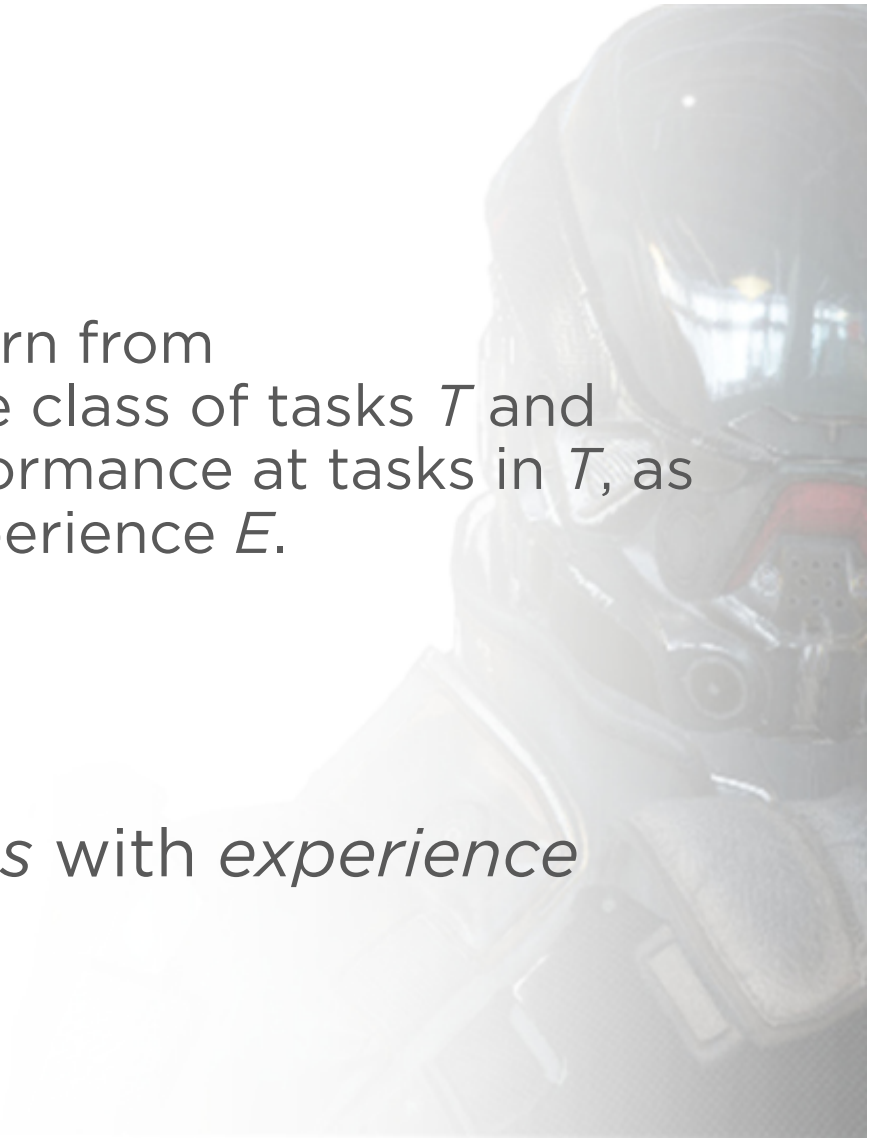
What Is Machine Learning?

Definition

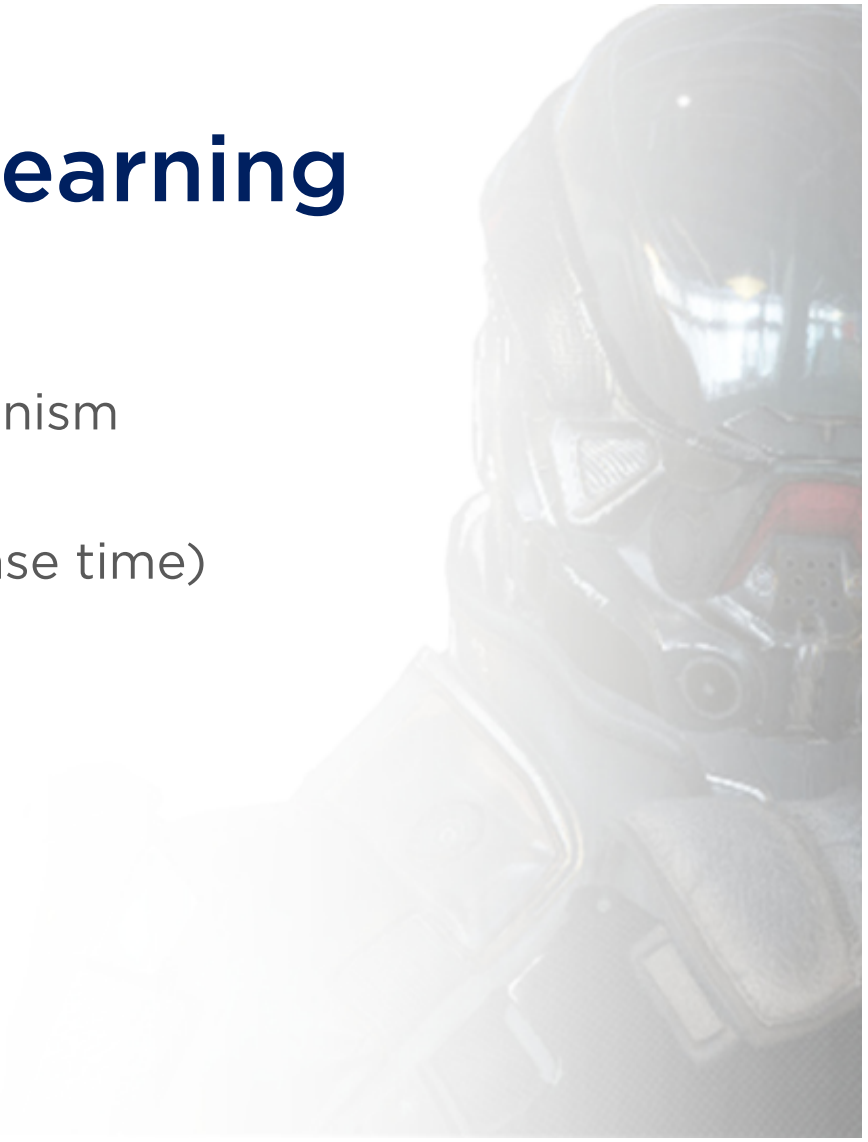
A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P* if its performance at tasks in *T*, as measured by *P*, improves with experience *E*.
– *Tom Mitchell*

A *program* learns if it *improves* with *experience*



Identifying Machine Learning

- Program
  - Some kind of computational mechanism
- Improvement
  - Any metric (e.g. accuracy or response time)
- Experience
  - What happens at runtime

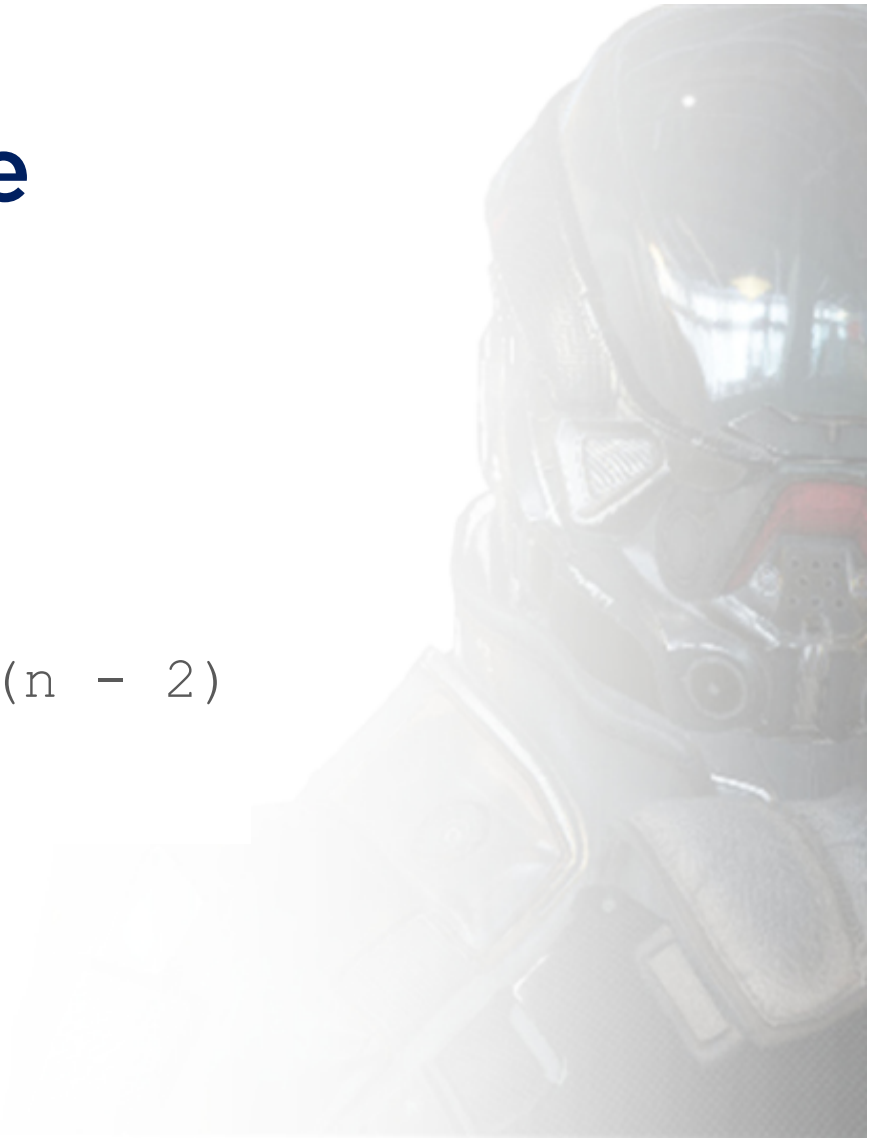


Case Study: Memoization

Store the result of an expensive operation

Memoization Example

```
define fib n
  if memo[n] return memo[n]
  if n = 0 return 0
  if n = 1 return 1
  memo[n] = fib(n - 1) + fib(n - 2)
  return memo[n]
```

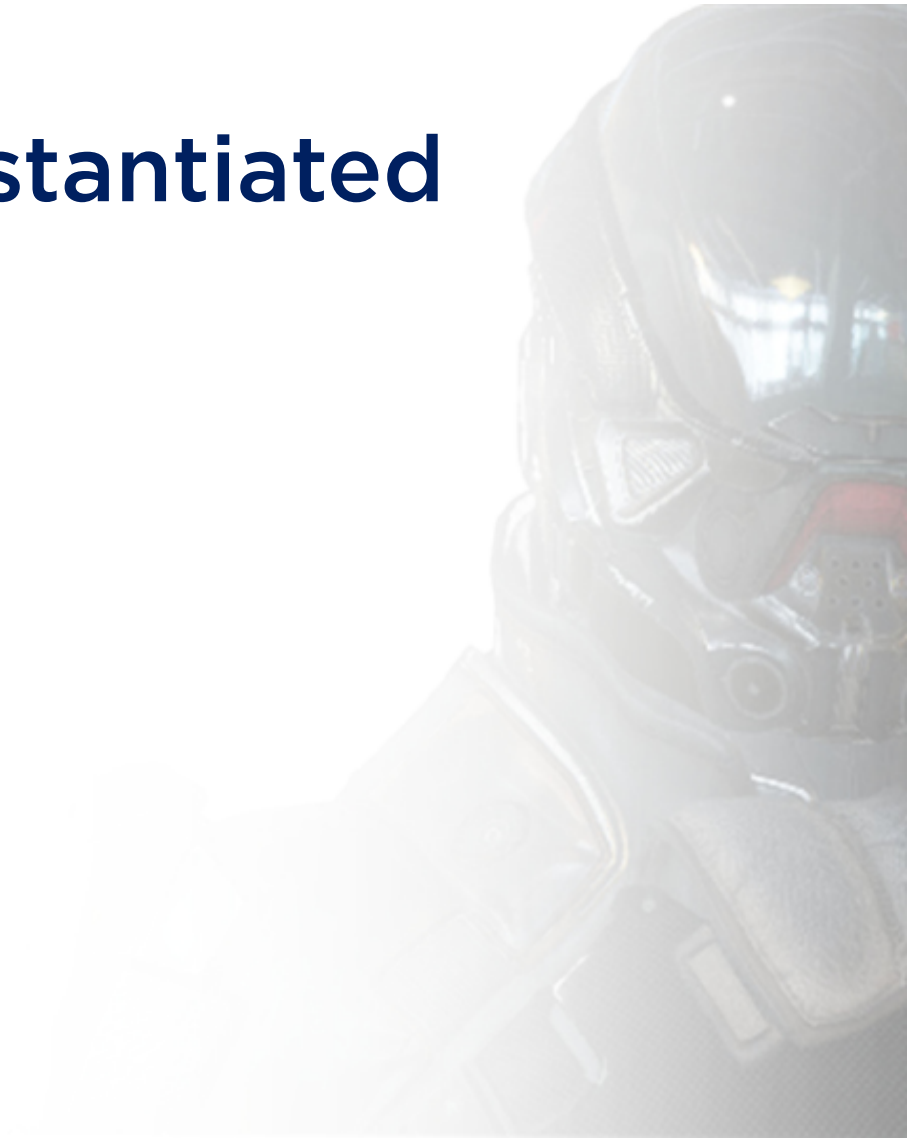


Case Study: Memoization

- Program
  - Fibonacci function
- Improvement
  - Execution time
- Experience
  - Calls to `fib`

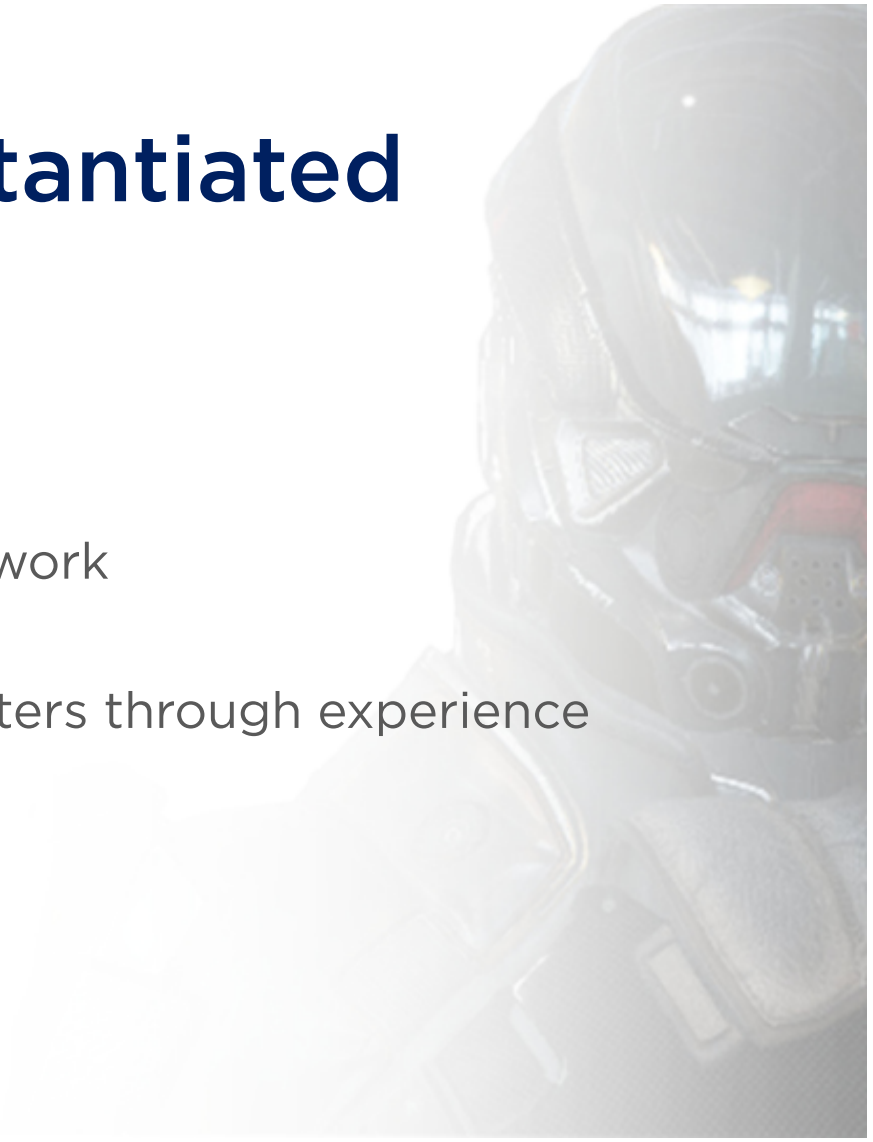
Machine Learning Instantiated

- Models
- Parameters
- Optimization



Machine Learning Instantiated

- Models
  - A static computational framework
- Parameters
  - Dynamic components of the framework
- Optimization
  - A mechanism for changing parameters through experience



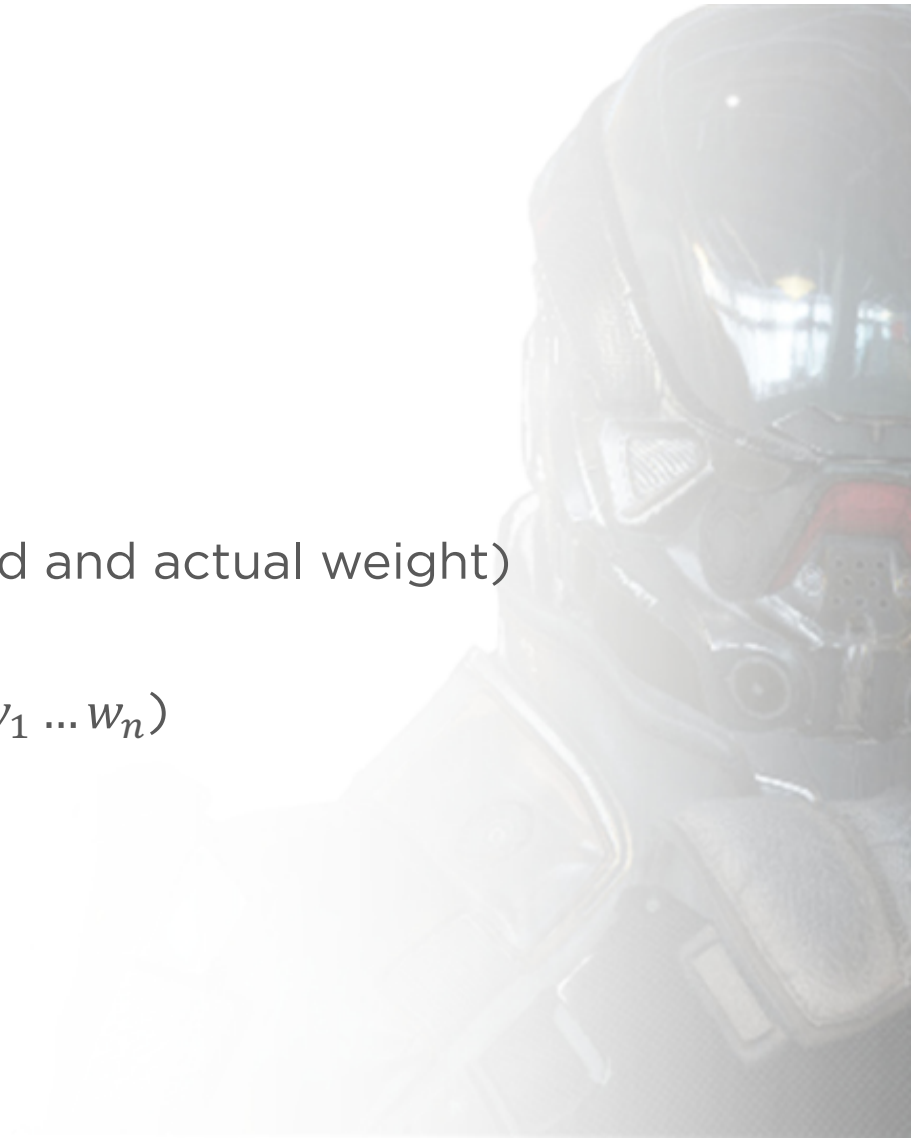
Case Study: Memoization

- Model
  - Hash table
- Parameters
  - Keys and values
- Optimization

Case Study: Guess Your Weight

Guess Your Weight

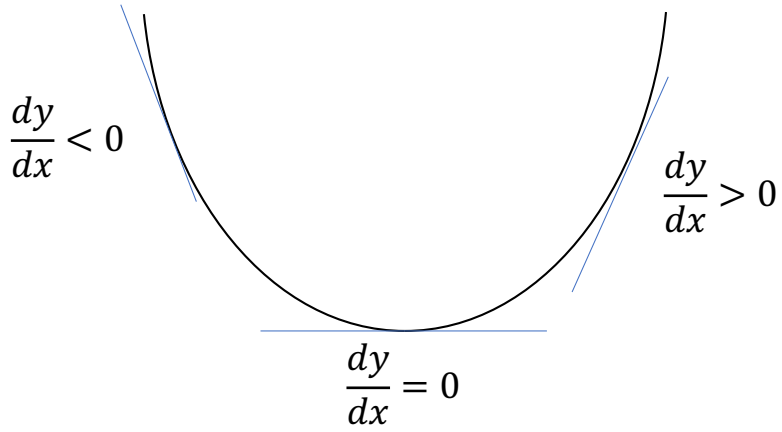
- Program
  - Return guessed weight
- Improvement
  - Error (difference between guessed and actual weight)
- Experience
  - Weights of previous customers ($w_1 \ldots w_n$)



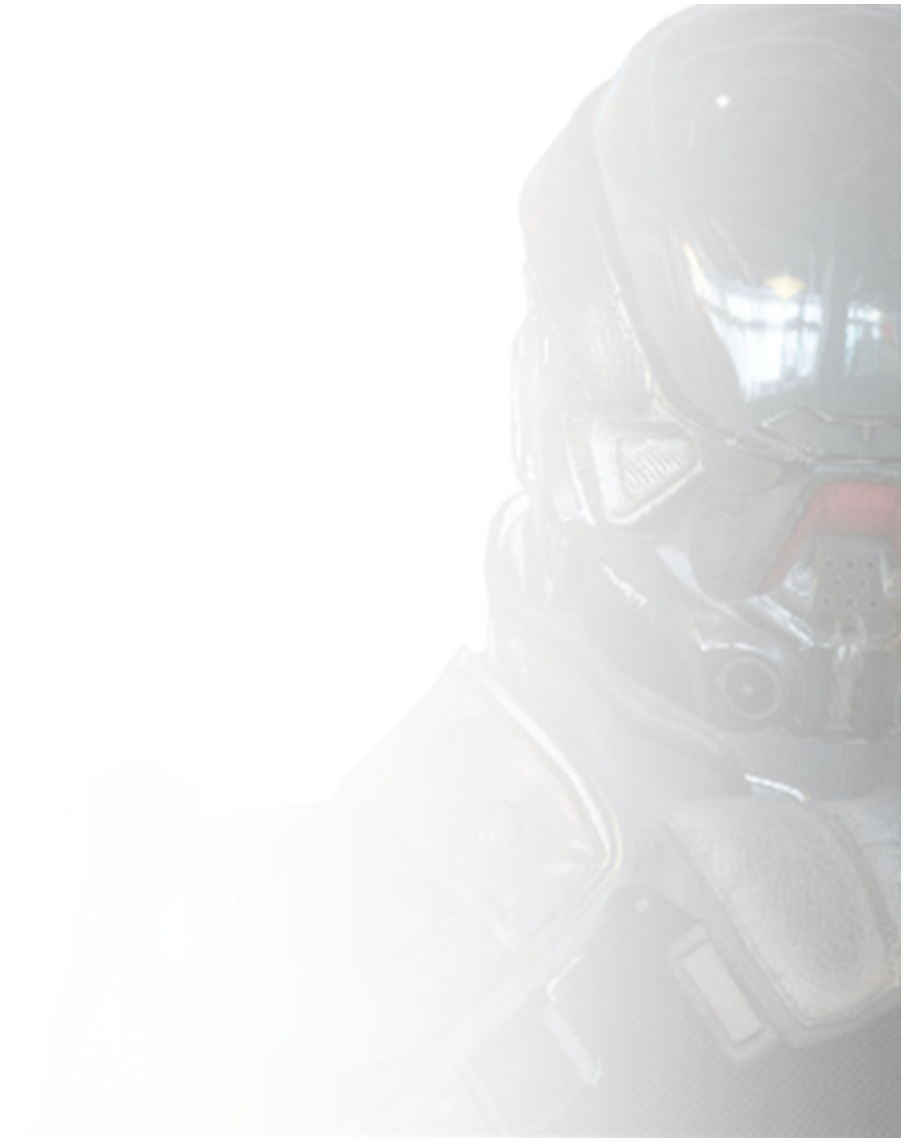
Guess Your Weight

- Model
  - The guessed weight, w^*
- Parameters
  - w^*
- Optimization
  - Find w^* that minimizes $\mathrm{E} = \sum_{i=1}^{n}(w_i - w^*)^2$

Calculus



Slope is zero at minimum



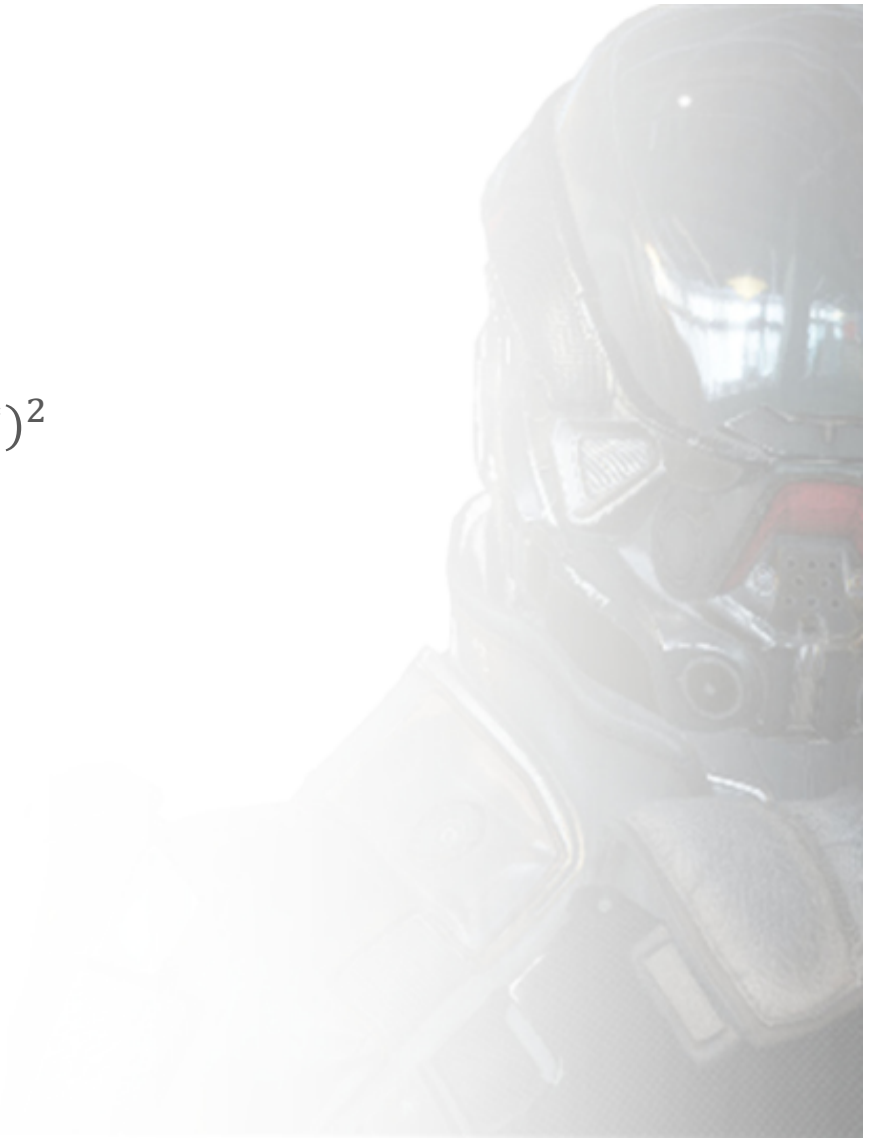
Guess Your Weight

Optimization

- Find w^* that minimizes $\mathrm{E} = \sum_{i=1}^{n}(w_i - w^*)^2$

$$0 = \frac{dE}{dw^*} = -2\sum_{i=1}^{n}(w_i - w^*)$$

$$w^* = \frac{1}{n}\sum_{i=1}^{n} w_i$$



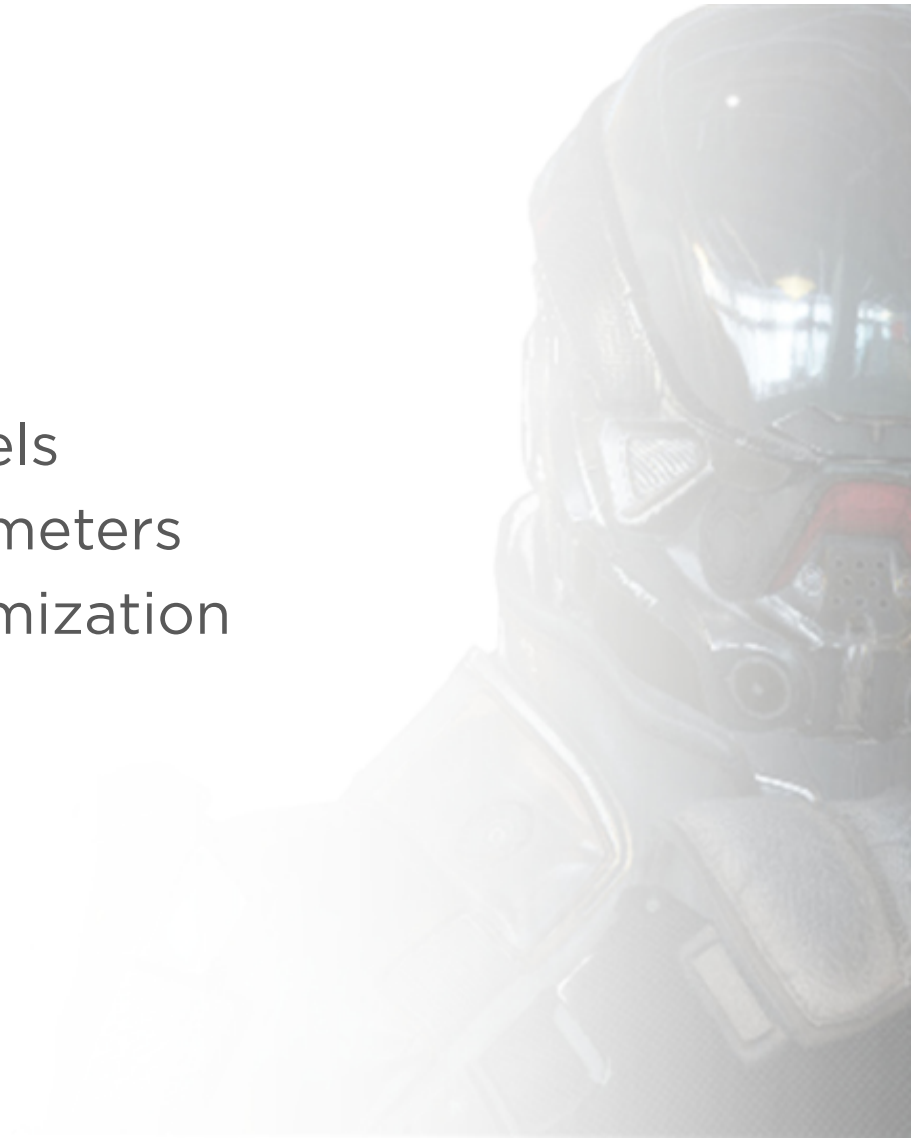
What is ML?

What

- Program
- Improvement
- Experience

How

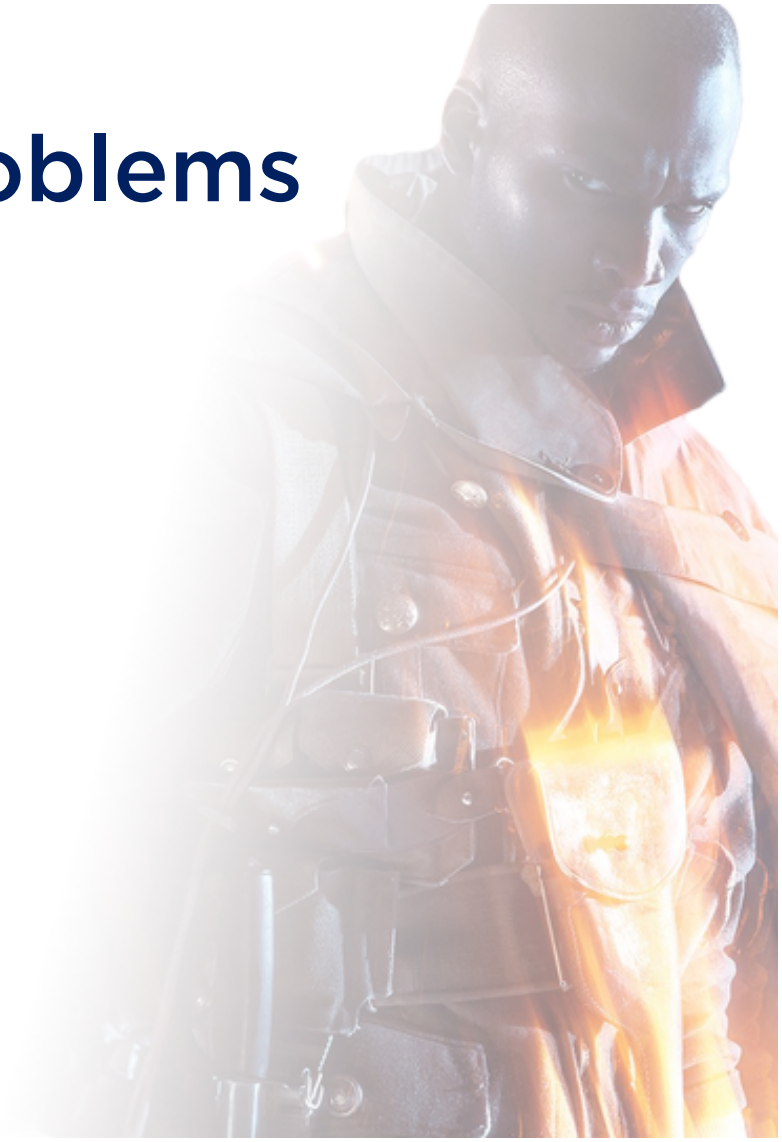
- Models
- Parameters
- Optimization



Types of Machine Learning Problems

Key Concerns of ML Problems

- Problem type
- Knowledge sources

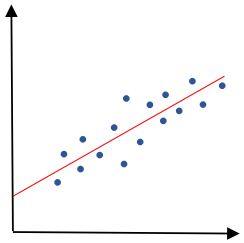


Three Problem Types

- Estimation
  - Tomorrow's temperature
- Labels
  - Cloudy, or not
- Neighborhoods
  - Days like yesterday

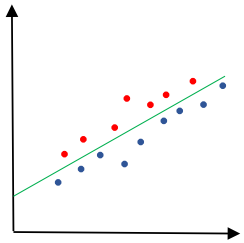
Three Problem Types

Estimation



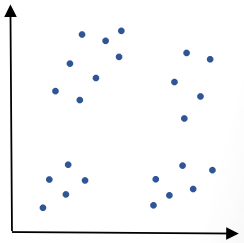
Regression

Labels



Classification

Neighborhoods



Clustering

Knowledge Sources

What do we know?

- Features

How much do we know?

- Everything
- Something
- Nothing

Supervised Learning

- We know everything
  - Values (regression)
  - Labels (classification)

- Regression
- Classification

- Error minimization

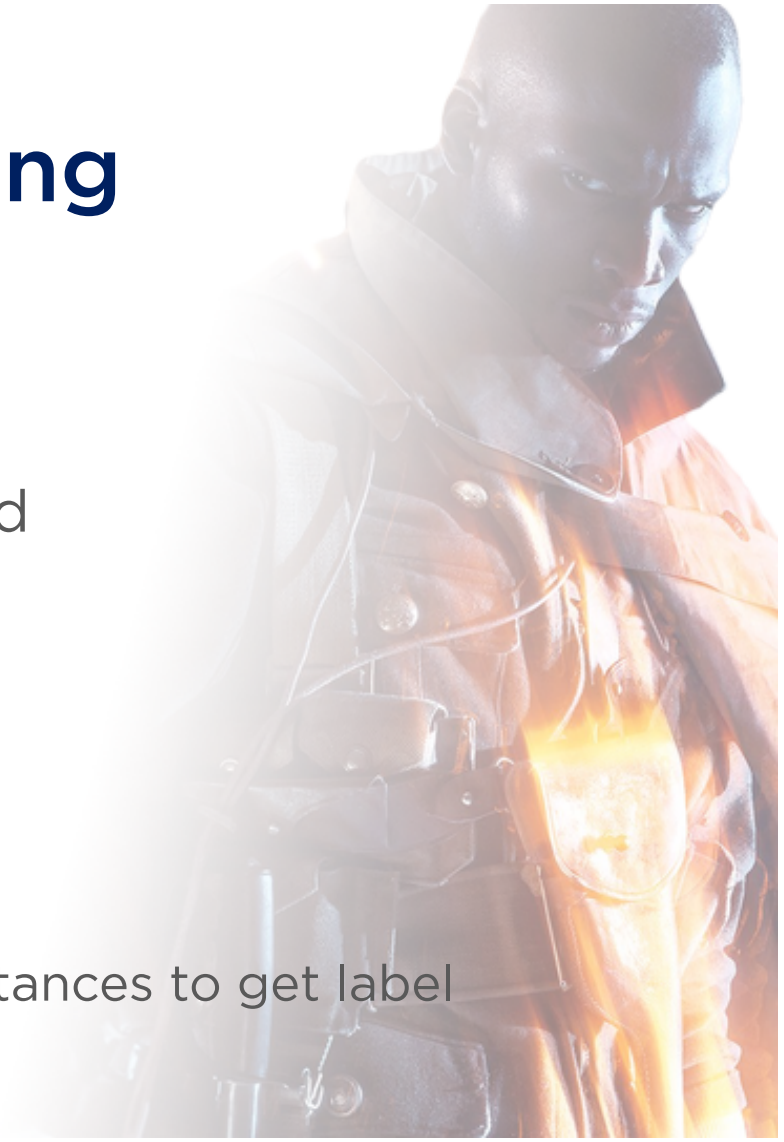
Unsupervised Learning

- We know nothing

- Clustering
- Error correction

Semi-supervised Learning

- We know something
- Labels are expensive
- Some labeled data, mostly unlabeled



- Inductive learning
  - Learn global labeling function
- Transductive learning
  - Map unlabeled instances to labeled instances to get label

Summary

Problem Types
- Regression (Estimate)
- Classification (Label)
- Clustering (Neighbors)

Knowledge Sources
- Supervised
- Unsupervised
- Semi-supervised

Good Old Fashioned Machine Learning

Trees and Workflows

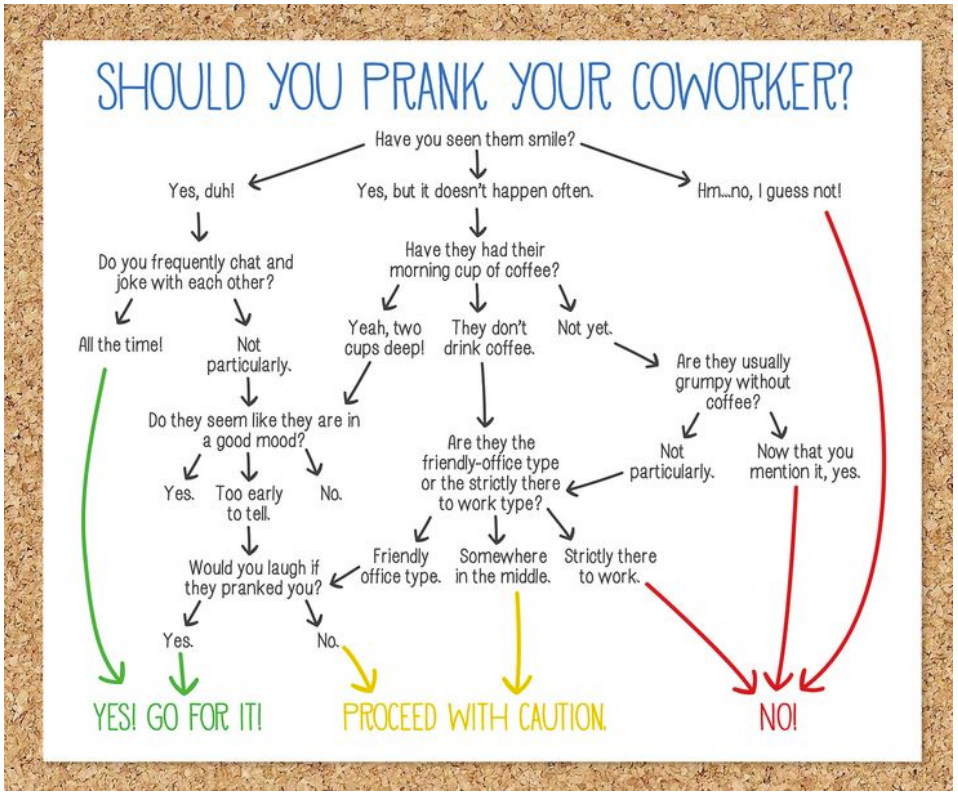
- Decision Trees
- Random Forests
- ML Workflow
- Spark

Decision Trees

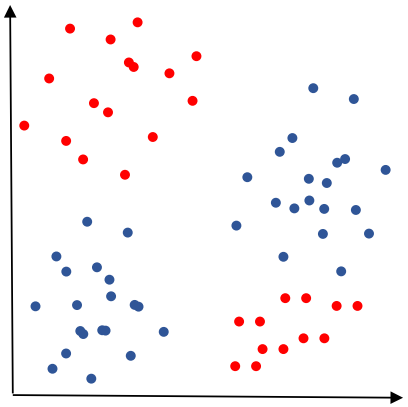
Decision Trees

Ask the right questions
and you shall receive
the right answer

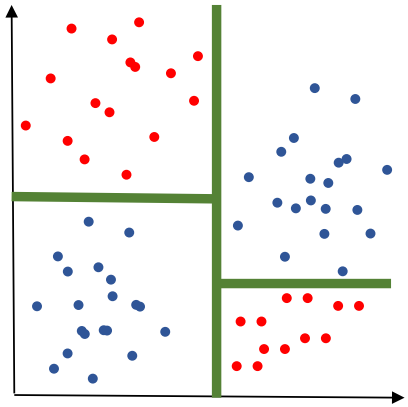
A Useful Decision Tree

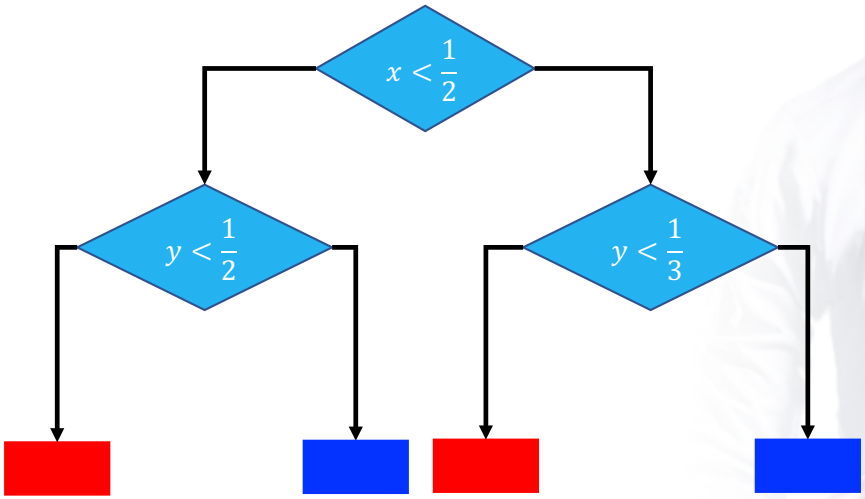


Decision Tree Classification



Decision Tree Classification





Learning Decision Trees

- Given a data set
- Find an optimal decision criteria
  - Maximize information gain
- Split training data by decision criteria
- Recursively learn subtrees

Decision Tree Pros & Cons

Pros
- Simple to understand
- Handles numerical and categorical data
- Scales to large data sets

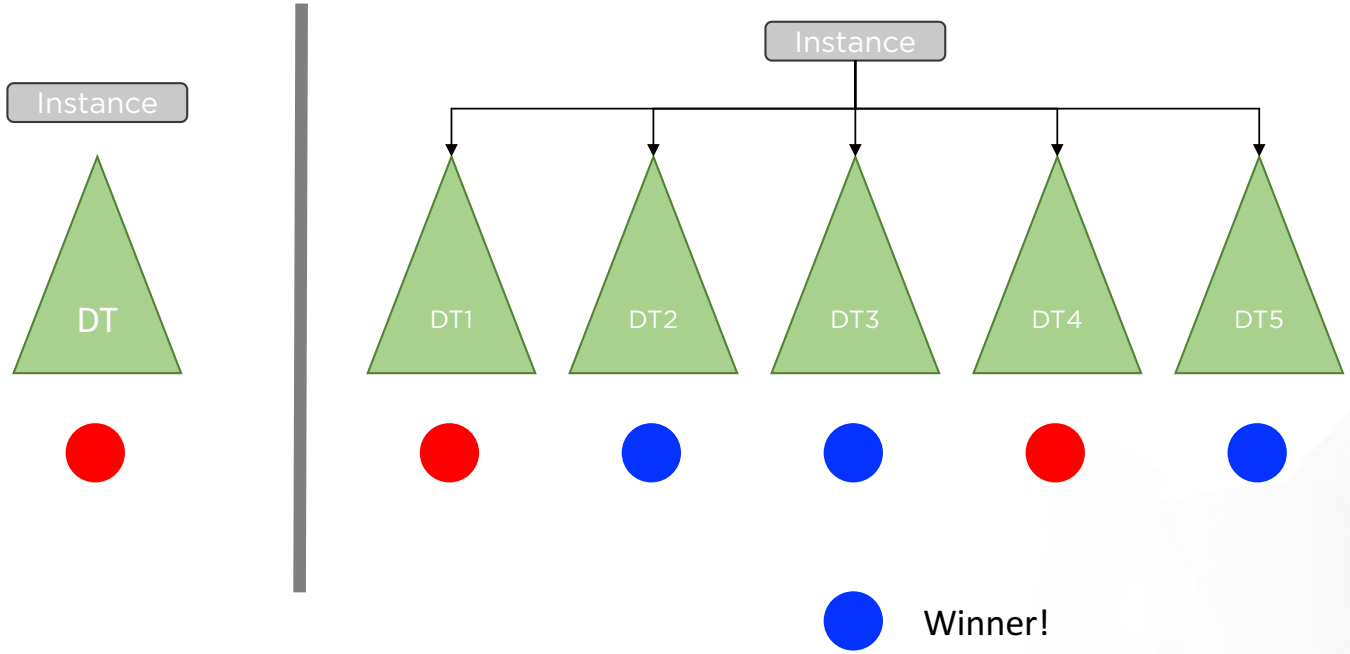
Cons
- Not as accurate as other approaches
- Not robust to data set changes
- Overfit data set with complex trees

Random Forests

Random Forest

If one tree is good,
a forest is better

Majority Rules in the Forest



Many Trees, One Data Set

How do you get multiple trees from the same data set?

- Train on subsets of data
- Train on subsets of features

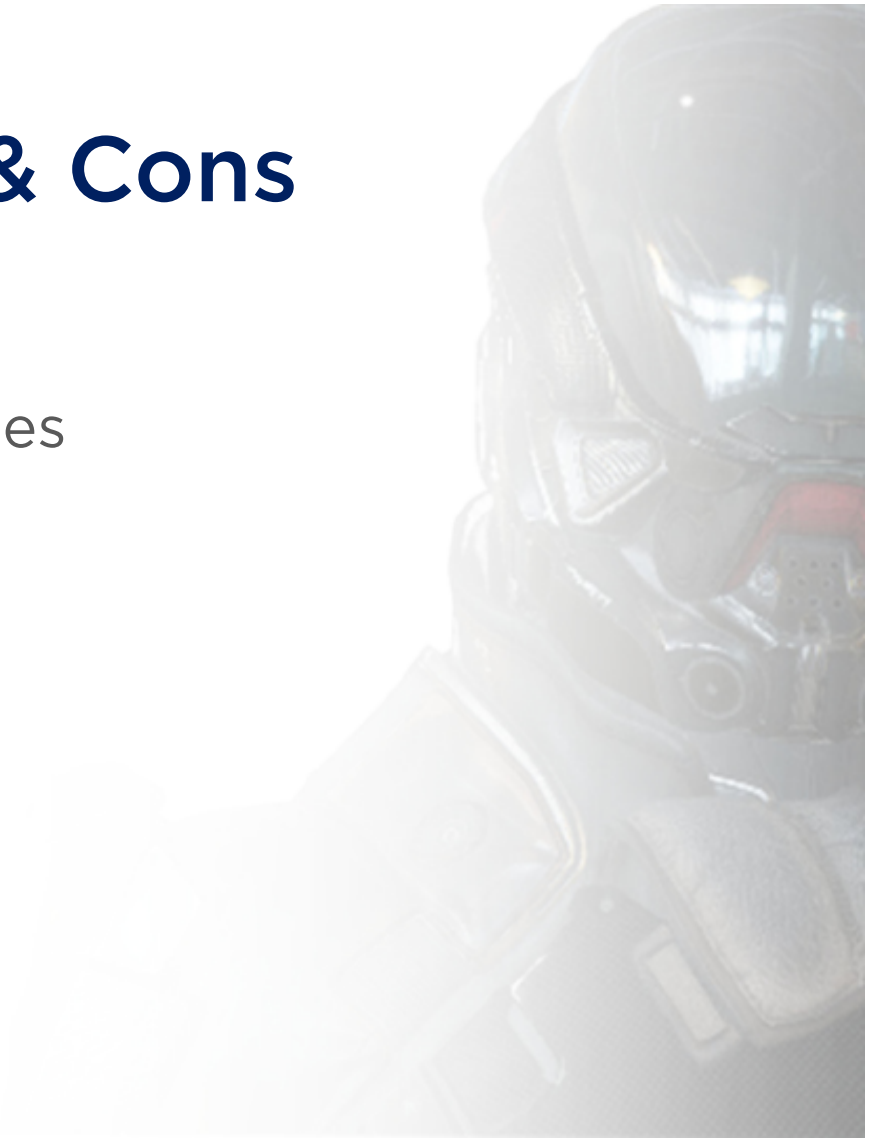
Random Forest Pros & Cons

Pros

- Feature bagging decorrelates trees
- Reduced variance

Cons

- Not easy to interpret visually



Working With Machine Learning

The ML Workflow

- Data preparation
- Model selection
- Training
- Evaluation
- Deployment

Data Preparation

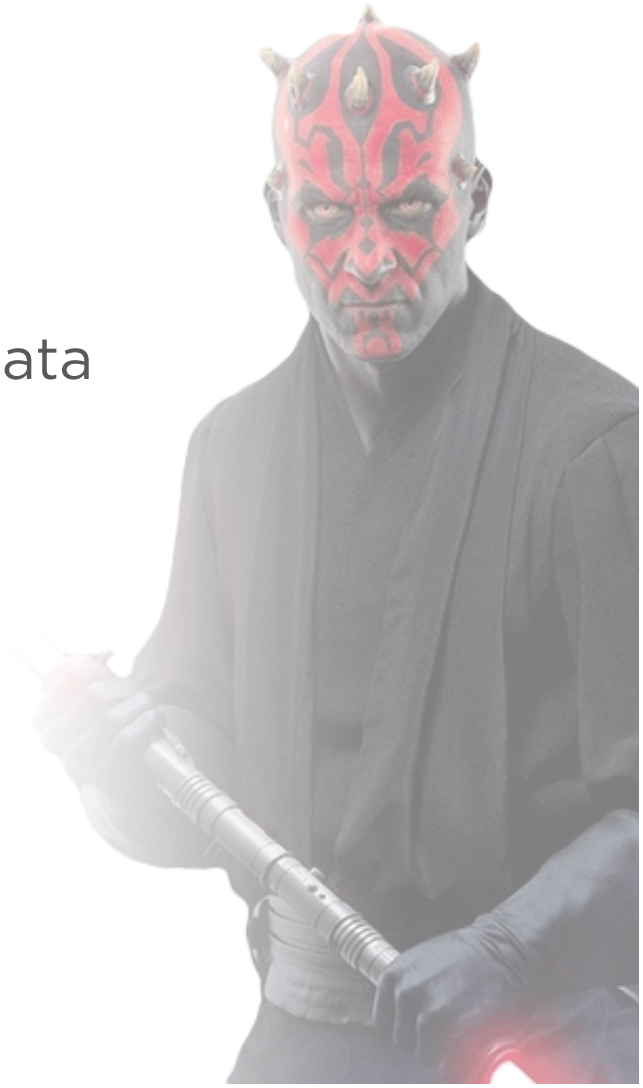
- Acquisition
- Quality Assurance
- Feature Selection
- Transformations

Acquiring Data

Many sources of data

- Player Insight Network (PIN) telemetry data
- Internal non-game sources
- Third-party sources
- ...

Getting data is the easy part

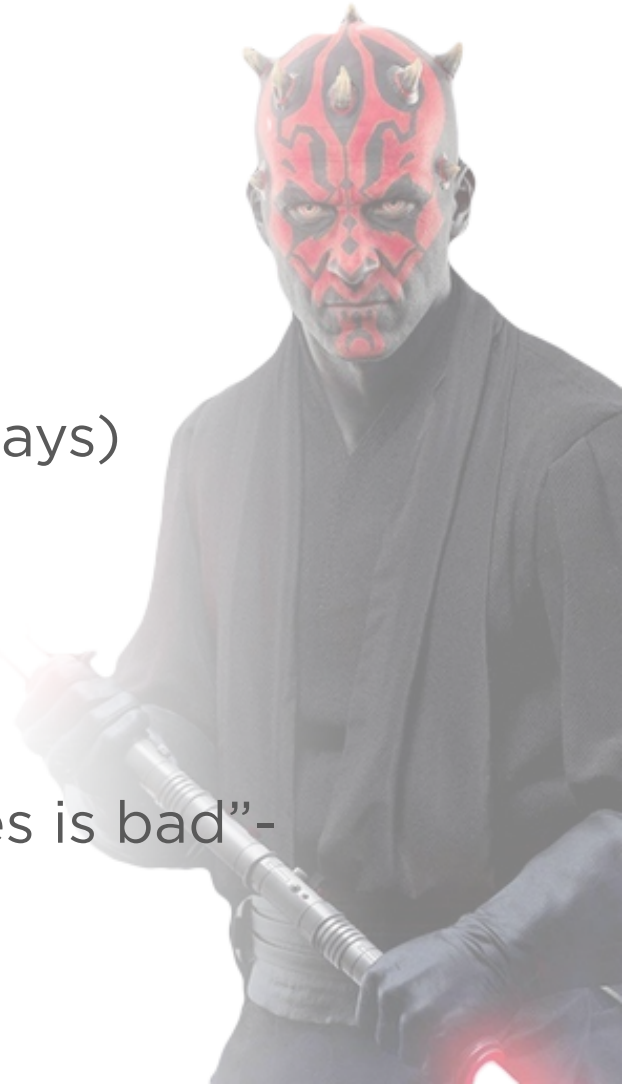


Quality Assurance

How bad can it be?

- Missing fields
- Out of range (e.g. 2,147,483,647 session days)
- Wrong field (e.g. player id “30 crystals”)
- Name/semantics mismatch
- ...

”At least 20% of all data in data warehouses is bad”- Michael Stonebraker



Feature Selection

Use features that make sense together

Name	A	B	C	D	E
Amy	2	9	18	4	13
Bob	2	8	16	8	16
Cat	2	3	6	4	7
Don	2	4	8	5	9
Ema	2	7	14	9	16

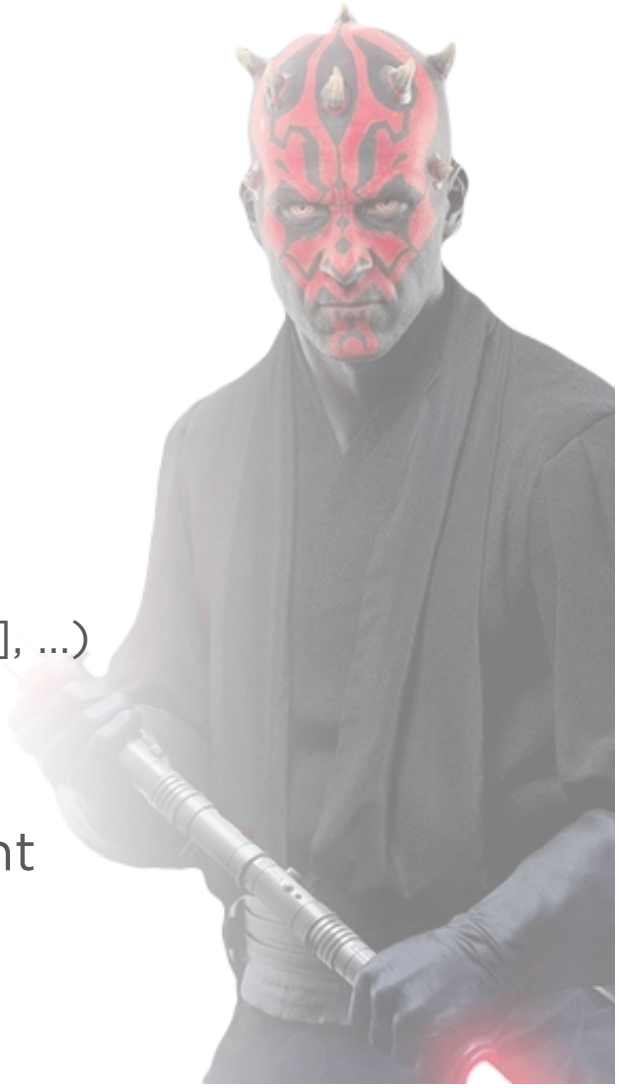
Transformations

Make it easier to find relevant abstractions

- Log space (order of magnitude is important)
- Standardization (z-score = $\frac{x-\bar{x}}{\sigma}$)
- Categories to numerics
  - Scalar (A=4, B=3, C=2, D=1, F=0)
  - One-hot (A=[1,0,0,0,0], B=[0,1,0,0,0], C=[0,0,1,0,0], …)
- …

Transformations as neighborhood management

- It's easier to treat similar things similarly



Data Preparation

- Acquisition
- Quality Assurance
- Feature Selection
- Transformations

Model Selection

Pick a model, any model

Supervised Learning Algorithms

AODE
Association rule learning algorithms
Apriori algorithm
Eclat algorithm
Case-based reasoning
Gaussian process regression
Gene expression programming
Group method of data handling (GMDH)
Inductive logic programming
Instance-based learning
Lazy learning
Learning Automata
Learning Vector Quantization
Logistic Model Tree
Minimum message length (decision trees, decision graphs, etc.)
Nearest Neighbor Algorithm
Analogical modeling
Probably approximately correct learning (PAC) learning
Ripple down rules, a knowledge acquisition methodology
Symbolic machine learning algorithms
Support vector machines
Random Forests
Ensembles of classifiers
Bootstrap aggregating (bagging)
Boosting (meta-algorithm)
Ordinal classification
Information fuzzy networks (IFN)
Conditional Random Field
ANOVA
Quadratic classifiers
k-nearest neighbor
Boosting
SPRINT

Bayesian networks
Naive Bayes
Hidden Markov models
Artificial neural network[edit]
Artificial neural network
Autoencoder
Backpropagation
Boltzmann machine
Convolutional neural network
Deep learning
Hopfield network
Multilayer perceptron
Perceptron
Radial basis function network (RBFN)
Restricted Boltzmann machine
Recurrent neural network (RNN)
Self-organizing map (SOM)
Spiking neural network
Bayesian[edit]
Bayesian statistics
Bayesian knowledge base
Naive Bayes
Gaussian Naive Bayes
Multinomial Naive Bayes
Averaged One-Dependence Estimators (AODE)

Bayesian Belief Network (BBN)
Bayesian Network (BN)
Decision tree[edit]
Decision tree algorithm
Classification and regression tree (CART)
Iterative Dichotomiser 3 (ID3)
C4.5 algorithm
C5.0 algorithm
Chi-squared Automatic Interaction Detection (CHAID)
Decision stump
Conditional decision tree
ID3 algorithm
Random forest
SLIQ
Linear classifier[edit]
Linear classifier
Fisher's linear discriminant
Linear regression
Logistic regression
Multinomial logistic regression
Naive Bayes classifier
Perceptron
Support vector machine

Training

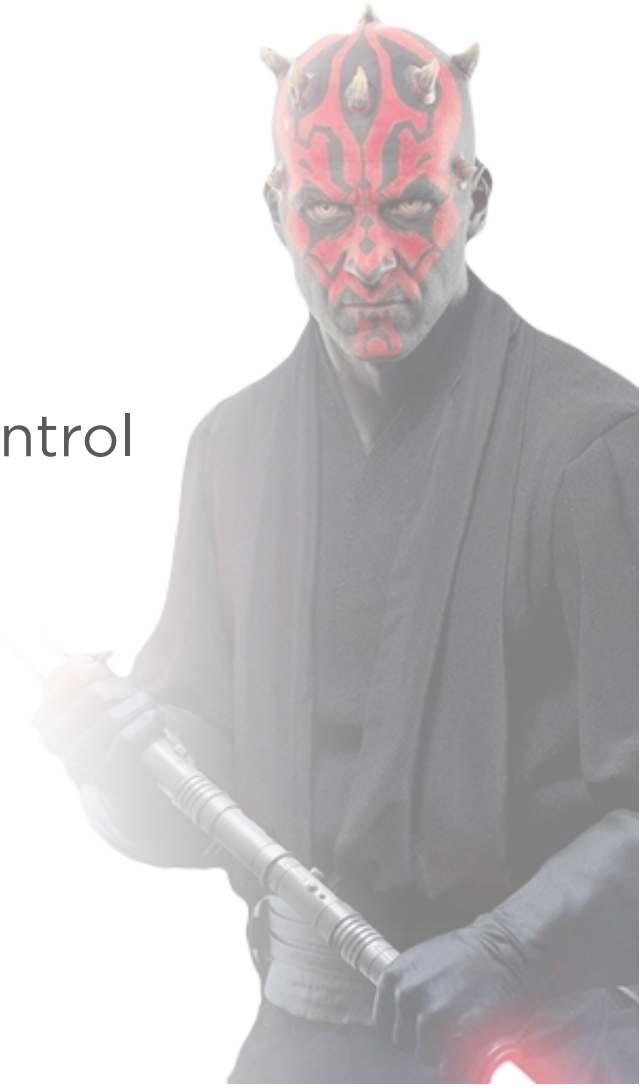
Data + Algorithm + Time = Model

Training

If it was only so easy...

Many algorithms require parameters to control optimization

- Max tree depth
- Sample size
- Learning rate
- Momentum
- ...



The Cursed Loop

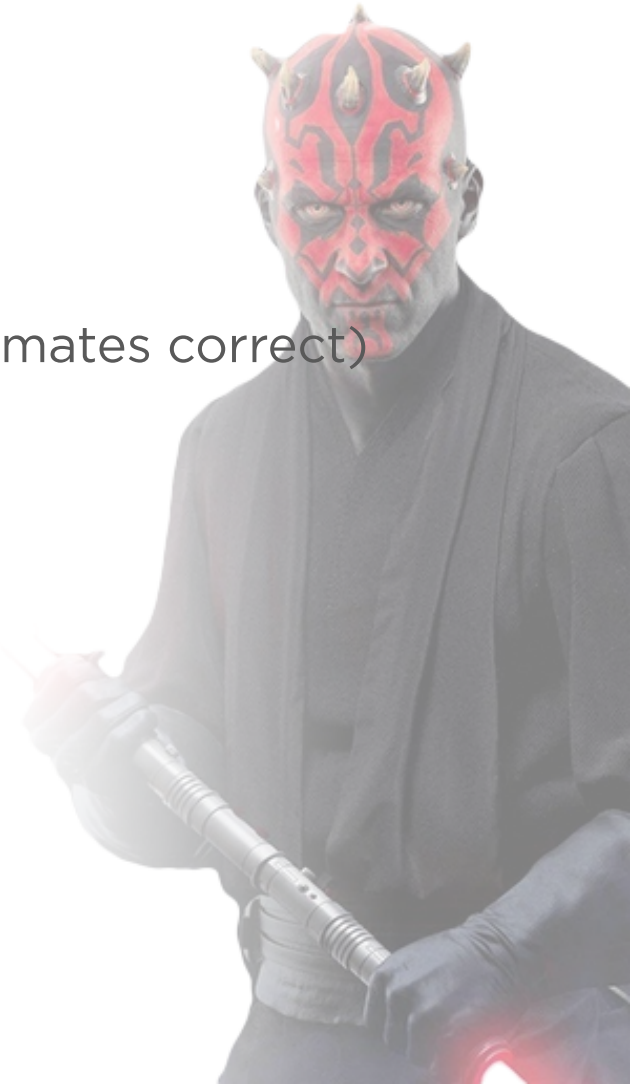
```
do
   select p from possible hyperparameters
   model = train(data, p)
until model is acceptable
```

Evaluation

- Metrics
- Overfitting and underfitting
- Estimators, Bias and Variance
- Cross Validation

Metrics

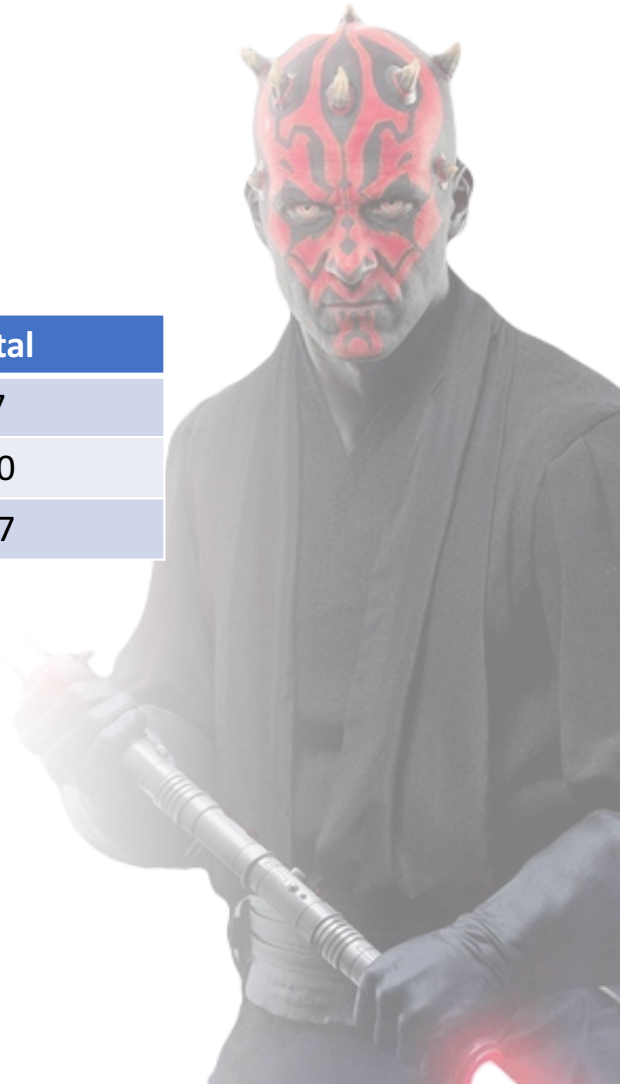
- Accuracy
  - Ratio of good classifications (e.g. 70% of estimates correct)
- Logarithmic Loss (logloss)
- Area Under ROC Curve (AUC)
- Confusion matrix
- Mean absolute error
- Mean square error
- R^2
- ...



Case Study: Cat Detector

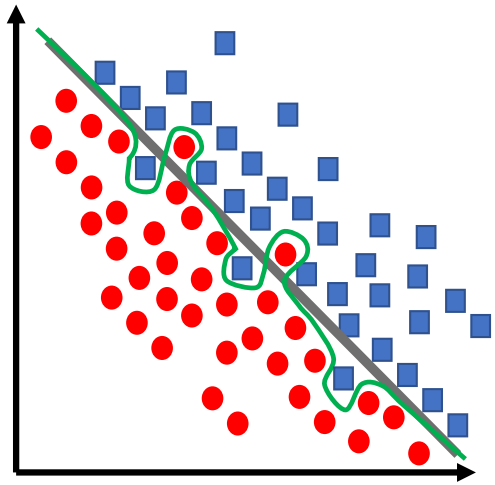
Actual Class

Predicted Class	Cat	Not-Cat	Total
Cat	5	2	7
Not-Cat	3	17	20
Total	8	19	27



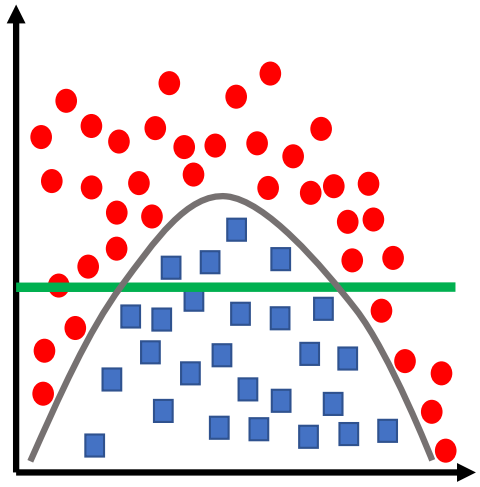
Accuracy = (5 + 17) / 27
Error Rate = (3 + 2) / 27
False Positives = 2
False Negatives = 3
Sensitivity (Recall) = 5 / 8
Precision = 5 / 7

Overfitting



Model memorizes at the cost of generalization

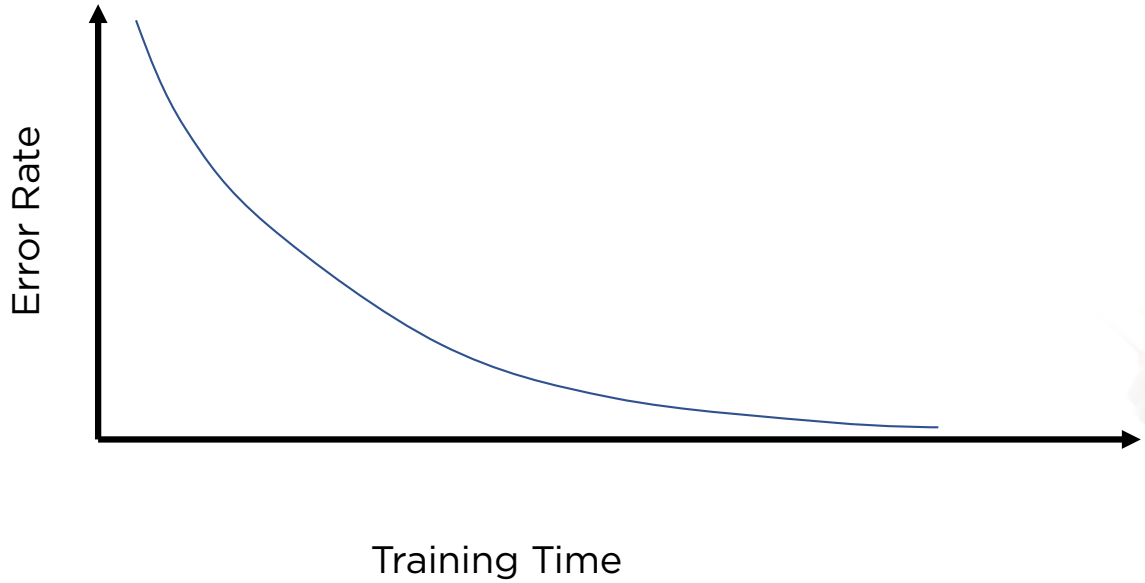
Underfitting



Linear Boundary
(Model)

Model over generalizes because boundary can't be represented

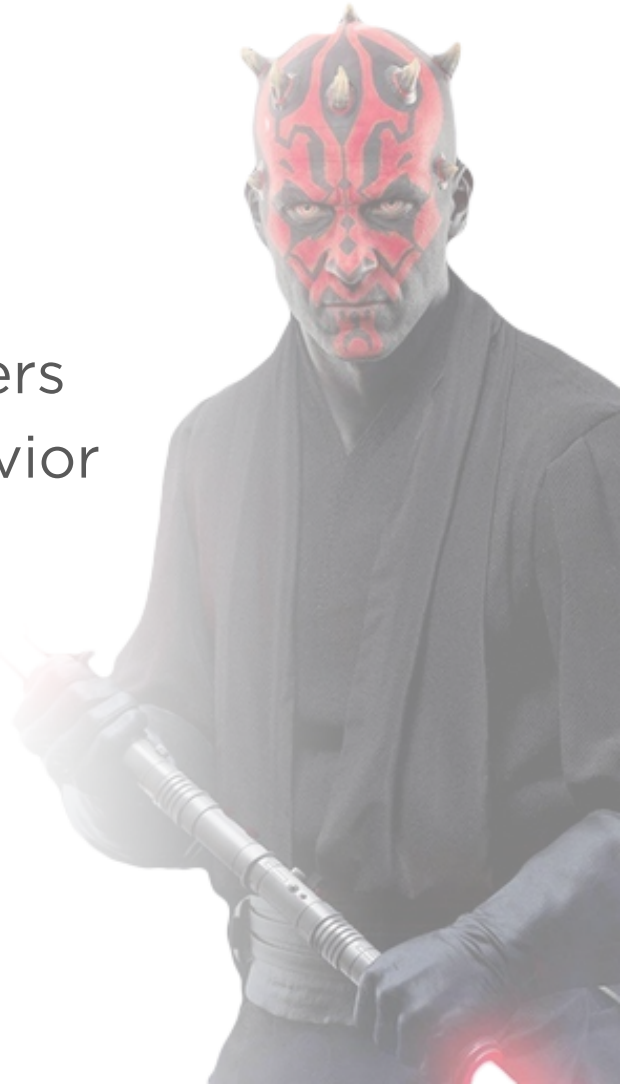
Error Rate During Training



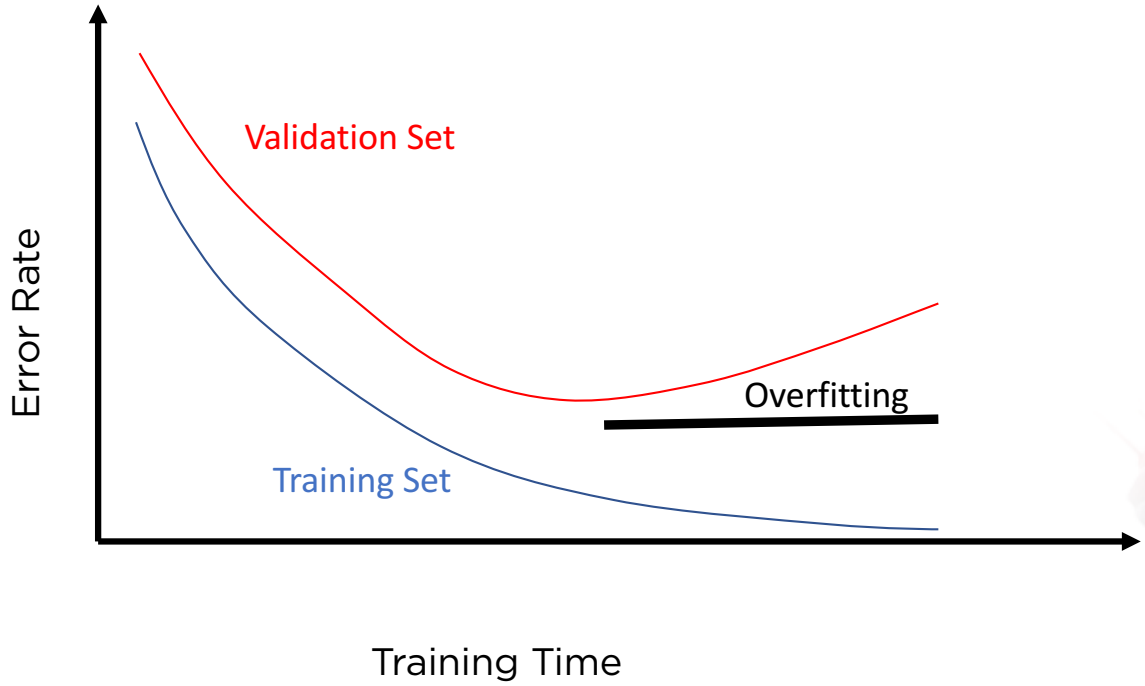
Training versus Validation

Separate data into two sets

- Training–Used to change model parameters
- Validation—Used to evaluate model behavior

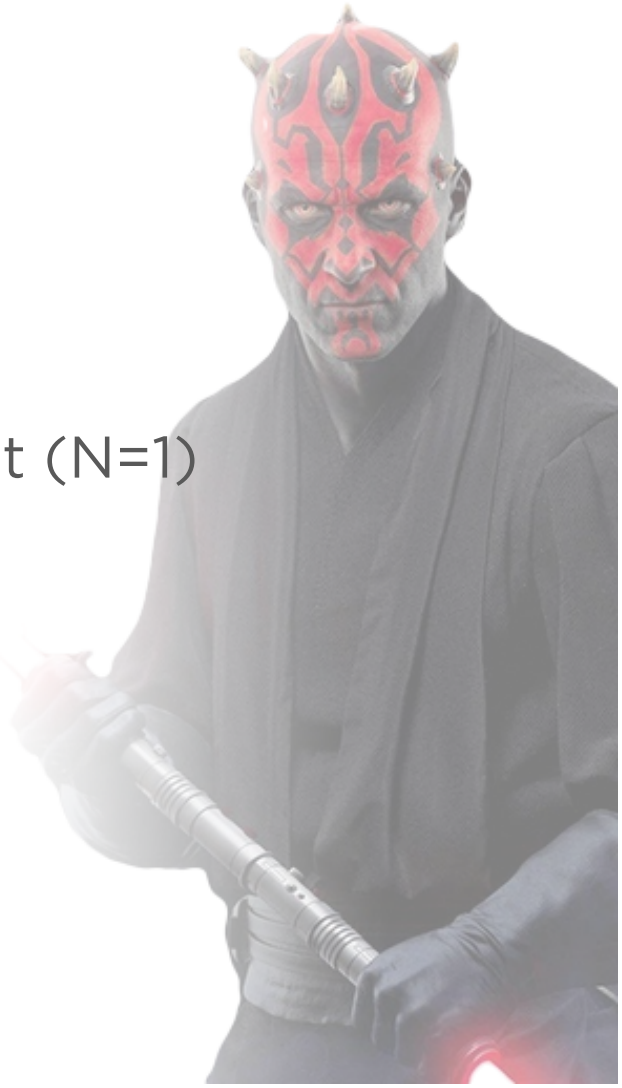


Validation Error During Training



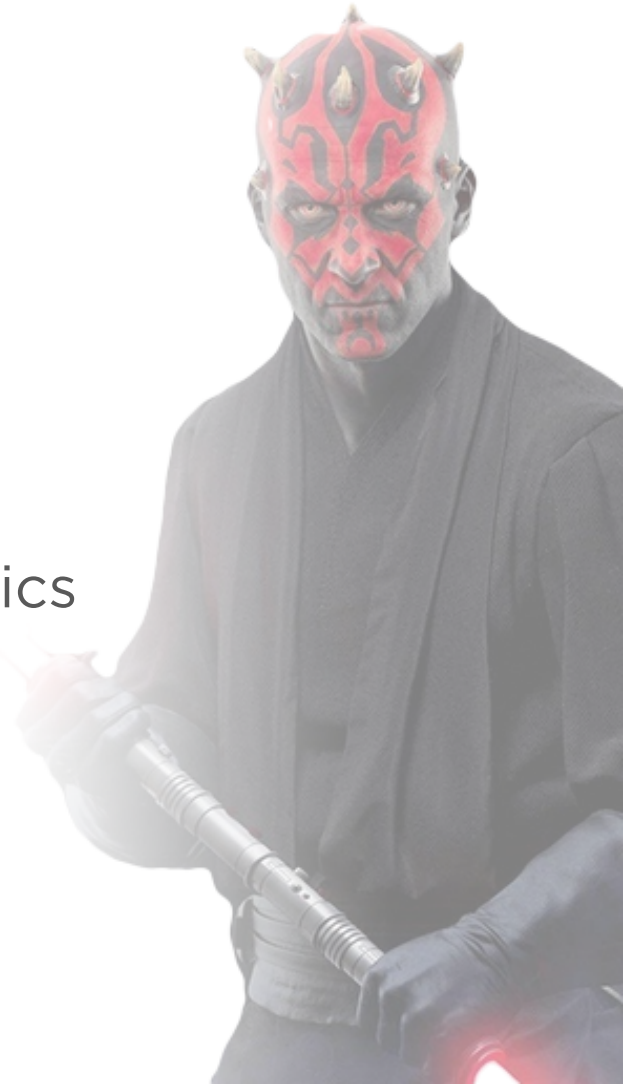
Is It Ready Yet?

- How will our model behave in the wild?
- Training metrics are not sufficient
- Validation metrics help, but not significant (N=1)
- Solution: increase N



Cross Validation

- Break data set into N folds
- For each fold
  - Train on other folds
  - Evaluate on this fold
- Report the distribution of validation metrics



Cross Validation

						Validiation Error Rate
Trial 1	V	T	T	T	T	0.75
Trial 2	T	V	T	T	T	0.83
Trial 3	T	T	V	T	T	0.62
Trial 4	T	T	T	V	T	0.73
Trial 5	T	T	T	T	V	0.78

5-fold Cross Validation

Mean = 0.74
SD = 0.08

Evaluation

- Metrics
- Overfitting and underfitting
- Estimators, Bias and Variance
- Cross Validation

The ML Workflow

- Data preparation
- Model selection
- Training
- Evaluation
- Deployment

Deployment

Building a model is only the beginning

- Model Store

- Systems over Models