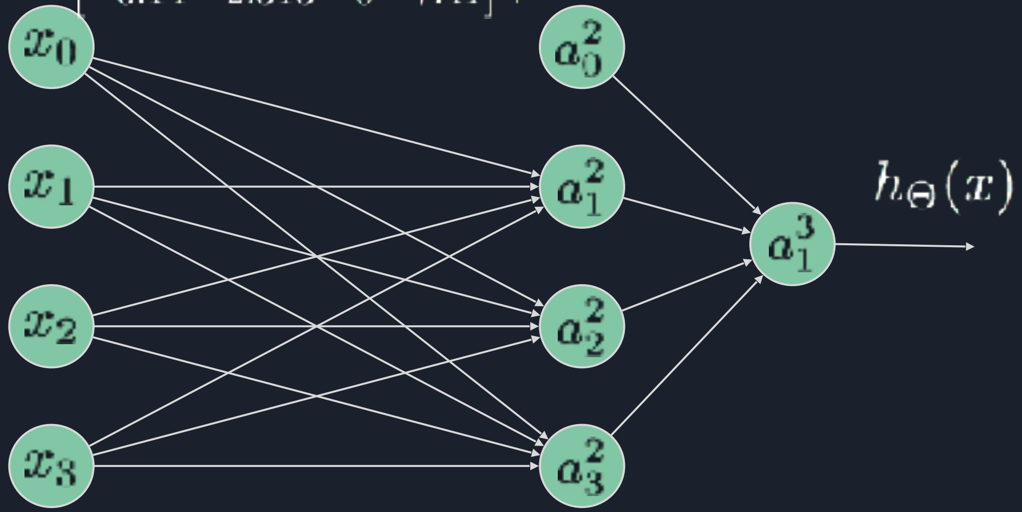
Neural Network - Parameters

$$\Theta^2 \in \mathbb{R}^{1x4}$$

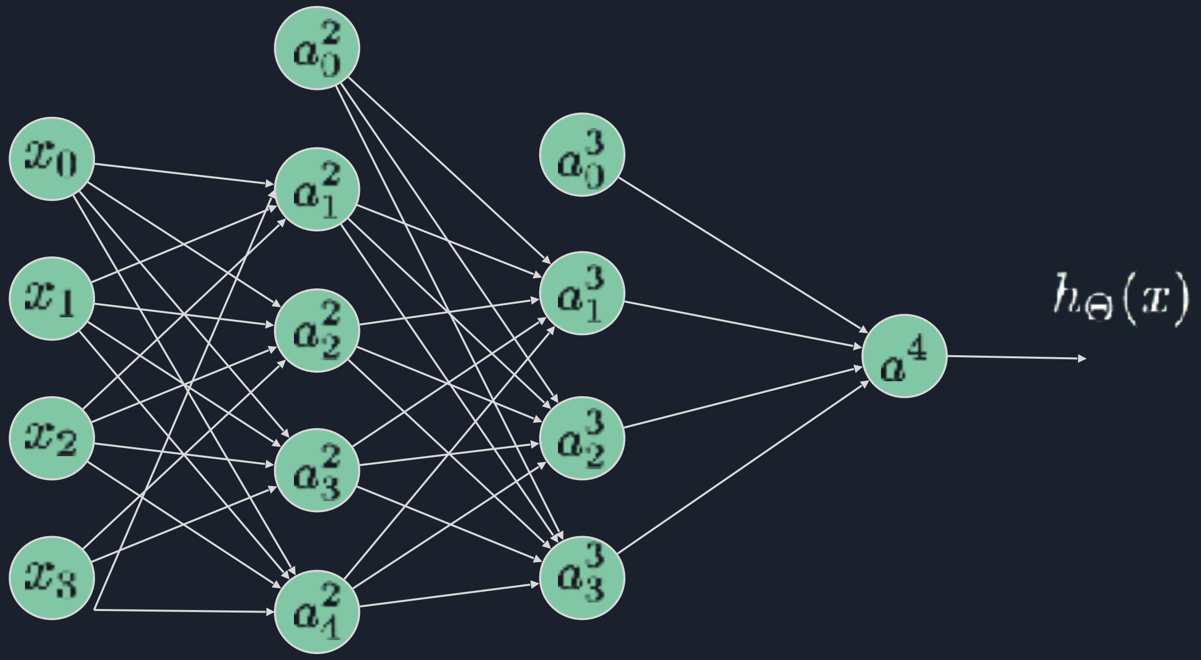
$$\begin{bmatrix} -0.14 & 2.313 & 0 & 7.41 \end{bmatrix} *$$

$$\Theta^1 \in \mathbb{R}^{3x4}$$

$$\begin{bmatrix} 1.14 & 2 & -2.73 & 4.0 \\ 5.32 & 666 & 7 & 9.1 \\ 2.3 & 0.77 & -12 & 3.14 \end{bmatrix} *$$



Deep Neural Network



Overfitting Is Good?!

The Case for Learned Index Structures

Tim Kraska*
MIT
Cambridge, MA
kraska@mit.edu

Alex Beutel
Google, Inc.
Mountain View, CA
alexbeutel@google.com

Ed H. Chi
Google, Inc.
Mountain View, CA
edchi@google.com

Jeffrey Dean
Google, Inc.
Mountain View, CA
jeff@google.com

Neoklis Polyzotis
Google, Inc.
Mountain View, CA
npolyzotis@google.com

pr 2018

- Research affiliated with Google
- Premise: Indexes resemble models
- Aim: replacing data structures with overfitting models
- Why overfitting?
    - Precision
    - No prediction is involved

* https://arxiv.org/abs/1712.01208

Learned Index Structures

- Tested against an in-memory DB
- Only covers index access, not index maintenance
- B-Tree (Range searches), Hash (Point Index), Bloom Filters (Existence)
- First attempt: Failure

  Replacing an index with one complicated Neural Network
  - Two order of magnitude slower than B-Tree index
  - Model implemented by TensorFlow => high latency
  - Index caching has no equivalent in ML solution

Recursive Regression Model

- Premise: It's easier to reduce the error in steps rather than in one go
- Doesn't have to be a tree or balanced
- Simple NN models in 1st stage
- Linear models in 2nd stage
- Faster training and inference on top levels
- Top level models can choose a variety of models in the lower level

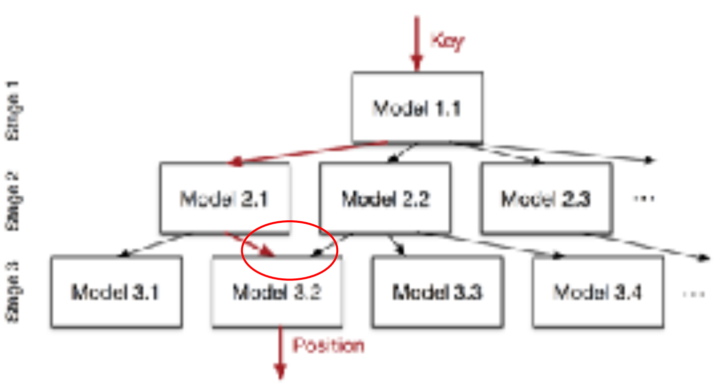


Figure 3: Staged models

Learned Index Structures

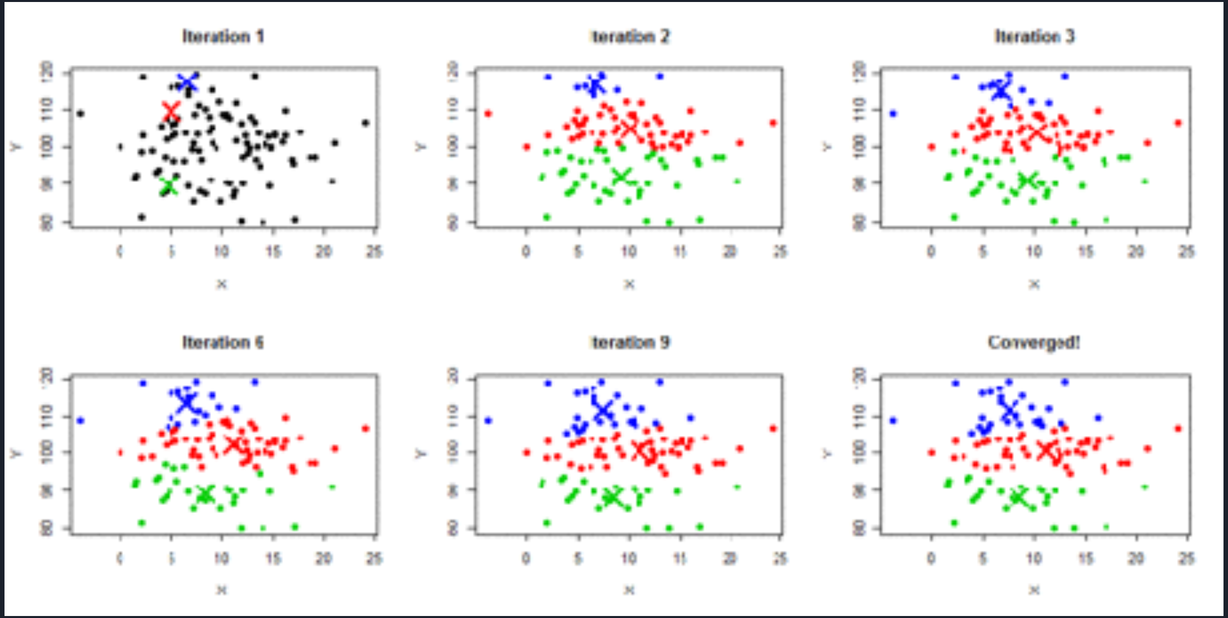
- Access speed: 3X faster
- Size: up to 10X smaller
- Currently in development for a Google KV store
- Prospects:
  - Query Optimizer
  - Storage Design
  - Sort/Join Algorithms

		Map Data		
Type	Config	Size (MB)	Lookup (ns)	Model (ns)
Btree	page size: 32	52.45 (4.00x)	274 (0.97x)	198 (72.3%)
	page size: 64	26.23 (2.00x)	277 (0.96x)	172 (62.0%)
	page size: 128	13.11 (1.00x)	265 (1.00x)	134 (50.8%)
	page size: 256	6.56 (0.50x)	267 (0.99x)	114 (42.7%)
	page size: 512	3.28 (0.25x)	286 (0.93x)	101 (35.3%)
Learned Index	2nd stage models: 10k	0.15 (0.01x)	98 (2.70x)	31 (31.6%)
	2nd stage models: 50k	0.76 (0.06x)	85 (3.11x)	39 (45.9%)
	2nd stage models: 100k	1.53 (0.12x)	82 (3.21x)	41 (50.2%)
	2nd stage models: 200k	3.05 (0.23x)	86 (3.08x)	50 (58.1%)

Machine Learning Categories

- Supervised Learning
  - Similar data exists
  - We know what that data represents
  - Find patterns in the existing data to help with predictions
- Unsupervised Learning
  - "Computer! Find out how 'these things' are related!"
- Reinforcement Learning
  - "Computer! Analyze data, make decision and take action!"

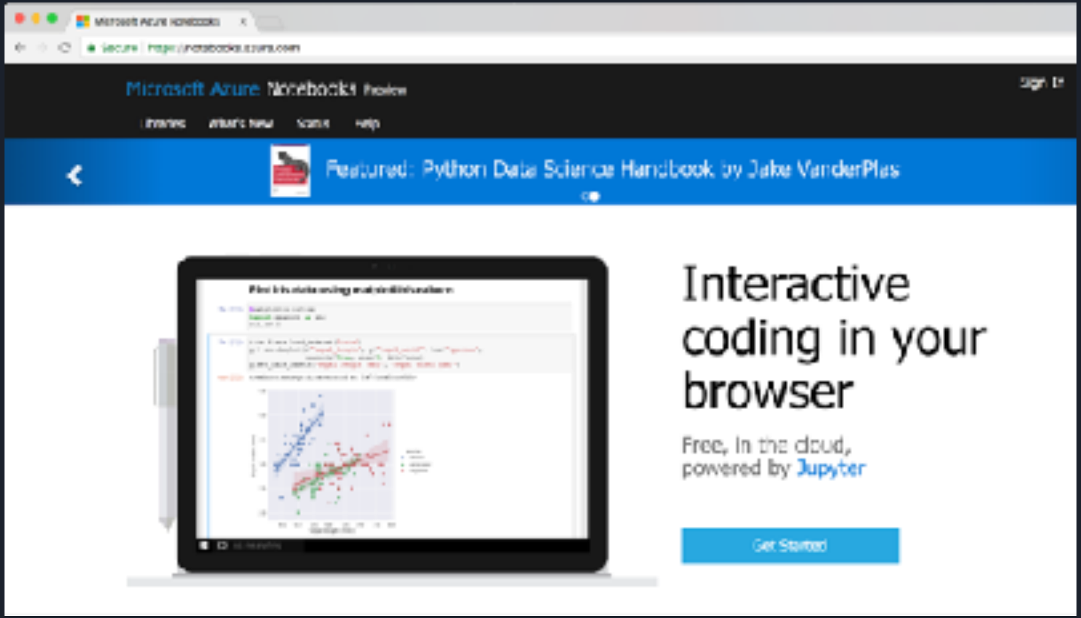
Clustering - K-means

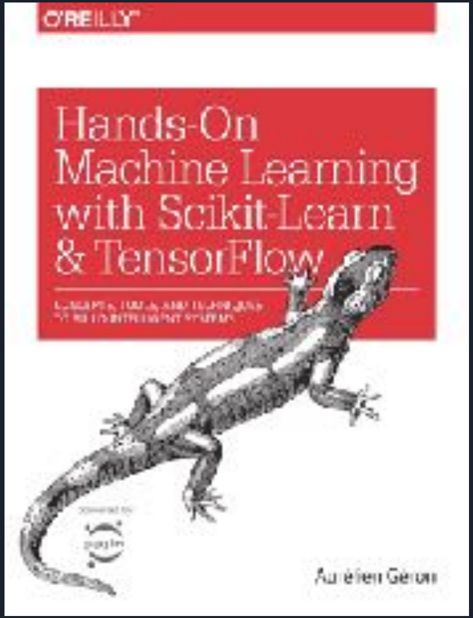


ML Algorithms

Algo	Regression	Classification	Handles Complexity	Training Speed	Precision	Amount of Data	Can be Explained?
Linear Regression	✓			high	low	low	yes
Logistic Regression		✓		fast	low	low	yes
Support Vector Machines	✓	✓	✓			low	no
Decision Trees	✓	✓	✓	moderate	high	high	not if Random Forest
Neural Networks		✓	✓	low	Very high	Very high	no
Naive Bayes		✓		high		low	maybe

Resources for Learning





Tools - Python Libraries

- Matrices and N-dimensional arrays
- Linear Algebra
- Matrix operations

- DataFrames
- Slice, merge, reshape,
  Join, pivot, aggregate
- I/O operations
  Files and databases

Tools - Python Libraries

- Algorithms: Classification, Regression, Clustering
- Preprocessing: Normalisation, Standardisation, ...
- Hyperparameter tuning, Model evaluation, ...

- Computation Framework
- Data flow graph processor
- Developed by Google
- Can run on nVidia GPUs and TPUs
- Convolutional/Recurrent Neural Network
- Classification, Regression

Resources for Learning

- Machine Learning by Andrew Ng (Stanford University - Coursera)
- OCD Level Machine Learning Guide podcast series
  http://ocdevel.com/podcasts/machine-learning
- Machine Learning Crash Course (Google)
  https://developers.google.com/machine-learning/crash-course/
- Kaggle
  https://www.kaggle.com/learn/overview
- Understanding Machine Learning with Python (Pluralsight)

Resources for Learning

- How to choose Algos for Azure ML https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-choice
- Choosing the right estimator http://scikit-learn.org/stable/tutorial/machine_learning_map/
- Which Algo for which problem https://recast.ai/blog/machine-learning-algorithms/2/

Neural Network - Clear Examples

- http://iamtrask.github.io/2015/07/12/basic-python-network/

- https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

Thank You!

@2ndhalf_oracle
https://www.linkedin.com/in/babak-tourani/