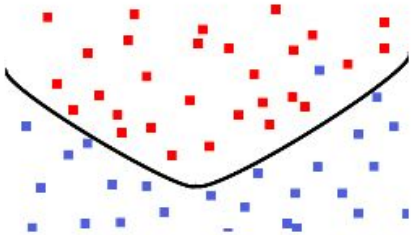
Framing an ML project

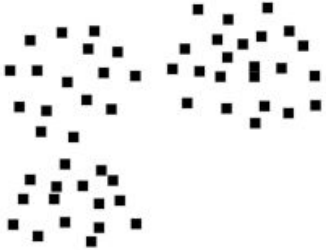
Exxon Training Day 1

Types of Machine Learning



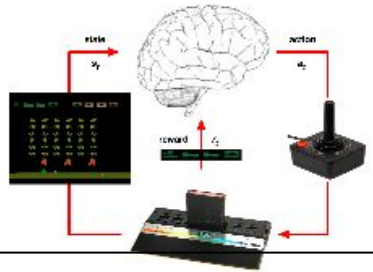
Supervised Learning

Reinforcement Learning



Unsupervised Learning

Learn the mapping function from input X to output Y

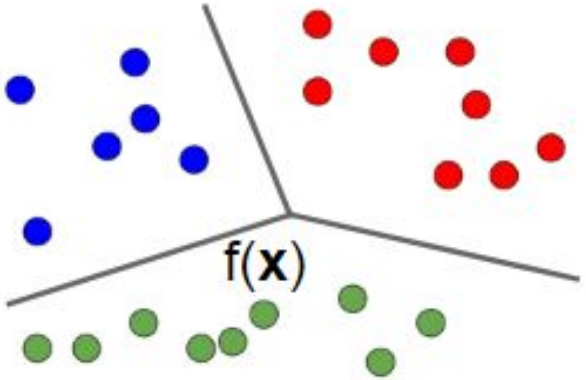


Learn by trial and error

Learn the structure of the data

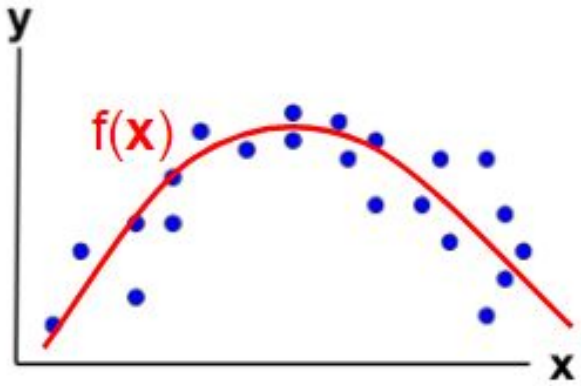
Classification and Regression

Supervised learning



Classification

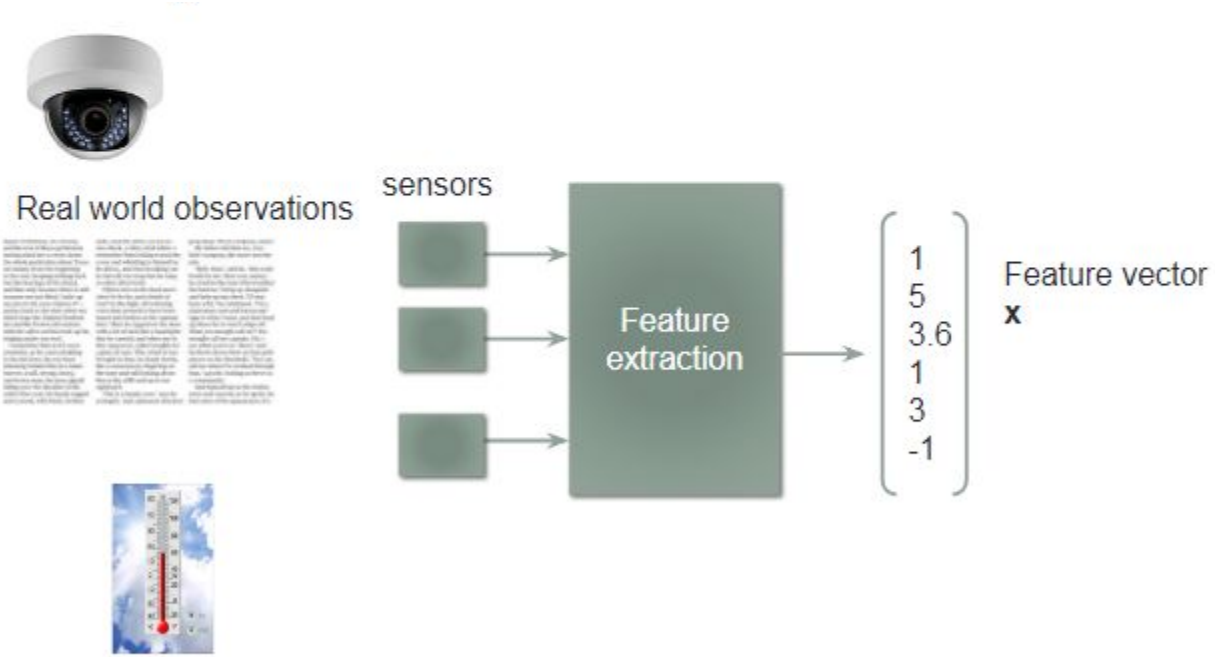
Cat vs dog
Age range of a person in the picture



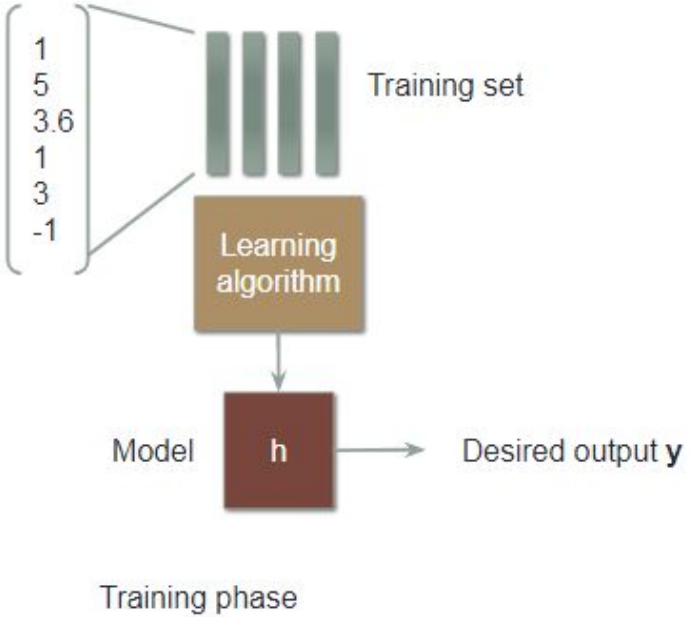
Regression

Rain amount
Age of a person in the picture

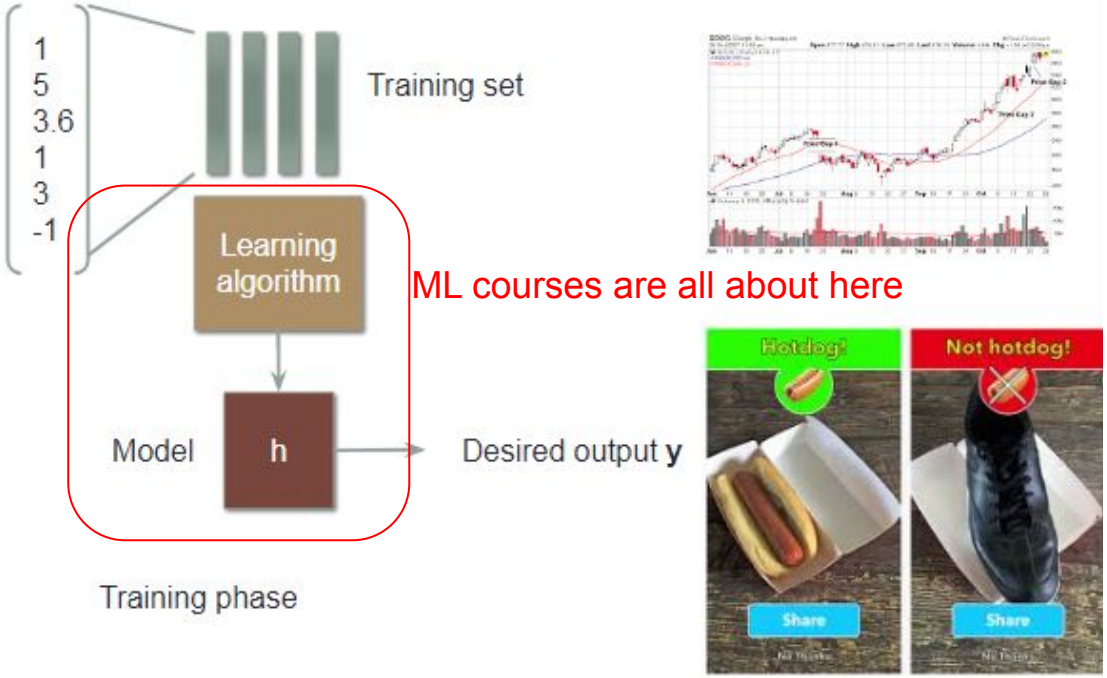
Typical ML workflow



Typical ML workflow

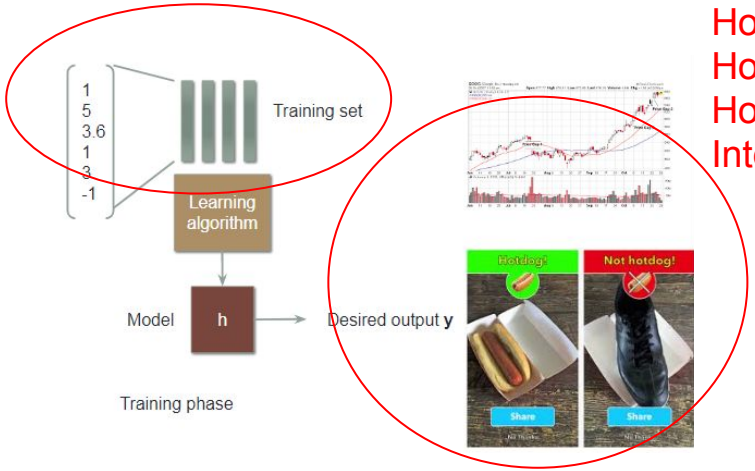


Typical ML workflow



Framing an ML application

What is the training set?
How to collect and curate them?



How to make use of the output?
How to evaluate the model?
How to iterate on your model?
Interpretation of results?

What does a data scientist do at work?

Kaggle competition winner

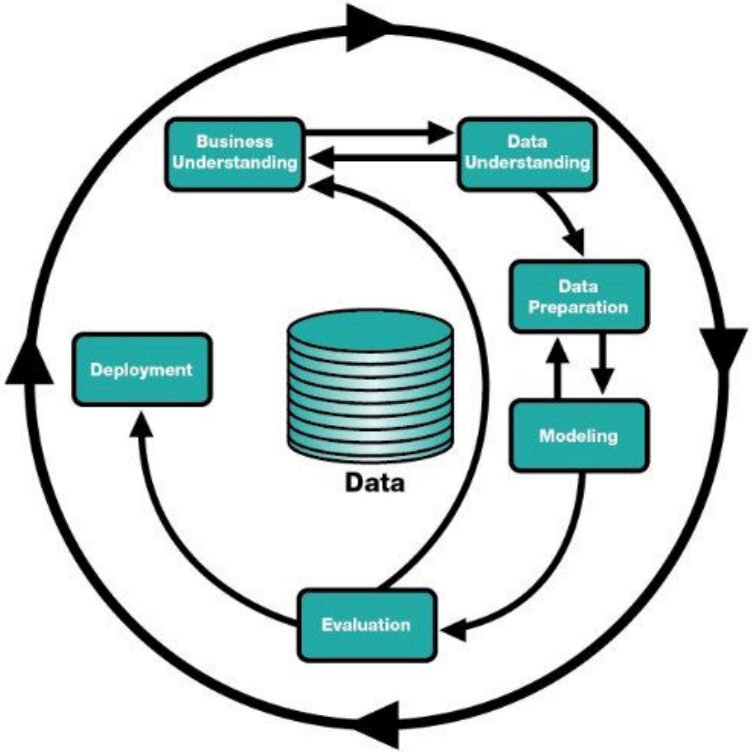
http://blog.kaggle.com/2015/12/21/rossmann-store-sales-winners-interview-1st-place-gert/

How did you spend your time on this competition?

I spent 50% on feature engineering, 40% on feature selection plus model ensembling, and less than 10% on model selection and tuning.

But this is not all of data science

What does a data scientist do at work?



Data science loop

In industry:

“data preparation and modeling (10%) and the rest (90%)”

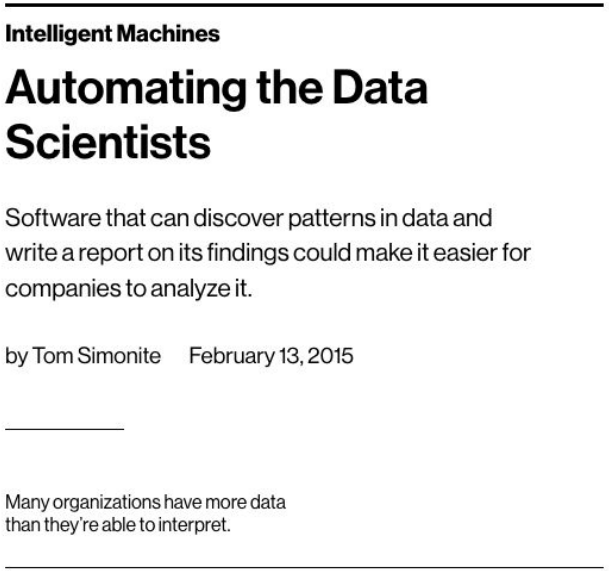
Kaggle competition pitfalls and automation

Given task

Given (closed-set) of inputs

Given performance metrics

Easy replaced by automation.



Intelligent Machines

Automating the Data Scientists

Software that can discover patterns in data and write a report on its findings could make it easier for companies to analyze it.

by Tom Simonite February 13, 2015

Many organizations have more data than they're able to interpret.

https://www.technologyreview.com/s/535041/automating-the-data-scientists/

The 90%

In real life, you will have to look for and decide your tasks, your inputs, your metrics.

Domain knowledge is important

Understand what's possible and what's not

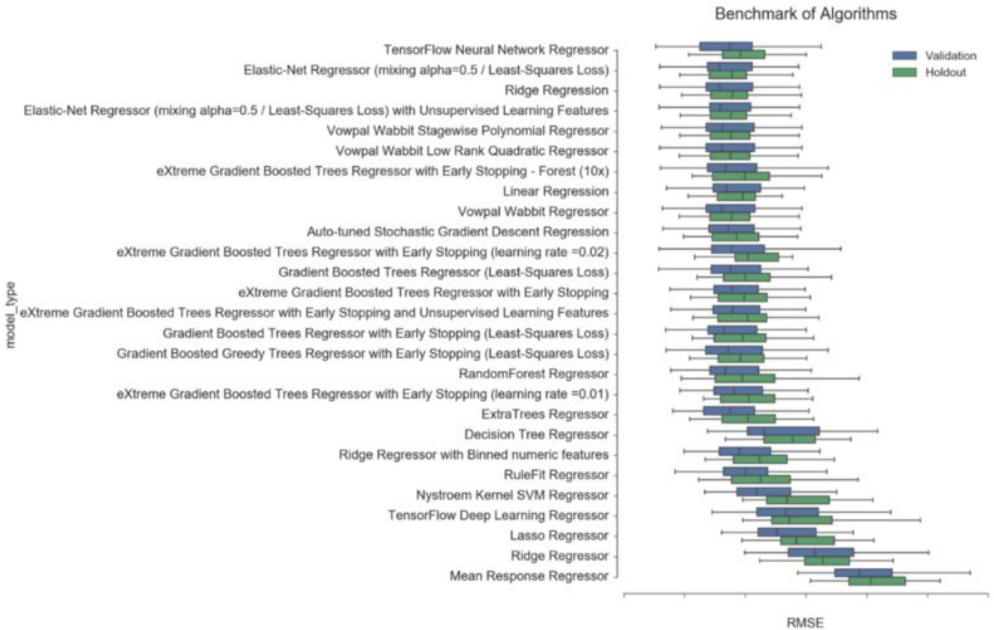
And be able to describe it to non-data scientists

Justify business usage and take care of deployment.

Data scientists + automation tools

Automate tools can remove boring tasks in data science

Give powerful benchmarks



https://www.kdnuggets.com/2017/07/automated-machine-learning-paradigm-shift.html

Rule#1

Don’t be afraid to launch a product without ML

https://www.technospot.net/blogs/gmail-doesnt-allow-you-to-forget-attachment/

Agenda

- Preparation
  - Data
  - Features
- Process
  - Pipeline
  - Metrics
- Analysis

Making use of your data

- Train/dev/test split
  - *80/10/10, 90/5/5, 5-fold CV, leave one out CV, etc. for academia*
- For real applications, get dev and test sets that represent your users.
  - *Reflects the data you want to do well on.*
  - *There can be a mis-match between train and dev data. But avoid mis-match between dev and test data.*
  - *If no users, recruit friends to pretend to be the users.*

- Example: Cat classifier.
  - *Should you use ImageNet cat pictures as train/dev/test?*
  - *Go pretend you're a user and take cat pictures for the dev/test set.*

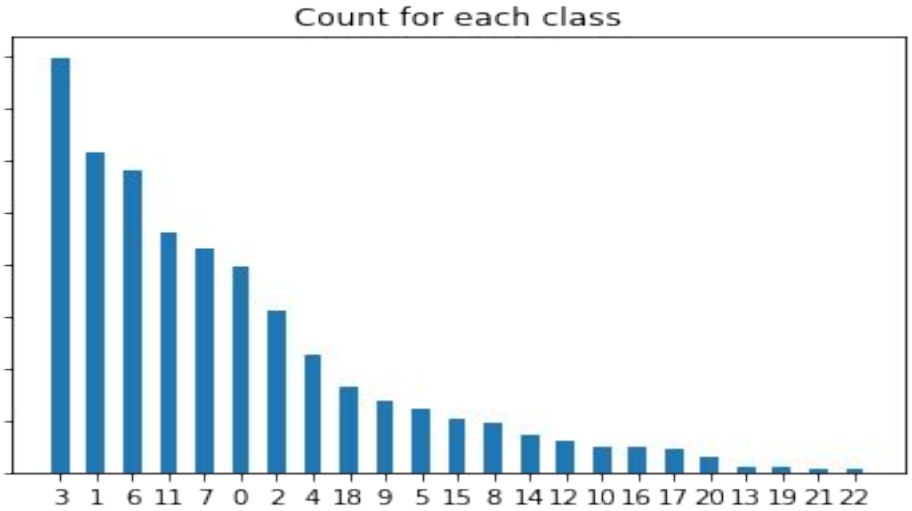
Dev and test set and size

- Dev - tune hyperparameters, select features, and make other decisions regarding the learning algorithm.
- Test - evaluate the performance of the algorithm, but not to make any decisions about regarding what learning algorithm or parameters to use.

- Dev – big enough to notice difference between algorithms (if you care about 0.1% difference, make sure you have enough dev set to spot it).
- Test – large enough to give confidence that your model will do well in real task

Is my task proper? Do you suffer from class imbalance?

- Throwing away
- Refactoring
  - Split
  - Merge
- Data augmentation
- Biasing
  - Weighting the loss function
  - Bias in mini-batch sampling



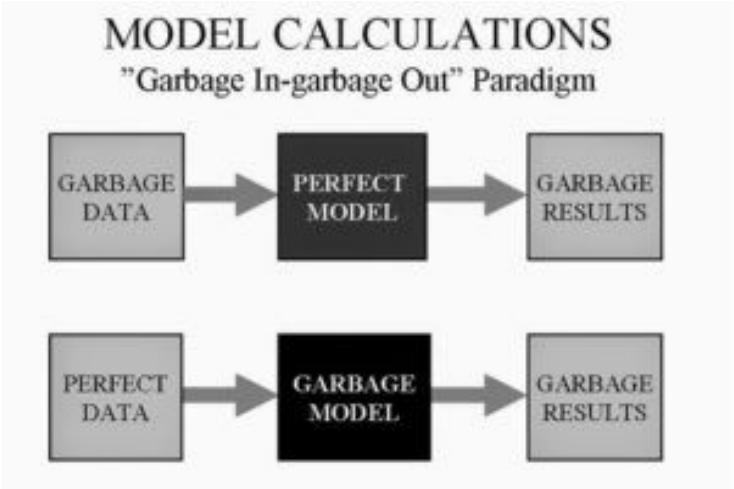
What are the features?

Feature extraction

- The process of extracting meaningful information related to the goal
- A distinctive characteristic or quality

Garbage in Garbage out

- The machine is as intelligent as the data/features we put in
- “Garbage in, Garbage out”
- Data cleaning is often done to reduce unwanted things



The need for data cleaning

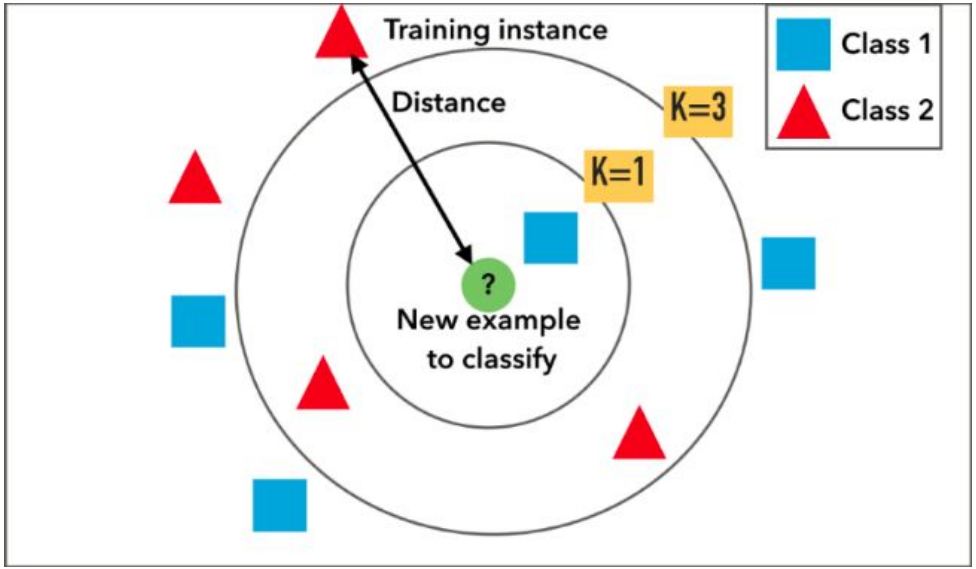
The devil is in the details.

https://www.linkedin.com/pulse/big-data-conundrum-garbage-out-other-challenges-business-platform

Nearest Neighbor Classifier

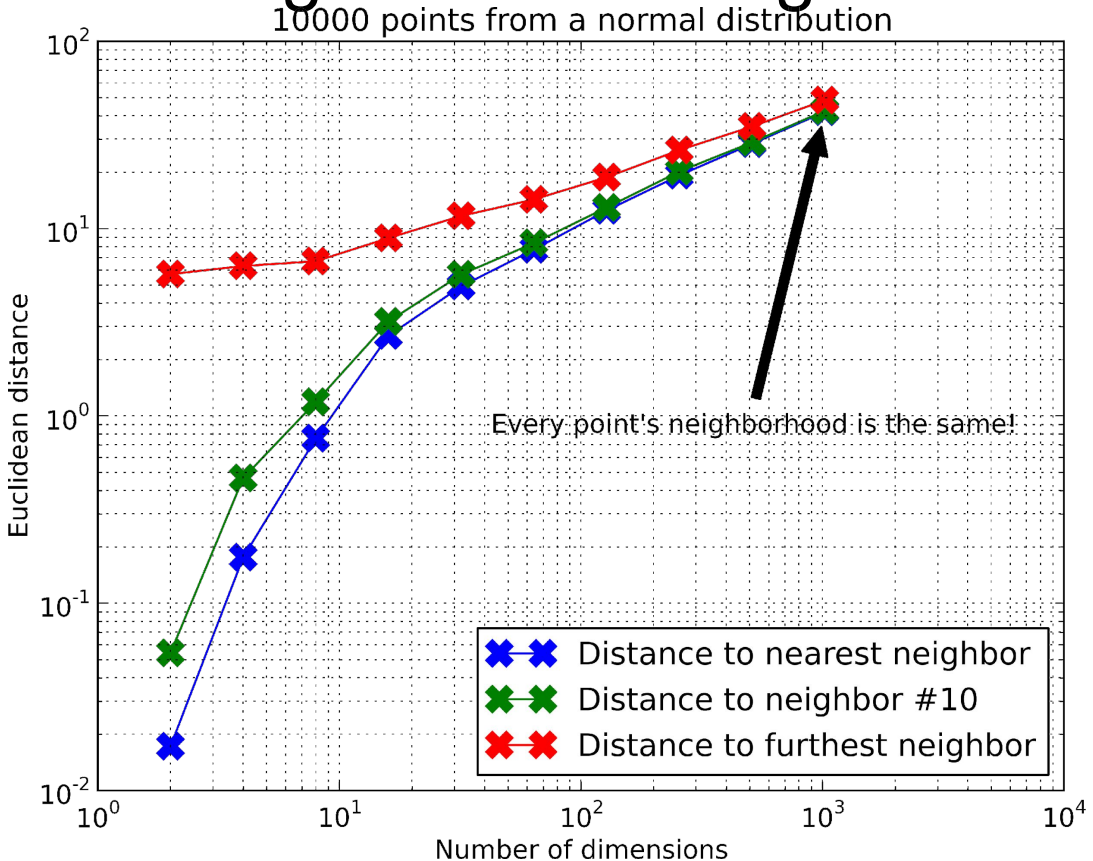
- The thing most similar to the test data must be of the same class

Find the nearest training data, and use that label



https://arifuzzamanfaisal.com/k-nearest-neighbor-regression/

What's wrong with knn in high dimension?



This is called curse of dimensionality

https://erikbern.com/2015/10/20/nearest-neighbors-and-vector-models-epilogue-curse-of-dimensionality.html

Combating the curse of dimensionality

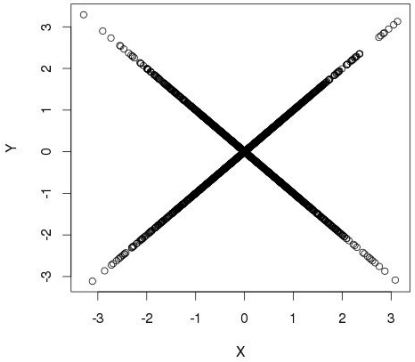
Feature selection

- Keep only “Good” features
- Correlation analysis, feature importance, bagging

Feature transformation

- Transform the original features into a smaller set of features
- PCA, LDA, Deep learning

Common feature selection techniques



- Drop missing features
- Drop low variance rows
  - A feature that is a constant is useless
- Drop features that have low correlation with target output
  - Measures only correlation between only two features. Might still carry useful information
- Model-based from feature importance
  - How often does a decision tree use the feature?
- Forward or backward feature elimination
  - Greedy algorithm: create a simple classifier with n-1 features, n times. Find which one has the best accuracy, drop that feature. Repeat.

Feature selection vs Feature transform

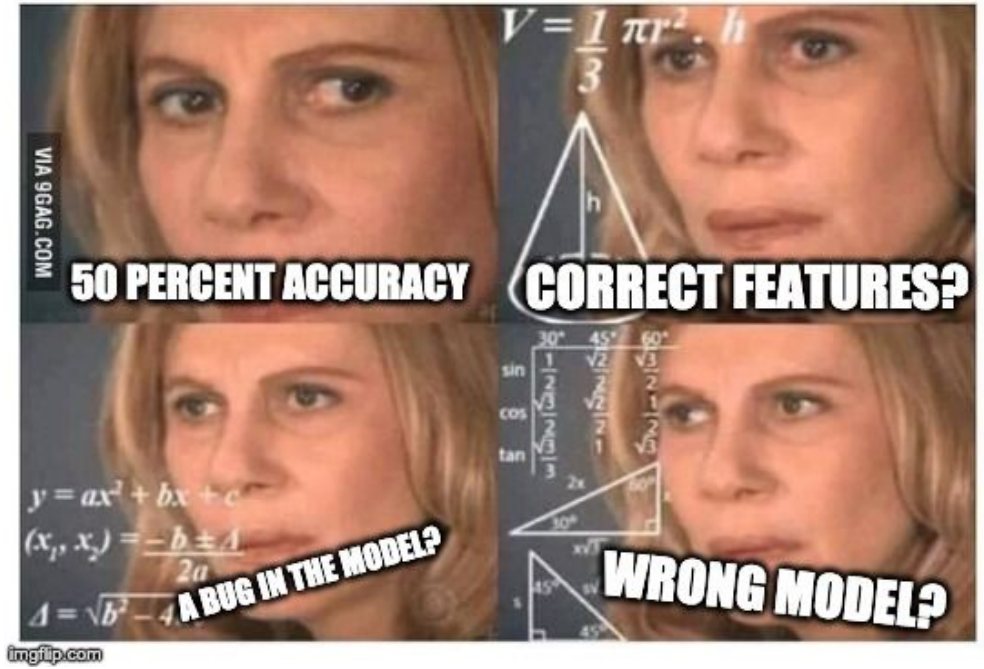
- Keep original features
- Useful for when the user wants to know which feature matters
- But, correlation does not imply causation…

- New features (a combination of old features)
- Usually more powerful
- Captures correlation between features

Agenda

- Preparation
  - Data
  - Features
- Process
  - Pipeline
  - Metrics
- Analysis

A recipe for disaster



A ML pipeline is complex with many pitfalls along the way.
Do not start with a complex model.

Rule#3 start simple

Keep the first model simple but with a complete infrastructure

Answer these questions

1. How to get data for your model?

2. What's "launchable"? How to measure that?

3. Integration?

Having simple models makes debugging easier.

Rule#2 Pick metrics

How to measure performance of the model

AND how to measure the effect on your business goal (A/B test)

Speech Recognition - Word Error Rate

Search – Recall rate

Machine Translation - BLEU score

Recommendation system – Click through rate

- Tasks can have multiple metrics
- Understand the relationship between the metric and the business goal

Example: Chatbots

- Task completion rate
- Number of turns to complete a task
- Satisfaction
- Word Error Rate (if speech input)

Understand the trade-offs

- Performance metrics
- Amount of computation power required
- Type of computation required
- Latency

Picking one guiding metric

- In order to develop fast, pick one summarizing metric to optimize
  - *Precision + recall -> F-score*
  - *Weighted average of multiple metrics*
- Task driven
- Instead of combining all metrics. Some metrics are optimizing metrics while some are constraints metrics
  - Must use less than X ram
  - Produce output in less than x milliseconds
  - Less than 1 false alarm per day
  - Note performance, metrics can sometimes be relaxed.
    - *Easier to make something fast, than make something good.*

Example: A detection problem

Identify whether an event occur

- A yes/no question
- A binary classifier

Smoke detector

Hotdog detector

Evaluating a detection problem

- 4 possible scenarios

		Detector	
		Yes	No
Actual	Yes	True positive	False negative (Type II error)
	No	False Alarm (Type I error)	True negative

True positive + False negative = # of actual yes
False alarm + True negative = # of actual no

Definition

- True positive rate (Recall, sensitivity)
  = # true positive / # of actual yes
- False positive rate (False alarm rate)
  = # false positive / # of actual no
- False negative rate (Miss rate)
  = # false negative / # of actual yes
- True negative rate (Specificity)
  = # true negative / # of actual no

- Precision = # true positive / # of predicted positive

Search engine example

A recall of 50% means?

A precision of 50% means?

When do you want high recall?
When do you want high precision?

Examples

When do you want high recall?
When do you want high precision?

Initial screening for cancer
Face recognition system for authentication
Detecting possible suicidal postings on social media

Thresholds

ML score	Detector decision	Real fire?	
0.4		Yes	
0.7		Yes	
0.6		No	
0.1		No	
0.2		No	

Thresholds

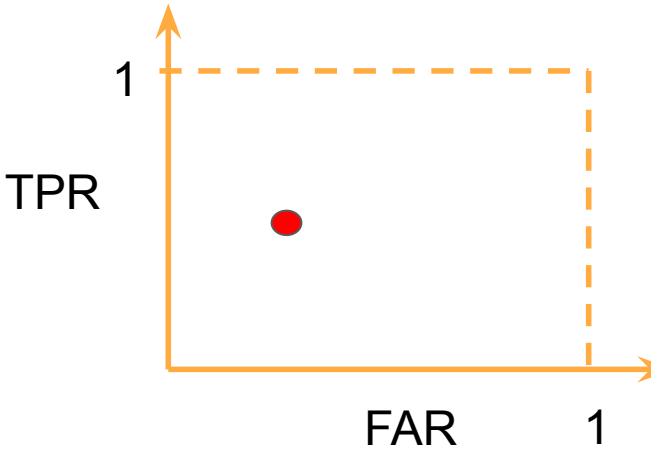
Threshold > 0.5 = yes

ML score	Detector decision	Real fire?	
0.4	No	Yes	Miss detection
0.7	Yes	Yes	True positive
0.6	Yes	No	False alarm
0.1	No	No	True negative
0.2	No	No	True negative

Thresholds

Threshold > 0.5 = yes

ML score	Detector decision	Real fire?	
0.4	No	Yes	Miss detection
0.7	Yes	Yes	True positive
0.6	Yes	No	False alarm
0.1	No	No	True negative
0.2	No	No	True negative

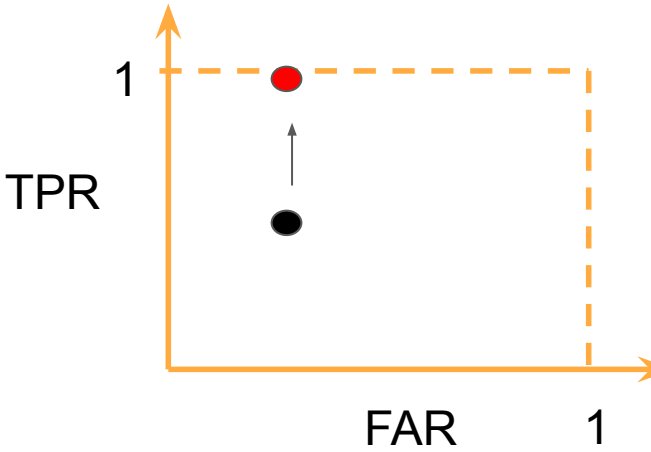


FAR = 1/3
TPR = 1/2

Thresholds

Threshold > 0.39 = yes

ML score	Detector decision	Real fire?	
0.4	Yes	Yes	True positive
0.7	Yes	Yes	True positive
0.6	Yes	No	False alarm
0.1	No	No	True negative
0.2	No	No	True negative

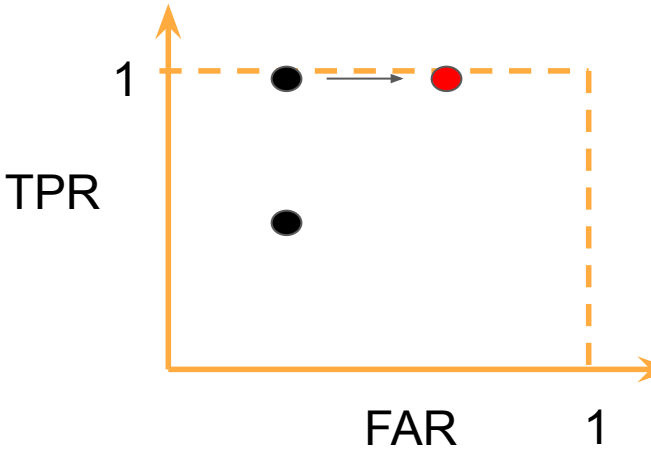


FAR = 1/3
TPR = 1

Thresholds

Threshold > 0.11 = yes

ML score	Detector decision	Real fire?	
0.4	Yes	Yes	True positive
0.7	Yes	Yes	True positive
0.6	Yes	No	False alarm
0.1	No	No	True negative
0.2	Yes	No	False alarm

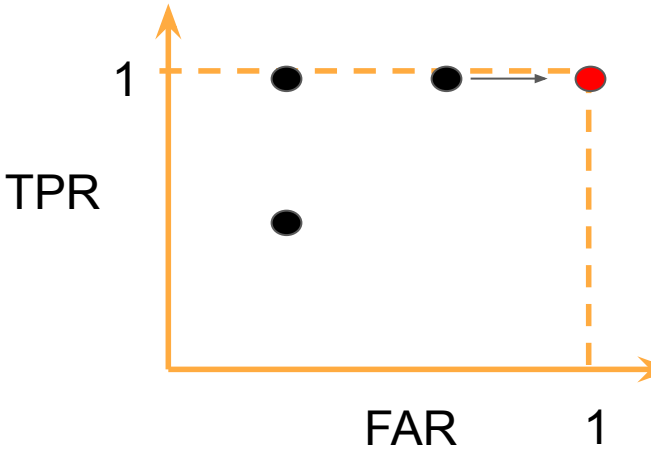


FAR = 2/3
TPR = 1

Thresholds

Threshold > 0 = yes

ML score	Detector decision	Real fire?	
0.4	Yes	Yes	True positive
0.7	Yes	Yes	True positive
0.6	Yes	No	False alarm
0.1	Yes	No	False alarm
0.2	Yes	No	False alarm

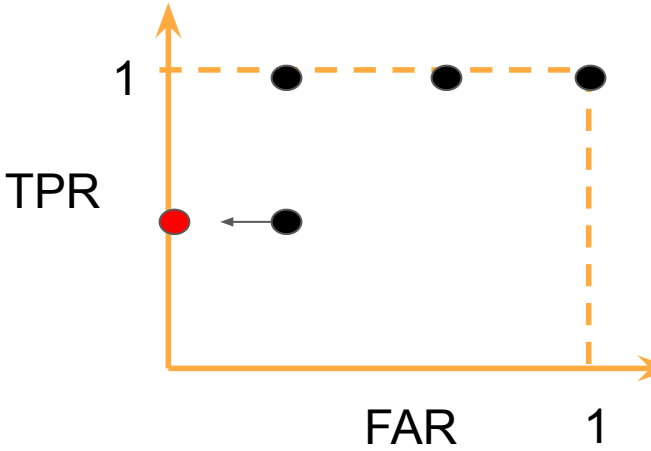


FAR = 1
TPR = 1

Thresholds

Threshold > 0.61 = yes

ML score	Detector decision	Real fire?	
0.4	No	Yes	Miss detection
0.7	Yes	Yes	True positive
0.6	No	No	True negative
0.1	No	No	True negative
0.2	No	No	True negative

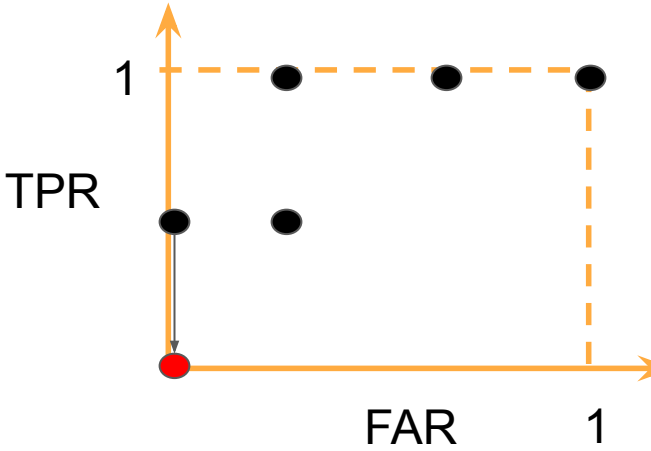


FAR = 0
TPR = 1/2

Thresholds

Threshold > 1 = yes

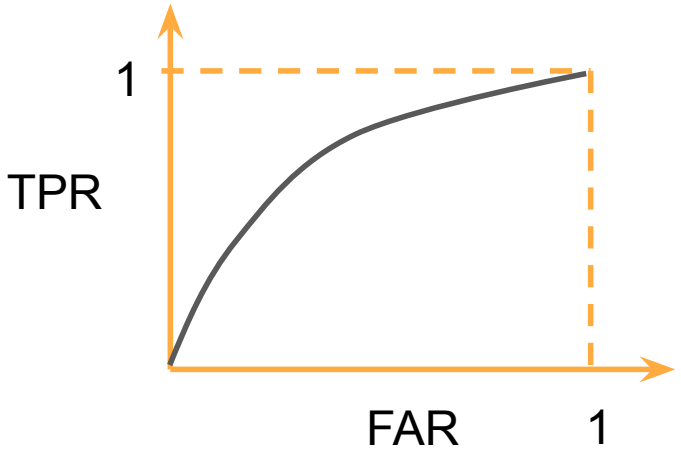
ML score	Detector decision	Real fire?	
0.4	No	Yes	Miss detection
0.7	No	Yes	Miss detection
0.6	No	No	True negative
0.1	No	No	True negative
0.2	No	No	True negative



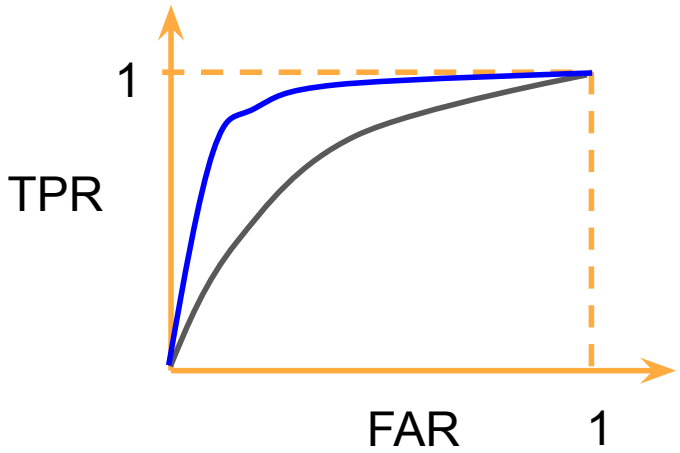
FAR = 0
TPR = 1/2

Receiver operating Characteristic (RoC) curve

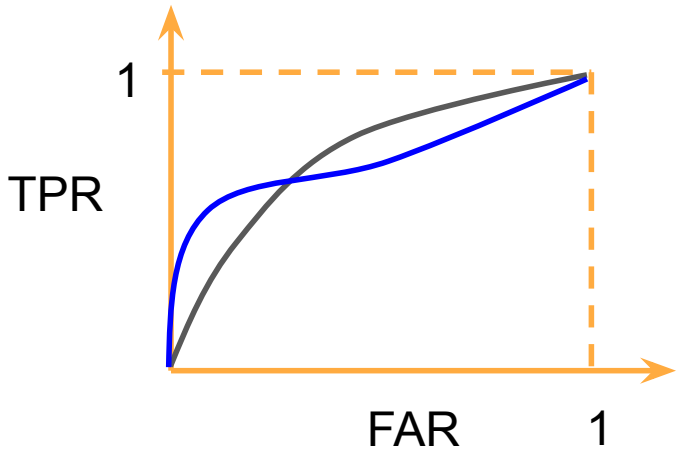
- What if we change the threshold?
- FA TP is a tradeoff
- Plot FA rate and TP rate as threshold changes
  - Each model will have one RoC line



Comparing models

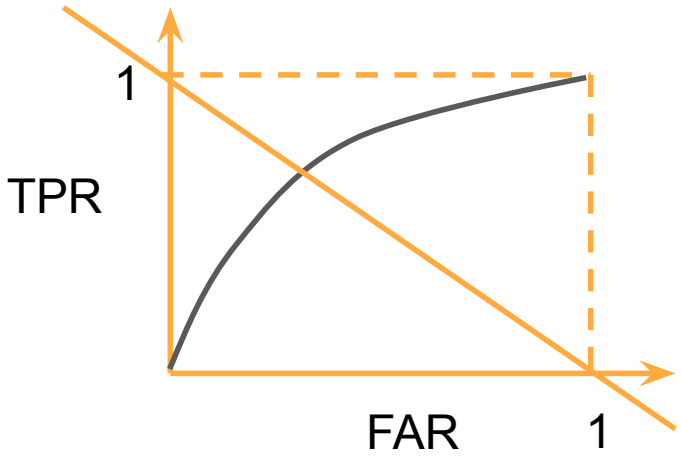


Comparing models



Selecting the threshold

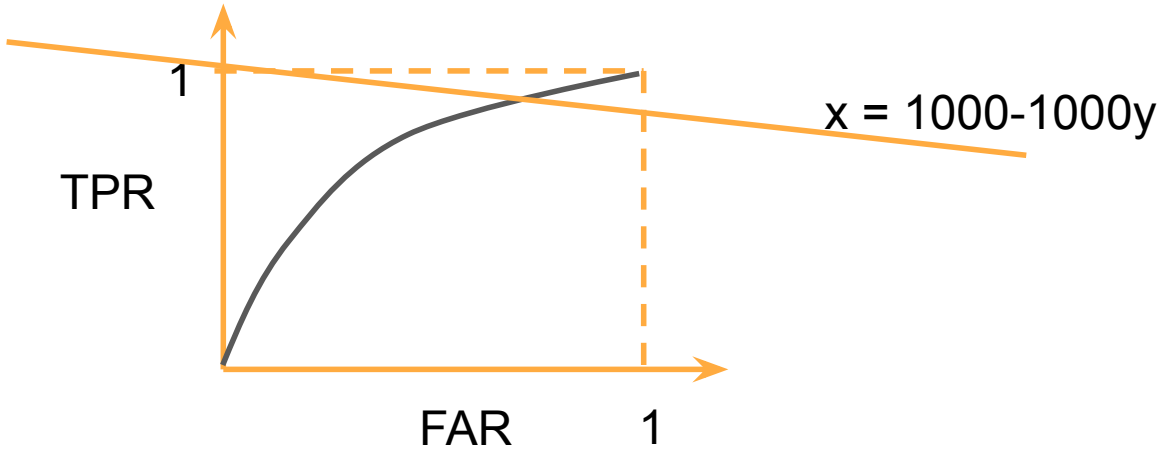
- Select based on the application
- Trade off between TP and FA. Know your application, know your users.
- A miss is as bad as a false alarm
  - FAR = 1-TPR => x = 1-y



This line has a special name
Equal Error Rate (EER)

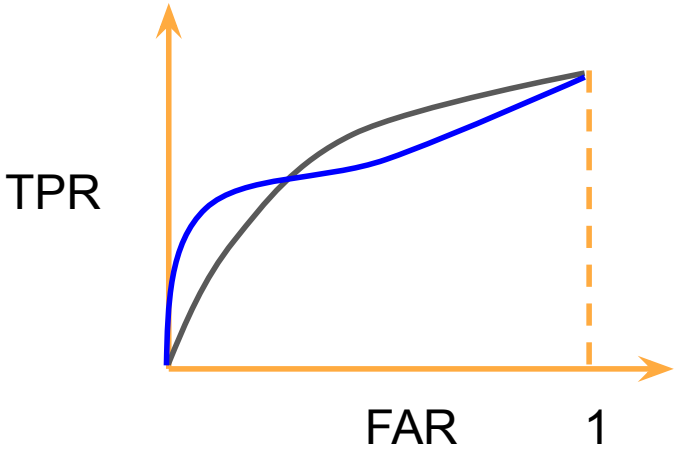
Selecting the threshold

- A miss is 1000 times more costly than false alarm.
  - FAR = 1000(1-TPR) => x = 1000-1000y



Comparing models

- Which is better?



- You want to send promotions to potential churn users in the system.
- You want have HR talk to staff who might leave the company

Agenda

- Preparation
  - Data
  - Features
- Process
  - Pipeline
  - Metrics
- Analysis
  - Error
  - Bias-variance

Error diagnosis

- You're making a cat classifier. (cat/not cat)
- It sucks.
- You heard of this super new hype algorithm (for example: capsule network). Should you spend months to try it out?

This is not a cat

https://www.vat19.com/item/not-a-cat-cat-the-cat-that-isnt

Looking at the errors

- Spend an hour or two looking at your errors. Identify why. Keep a table.

	Blurred	Weird angle	Notes
Pic1	x		Stuffed toy
Pic2	x		
Pic3	x		
Pic4		x	Top view
…	…	…	…
	68%	2%	

Solution: Use a method to sharpen the image. Train on blurry images.

The table categories can expand as you look through more pictures and see frequently occurring error cases. So keep notes.

Rule#4 always perform error diagnosis

- Simple analysis of the data can help you notice underlying problems
- Error analysis can help guide your model improvements

Taking the time to do diagnosis can help you save months of trying random things.

Bias-Variance Diagnosis

Bias: (Underfitting)

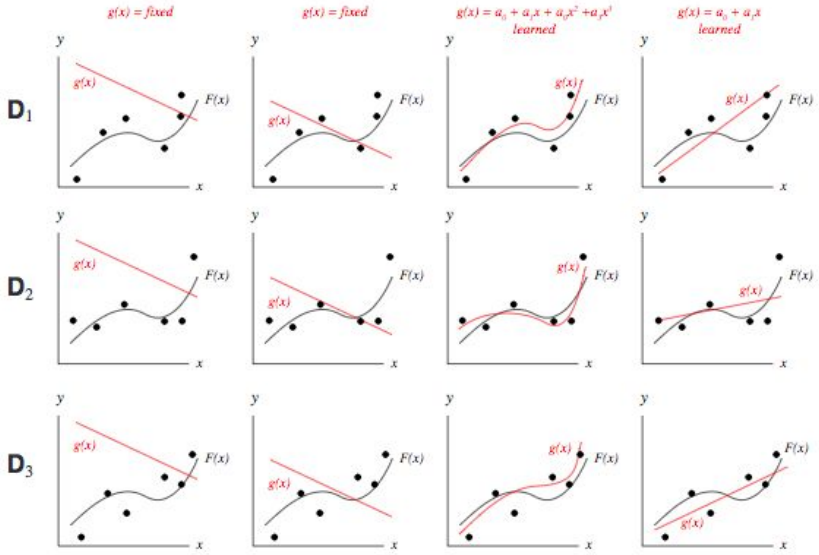
"Error of the model with infinite amount of data"

Measure from training error

Variance: (Overfitting)

"Error of the model that comes from limited data"

Measure from validation error



Duda et al. Pattern Classification

Bias-Variance Diagnosis

Bias: (Underfitting)

- Measure from training error
- Training error is high

Solved by

- More features
- Better techniques

Variance: (Overfitting)

- Measure from validation error
- High gap between training and validation error

Solved by

- More data
- Techniques to reduce overfitting

Bias-Variance Diagnosis Example

Text-to-speech

- Human level performance: 5% error
- Training error: 6%
- Validation error: 10%

Bias-Variance Diagnosis Example

Churn prediction

- Target accuracy: 10% error
- Training error: 30%
- Validation error: 32%

This is also related to rule#2

Summary

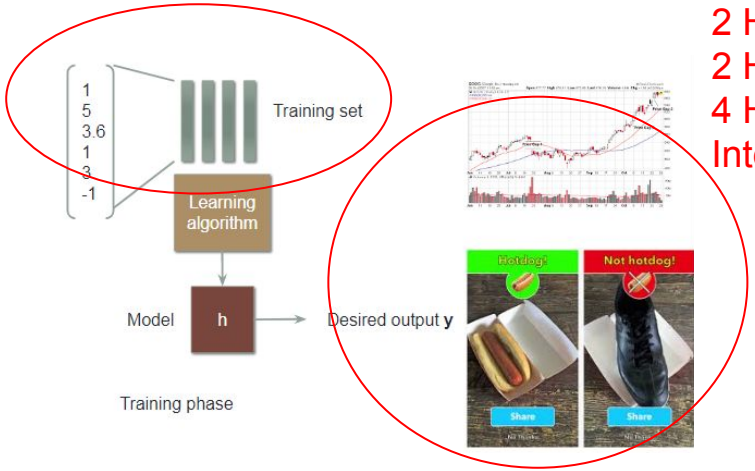
Rule#1

Rule#2

3 What is the training set?
3 How to collect and curate them?

Rule#3

Rule#4



2 How to make use of the output?
2 How to evaluate the model?
4 How to iterate on your model?
Interpretation of results?