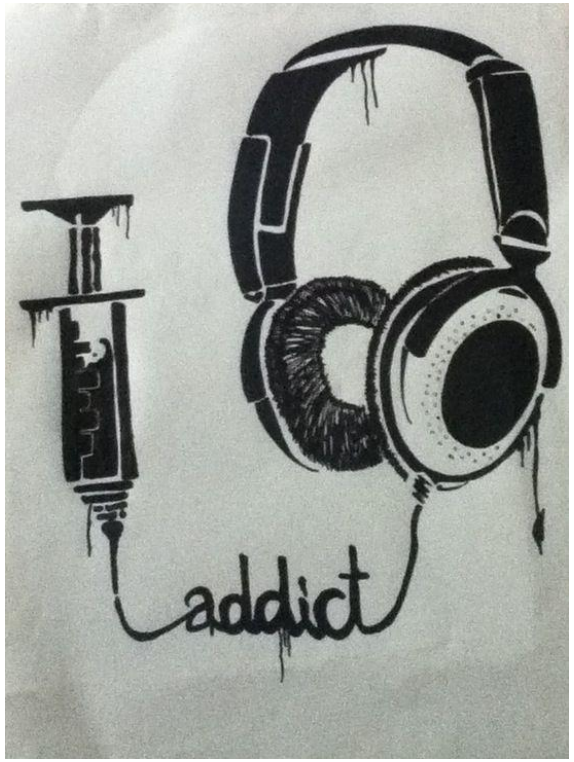
Machine learning in action at Expedia

Anupama Bhati - Developer and AI enthusiast

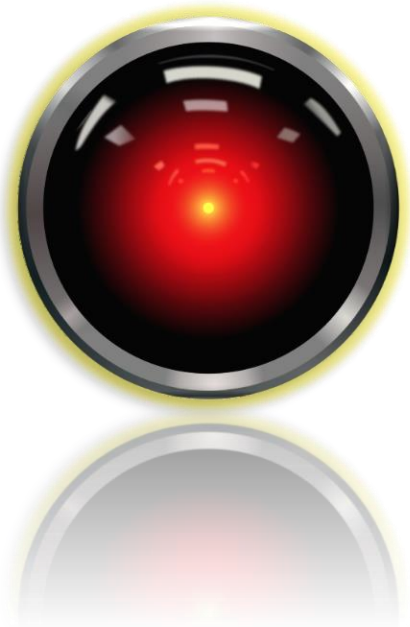
Expedia

What I want to do?

What I actually do



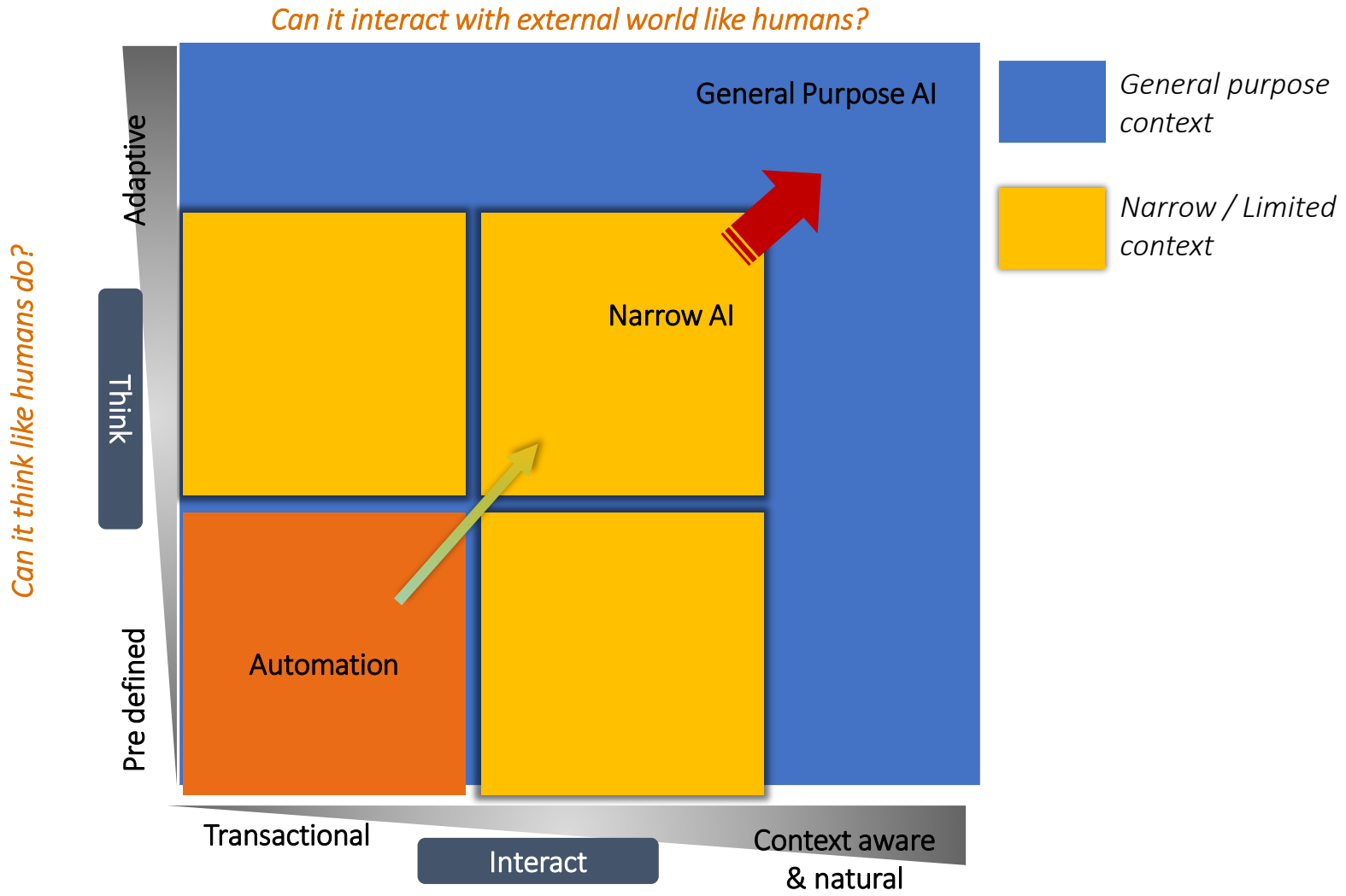
Why did HAL 9000 refuse to follow the order ?



“I am sorry Dave, I’m afraid I can’t do that”

- HAL 9000,
2001 - A Space Odyssey
(Directed by Stanley Kubrick)

For any machine to be called AI, it should either think or interact like humans do?

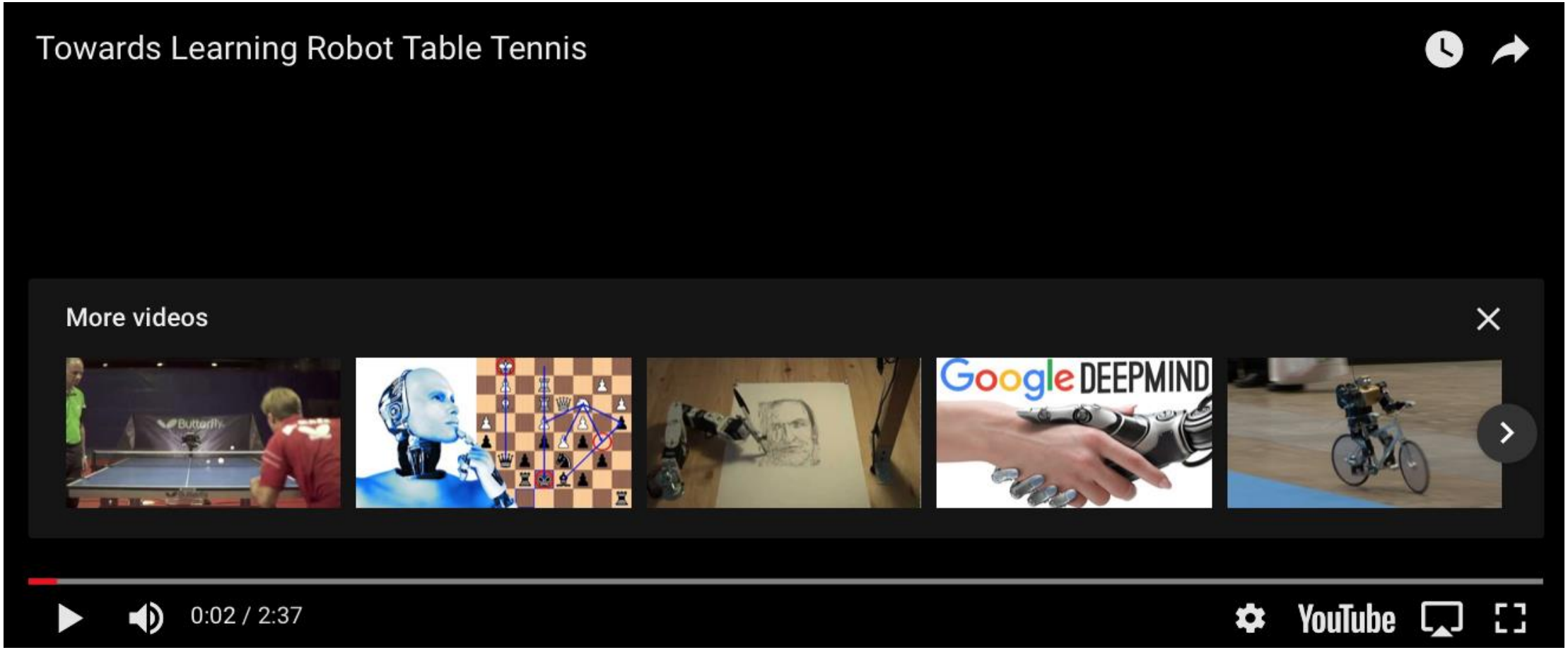


What is machine learning?

Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. **Machine learning** focuses on the development of computer programs that can change when exposed to new data. The process of **machine learning** is similar to that of data mining.

> “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P,if its performance at tasks in T, as measured by P, improves with experience ”
> - T.Mitchell(1997)

Towards learning robot table tennis



Types of machine learning

Supervised Learning

- Makes machine learn explicitly
- Data with clearly defined output is given
- Direct feedback is given
- Predicts outcome/future
- Classification/Regression

Unsupervised Learning

- Machine understands the data(Identifies patterns/structures)
- Evaluation is qualitative or indirect
- Does not predict or find anything specific

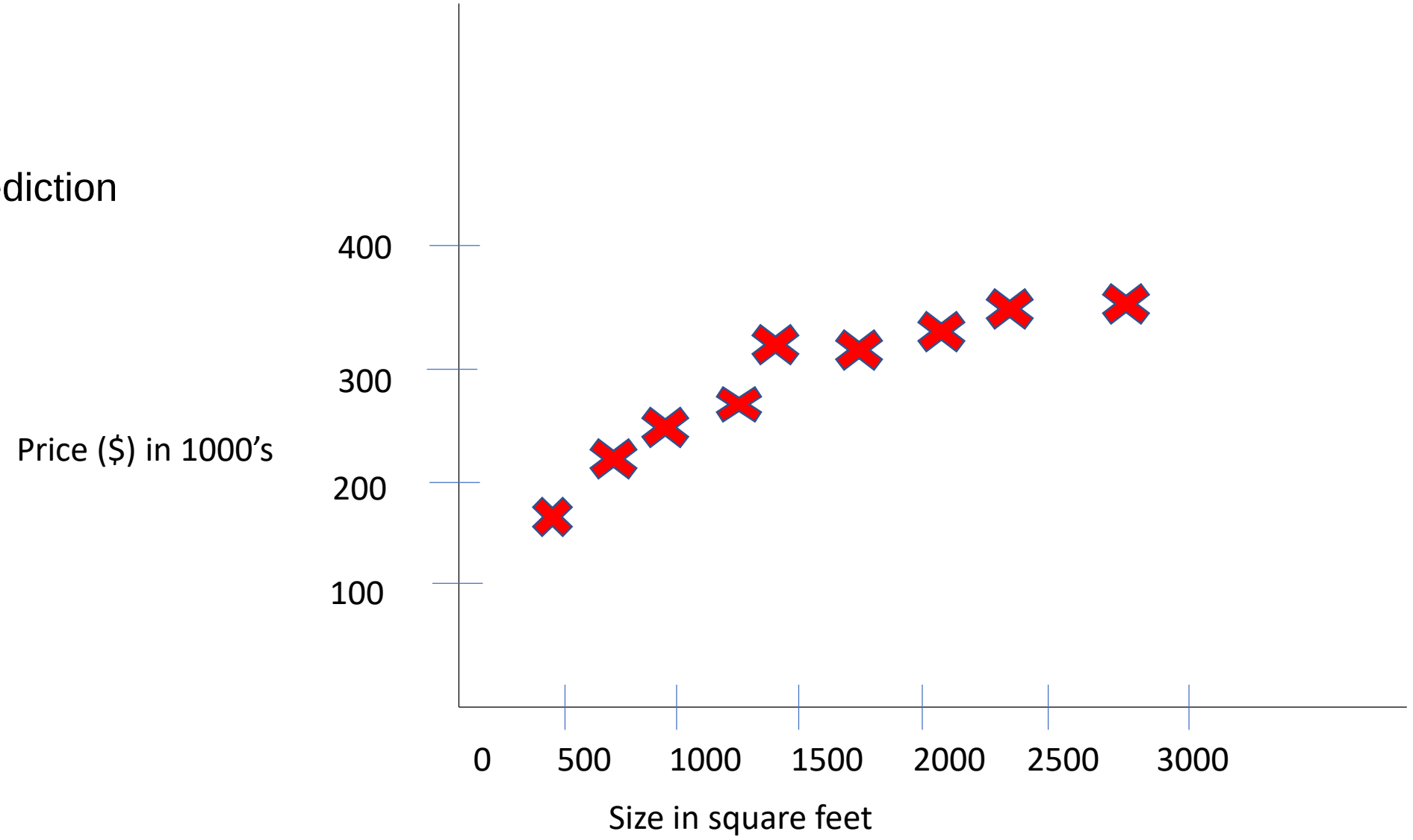
Reinforcement Learning

- Reward based learning
- Learning from +ve and –ve reinforcement
- Machine learns how to act in a certain environment
- To maximize rewards

Supervised Learning

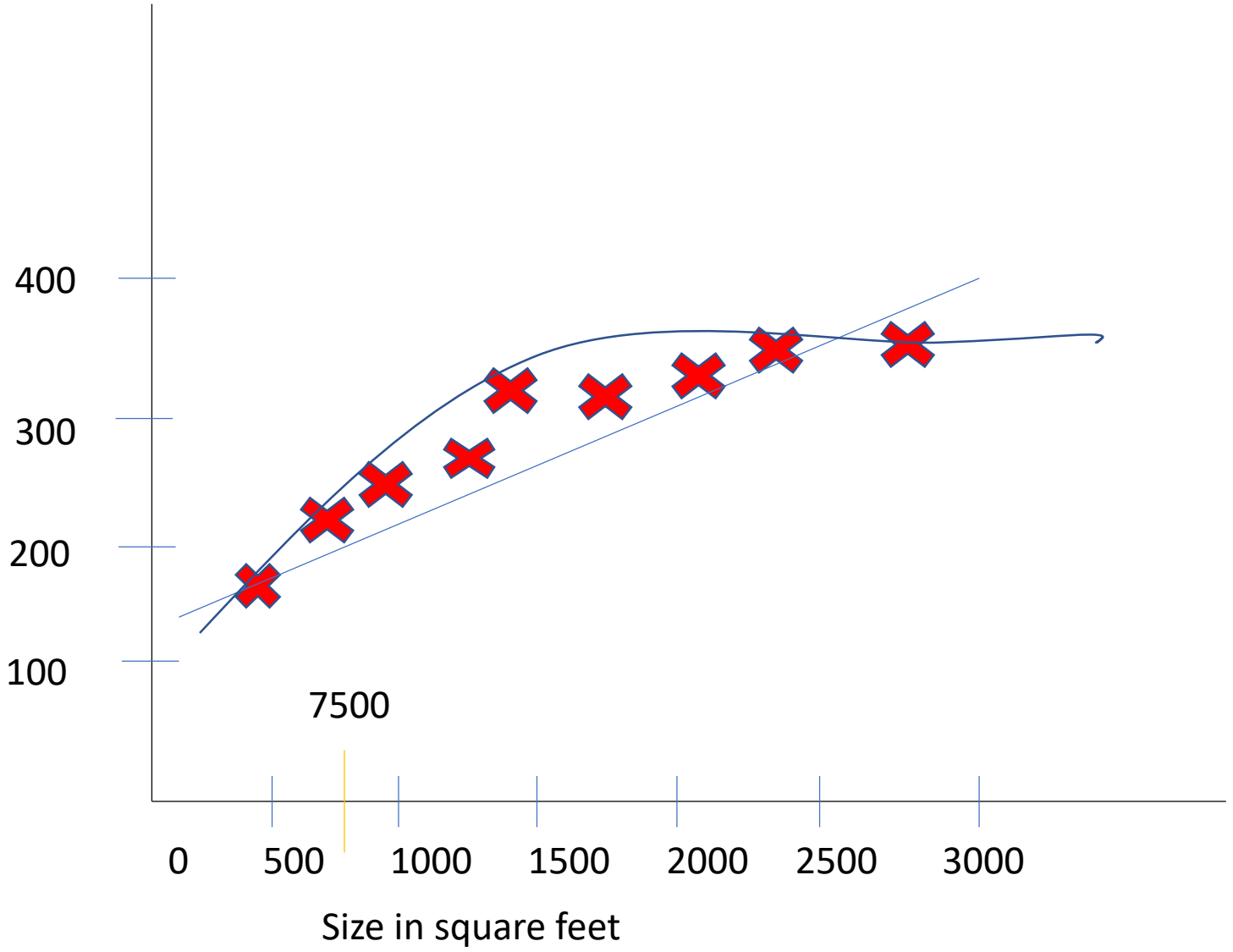
Regression

Housing price prediction



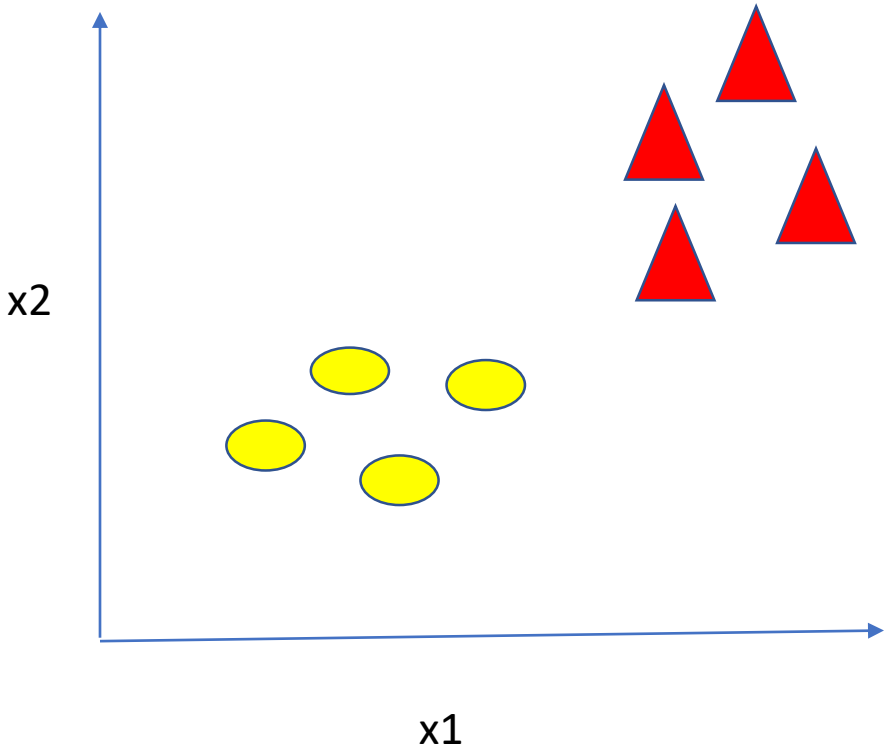
Regression

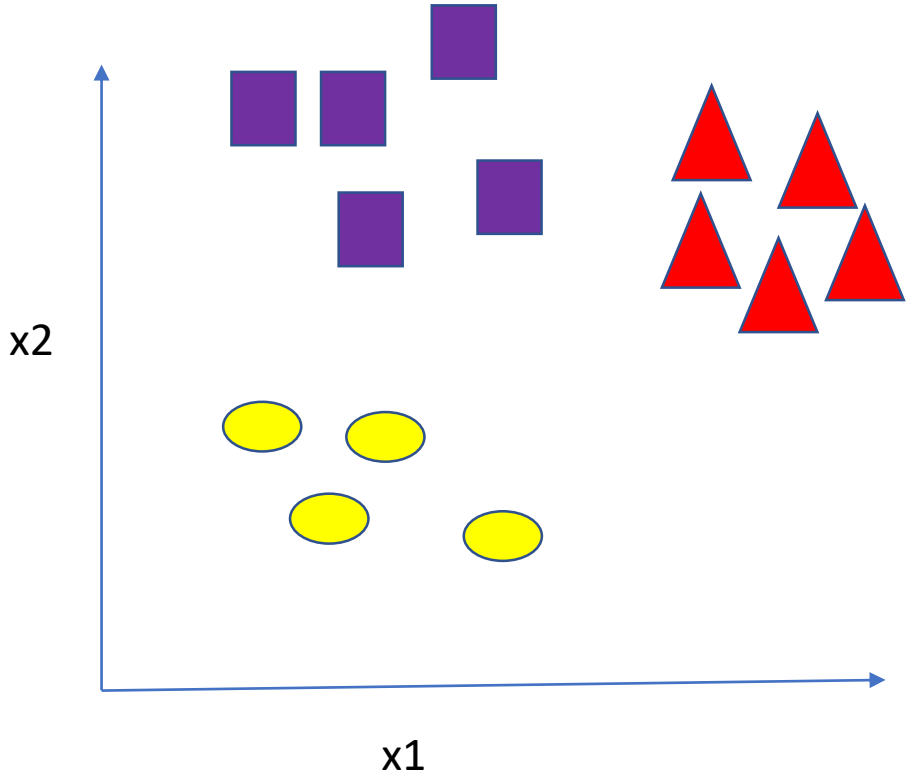
Housing price prediction



- Rights answers given
- Regression - Predict continuous values output(price)

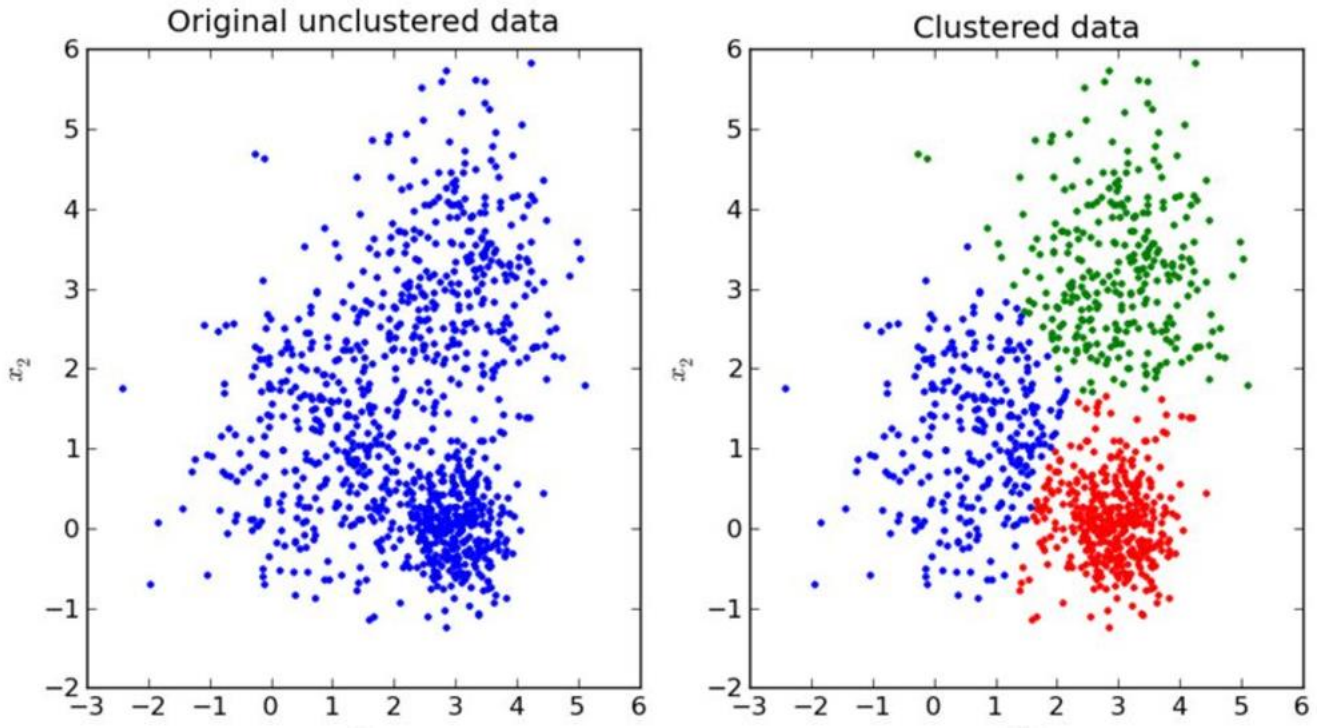
Classification Problem



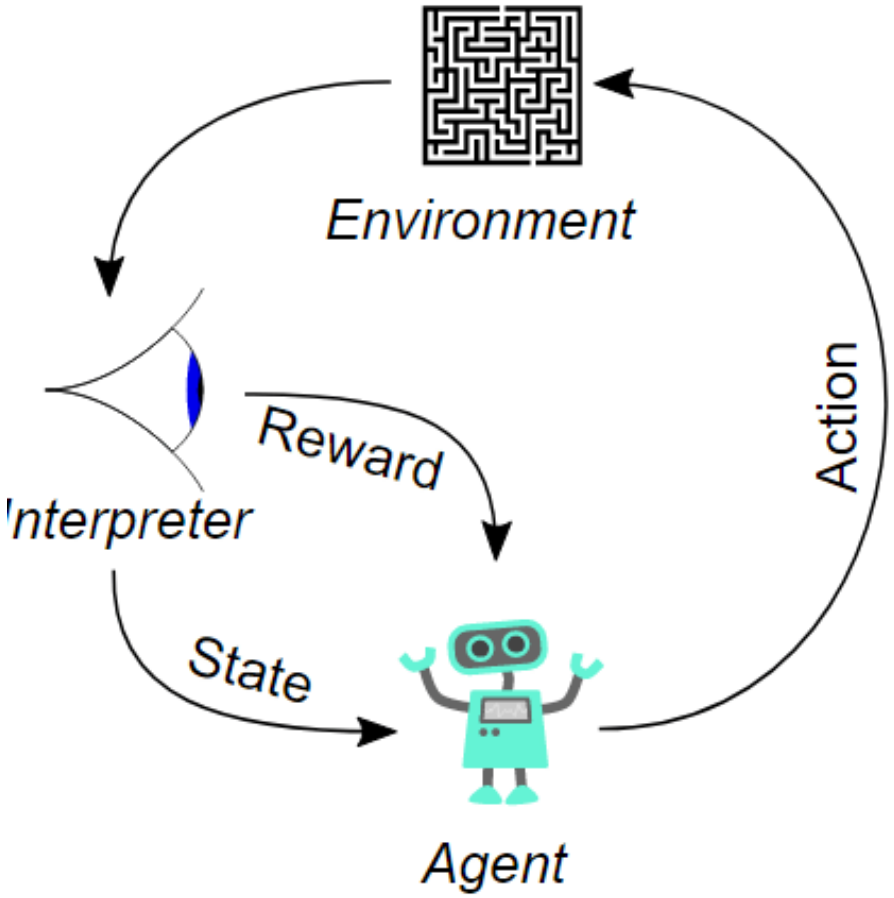


Unsupervised Learning

Given the data give me some description of data – Structure the data



Reinforcement Learning



Exercise

1. Suppose you have to predict whether sun will rise today, based on the trend that you know of for the past 2 months based on certain parameters. What kind of problem it is?
   Classification/Regression?

2. Suppose you have patient history based on age, race and other parameters it is known whether the a tumor was cancerous or benign. Given a patient with a tumor predict whether he/she has cancer or not

3. Suppose you are working on stock market prediction, and you would like to predict the price of a particular stock tomorrow (measured in dollars). You want to use a learning algorithm for this. Would you treat this as a classification or a regression problem?

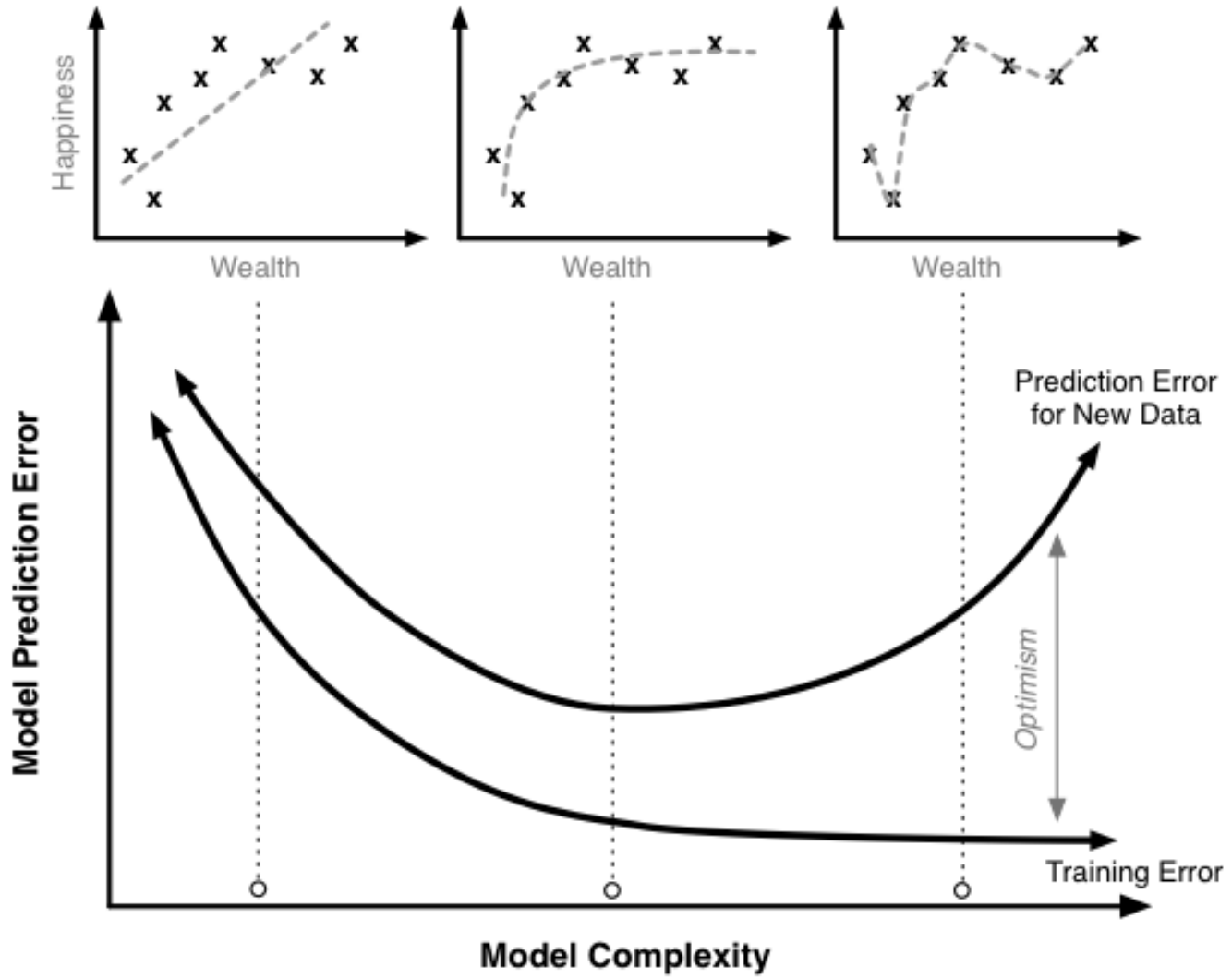
4. Suppose if robot had to be taught to learn to walk on a rough terrain, what kind of learning will you use.

Generalization

The whole point of machine learning is generalization.

Generalization usually refers to a ML model's ability to perform well on new unseen data rather than just the data that it was trained on. It is strongly related to the concept of overfitting. If your model is over fitted then it will not generalize well.

Overfitting/ Under fitting



Inductive Bias

- The theoretical assumption that must be added to transform the algorithm's outputs into logical deductions.

Examples of Applications in Expedia

- Auto-moderation of reviews
- Entity extraction /sentiment analysis - reviews
- Image classification
- Image quality analysis
- Open search
- Expedia Search and Suggest
- Hotel search result optimization
- Package price prediction
- Hotel shell deduplication
- City/Hotel clustering
- Creating ad-groups – SEM
- Prize prediction/User segmentation insurance

Applications in User Generated Content

UGC Team @Expedia Inc.

Collection

- All Line of Business's
- All brands of Expedia Inc.
- Not just reviews

Curation

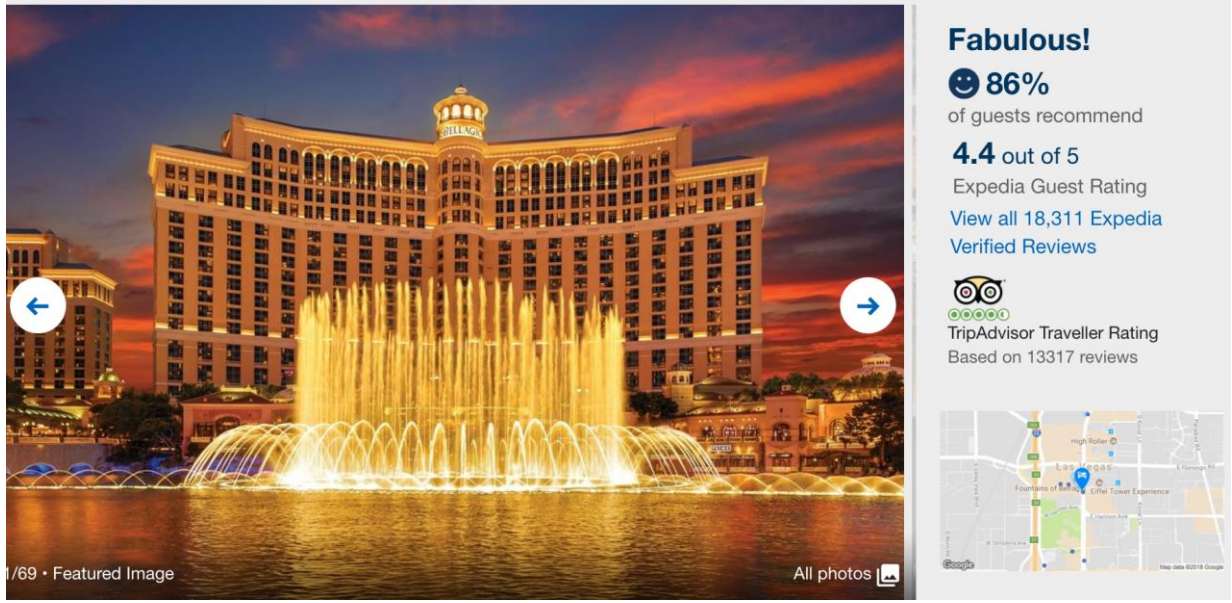
- Draw insights from User Generated Content

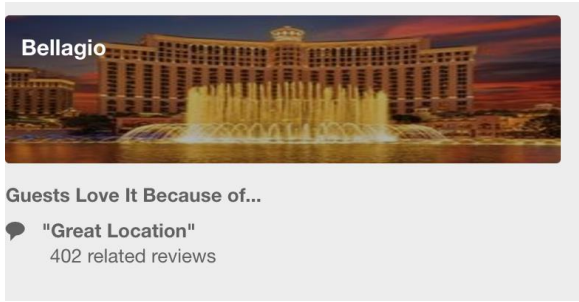
Playback

- Help people go places
- Drive conversion via social proofing

Sentiment Analysis

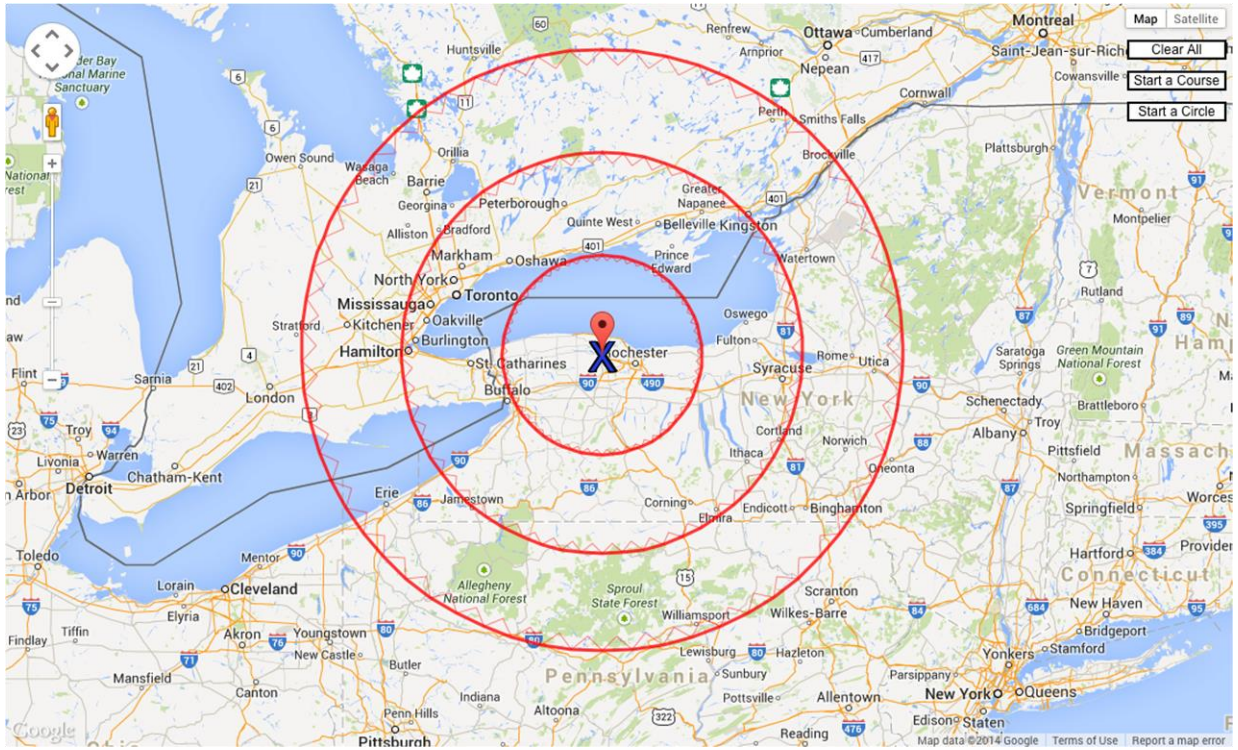
Why sentiment analysis?





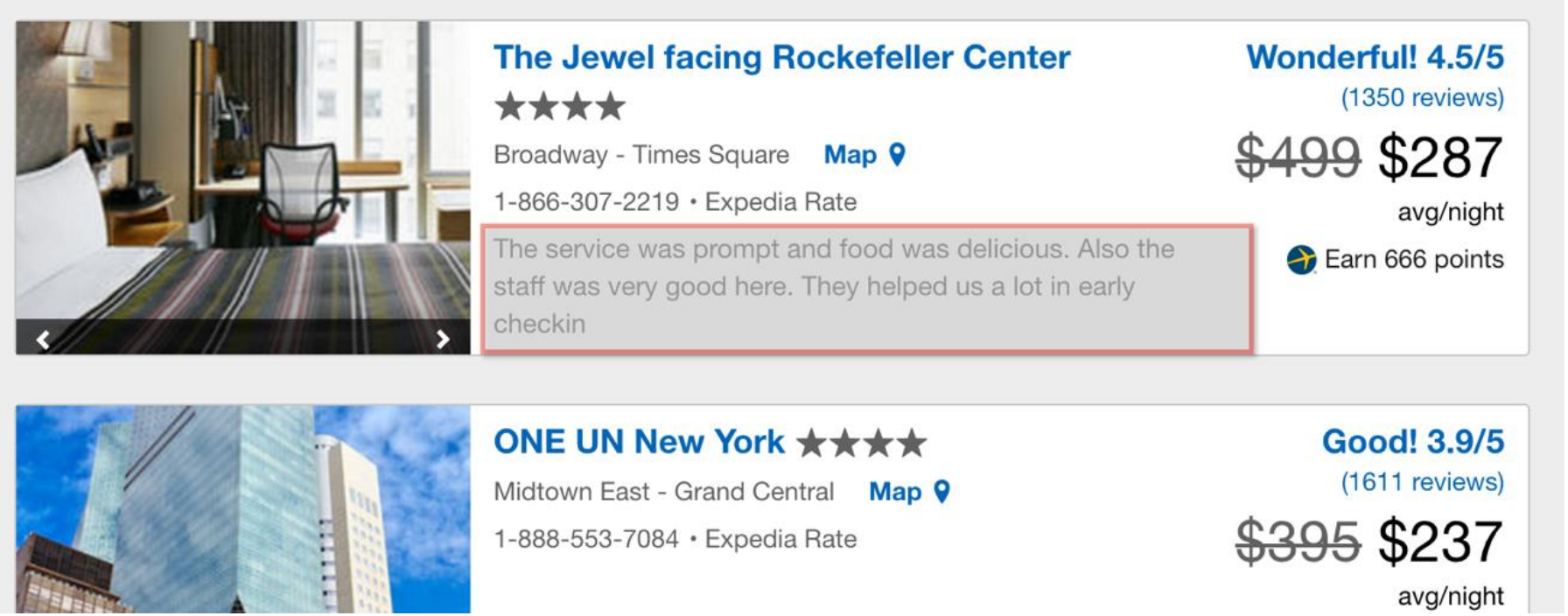
Top beach hotels in Phuket

Top rated hotels within 10km of my radius



Sentiment of voucher redemption for an activity

Snippets on Hotel Search Results page

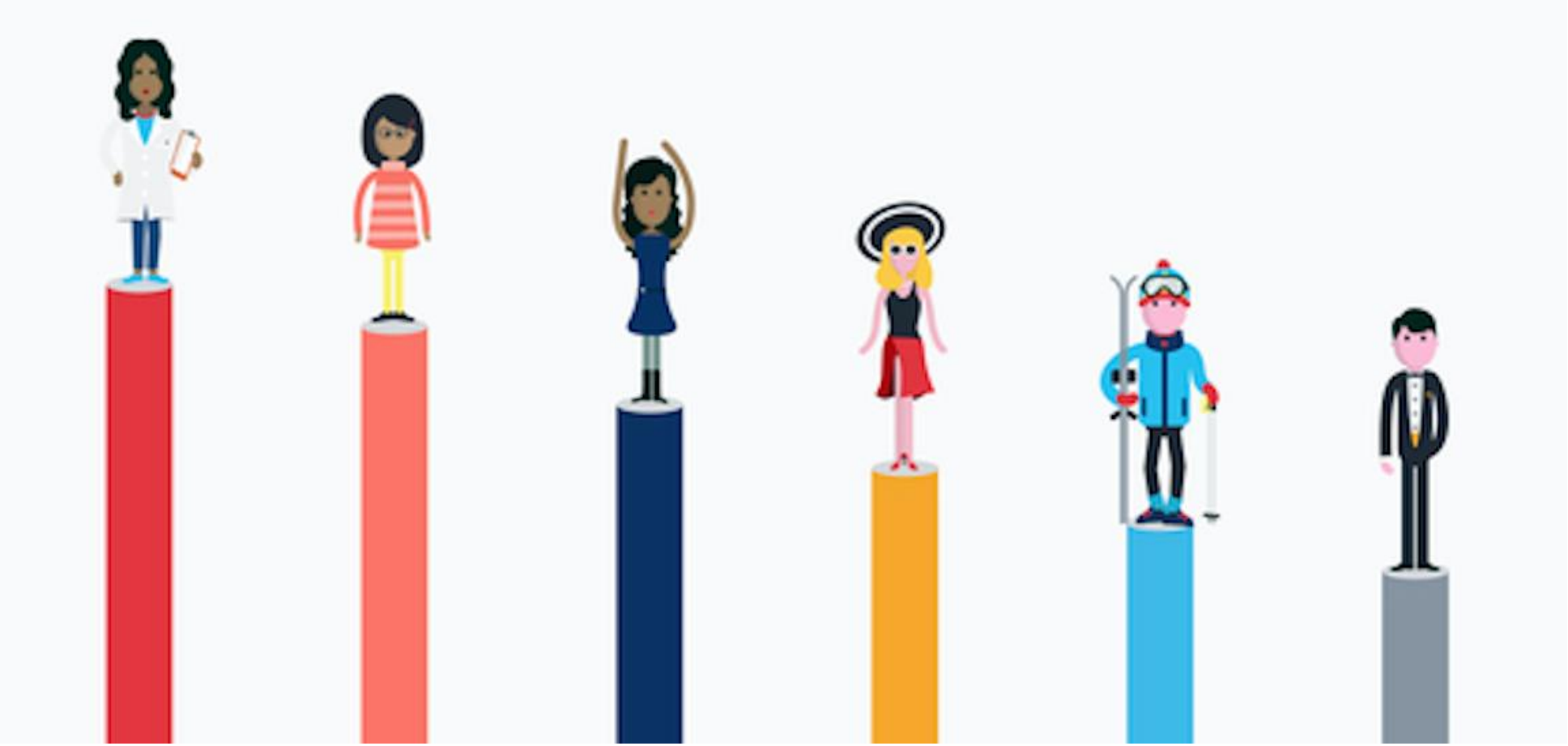


Snippets Trends

Monthly trends of tags for a particular hotel

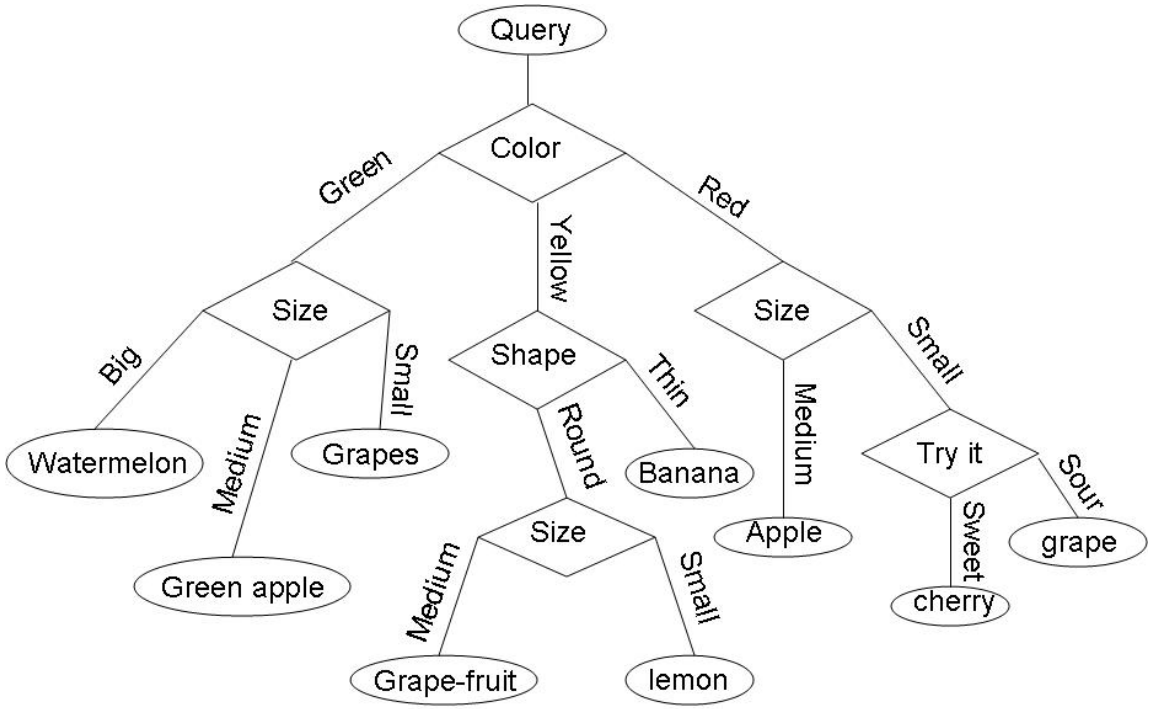
Example: Breakfast was great in January but was horrible in February

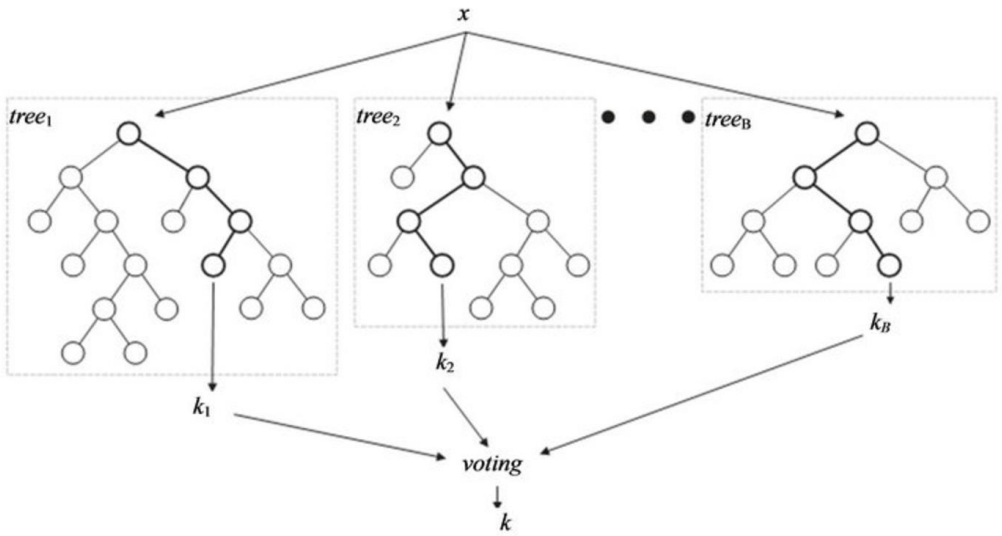
Stack rank of hotels in a region



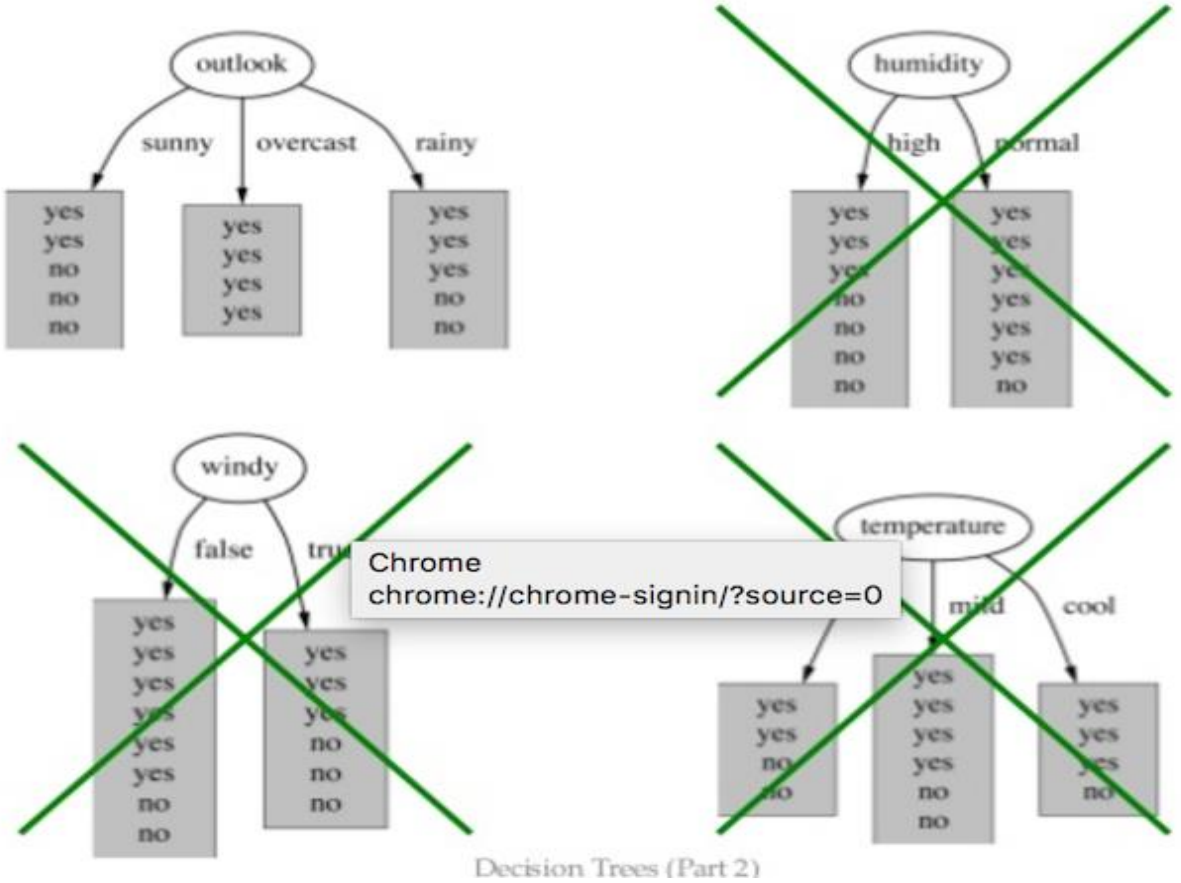
Different types of machine learning models

Decision Tree





Play golf


outlook
sunny
overcast
rainy
yes yes no no no
yes yes yes yes
yes yes yes no no
humidity
high
normal
yes yes yes no no no no
yes yes yes yes yes no
windy
false
yes yes yes yes yes no no
yes yes yes no no no
temperature
mild
cool
yes yes no no
yes yes yes yes no no
yes yes yes no
Chrome
chrome://chrome-signin/?source=0

Linear Regression

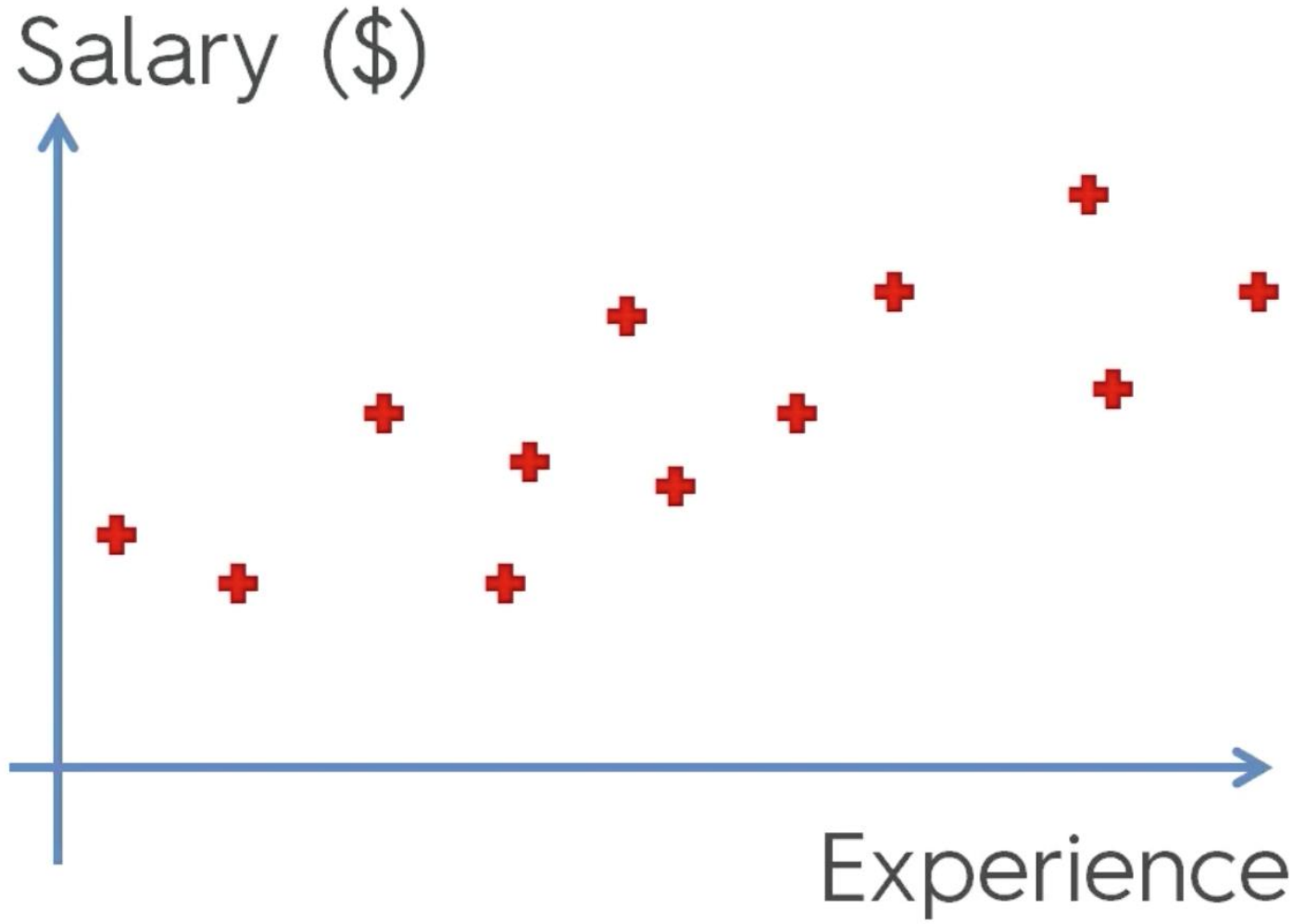
- **Simple:**

$$y = b_0 + b_1 * x$$

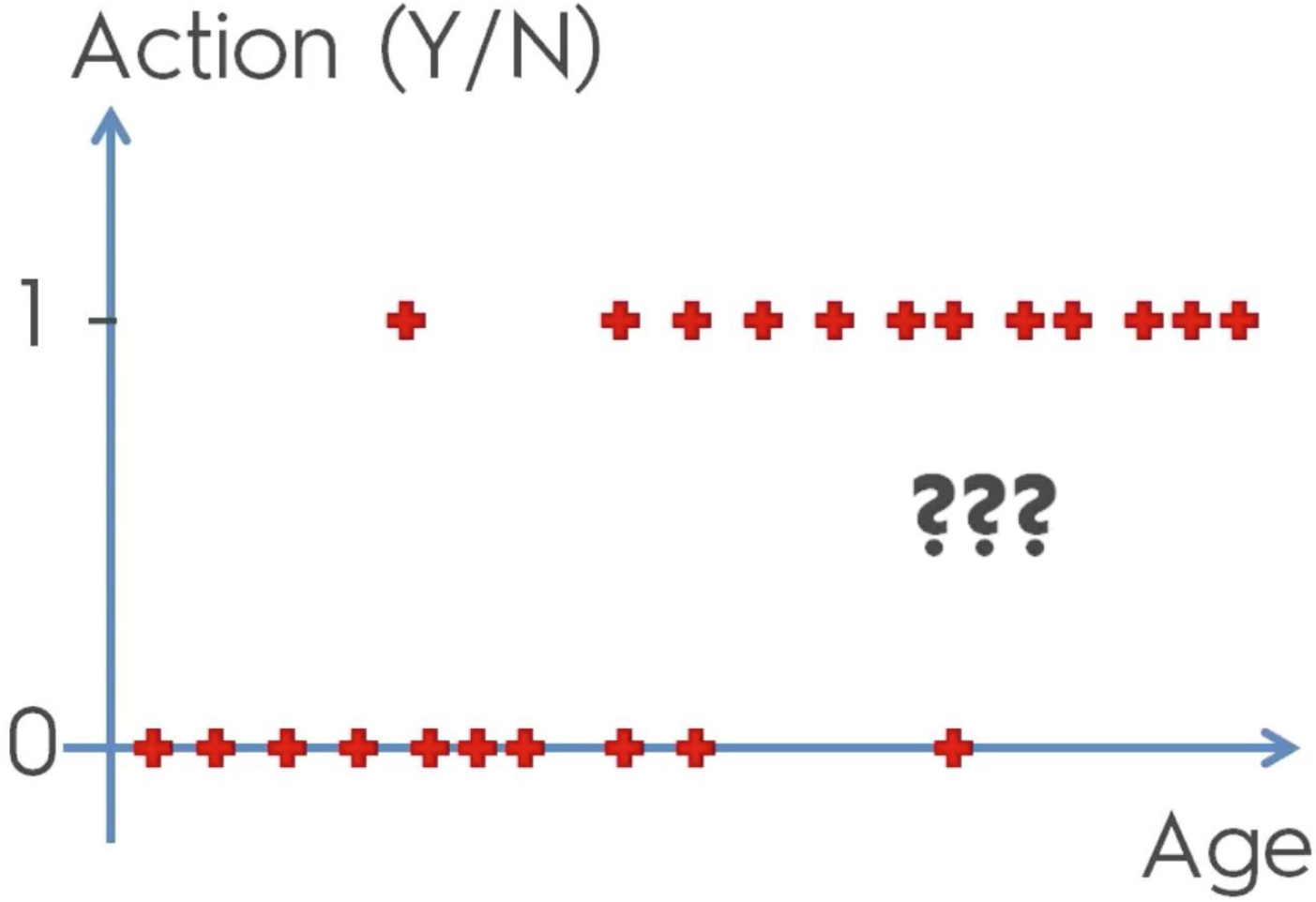
- **Multiple:**

$$y = b_0 + b_1 * x_1 + \ldots + b_n * x_n$$

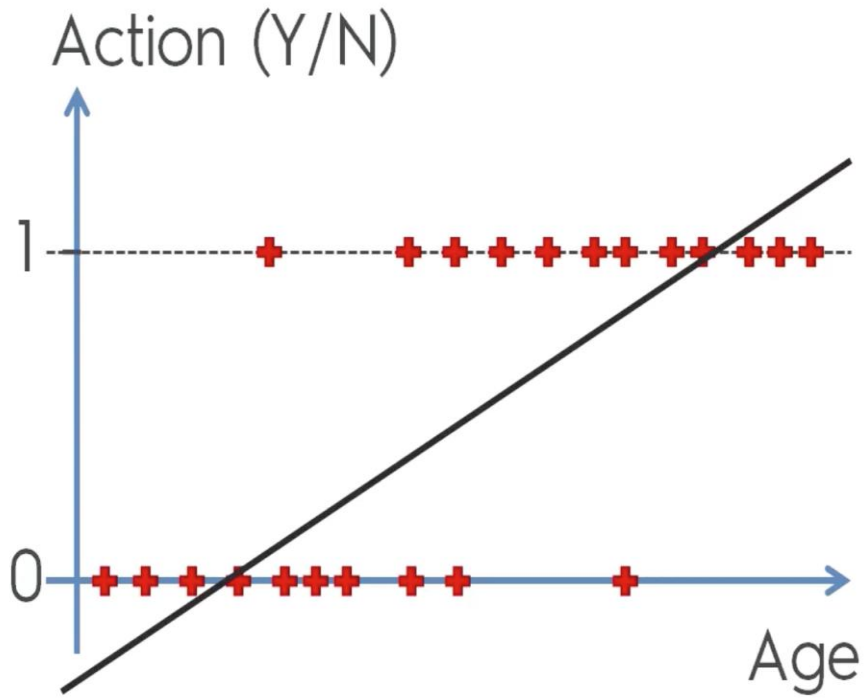
We know this ..

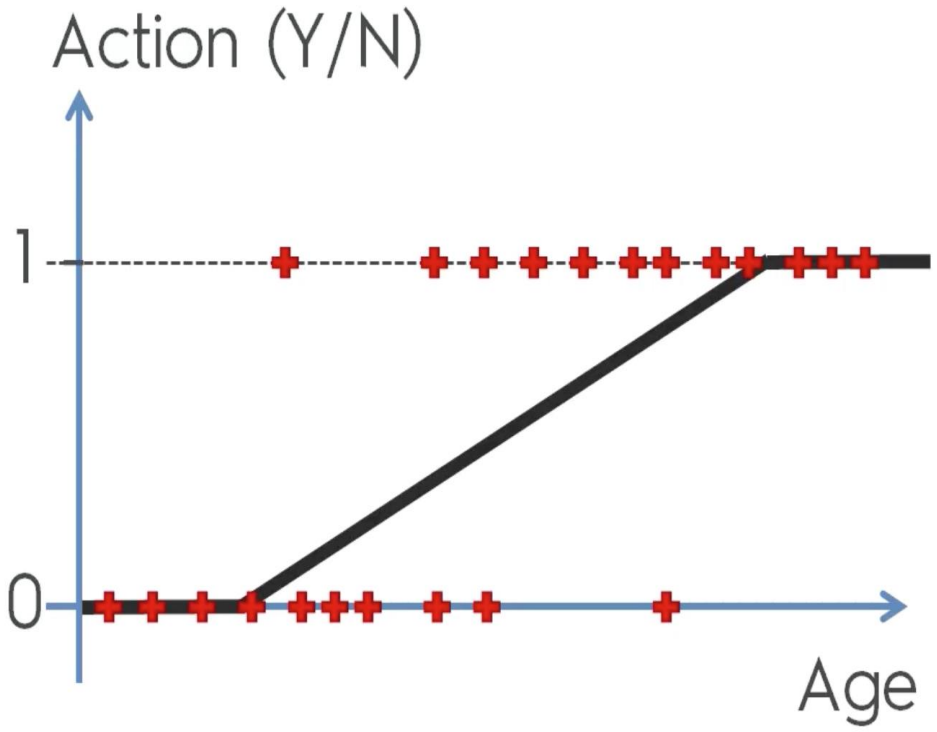


This is new ..

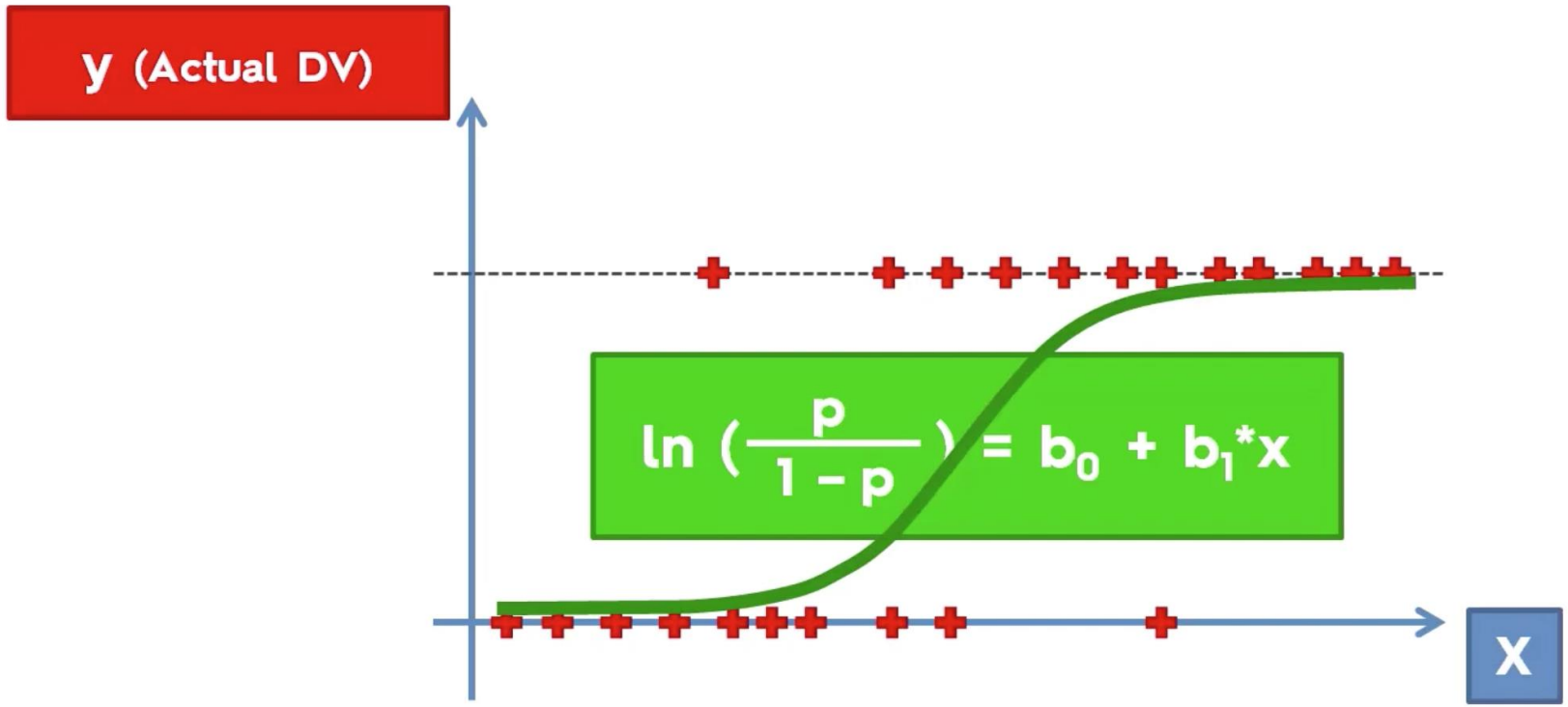


Logistic Regression

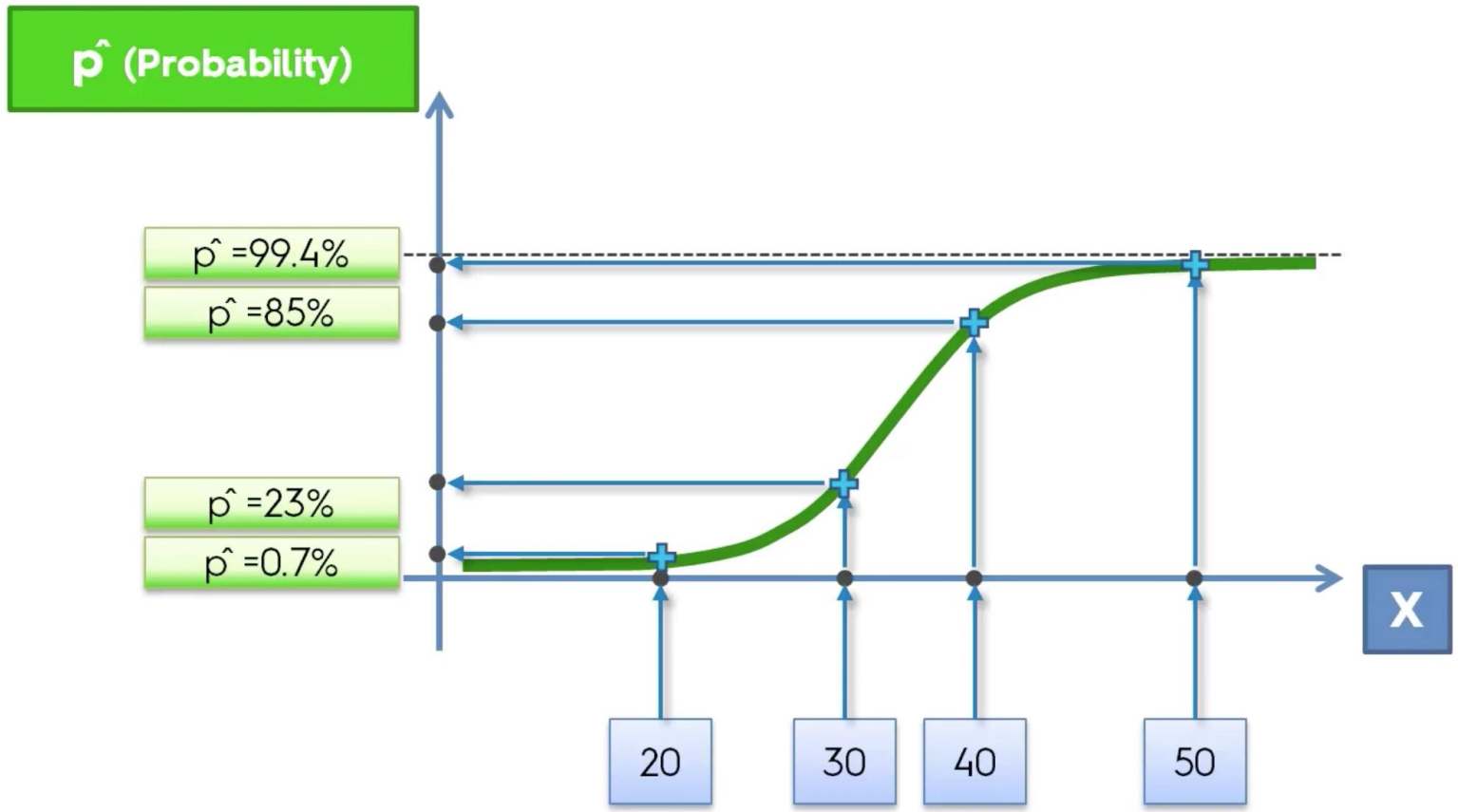




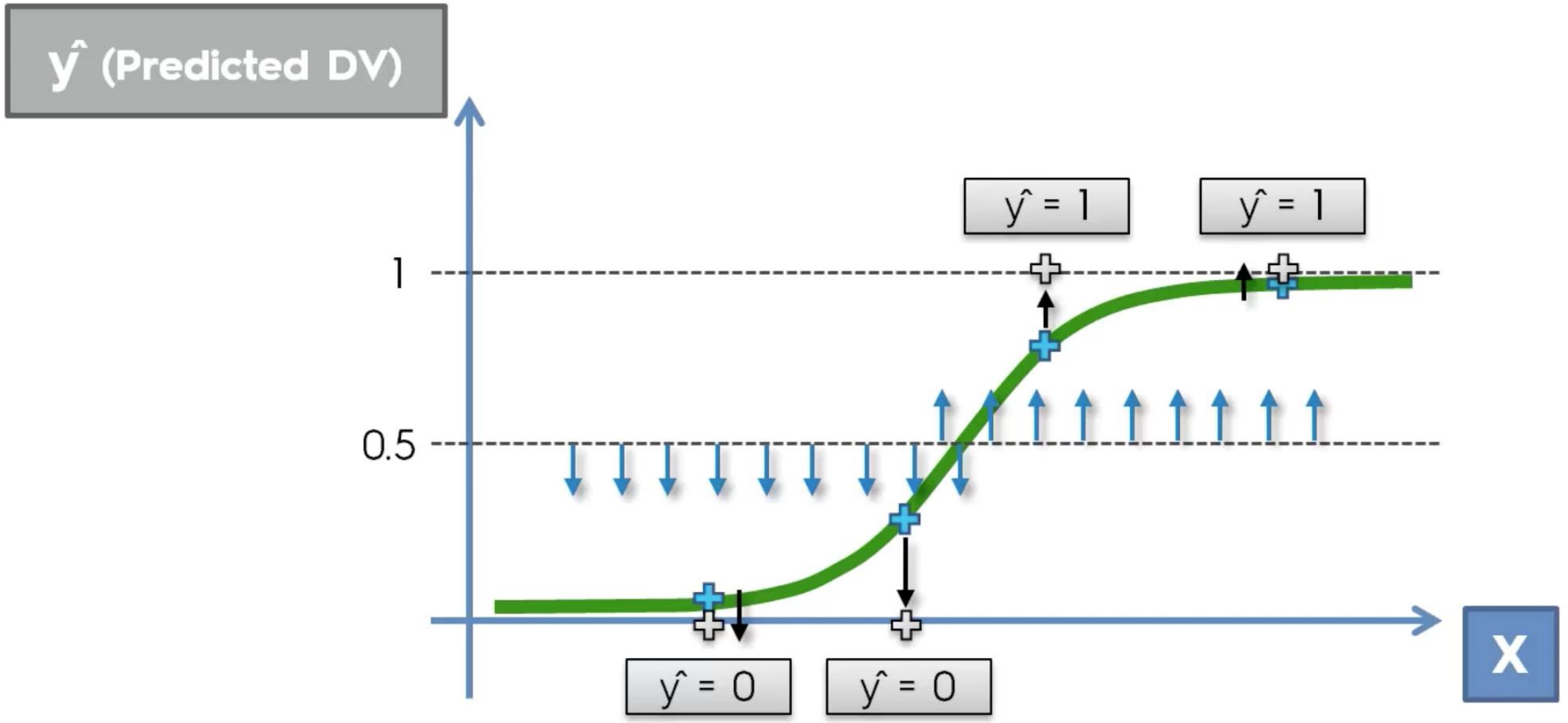
Logistic Regression



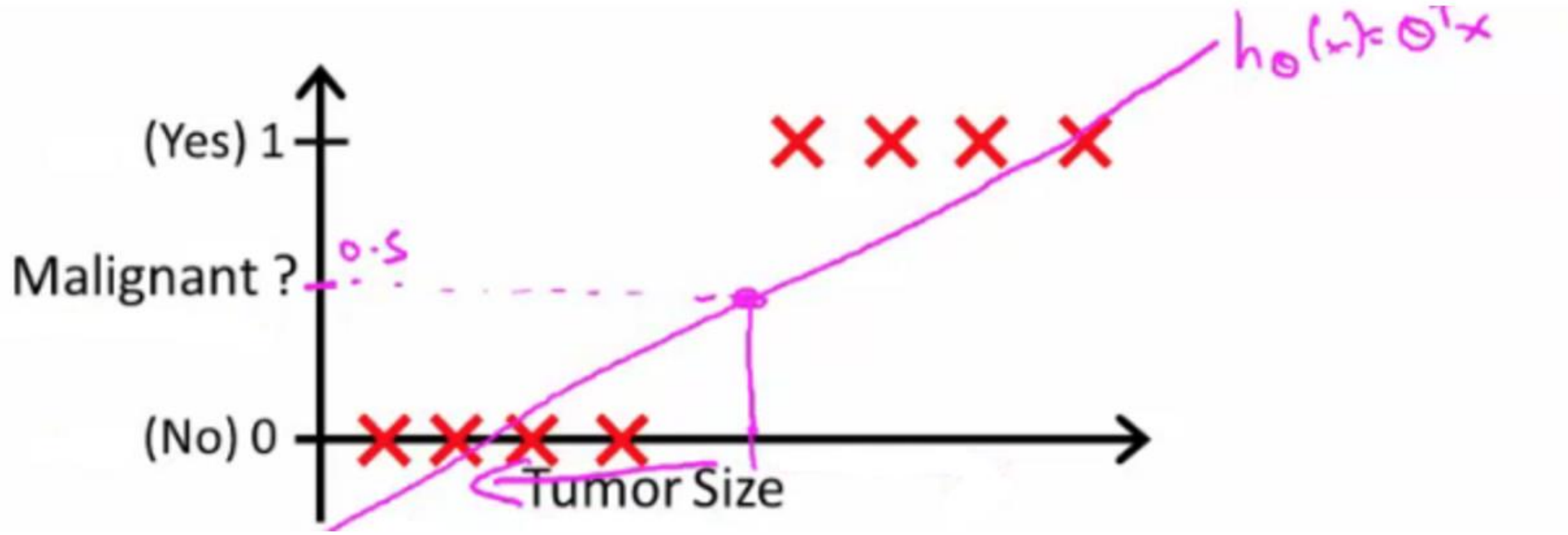
Logistic Regression



Logistic Regression

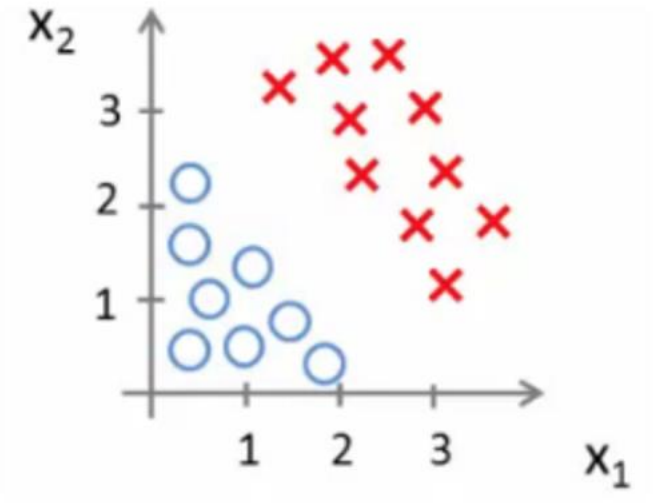


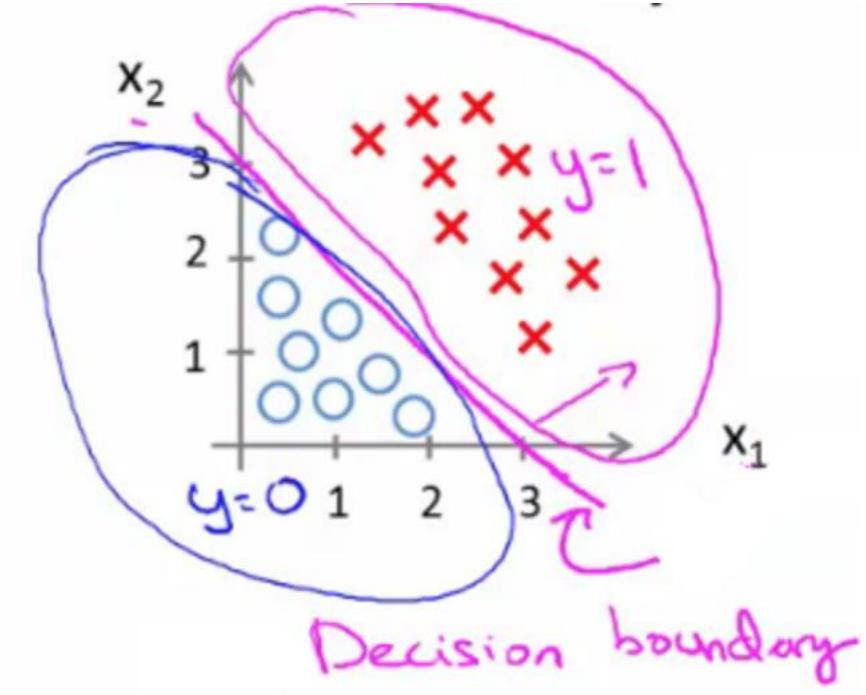
Decision Boundary



Decision Boundary contd..

$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$





Non linear decision boundary

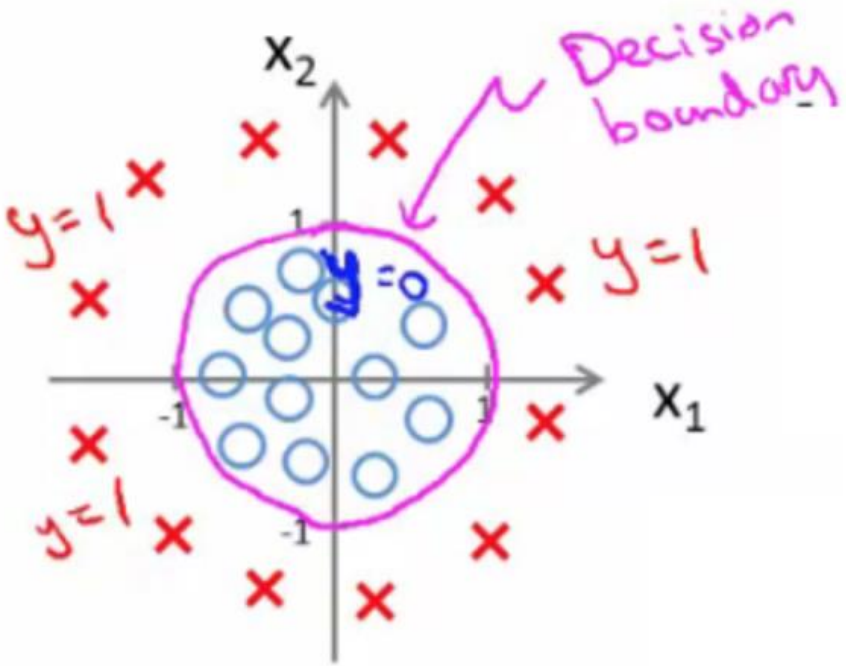
$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

Say θ^T was [-1,0,0,1,1] then we say;
Predict that "y = 1" *if*

$-1 + x_1^2 + x_2^2 >= 0$

or

$x_1^2 + x_2^2 >= 1$



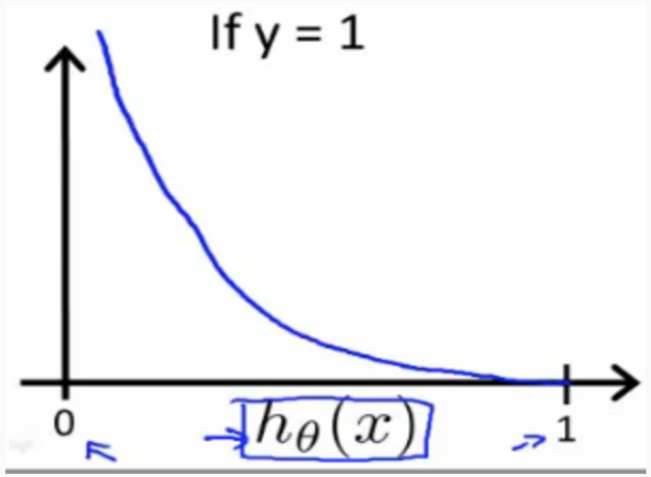
Logistic Regression – Cost Function

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

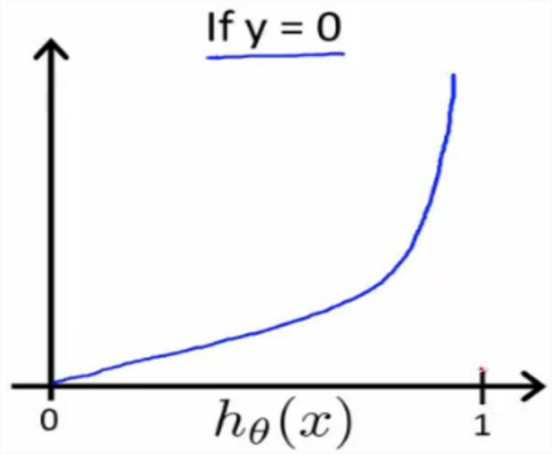
$$\text{Cost}(h_\theta(x), y) = -\log(h_\theta(x)) \qquad \text{if y} = 1$$

$$\text{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x)) \qquad \text{if y} = 0$$

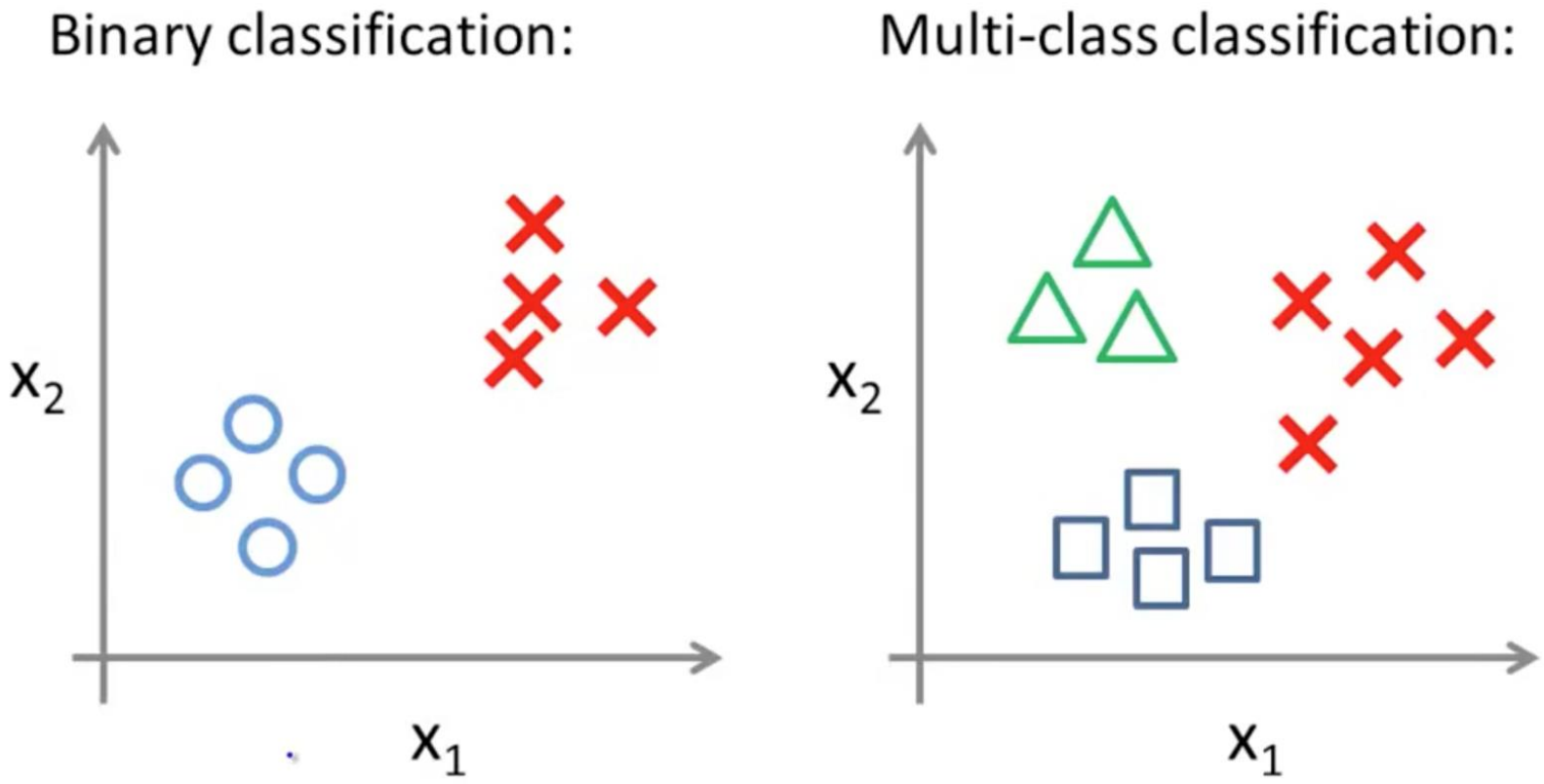
When y = 1, we get the following plot for $J(\theta)$ vs $h_\theta(x)$:



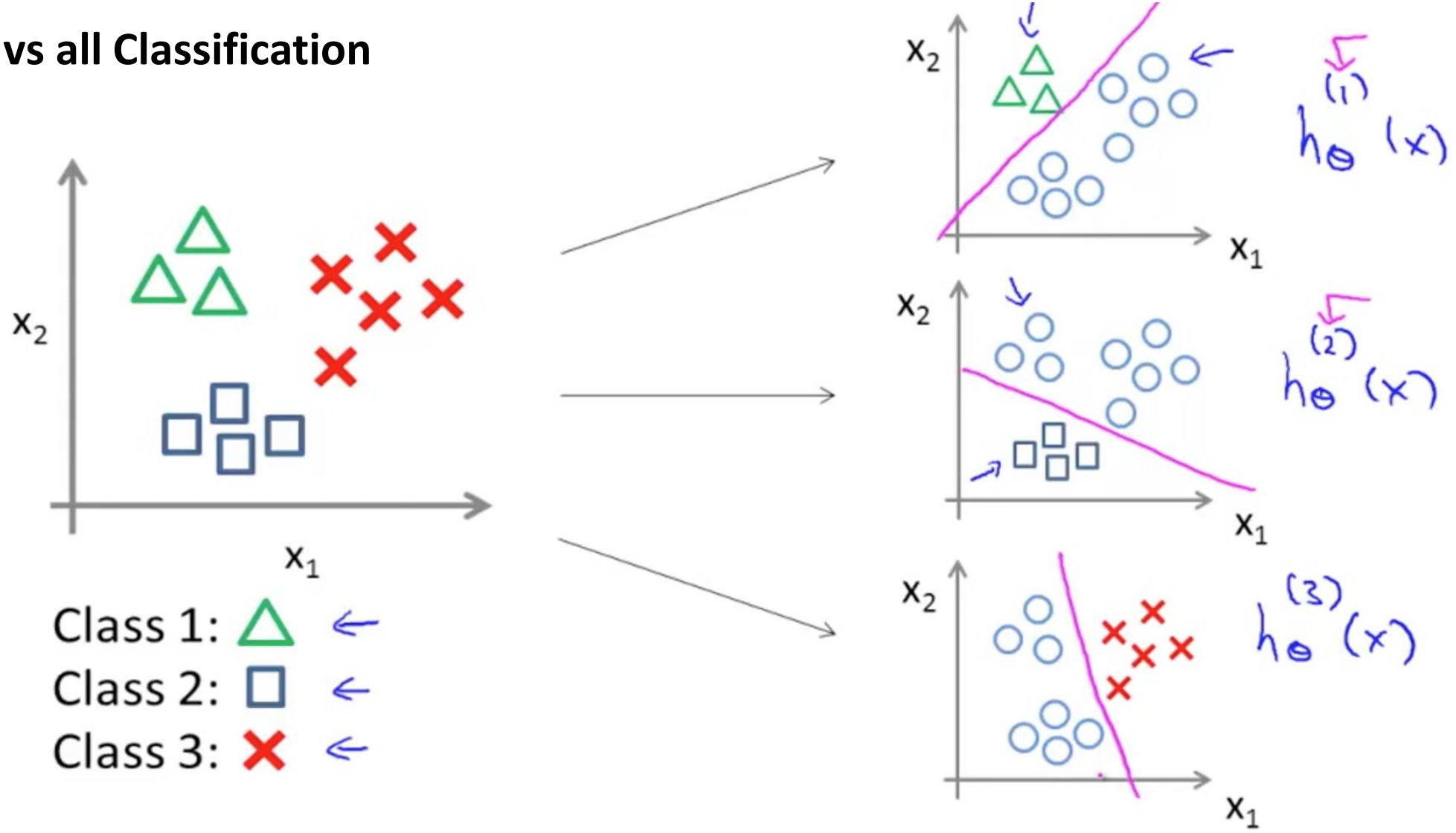
Similarly, when y = 0, we get the following plot for $J(\theta)$ vs $h_\theta(x)$:



Binary vs Multi Class Classification



One vs all Classification



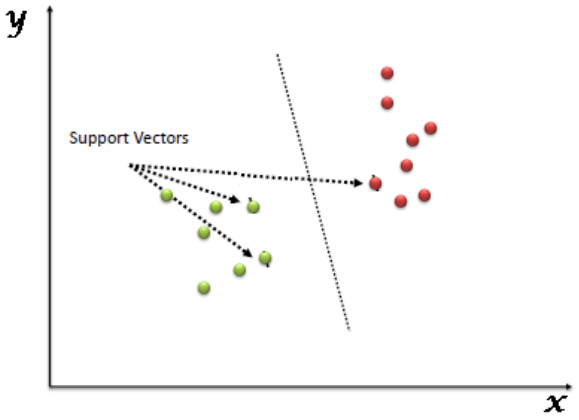
Support Vector Machine

Intuition

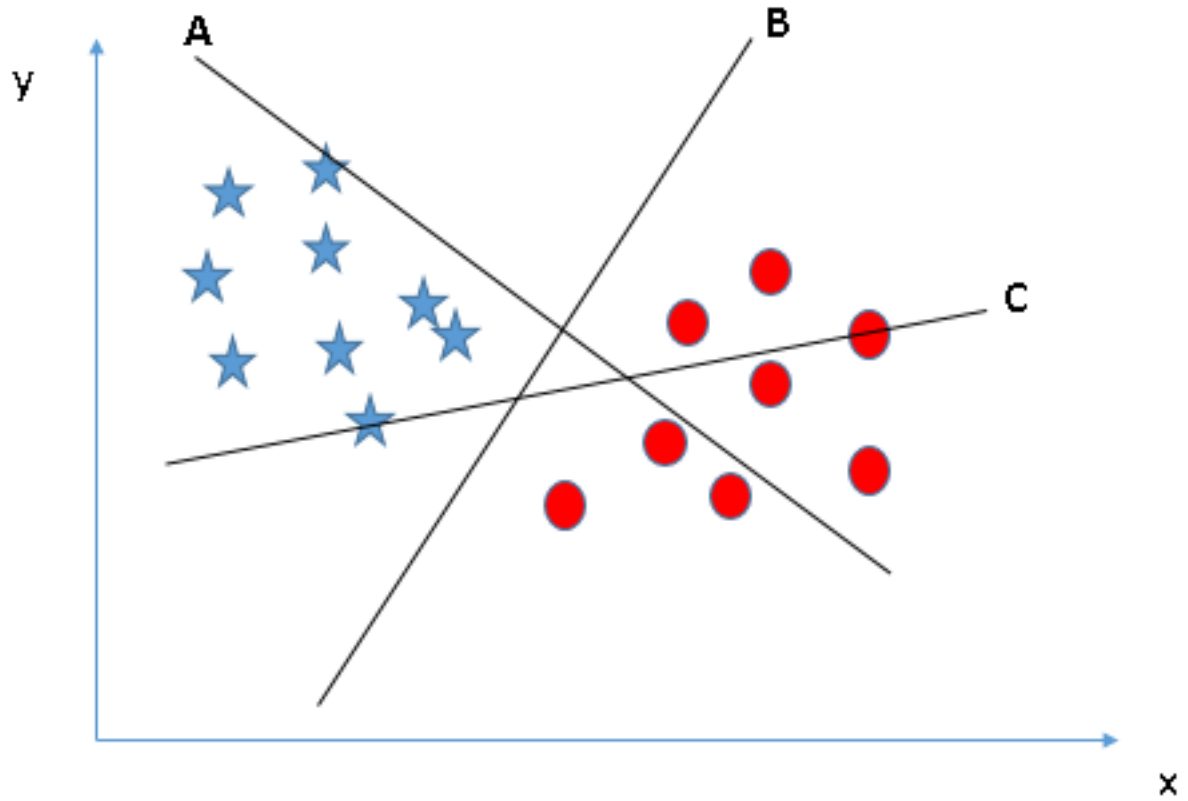
- Think of machine learning algorithms as an armory packed with axes, sword, blades, bow, dagger etc.
- You have various tools, but you ought to learn to use them at the right time
- As an analogy, think of ‘Regression’ as a sword capable of slicing and dicing data efficiently, but incapable of dealing with highly complex data
- On the contrary, ‘Support Vector Machines’ is like a sharp knife – it works on smaller datasets, but on them, it can be much more stronger and powerful in building models.

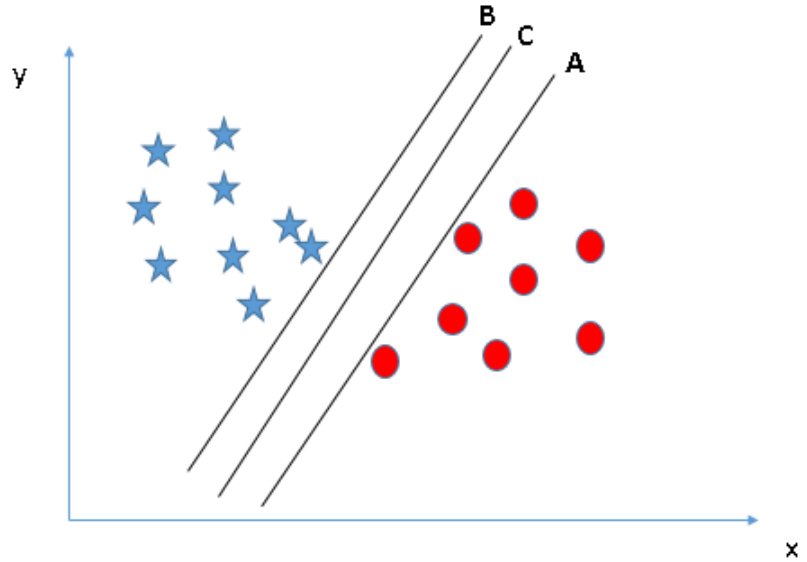
What is SVM

- “Support Vector Machine” (SVM) is a supervised machine learning algorithm, it is mostly used in classification problems
- In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate
- Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).
- Support Vectors are simply the co-ordinates of individual observation
- Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

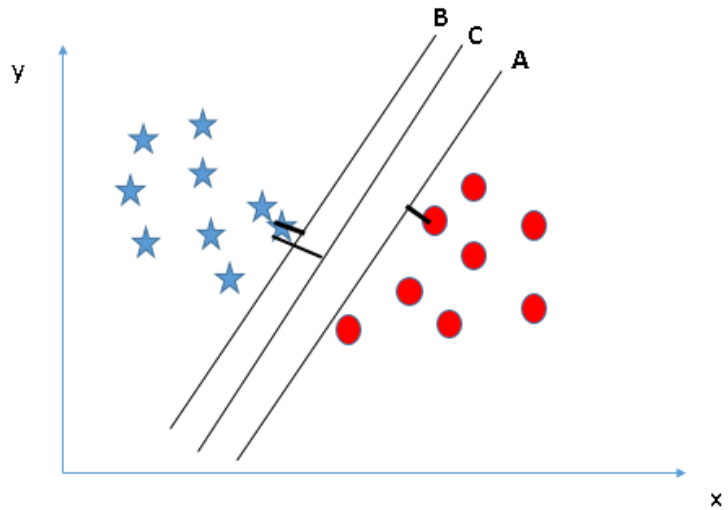


Which is the right Hyperplane



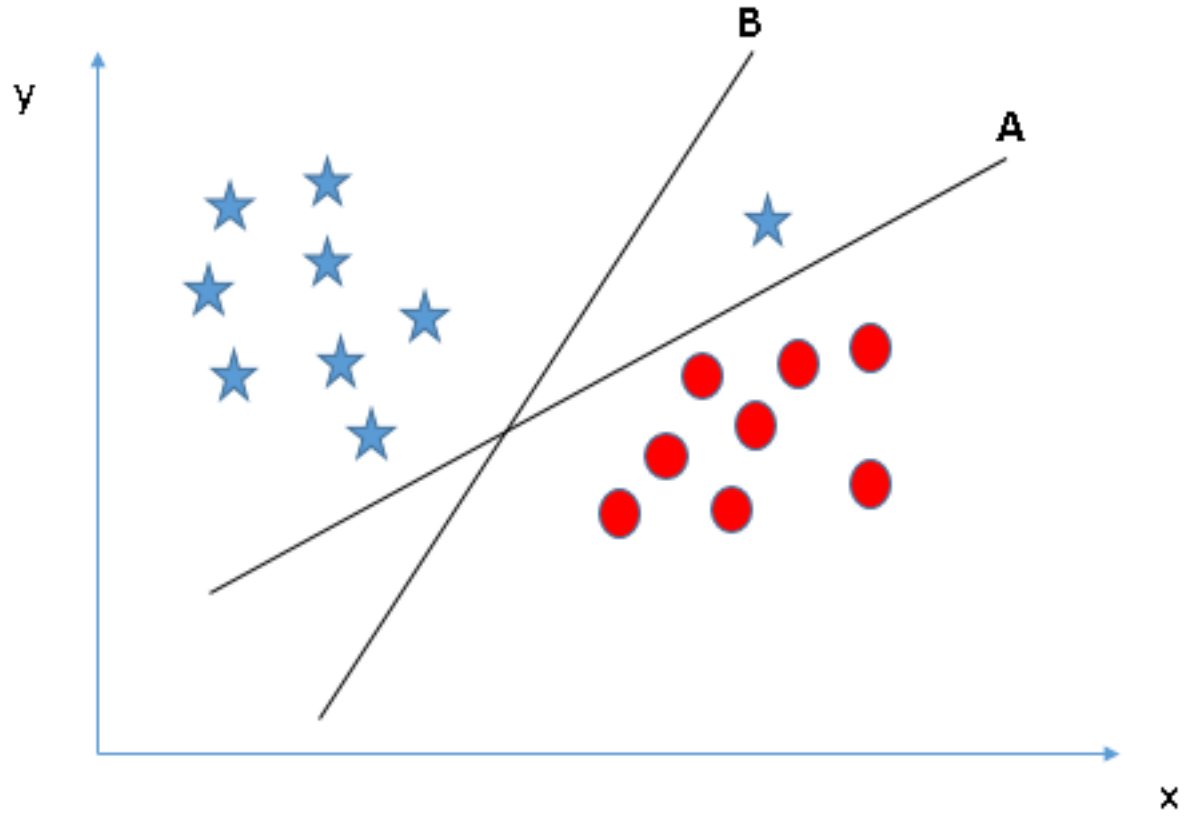


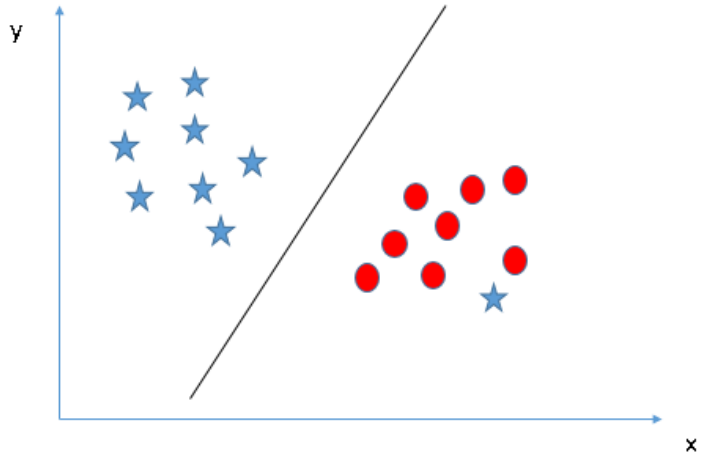
Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as **Margin**



If we select a hyper-plane having low margin then there is high chance of miss-classification.

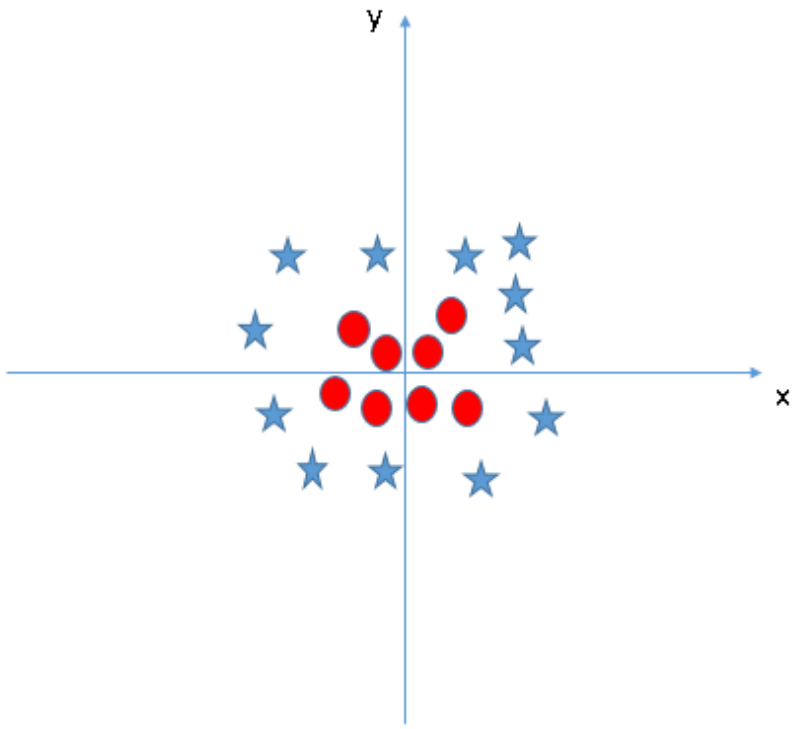
Which is the right Hyperplane



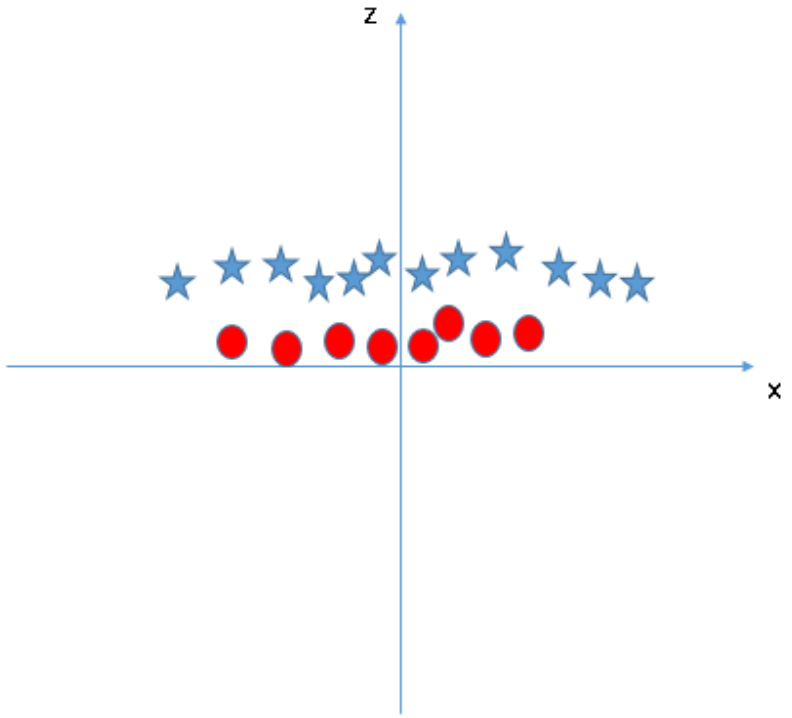


SVM has a feature to ignore ***outliers*** and find the hyper-plane that has maximum margin. Hence, we can say, SVM is robust to ***outliers***.

Can we have a linear line/hyperplane

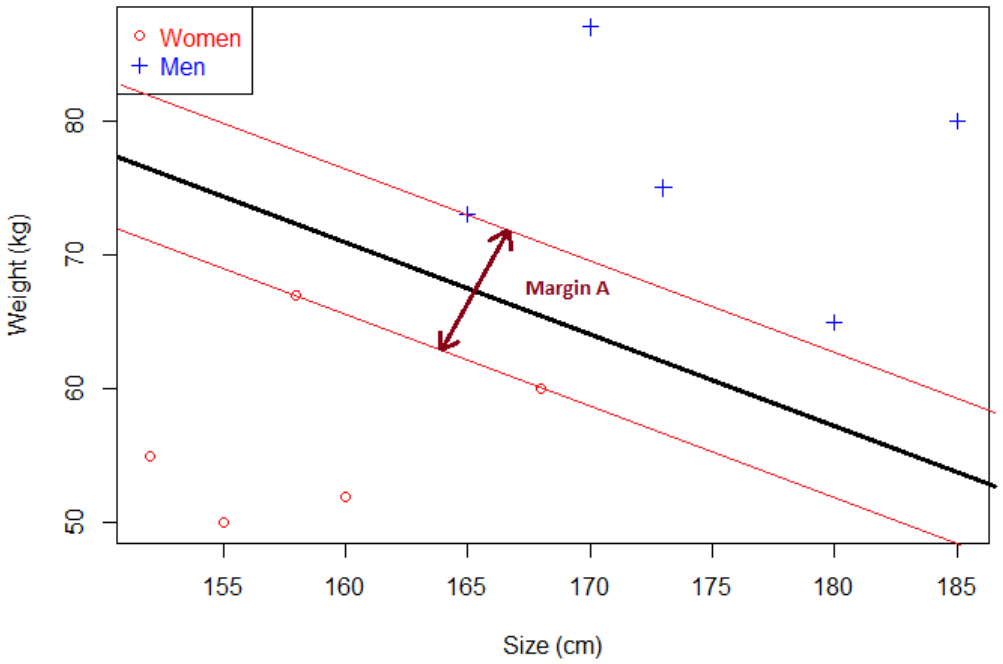


SVM can solve this problem. Easily! It solves this problem by introducing additional feature. Here, we will add a new feature z=x^2+y^2. Now, let's plot the data points on axis x and z:



We will not add the feature manually to have a hyper-plane. SVM has a technique called the **kernel trick. I**t is mostly useful in non-linear separation problem

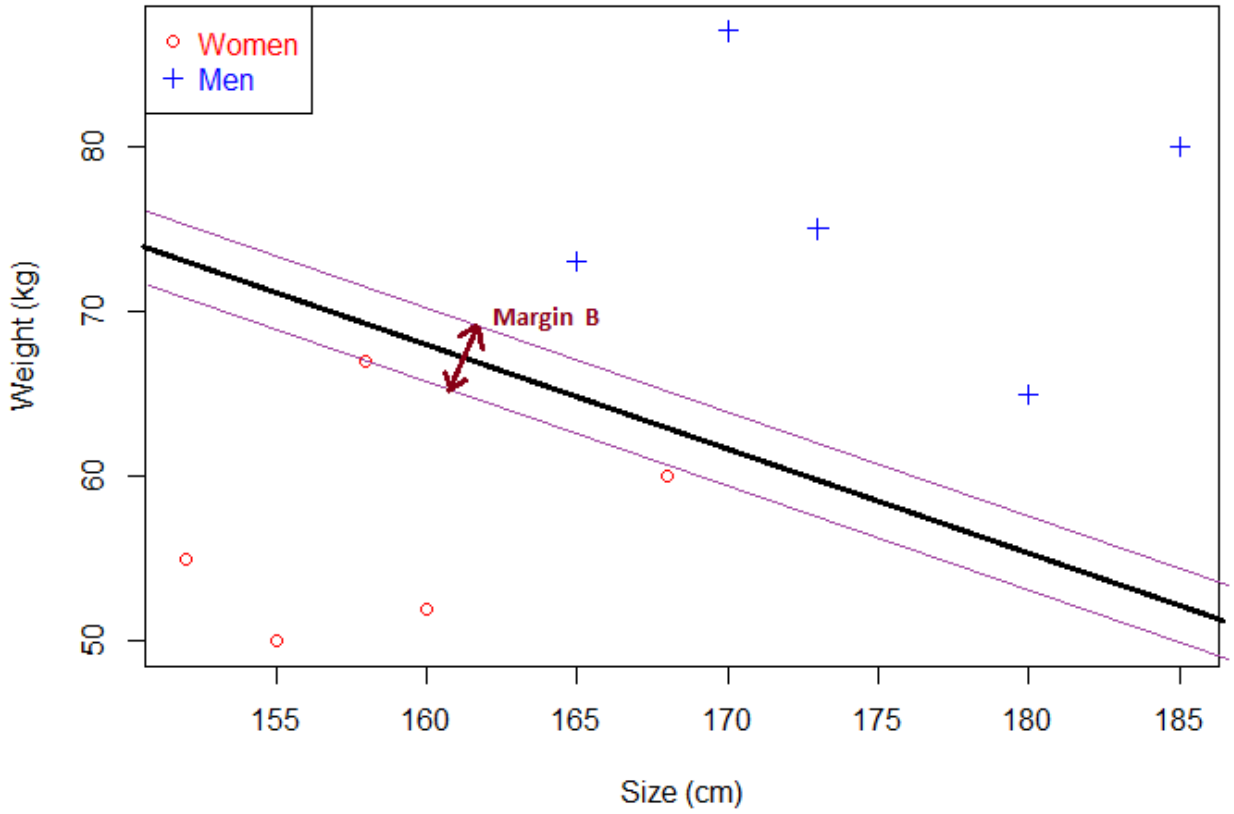
What is Margin and how does it give optimal hyperplane



- Given a particular hyperplane, we can compute the distance between the hyperplane and the closest data point. Once we have this value, if we double it we will get what is called the **margin**

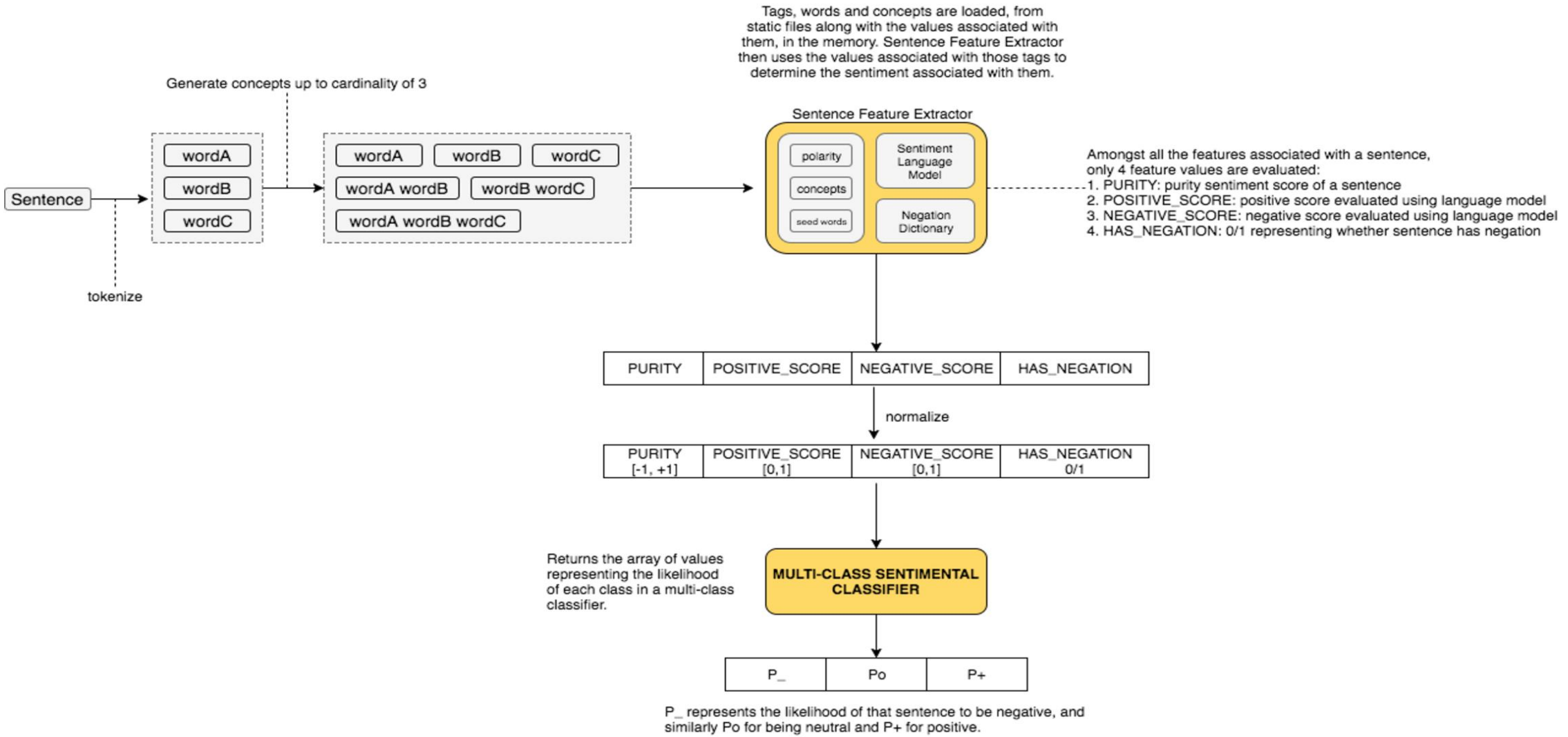
- Basically the margin is a ***no man's land***. There will never be any data point inside the margin

Not the best hyperplane



- If an hyperplane is very close to a data point, its margin will be small.
- The further an hyperplane is from a data point, the larger its margin will be.

How do we do it?



Sentiment classifier - Future - Information Extraction Pipeline

- Concept Extraction
- Concept Resolution
- Sentiment Analysis

Categories

Room Amenities

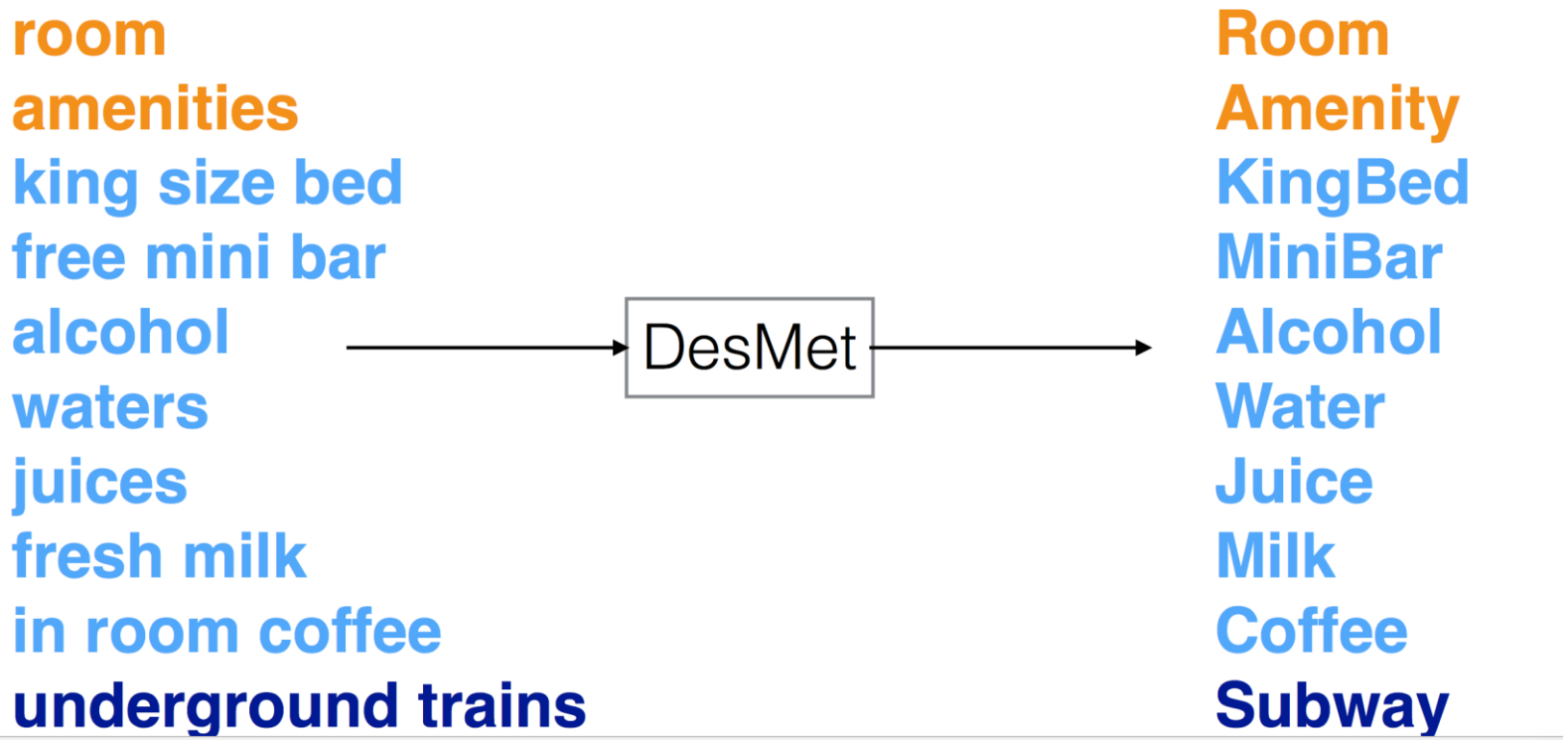
Property Amenities

Location Highlights

Miscellaneous

Room was spacious and well appointed with a deluxe **king size bed** and **amenities**. They offer a **free mini bar**, no **alcohol** but **waters juices** and **fresh milk** for your **in room coffee**. Only problem was constant background noise of the **underground trains** … too close to the **subway**.


Concept Extraction
Concept Resolution
Sentiment Analysis


room
amenities
king size bed
free mini bar
alcohol
waters
juices
fresh milk
in room coffee
underground trains
DesMet
Room
Amenity
KingBed
MiniBar
Alcohol
Water
Juice
Milk
Coffee
Subway

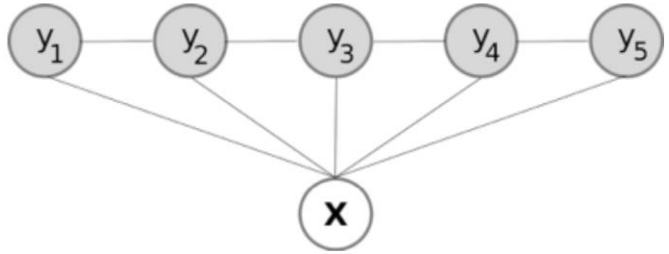
Sentiment

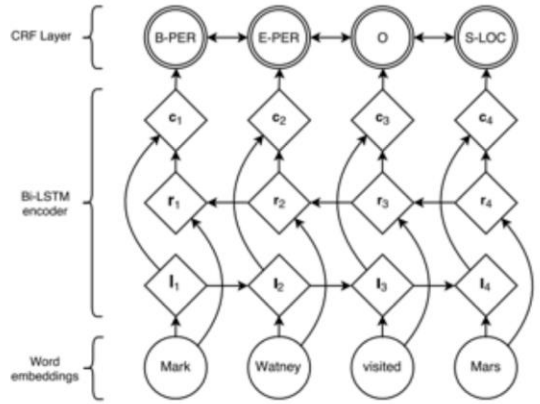
Positive
Neutral
Negative

Room was spacious and well appointed with a deluxe **king size bed** and **amenities**. They offer a **free mini bar**, no **alcohol** but **waters juices** and **fresh milk** for your **in room coffee**. Only problem was constant background noise of the **underground trains** … too close to the **subway**.

Concept Extraction

- Conditional random field (CRF) classifier*
- Long short-term memory (LSTM) CRF classifier**





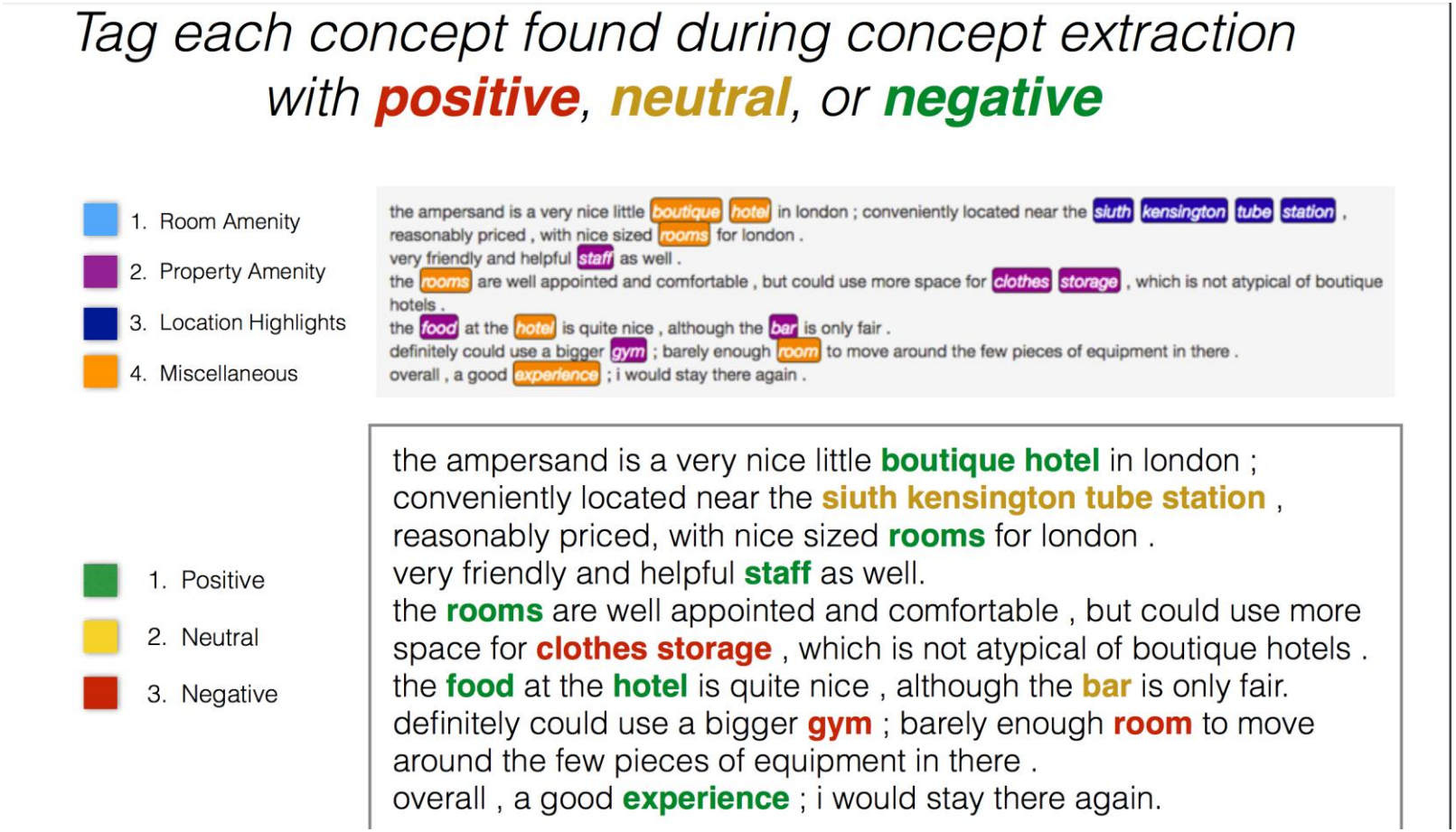
* http://nlp.stanford.edu/software/CRF-NER.shtml

** https://github.com/glample/tagger

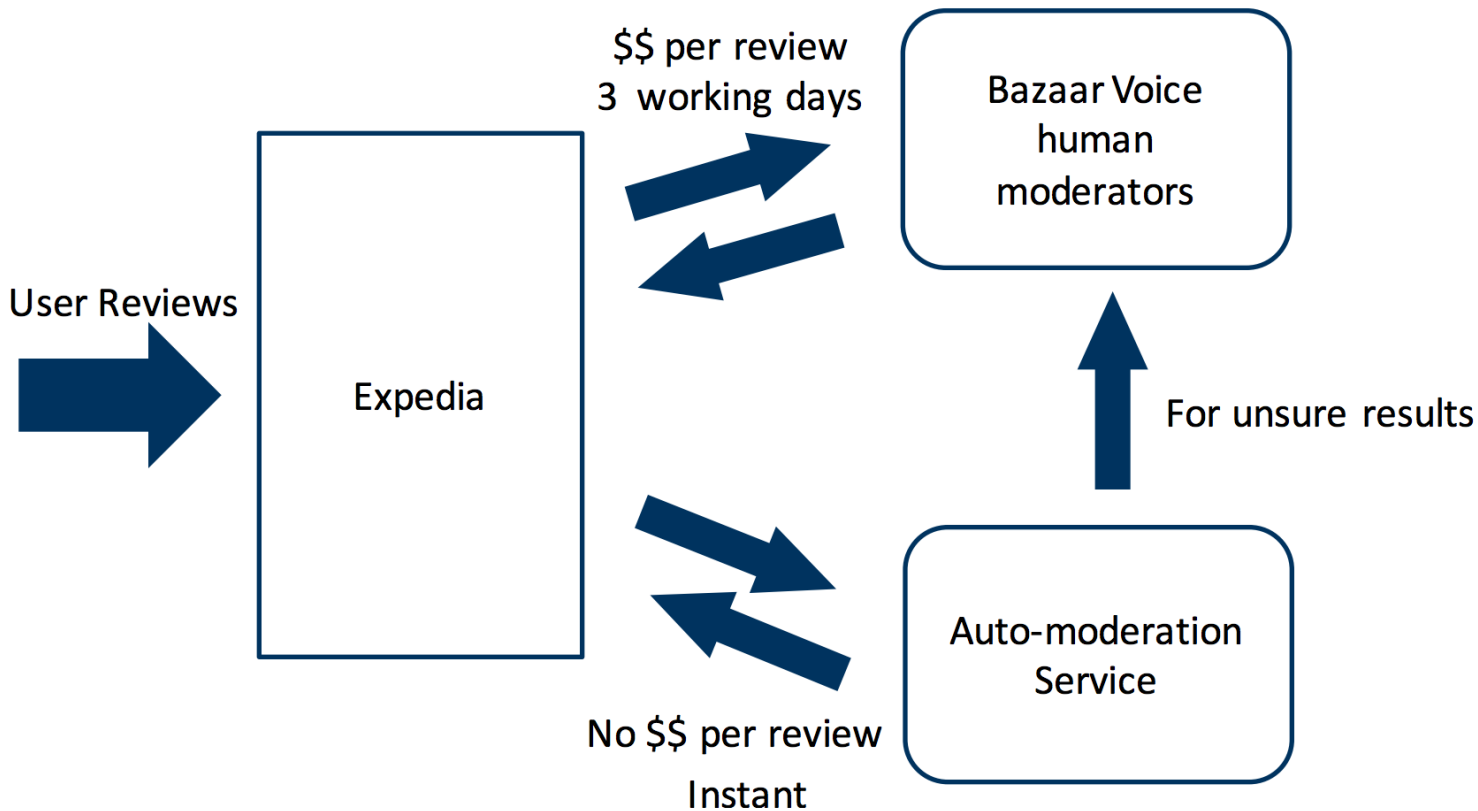
Concept Extraction

	#	Examples
Property (LCM)	150	airport drop off, atm, babysitting, coffee shop, fax machine, health club, lounge, steam room
Room (LCM)	316	air conditioning, television, wheelchair accessibility, hypo-allergenic bedding
Activity (LCM)	66	basketball, beach, golf, shopping, ski, tennis lessons, yoga classes
Breakfast (Wikipedia)	209	aloo paratha, bacon, bread, cinnamon roll, collops

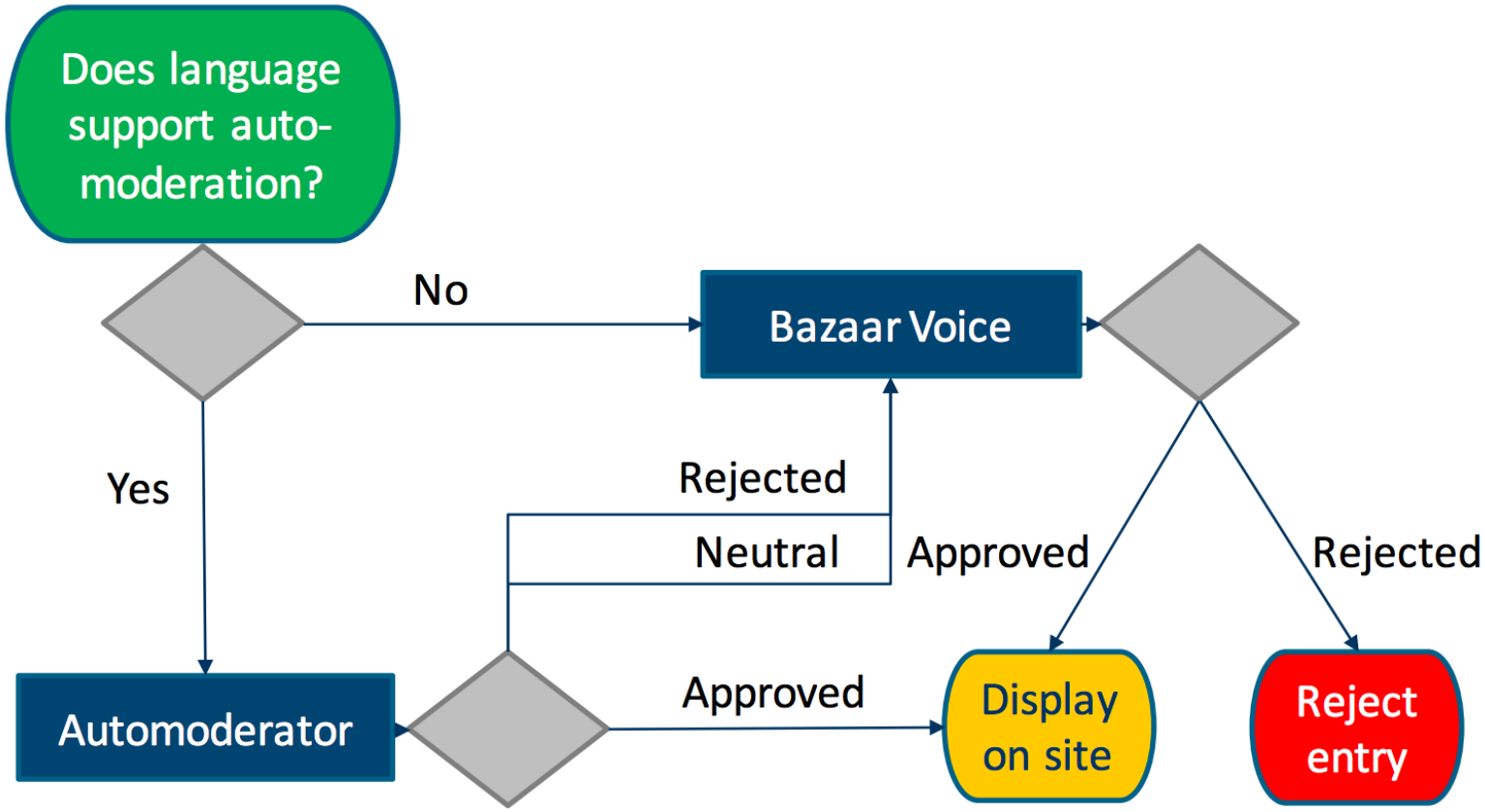
Sentiment Analysis



Auto-moderation is a cost and time-effective complement to the manual moderation system



Auto-moderation is used to quickly approve reviews the system is confident about to save costs and improve user experience

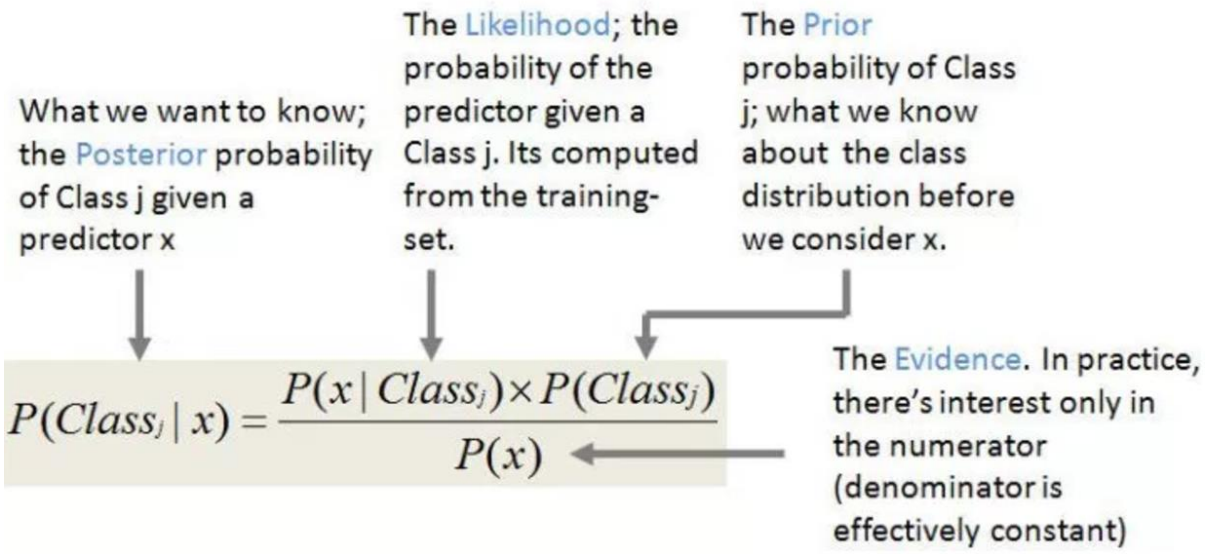


Auto-moderation Service - How do we do it?

We used Naïve Bayes Classifier – But why?

- It is simple and if the conditional independence assumption actually holds, a Naive Bayes classifier will converge quicker than discriminative models like logistic regression, so you need less training data.
- It requires less model training time
- It is an anyhow decent choice when working with text classification
- Handles missing data well using smoothing algorithms

How does it work?



Applying the independence assumption

$$P(x \mid Class_j) = P(x_1 \mid Class_j) \times P(x_2 \mid Class_j) \times \ldots \times P(x_k \mid Class_j)$$

Substituting the independence assumption, we derive the Posterior probability of Class j given a new instance x' as...

$$P(Class_j \mid x') = P(x'_1 \mid Class_j) \times P(x'_2 \mid Class_j) \times \ldots \times P(x'_k \mid Class_j) \times P(Class_j)$$

How does it work?

$$p(approved|review) = \frac{p(review|approved)p(approved)}{p(review)}$$

where: p(approved|review) = probability of the review being in approved

p(review|approved) = probability of observing the review given the pool of approved reviews

p(approved) = probability of finding an approved review

p(review) = probability of observing the review. Since p(review) is constant for both approved and rejected, this can be ignored.

$$p(word_i|approved) = \frac{\text{word frequency in all approved reviews}}{\text{sum of frequency of all words in approved reviews}}$$

Furthermore, since we are assuming that the occurrences of words are independent, p(review|approved) can be calculated using the formula:

$$p(review|approved) = \Pi_i P(word_i|approved)$$

How does it work?

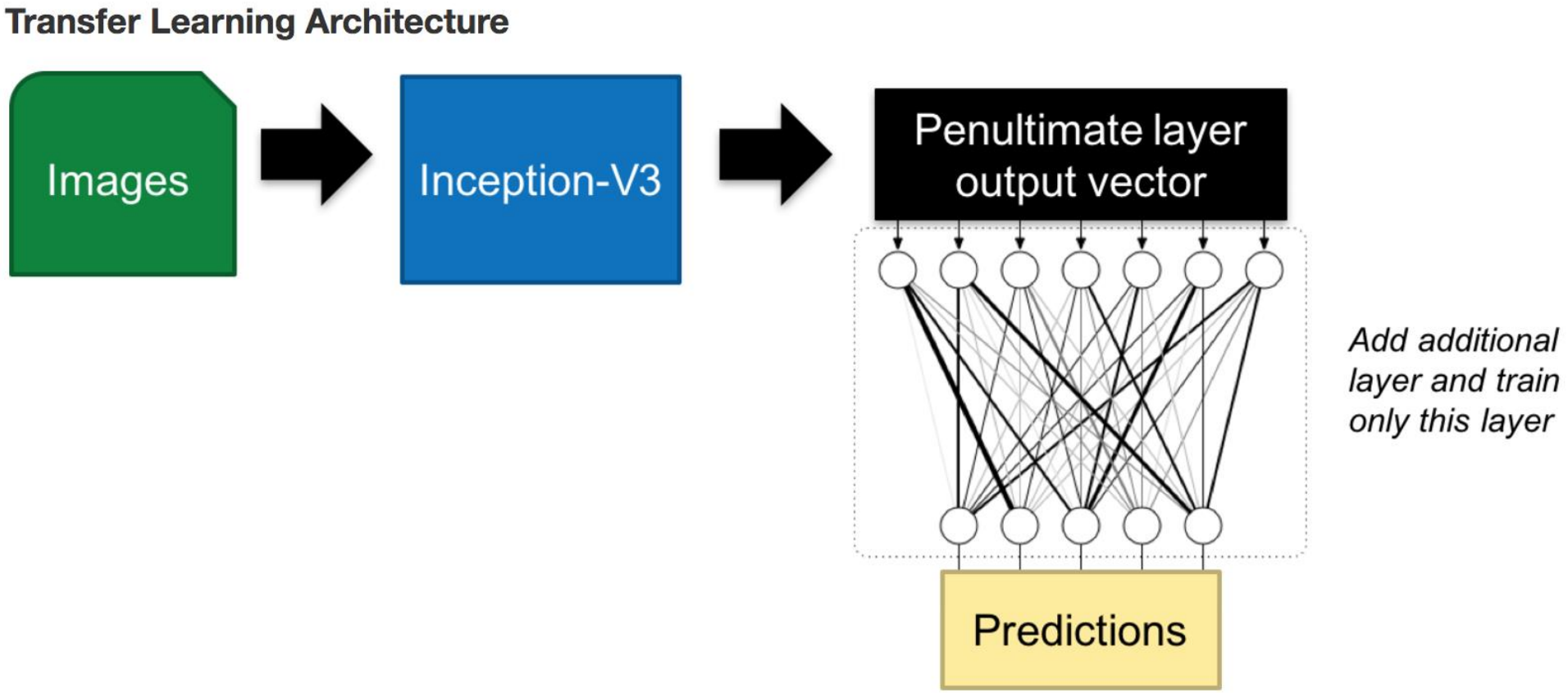
The general chain rule (always true):

$$P(x_1, x_2, \ldots, x_n) = P(x_1 \mid x_2, x_3, \ldots, x_n) P(x_2, x_3, \ldots, x_n) =$$
$$P(x_1 \mid x_2, x_3, \ldots, x_n) P(x_2 \mid x_3, x_4, \ldots, x_n) P(x_3, x_4, \ldots, x_n) = \cdots$$
$$= \prod_{i=1}^{n} P(x_i \mid x_{i+1}, \ldots, x_n)$$

The Bayesian network chain rule:

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid parents(X_i))$$

UGC - Image recognition using Tensorflow/Inception V3



PolloLoco -Image recognition using Caffe

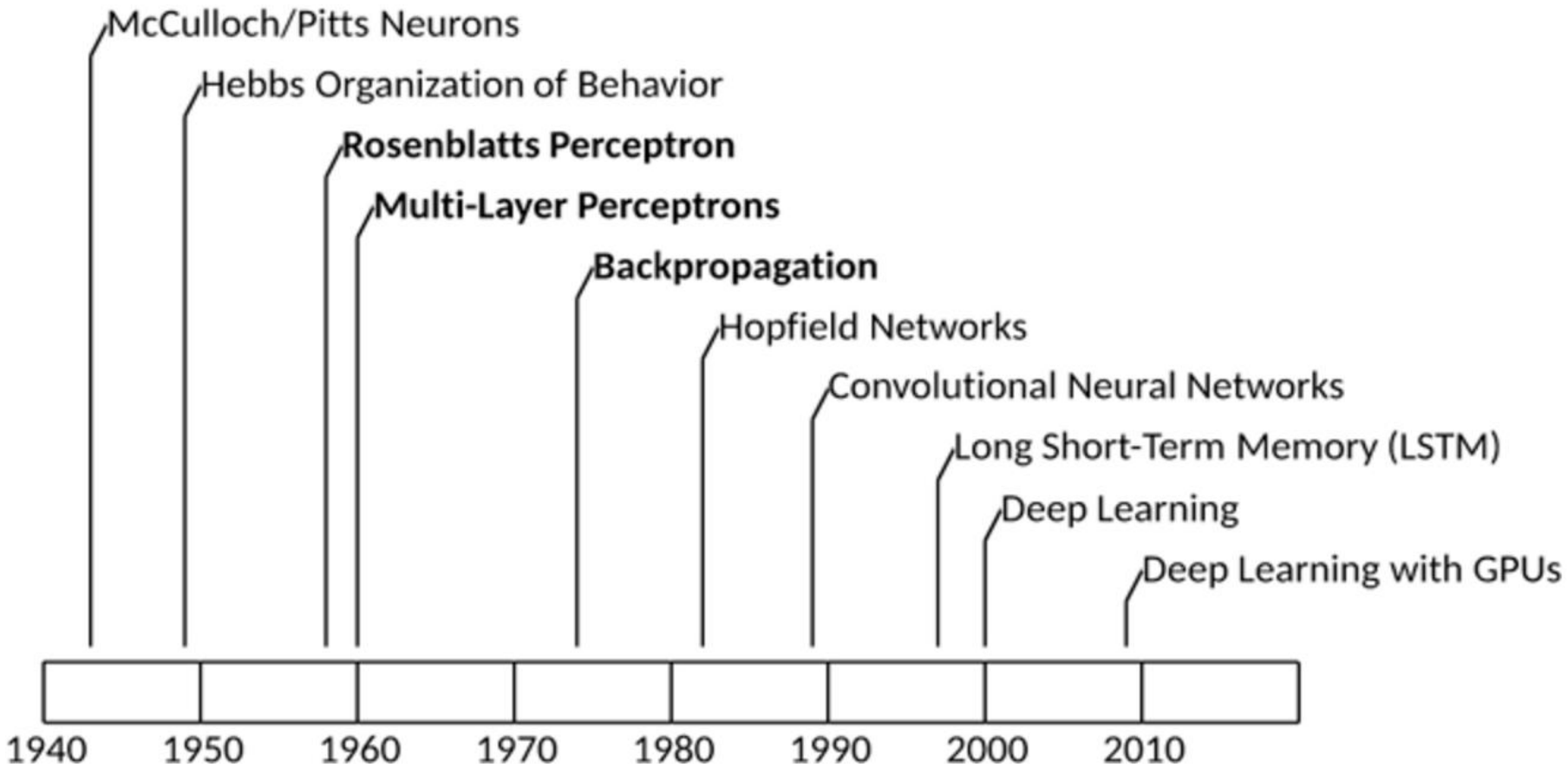
- Out of domain classification
- Scene classification
- Attractiveness ranking

- Based on Caffe and Alexnet

- http://image-classifier.pollo-loco.test.expedia.com/

Neural Networks

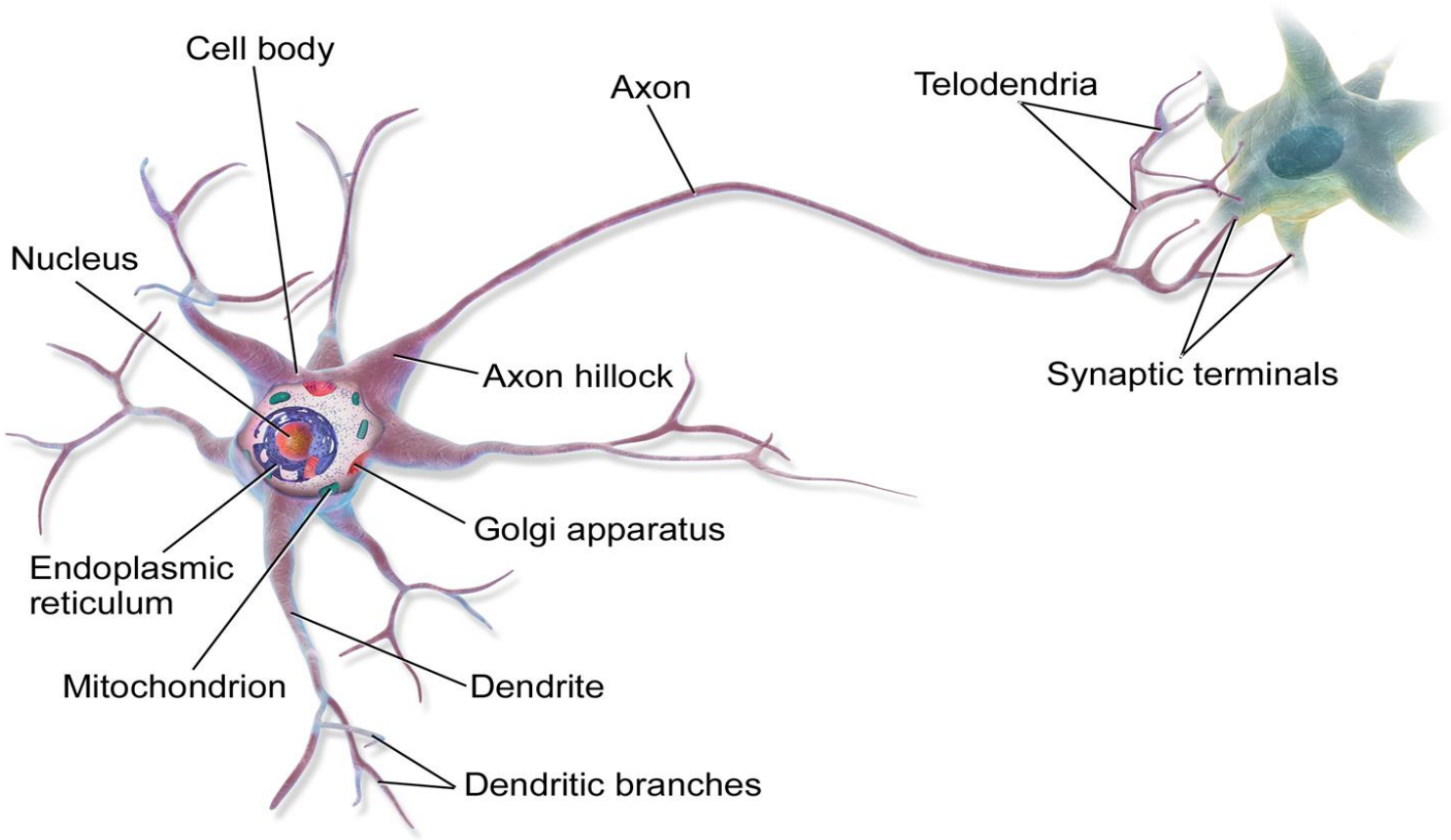
History of Neural Networks



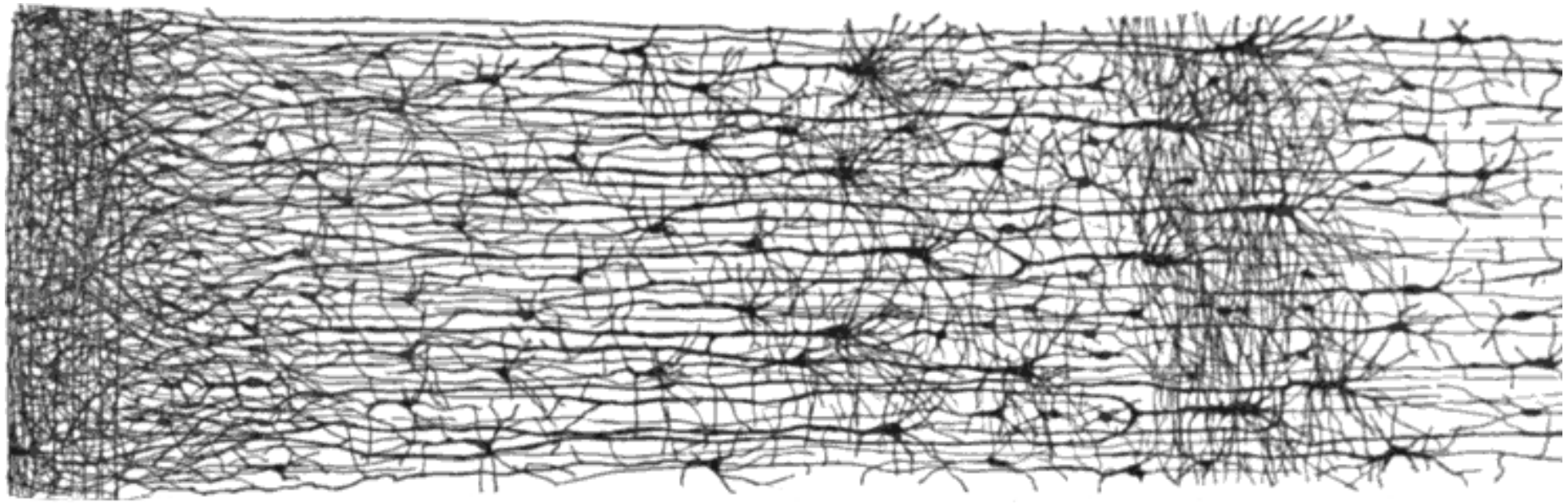
Why popular today?

- Lot of data around to train
- Computing power – Gaming industry producing GPU's and Moore's law
- Small tweaks – Huge impact as compared to what was there
- Assumptions against proved benign – ANN might get stuck in local minimum
- Popularity - ANNs seem to have entered a virtuous circle of funding and progress

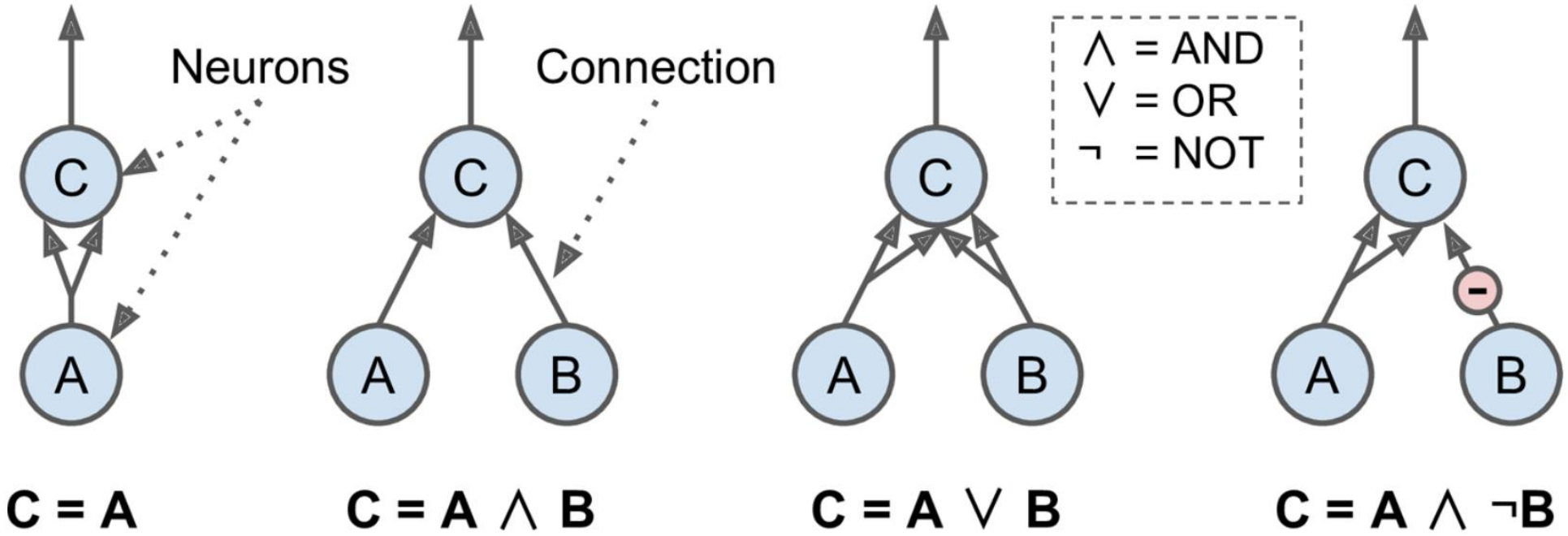
Biological Neurons



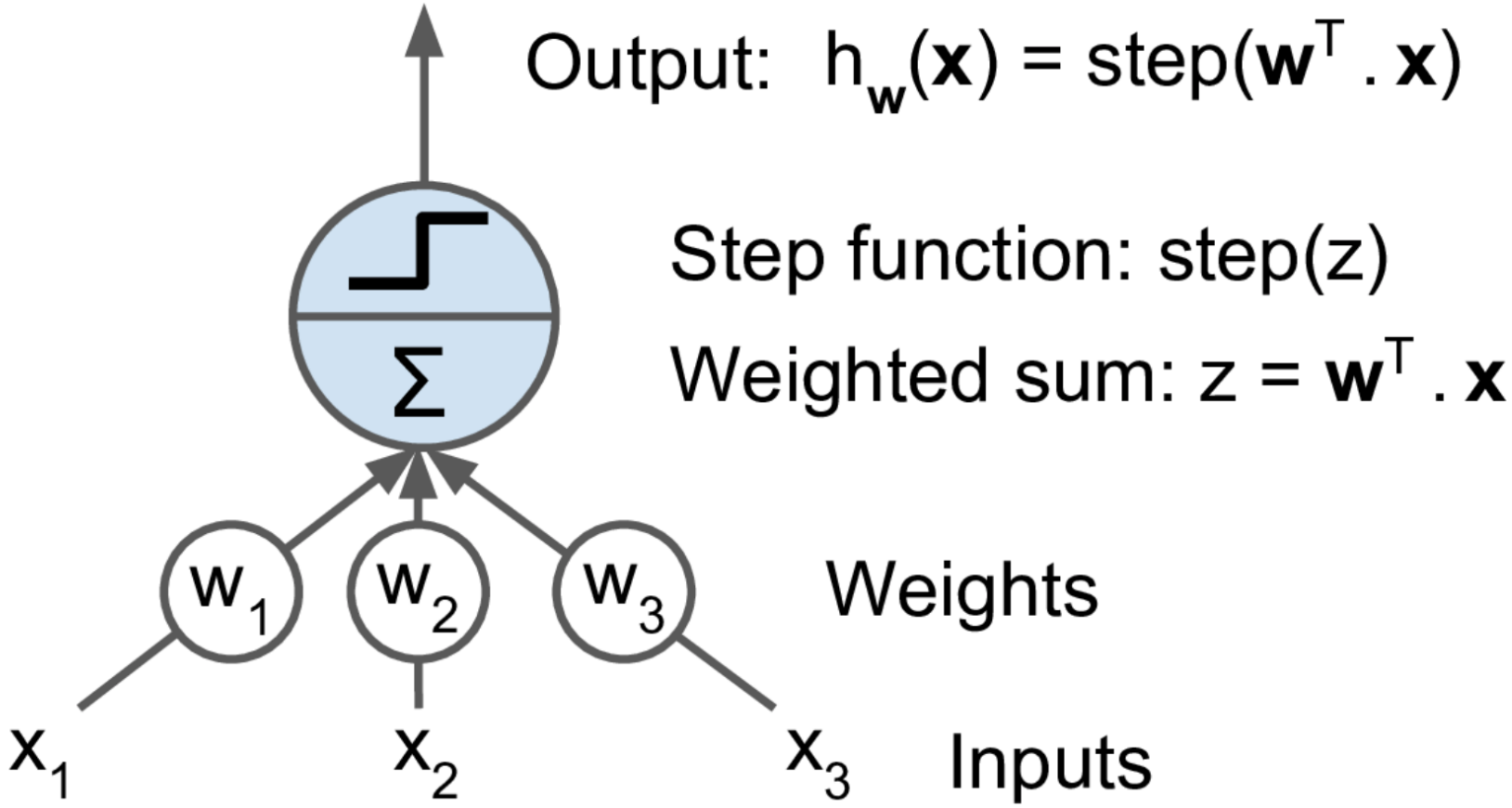
Biological Neurons



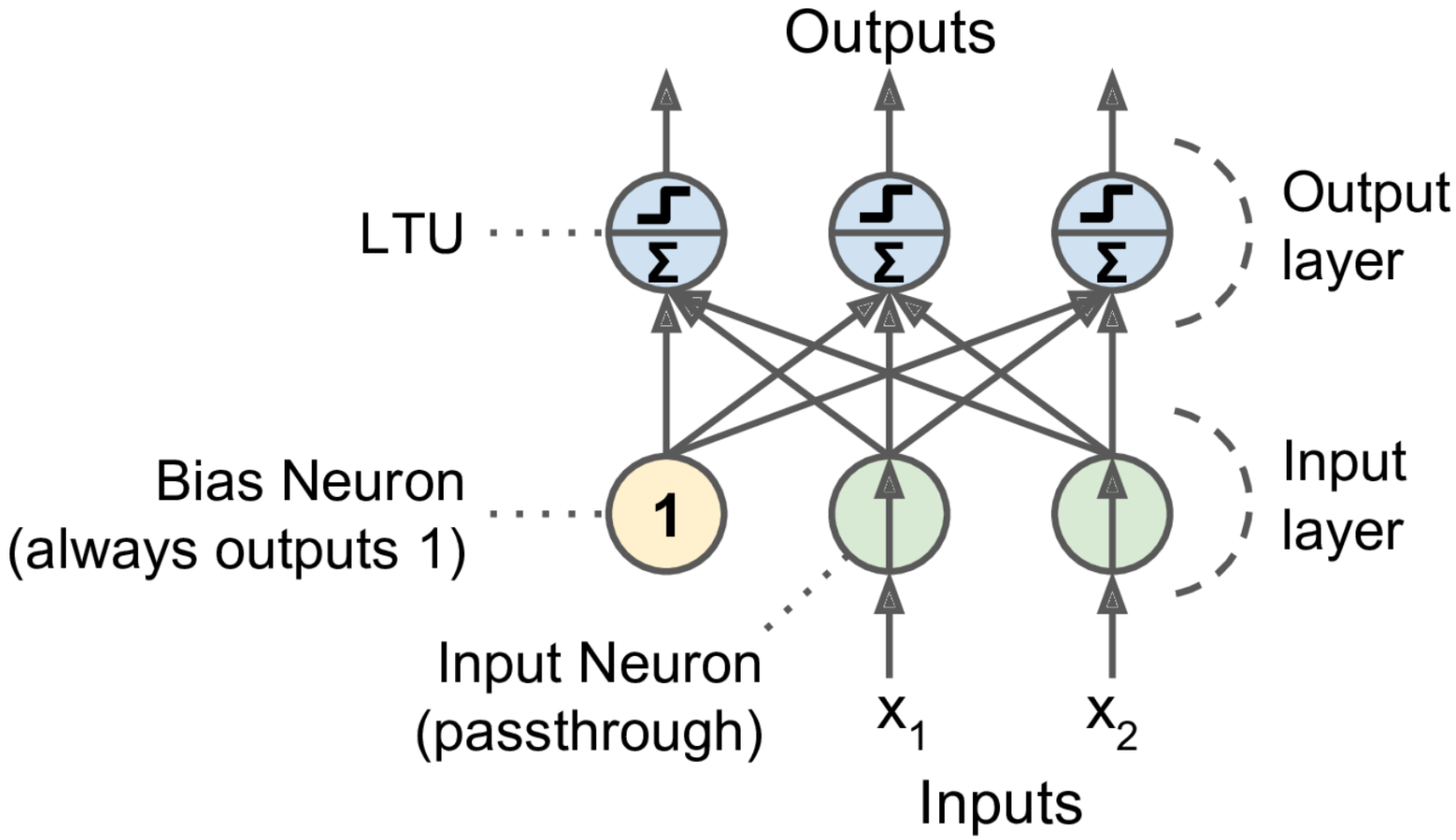
Basic form of ANN



Linear Threshold Unit



Perceptron

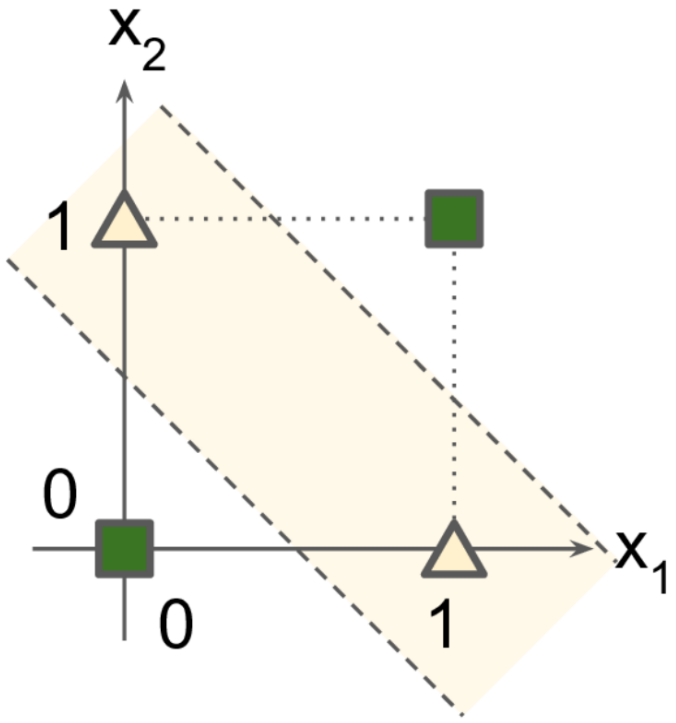

Outputs
LTU
Output layer
Bias Neuron (always outputs 1)
1
Input layer
Input Neuron (passthrough)
x_1
x_2
Inputs

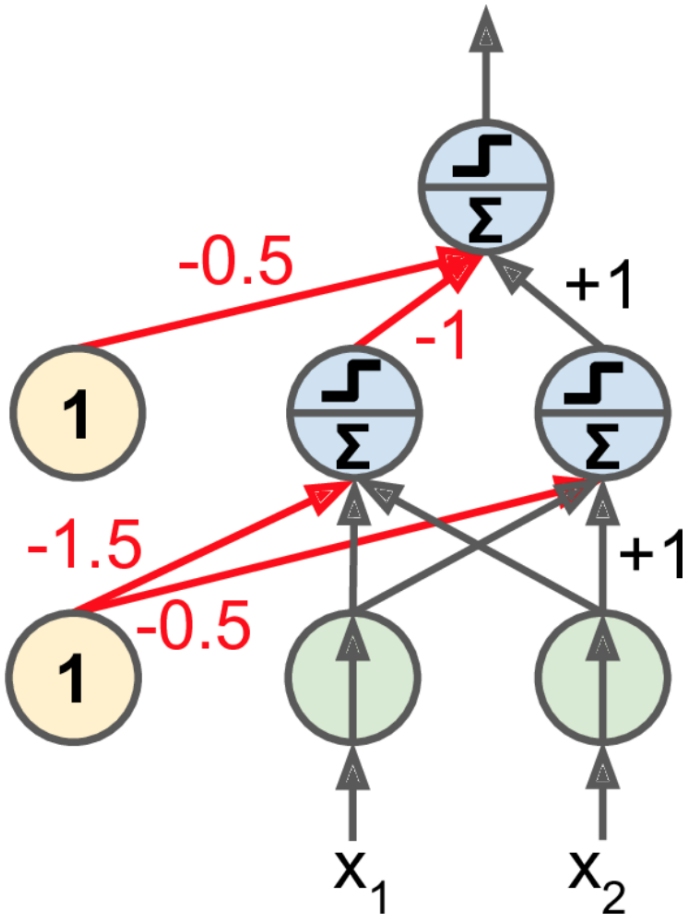
Hebb's rule - Cells that fire together wire together

$$w_{i,j}^{(\text{next step})} = w_{i,j} + \eta(y_j - \hat{y}_j)x_i$$

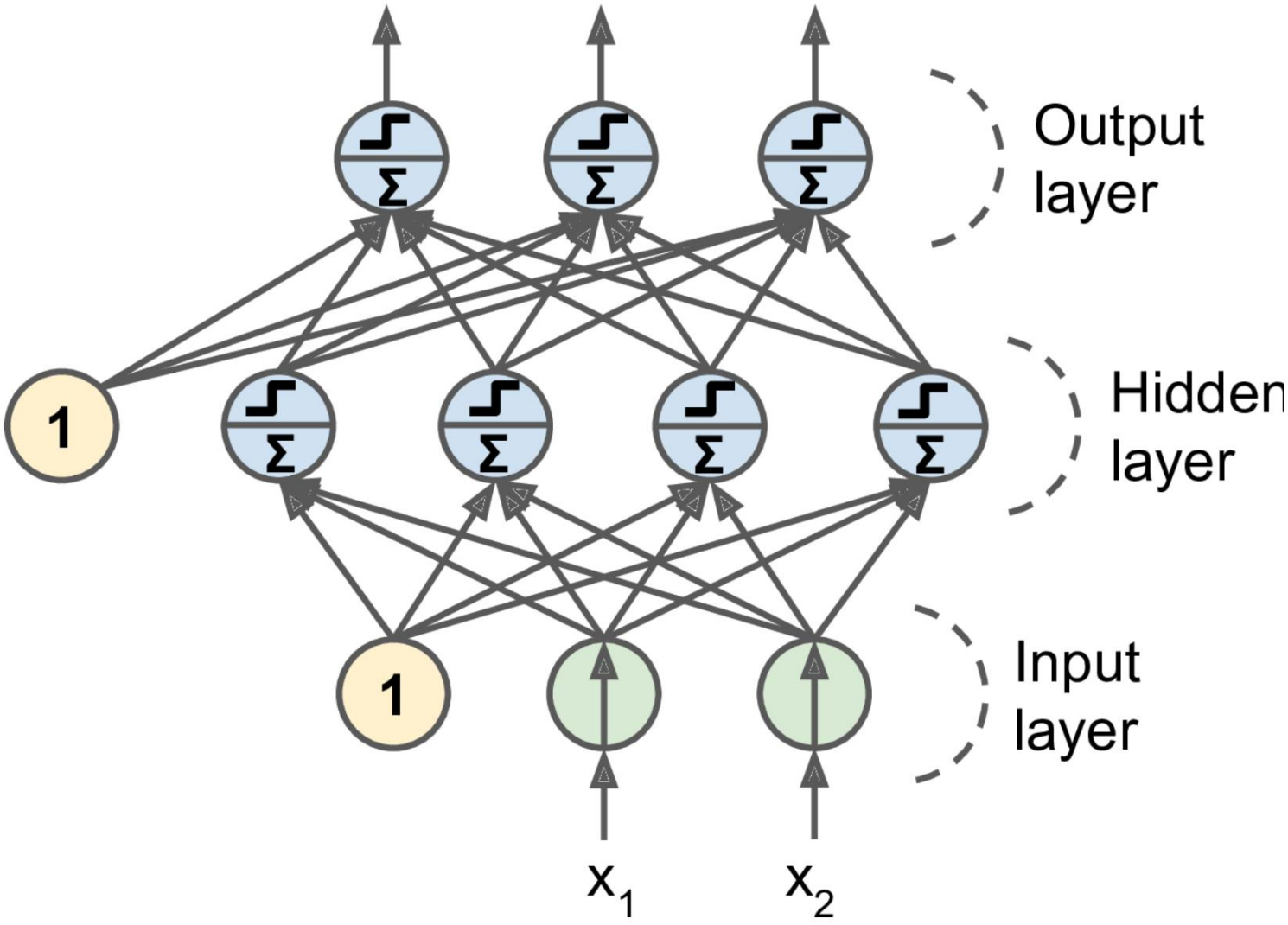
- $w_{i,j}$ is the connection weight between the i^{th} input neuron and the j^{th} output neuron.
- x_i is the i^{th} input value of the current training instance.
- $\hat{y}_j$ is the output of the j^{th} output neuron for the current training instance.
- y_j is the target output of the j^{th} output neuron for the current training instance.
- η is the learning rate.

XOR Problem and Multi-Layer Perceptron





MLP/ANN



How does it work?

1. **Input** x**:** Set the corresponding activation a^1 for the input layer.

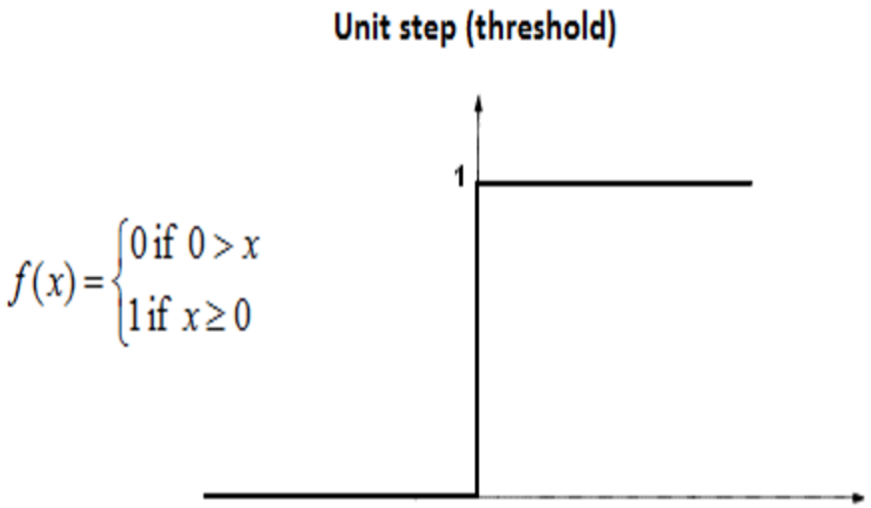
2. **Feedforward:** For each $l = 2, 3, \ldots, L$ compute $z^l = w^l a^{l-1} + b^l$ and $a^l = \sigma(z^l)$.

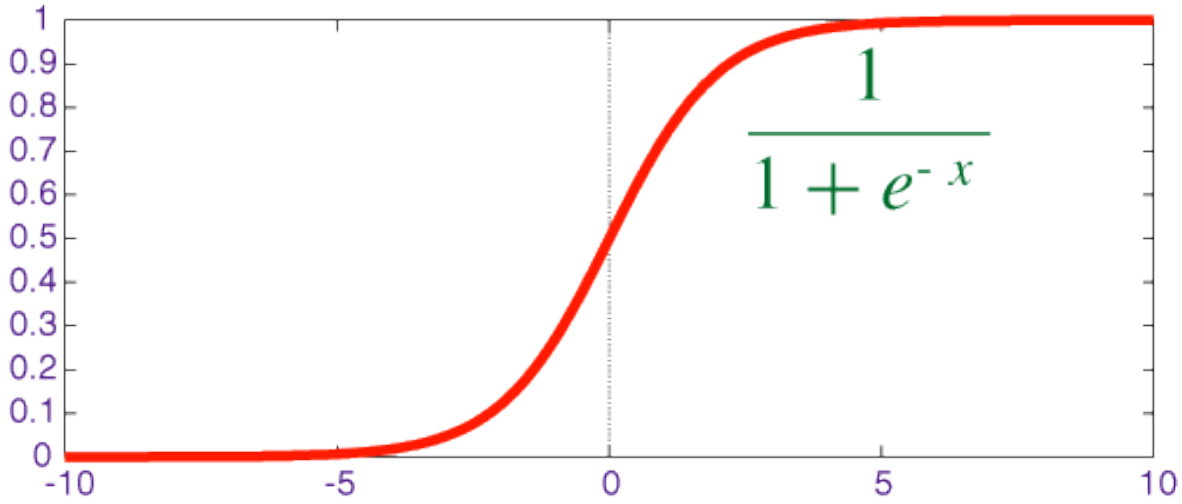
3. **Output error** δ^L**:** Compute the vector $\delta^L = \nabla_a C \odot \sigma'(z^L)$.

4. **Backpropagate the error:** For each $l = L-1, L-2, \ldots, 2$ compute $\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l)$.

5. **Output:** The gradient of the cost function is given by $\frac{\partial C}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j$ and $\frac{\partial C}{\partial b^l_j} = \delta^l_j$.

Better activation function?





Different type of activation functions

$$sigmoid: \quad \sigma(x) = \frac{1}{1+e^{-x}}$$

$$tanh: \quad \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

$$RELU: \quad relu(x) = \max(0, x)$$

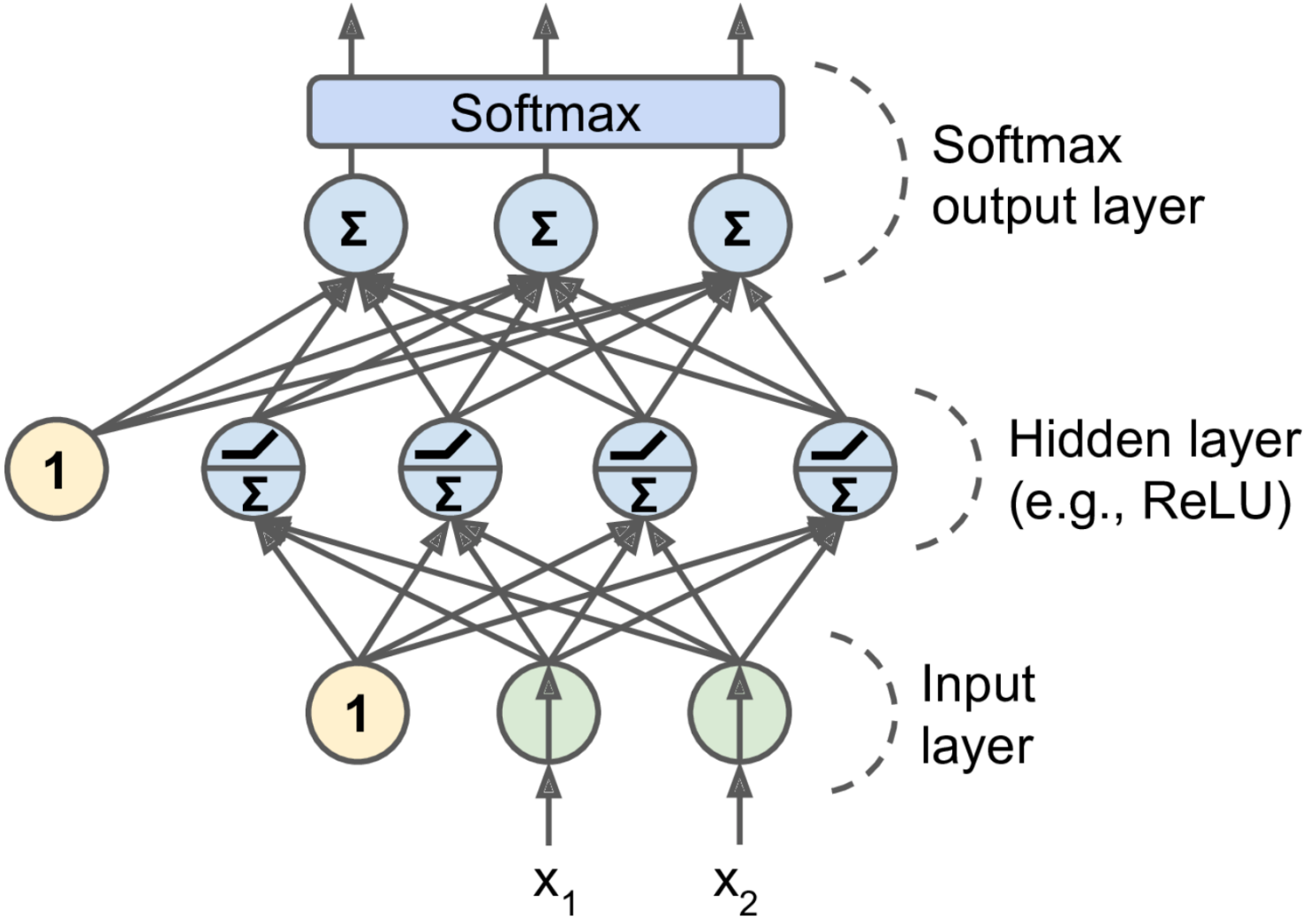
Different type of activation functions

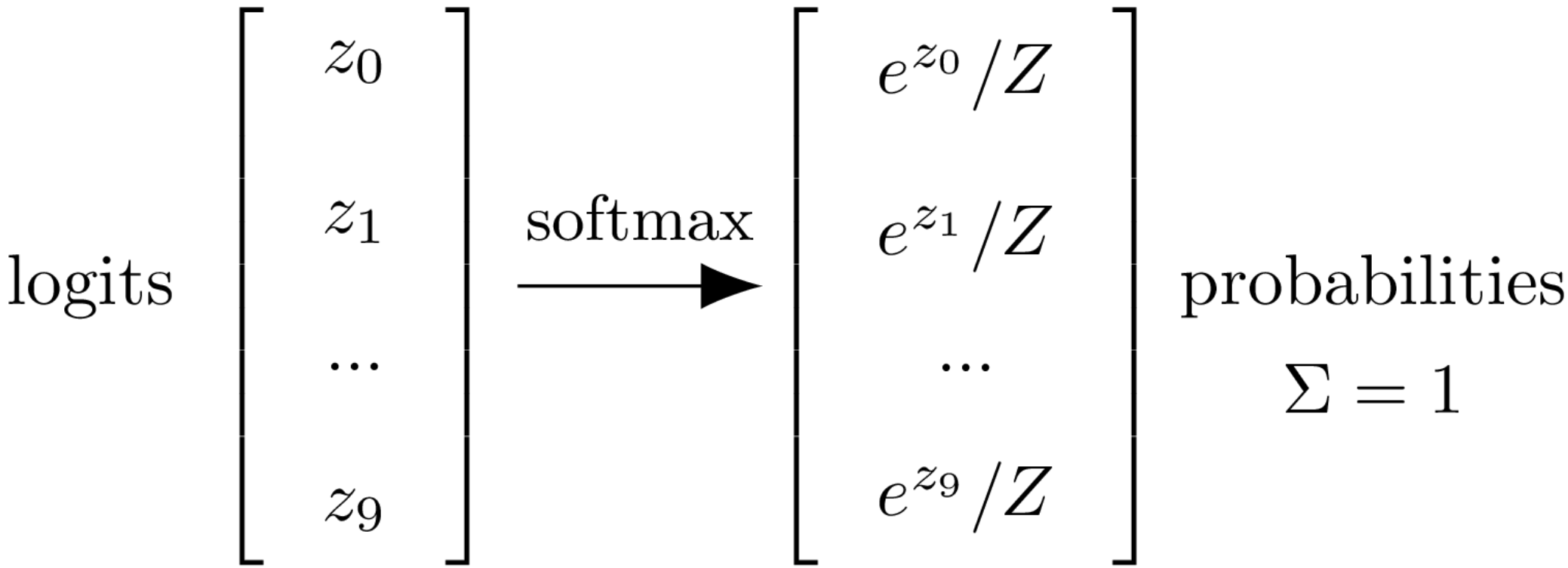
$$sigmoid: \quad \sigma(x) = \frac{1}{1+e^{-x}}$$

$$tanh: \quad \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

$$RELU: \quad relu(x) = \max(0, x)$$

Shared Softmax

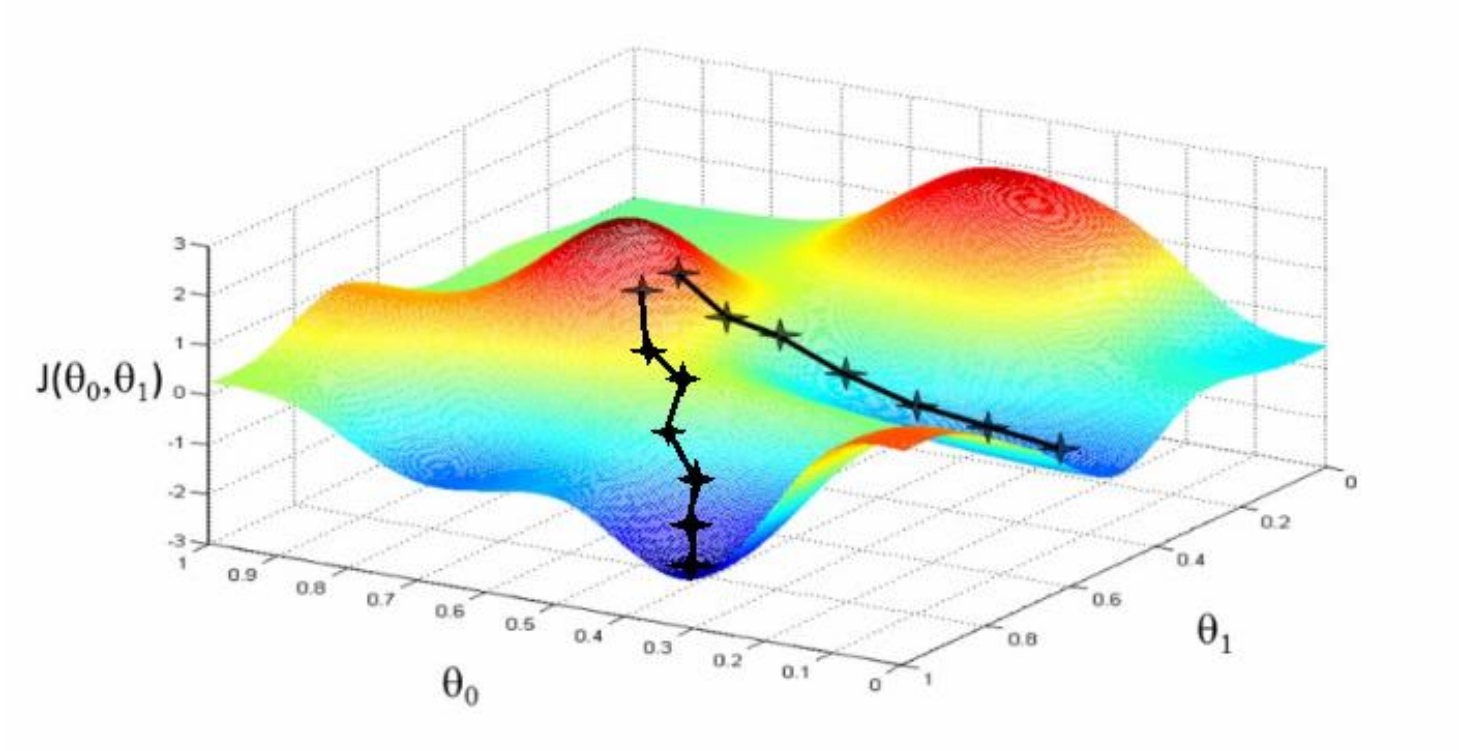



logits
z_0
z_1
...
z_9
softmax
e^{z_0}/Z
e^{z_1}/Z
...
e^{z_9}/Z
probabilities
$\Sigma = 1$

Why neural networks?

- As the number of quadratic, cubic or high order independent variables increase the regression model becomes more computationally expensive. For Ex if we have say 100 features and we want to include the combination of features such as (x1^2,x1.x2,x1.x3..x1.x100,x2^2,.....,x100^x100) then 100 features are now appx. 5000 . Regression is not good with so much quadratic or higher order features.

Gradient Descent



Types of Gradient Descent

1.Batch Gradient Descent

- we use the complete dataset available to compute the gradient of cost function.
- batch gradient descent can be very slow and is intractable for datasets that don't fit in memory

Repeat until convergence

{

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

}

2. Stochastic Gradient Descent

- we use the 1 training Example to compute the gradient of cost function.

- Stochastic gradient descent is very fast but accuracy is low

Randomly shuffle (reorder) training examples

Repeat {
  for $i := 1, \ldots, m$ {
    $\theta_j := \theta_j - \alpha(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$
      (for every $j = 0, \ldots, n$)
  }
}

3. Mini Batch Gradient Descent

- we use the a batch of m training Example to compute the gradient of cost function.

- Mini Batch gradient descent is faster then batch and accuracy is higher than Stochastic

Say $b = 10, m = 1000.$
Repeat {
  for $i = 1, 11, 21, 31, \ldots, 991$ {

$$\theta_j := \theta_j - \alpha \frac{1}{10} \sum_{k=i}^{i+9} (h_\theta(x^{(k)}) - y^{(k)}) x_j^{(k)}$$

  (for every $j = 0, \ldots, n$) } }

PARAMETERS	BATCH GD ALGORITHM	MINI BATCH ALGORITHM	STOCHASTIC GD ALGORITHM
ACCURACY	HIGH	MODERATE	LOW
TIME CONSUMING	MORE	MODERATE	LESS

Tuning neural network

- Random Search/Grid Search/Use Oscar tool

- Number of Hidden layer – Most cases 1 or 2 hidden layers suffice
- Number of neurons per hidden layer – Common practice to form a tunnel
- Choice of activation function – ReLu works most of the time

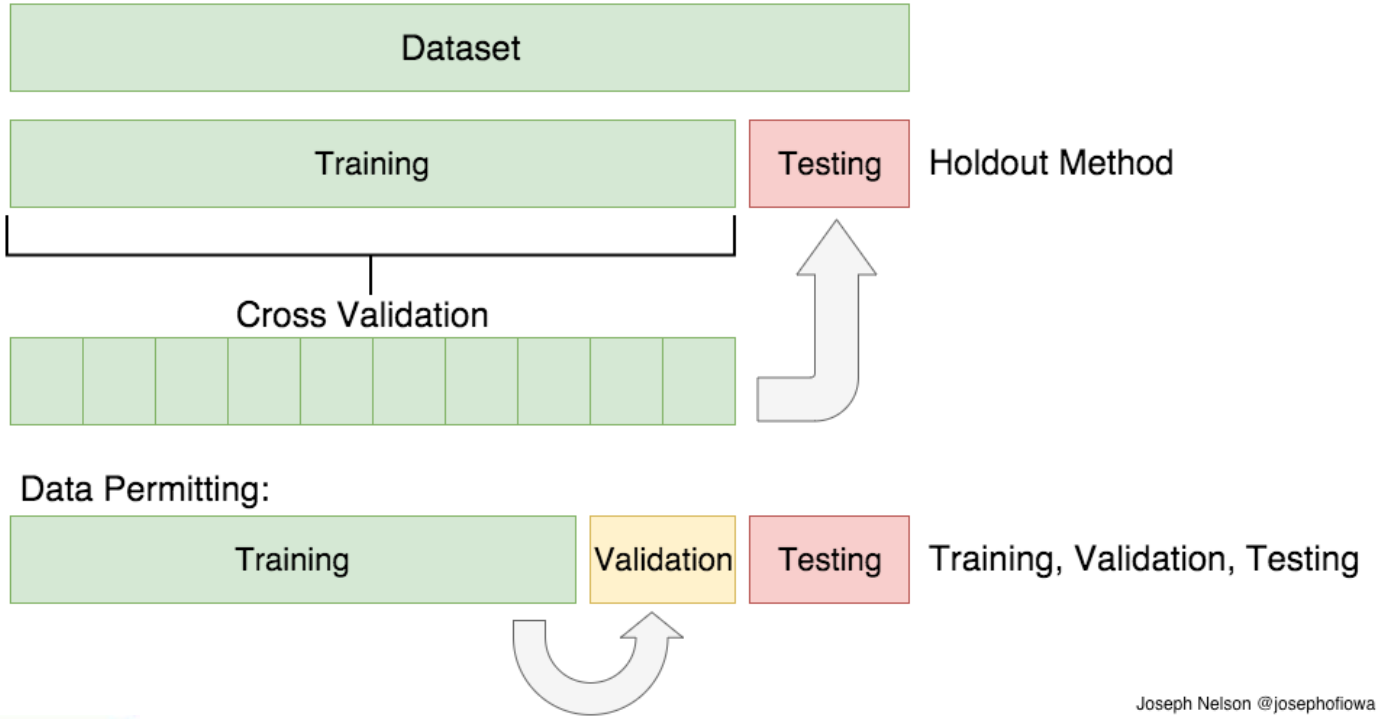
Expedia Kaggle Competitions in the past

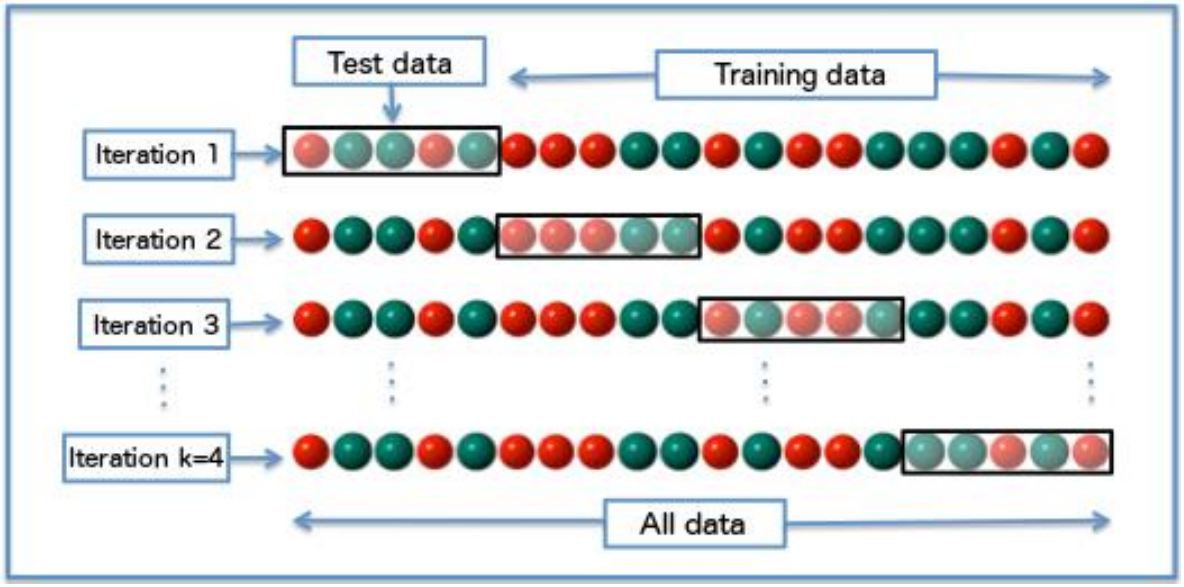
- https://www.kaggle.com/c/expedia-hotel-recommendations
- https://www.kaggle.com/c/expedia-personalized-sort/data

Create environment

- https://conda.io/docs/user-guide/tasks/manage-environments.html

Test/Train/Cross validation





Accuracy profiling

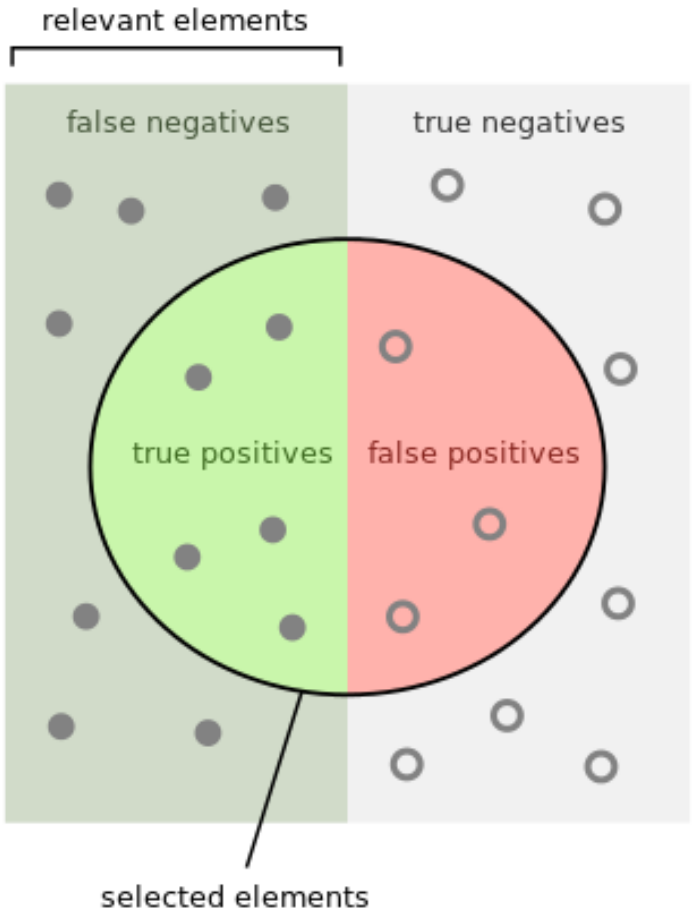
Predictive Model: Evaluation

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

predictive result / classification		actual result / classification: yes	actual result / classification: no
	yes	**tp** (true positive)	**fp** (false positive) — Type 1 error
	no	**fn** (false negative)	**tn** (true negative)

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Recall} = \frac{tp}{tp + fn}$$

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \qquad \text{True Negative Rate} = \frac{tn}{tn + fp}$$



How many selected items are relevant?

Precision =

How many relevant items are selected?

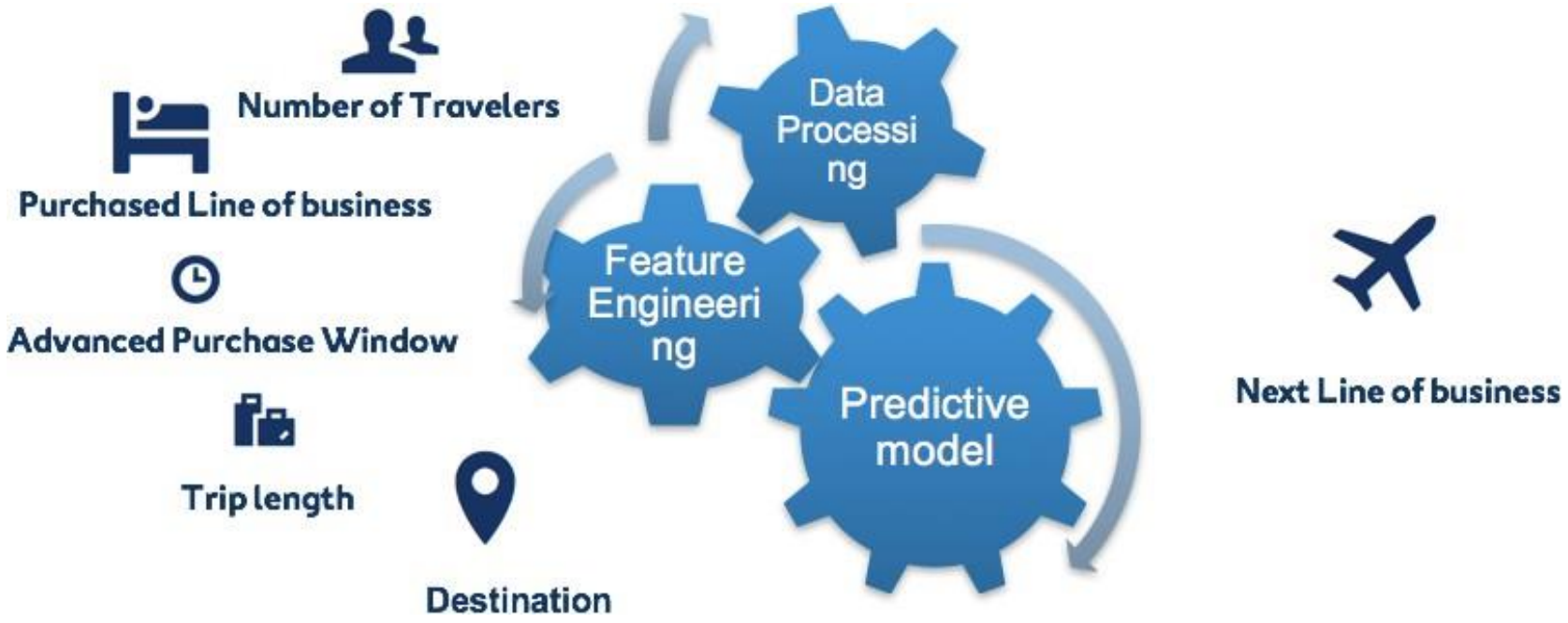
Recall =

Hands on time – One problem

- Krazy glue

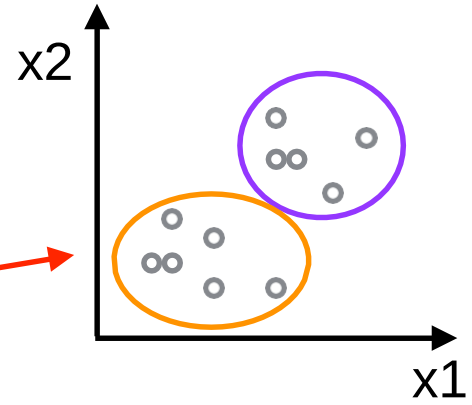
Problem Statement

Predict what LOB a customer will purchase next

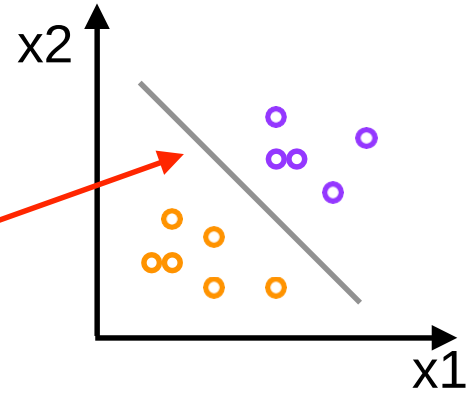


Approach

- Unsupervised learning
  - Data is unlabeled
  - Induce a clustering



- **Supervised learning**
  - **Data is labeled**
  - **Learn a decision boundary**



Data

Name	Description
date_time	Date-time of transaction
site_name	ID of the Expedia point-of-sale (i.e., expedia.com, expedia.co.uk, expedia.co.jp, …)
booking_platform	Desktop, tablet, mobile
booking_start_date booking_end_date	Start and end dates for the trip
user_id	ID of user
user_location_country	ID of the country where customer located. Derived from IP.
user_location_region	ID of the region where customer located. Derived from IP.
user_location_city	ID of the city where customer located. Derived from IP.
origin_destination_distance	Physical distance between a destination customer's location at the time of search.
line_of_business	Line of business booked (Hotel, Car, Flight, etc.)

Purchase Histories: Attach Definition

Purchased within 12mo

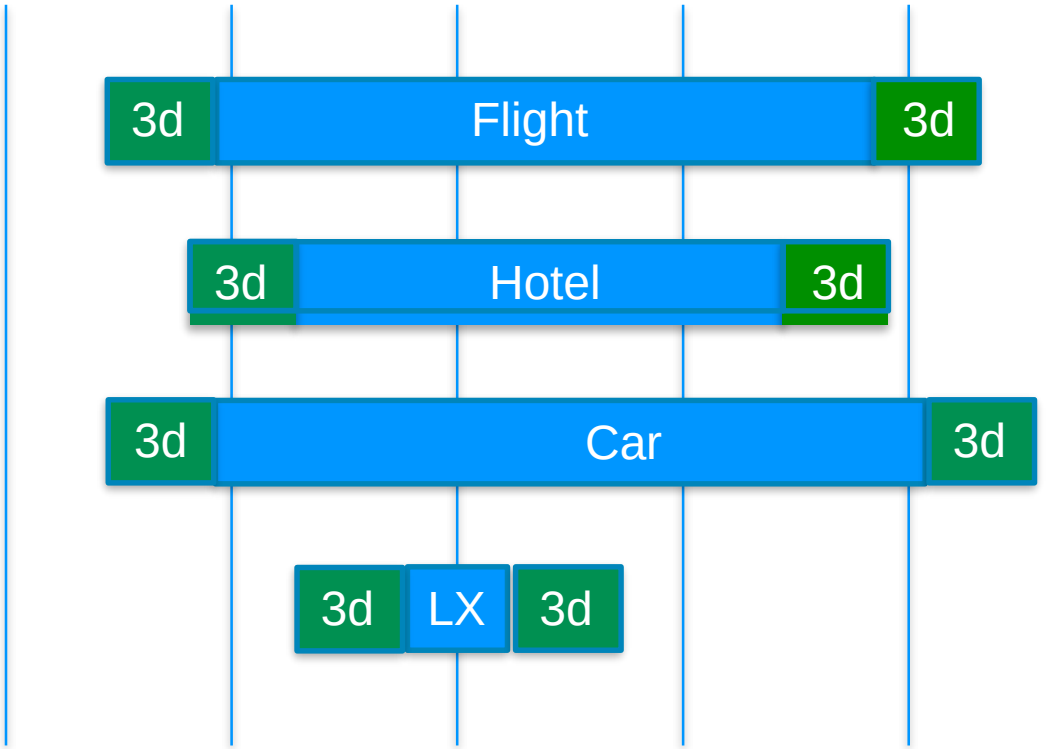
Unique email address

expedia_user@gmail.com

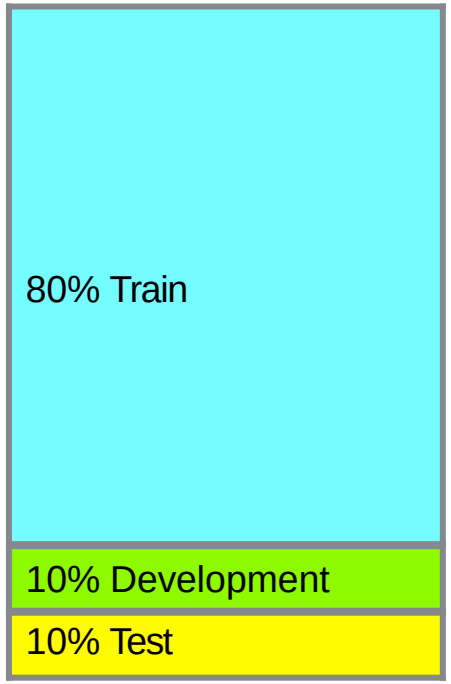
expedia_user@gmail.com

expedia_user@gmail.com

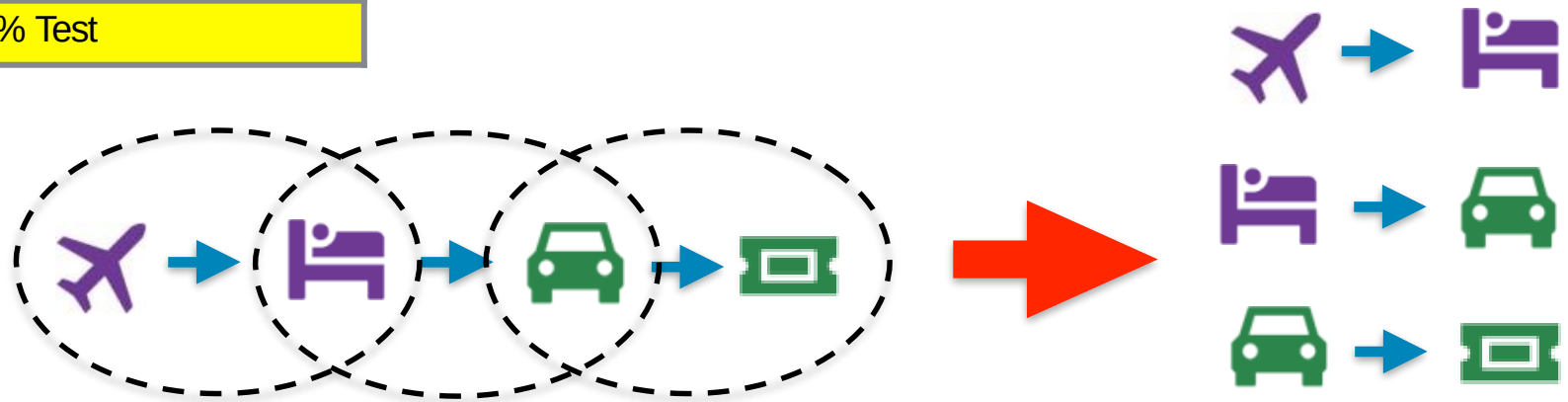
expedia_user@gmail.com



Data Sets



	Train	Dev	Test
Krazy Machine	16M purchase pairs	2M purchase pairs	2M purchase pairs



Features

	Feature	Description	Status
Current purchase	**Current LOB**	What did the user just purchase?	✔
	Advanced purchase window	# days until start date of current LOB	✔
	Trip length	Duration of current LOB purchase	✔
	Trip spans Saturday?	Current LOB purchase spans a Sat?	✔
	International purchase	Current LOB purchase is international?	✔
	# of travelers, adults, kids	#/type of travelers for current LOB purchase	
	Origin city, state, region, …	Origin-location features for current LOB	
	Dest city, state, region, …	Dest-location features for current LOB	
	Platform type	Desktop, mobile, tablet for current LOB	
	Logged-in user?	User logged in when purchased curr LOB?	✗
User history	**Reward member status**	User's reward member status	✗
	Search history	User's tendency to make purchasing decisions	✗
	Time since last purchase	How long has it been since user made last purchase?	✗
	Attach history	What has the user already purchased for this trip?	✗
	Package-specific	Is the package hotel+flight, hotel+flight+car, …?	✗

Experimental Setup

- Purchase histories
  1. Chronological
  2. Attach definition
- Baseline
  - For each input, pick most popular next purchase
  - Use historical data to calculate most popular
- Features
  - Advanced purchase window
  - Trip length
  - Weekend trip?
  - International purchase?
- Metrics
  - F1 (harmonic mean of precision and recall)

Results on Development Sets

	Model	F1 Score	
1	Chronological, Baseline	73.7%	
2	Chronological, All Features	74.8%	← 1-pt gain
3	Attach, Baseline	73.3%	
4	Attach, All Features	73.5%	← no gain
5	Attach, Baseline	69.0%	← New dataset! *
6	Attach, All Features	?	

** Rerunning all classifiers on the same dev set for comparability.*

https://confluence/display/POS/Krazymachine+-+Machine+learning+in+krazyglue

Microsoft Azure Machine Learning: Algorithm Cheat Sheet

This cheat sheet helps you choose the best Azure Machine Learning Studio algorithm for your predictive analytics solution. Your decision is driven by both the nature of your data and the question you're trying to answer.

START

Finding unusual data points

Discovering structure

Predicting categories

Predicting values

Three or more

Two

ANOMALY DETECTION

- **One-class SVM** — >100 features, aggressive boundary
- **PCA-based anomaly detection** — Fast training

CLUSTERING

- **K-means**

MULTI-CLASS CLASSIFICATION

- Fast training, linear model — **Multiclass logistic regression**
- Accuracy, long training times — **Multiclass neural network**
- Accuracy, fast training — **Multiclass decision forest**
- Accuracy, small memory footprint — **Multiclass decision jungle**
- Depends on the two-class classifier, see notes below — **One-v-all multiclass**

REGRESSION

- **Ordinal regression** — Data in rank ordered categories
- **Poisson regression** — Predicting event counts
- **Fast forest quantile regression** — Predicting a distribution
- **Linear regression** — Fast training, linear model
- **Bayesian linear regression** — Linear model, small data sets
- **Neural network regression** — Accuracy, long training time
- **Decision forest regression** — Accuracy, fast training
- **Boosted decision tree regression** — Accuracy, fast training, large memory footprint

TWO-CLASS CLASSIFICATION

- **Two-class SVM** — >100 features, linear model
- **Two-class averaged perceptron** — Fast training, linear model
- **Two-class logistic regression** — Fast training, linear model
- **Two-class Bayes point machine** — Fast training, linear model

- Accuracy, fast training — **Two-class decision forest**
- Accuracy, fast training, large memory footprint — **Two-class boosted decision tree**
- Accuracy, small memory footprint — **Two-class decision jungle**
- >100 features — **Two-class locally deep SVM**
- Accuracy, long training times — **Two-class neural network**