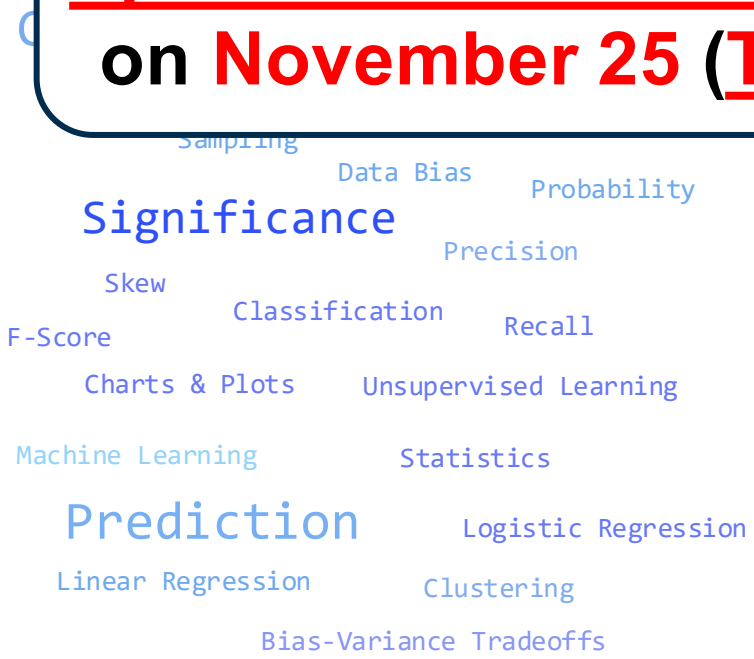
Super-Important Note:
Special Guest Lecture will be on November 25 (Tuesday)!



Data Science 1:
Introduction to Data Science

Building & Evaluating Models

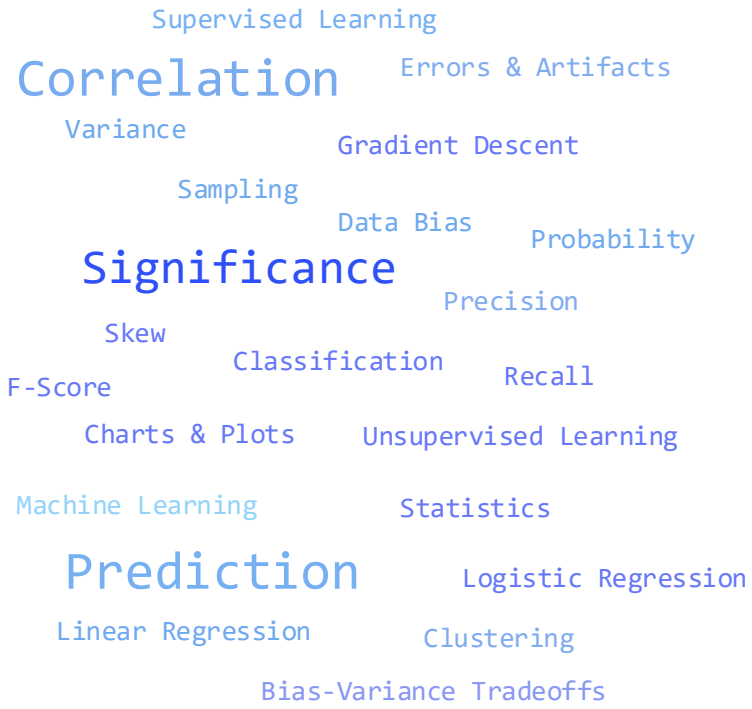
Winter 2025

Wolfram Wingerath, Jannik Schröder

Department for Computing Science
Data Science / Information Systems

Lecture slides based on content from "The Data Science Design Manual" (Steven Skiena, 2017) and associated course materials generously made available online by the author at https://www3.cs.stonybrook.edu/~skiena/data-manual/.

Special thanks to Professor Skiena for sharing these valuable teaching resources!



Data Science 1:
Introduction to Data Science

Building & Evaluating Models

Winter 2025

Wolfram Wingerath, Jannik Schröder

Department for Computing Science
Data Science / Information Systems

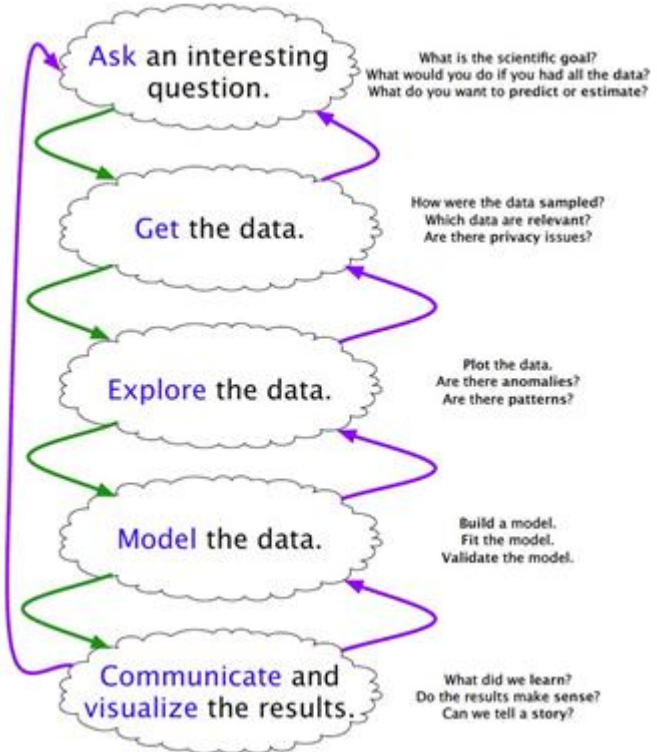
Semester Schedule

CW 42	14. Oct	Lecture	1	Orga & Intro	1-26
CW 43	21. / 23. Oct	Lecture + Exercises	2	Probability, Statistics & Correlation	27-56
CW 44	28. Oct	Lecture	3	Data Munging, Cleaning & Bias	57-94 / “Invisible Women”
CW 45	04. / 06. Nov	Lecture + Exercises	4	Scores & Rankings	95-120
CW 46	11. Nov	Lecture	5	Statistical Distributions & Significance	121-154
CW 47	18. / 20. Nov	Lecture + Exercises	6	Building & Evaluating Models	201-236
CW 48	25. Nov	Guest Lecture	7	Data Visualization	155-200
CW 49	02. / 04. Dec	Lecture + Exercises	8	Intro to Machine Learning	351-390
CW 50	09. Dec	Lecture	9	Linear Algebra	237-266
CW 51	16. / 18. Dec	Lecture + Exercises	10	Linear Regression & Gradient Descent	267-288
CW 02	06. Jan	Lecture	11	Logistic Regression & Classification	289-302
CW 03	13. / 15. Jan	Lecture + Exercises	12	Nearest Neighbor Methods & Clustering	303-350
CW 04	20. Jan	Lecture	13	Data Science in the Wild	391-426
CW 05	27. / 29. Jan	Lecture + Exercises	14	Q&A / Feedback	
CW 06	03. / 04. Feb	Oral Exams (Block 1)		Preparation in our last session („Oral Exam Briefing“)	
CW 13	24. / 25. Mar	Oral Exams (Block 2)			

The Data Science Analysis Pipeline

Modeling is the process of encapsulating information into a tool which can make forecasts/predictions.

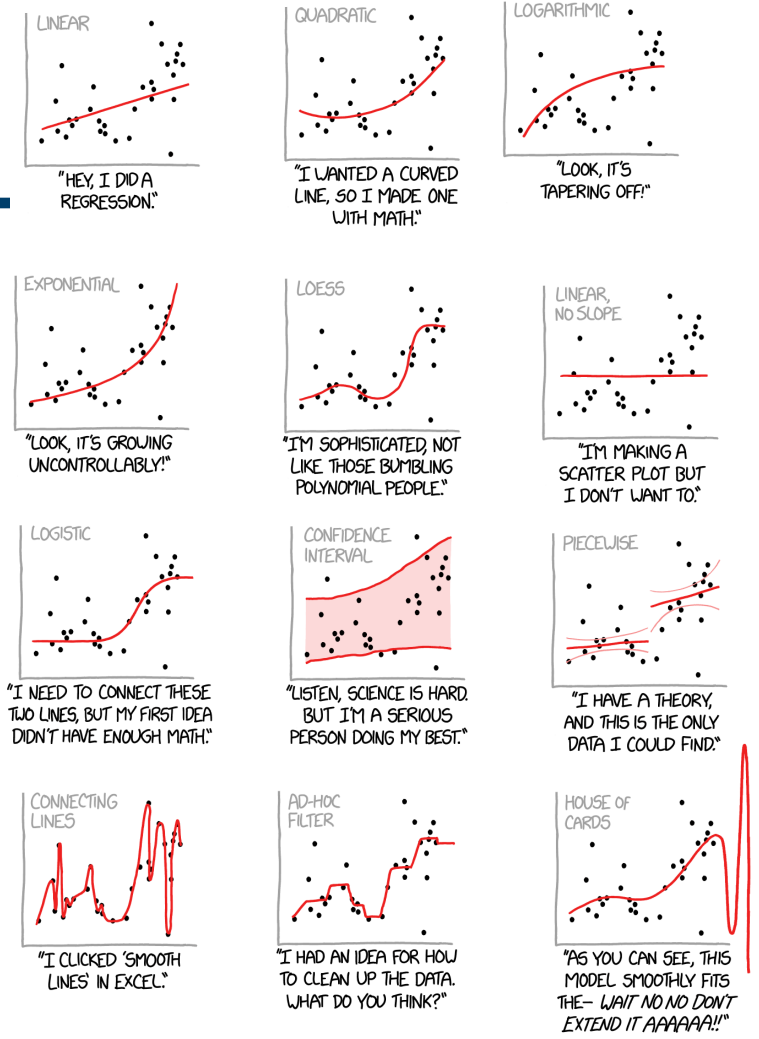
The key steps are building, fitting, and validating the model.



Which is Best?

There are many ways to model any given data set.

How can we decide which approach is better?



XKCD: Curve-Fitting Methods and the Messages They Send (https://xkcd.com/2048/)

Philosophies of Modeling

We need to think in some fundamental ways about modeling to build them in sensible ways.

- Occam's Razor
- Bias-Variance trade offs
- Nate Silver: The Signal and Noise

Occam's Razor

This philosophical principle states that "the simplest explanation is best".

With respect to modeling, this often means minimizing the parameter count in a model.

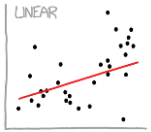
Machine learning methods like LASSO/ridge regression employ penalty functions to minimize features, but also do a "sniff test".

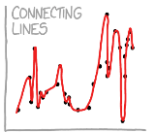
Bias-Variance Tradeoffs

"All models are wrong, but some models are useful."

– George Box (1919-2013)

- ***Bias*** is error from erroneous assumptions in the model, like making it linear. (underfitting)
- ***Variance*** is error from sensitivity to small fluctuations in the training set. (overfitting)

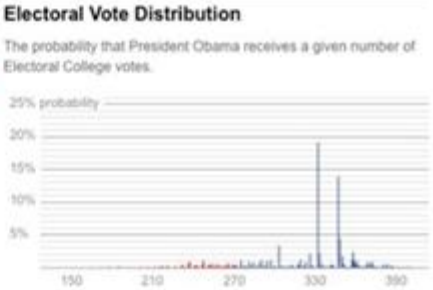




First-principle models likely to suffer from bias, with data-driven models in greater danger of overfitting.

XKCD: Curve-Fitting Methods and the Messages They Send (https://xkcd.com/2048/)

What would Nate Silver do?



Principles of Nate Silver

- Think probabilistically
- Change your forecast in response to new information.
- Look for consensus
- Employ Bayesian reasoning

The Output of Your Models

Demanding a single deterministic “prediction” from a model is a fool’s errand.

Good forecasting models generally produce a probability distribution over all possible events.

Good models do better than baseline models, but you could get rich predicting if the stock market goes up/down with $p>0.55$.

Properties of Probabilities

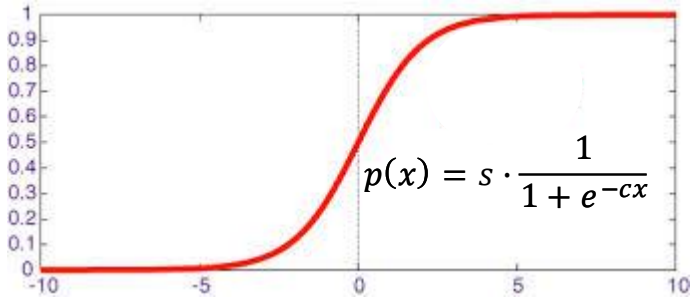
- They sum to 1.
- They are never negative.
- Rare events do not get probabilities of zero.

Probabilities are a measure of humility in the accuracy of the model, and the uncertainty of a complex world.

Models must be honest in what they do/don't know.

Scores to Probabilities

The logistic function maps a score x to a probability using a parameter.



Summing up the “probabilities” over all events, s defines the constant *1/s* to multiply each so they sum up to 1.

Logistic vs. Logit Function

Error in the Book: The *Data Science Design Manual* does not separate these concepts clearly!

Logistic & logit function are inverse to each other:

Logistic function
(score → probability)

$$logistic(x) = \frac{1}{1 + e^{-x}}$$

Logit function
(probability → score)

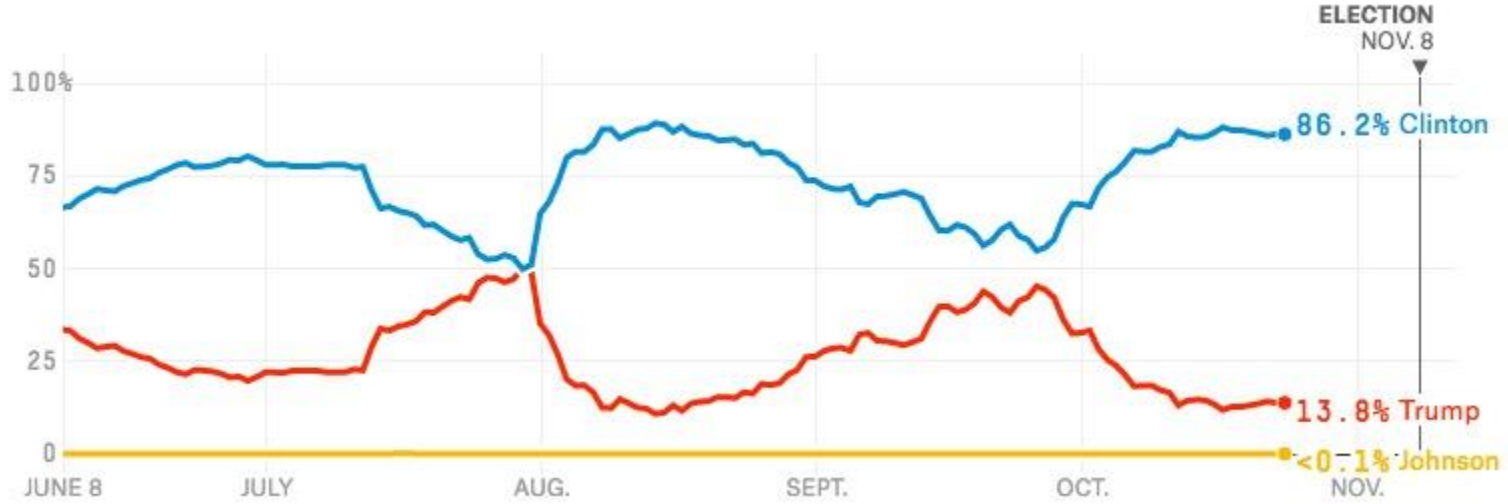
$$logit(p) = log\left(\frac{p}{1-p}\right)$$

Live Models

A model is *live* if it continually updating predictions in response to new information.

- Does the forecast ultimately converge on the right answer?
- Does it display past forecasts so the user can judge the consistency of the model?
- Does the model retrain on fresher data?

Presidential Election Forecast, 2016



Look for Consensus

- Are there competing forecasts you can compare to, e.g. prediction markets?
- What do your baseline models say?
- Do you have multiple models which use different approaches to making the forecast?

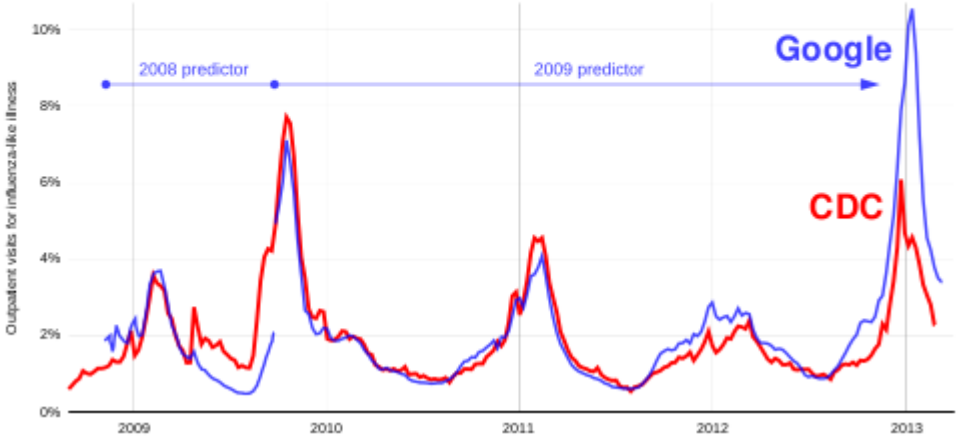
Boosting is a machine learning technique which explicitly combines an ensemble of classifier.

Google Flu Trends

Predicted flu outbreaks using query frequency of illness terms.

The model failed after Google added search suggestions

Second divergence in 2012–2013 for U.S.



Modeling Methodologies

- First principle models: based on a theoretical explanation of how the system works (like simulations, scientific formulae)
- Data-driven models: based on observed data correlations between input parameters and outcome variables.

Good models are typically a mixture of both.

Baseline Models

"A broken clock is right twice a day."

The first step to assess whether your model is any good is to build baselines: the simplest *reasonable* models to compare against.

Only after you decisively beat your baselines can your models be deemed effective.

Representative Baseline Models

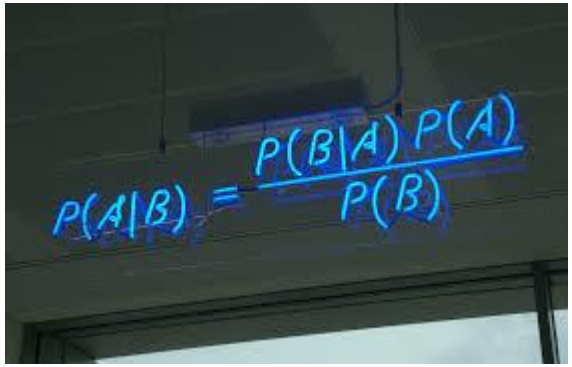
- Uniform or random selection among labels.
- The most common label in the training data.
- The best performing single-variable model.
- Same label as the previous point in time.
- Rule of thumb heuristics.

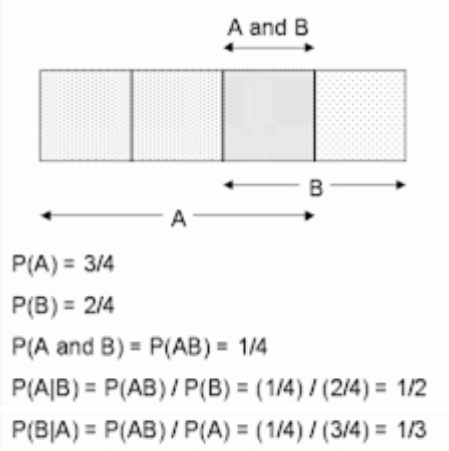
Baseline models must be fair: they should be simple but not stupid.

Bayesian Reasoning

Bayes' Theorem lets us update our confidence in an event in response to fresh evidence.

Bayesian reasoning reflects how a **prior probability** P(A) is updated to given the **posterior probability** P(A|B) in the face of a new observation B according to the ratio of the likelihood P(B|A) and the **marginal probability** P(B)





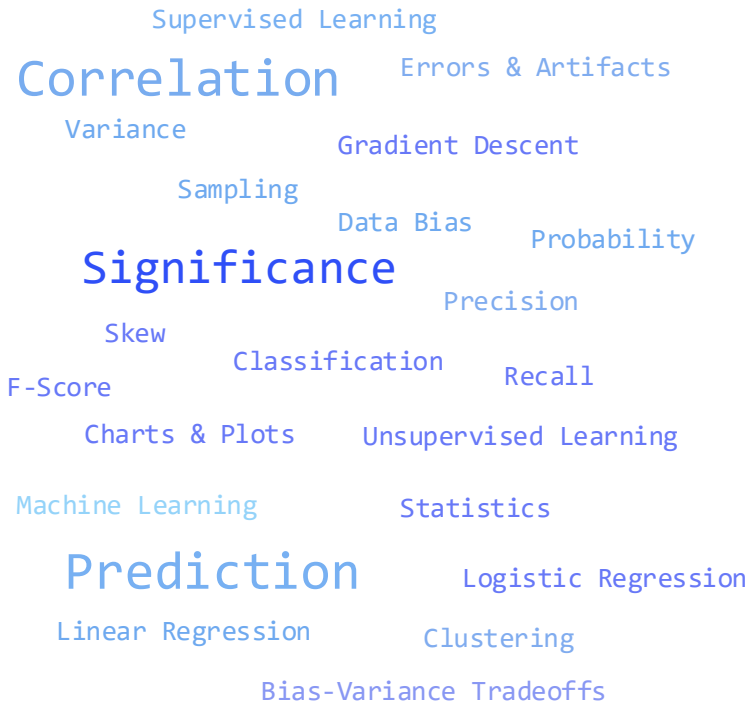
Taxonomy of Models

Models have different properties inherent in how they are constructed:

- Linear vs. nonlinear
- Discrete vs. continuous models
- Black box vs. descriptive
- Stochastic vs. deterministic
- Flat vs. hierarchical

Steps to Build Effective Models

- Identify the best output type for your model, likely a probability distribution.
- Develop reasonable baseline models.
- Identify the most important levels to build submodels around.
- Test models with out-of-sample predictions.



Data Science 1:
Introduction to Data Science

Building &

Evaluating Models

Winter 2025

Wolfram Wingerath, Jannik Schröder

Department for Computing Science
Data Science / Information Systems

How Good is Your Model?

After you train a model, you need to evaluate it on your testing data.

What statistics are most meaningful for:

- Classification models (which produce labels)
- Regression models (which produce numerical value predictions)

Evaluating Classifiers

		Predicted Class: Yes	Predicted Class: No
Actual Class	Yes	TP	FN
	No	FP	TN

There are four possible outcomes for a binary classifier:

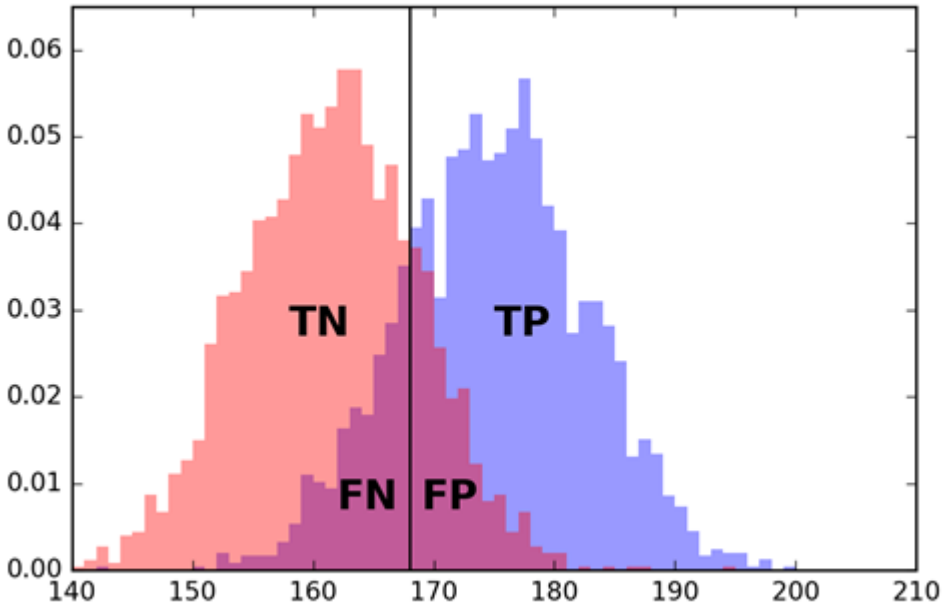
- True positives (TP) where + is labeled +
- True negative (TN) where - is labeled -
- False positives (FP) where - is labeled +
- False negatives (FN) where + is labeled -

Evaluating Classifiers

		Predicted Class	
		Yes	No
Actual Class	Yes	True Positives (TP)	False Negatives (FN)
	No	False Positives (FP)	True Negatives (TN)

Threshold Classifiers

Identifying the best threshold requires deciding on an appropriate evaluation metric.



Accuracy

The accuracy is the ratio of correct predictions over total predictions:

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

A monkey would randomly guess coin flips with p=0.5 correctly, with accuracy 50%.

Picking the biggest of 2 classes yields >=50%.

Precision

With imbalanced classes, accuracy is not helpful.

Imagine only 5% of tests indicate cancer:

Never diagnosing cancer is 95% accurate!

$$precision = \frac{TP}{(TP + FP)}$$

The monkey would achieve 5% precision,
as would a static cancer diagnosis.

Recall

In the cancer case, we would tolerate some false positive (scares) to identify real cases:

$$recall = \frac{TP}{(TP + FN)}$$

Recall measures being right on positive instances.

Saying everyone has cancer gives perfect recall!

F-Score

To get a meaningful single score balancing precision and recall use F-score:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

The harmonic mean is always less than/equal to the arithmetic mean, making it tough to get a high F-score. Weighted variants exist.

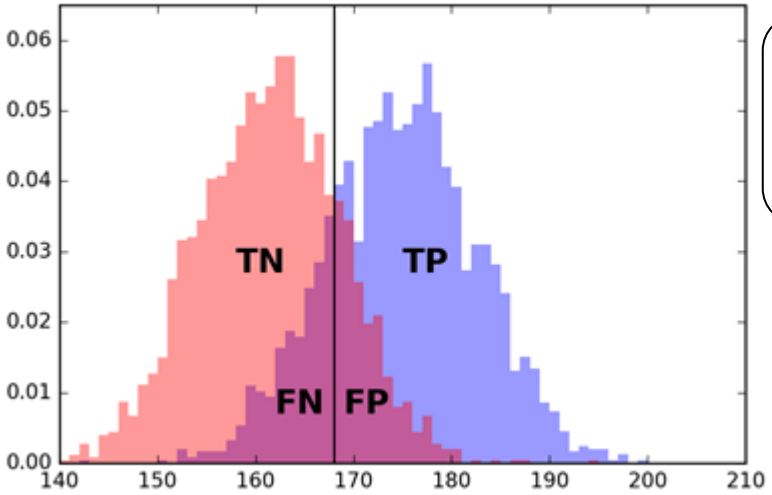
Take Away Lessons

- Accuracy is misleading when the class sizes are substantially different.
- High precision is very hard to achieve in unbalanced class sizes.
- F-score does the best job of any single statistic, but all four work together to describe the performance of a classifier.

Receiver-Operator (ROC) Curves

Varying the threshold changes recall/precision.

Area under ROC is a measure of accuracy.

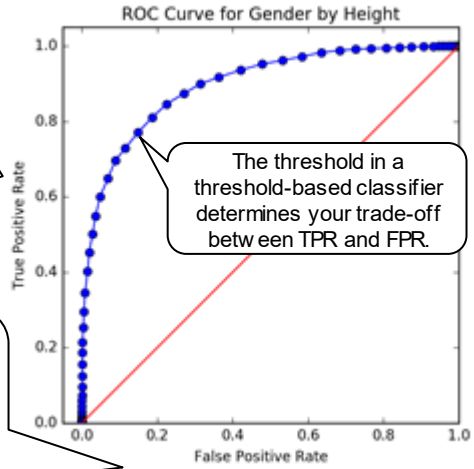


$$TPR = \frac{TP}{TP + FN}$$

(Did you get all the **true positives**?)

$$FPR = \frac{FP}{FP + TN}$$

(Did you get all the **false positives**?)



Evaluating Multiclass Systems

Classification gets harder with more classes.

The confusion matrix shows where the mistakes are being made: e.g. 5→3, 8→2

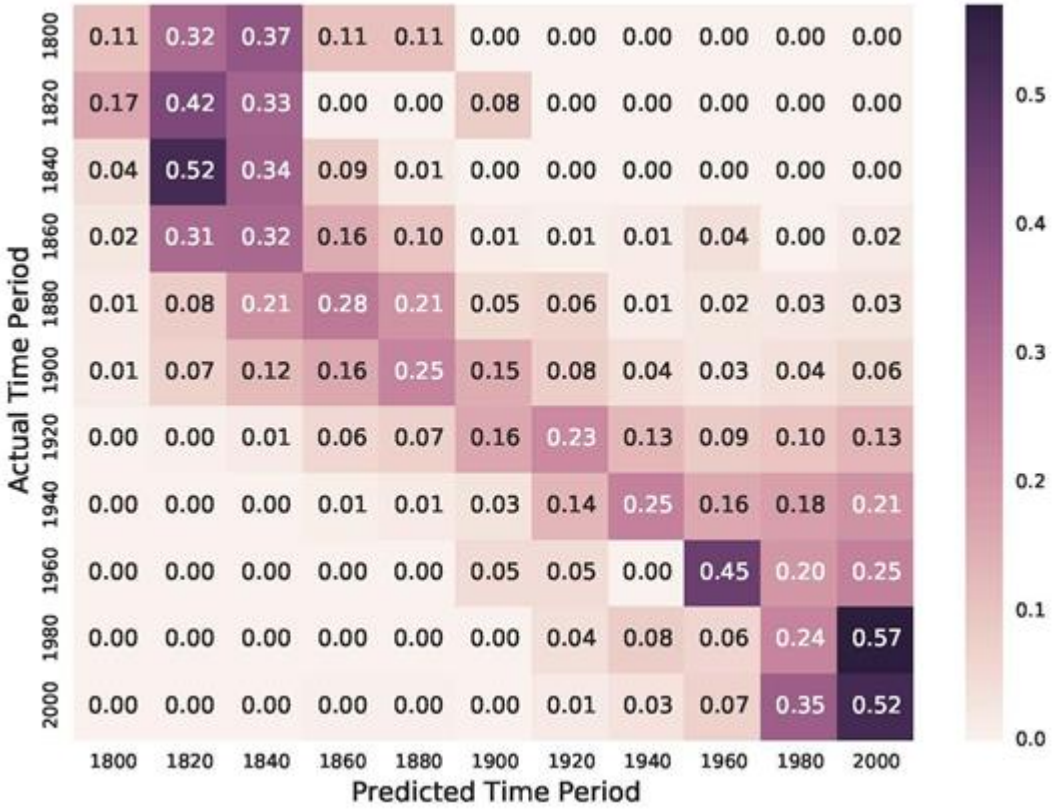
Digits	0	1	2	3	4	5	6	7	8	9
0	351	0	5	4	2	7	2	1	6	0
1	0	254	0	0	2	0	0	1	1	2
2	1	1	166	4	5	1	3	2	2	1
3	1	2	4	142	0	5	0	1	4	0
4	3	3	8	1	180	3	2	5	4	4
5	0	0	3	11	0	140	3	0	7	1
6	0	2	2	0	4	0	158	0	1	0
7	0	0	2	2	1	0	0	132	2	1
8	2	1	8	0	0	0	2	1	137	1
9	1	1	0	2	6	4	0	4	2	167

Figure 7.7: Confusion matrix for digits in a zip code OCR program.

Confusion Matrix: Dating Documents

What periods are most often confused with each other?

The main diagonal is not exactly where the heaviest weight always is.



Scoring Hard Problems Easier

Too low a classification rate is discouraging and often misleading with multiple classes.

The *top-k* success rate gives you credit if the right label would have been one of your first *k* guesses.

It is important to pick *k* so that real improvements can be recognized.

Summary Statistics: Numerical Error

For numerical values, error is a function of the delta between forecast *f* and observation *o*:

- Absolute error: *(f - o)*
- Relative error: *(f - o) / o* (typically better)

These can be aggregated over many tests:

- Mean or median squared error $\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2.$
- Root mean squared error $\mathrm{RMSD}(\hat{\theta}) = \sqrt{\mathrm{MSE}(\hat{\theta})} = \sqrt{\mathrm{E}((\hat{\theta} - \theta)^2)}.$

Evaluation Data

The best way to assess models involve out-of-sample predictions, results on data you never saw (or even better did not exist) when you built the model.

Partitioning the input into training (60%), testing (20%) and evaluation (20%) data works only if you never open evaluation data until the end.

Sins in Evaluation

Formal evaluation metrics reduce models to a few summary statistics.

But many problems can be hidden by statistics:

- Did I mix training and evaluation data?
- Do I have bugs in my implementation?

Revealing such errors requires understanding the types of errors your model makes.

Building an Evaluation Environment

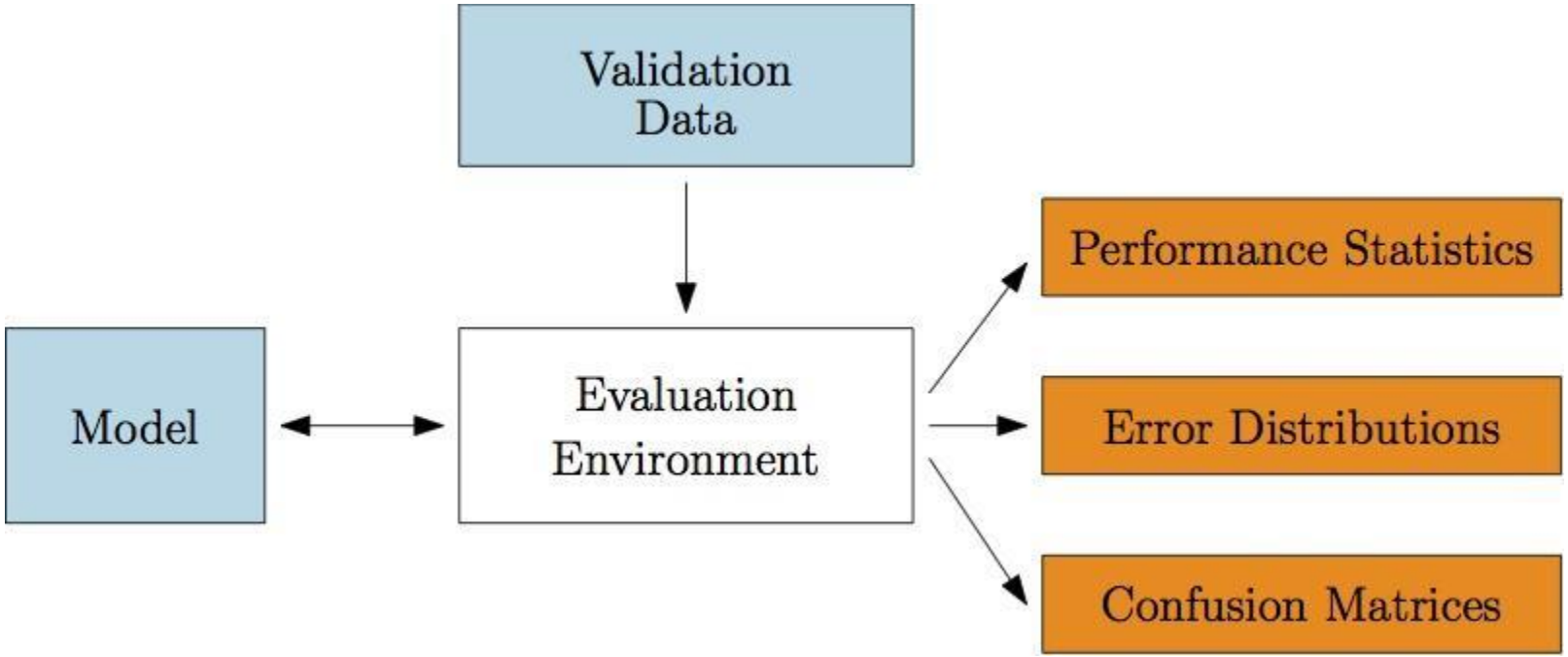
You want to have a *single-command* program to run your model on the evaluation data, and produce plots/reports on its effectiveness.

Input: evaluation data with outcome variables.

Embedded: function coding current model

Output: summary statistics and distributions of predictions on data vs. outcome variables.

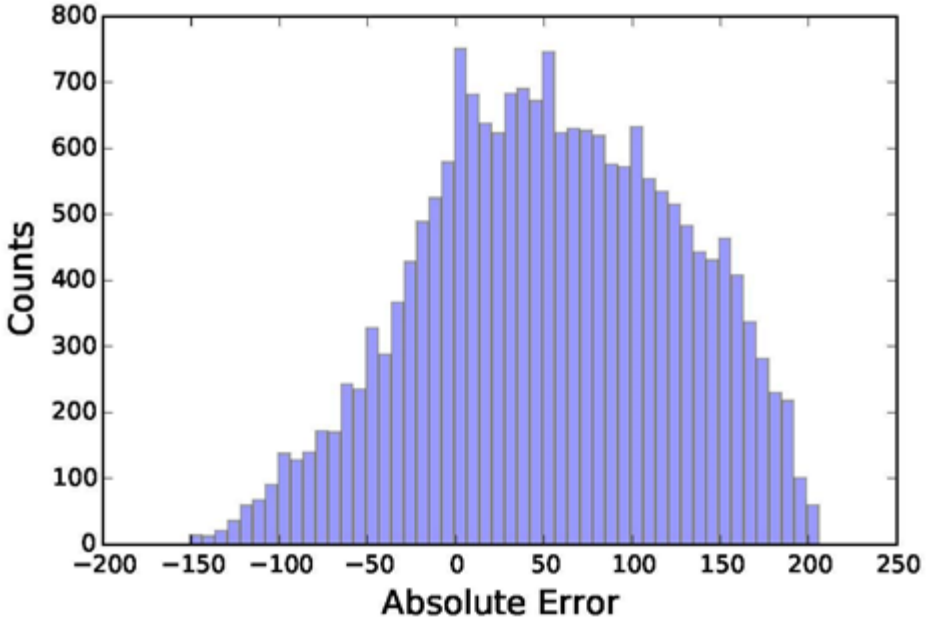
Evaluation Environment Architecture

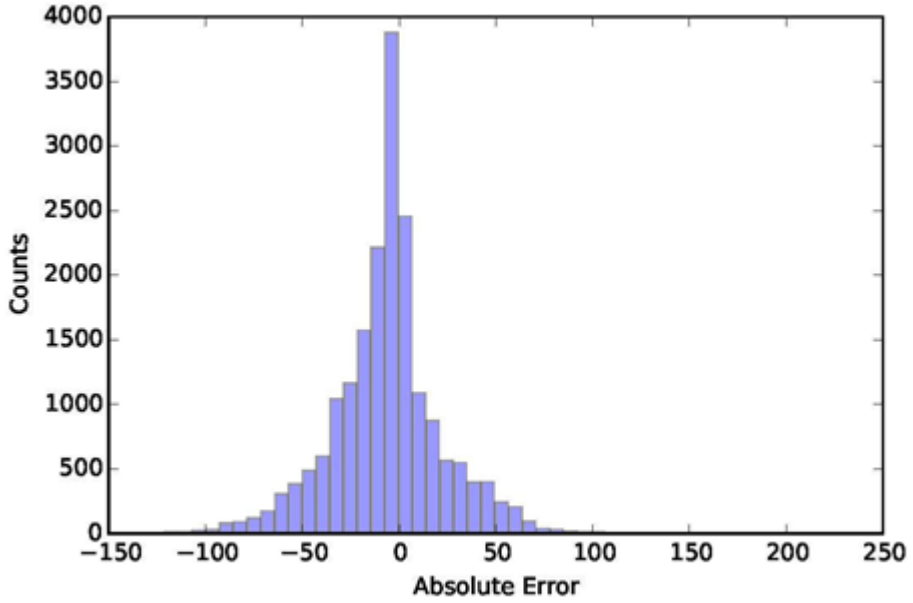


Designing Good Evaluation Systems

- Produce error distributions in addition to binary outcomes (how close was your prediction, not just right or wrong).
- Produce a report with multiple plots / distributions automatically, to read carefully.
- Output relevant summary statistics about performance to quickly gauge quality.

Error Histograms: Dating Documents





Performance of Random vs. Naive Bayes models

Evaluation Environment: Results Table

	Dataset	Method	MAE	MedAE	Acc
0	NYTimes	Random	73.335463	65.0	0.004895
1	COHA_Fiction_100	Random	79.865017	72.0	0.005287
2	COHA_Fiction_500	Random	80.505849	74.0	0.003825
3	COHA_Fiction_1000	Random	80.604837	72.0	0.003825
4	COHA_Fiction_2000	Random	79.845332	72.0	0.005737
5	COHA_News_100	Random	66.539239	59.0	0.005461
6	COHA_News_500	Random	66.267091	59.0	0.005461
7	COHA_News_1000	Random	66.077670	57.5	0.004956
8	COHA_News_2000	Random	66.225526	58.0	0.005057

random

	Dataset	Method	MAE	MedAE	Acc
0	NYTimes	NB	21.306301	14	0.029728
1	COHA_Fiction_100	NB	32.302025	22	0.041732
2	COHA_Fiction_500	NB	25.428234	14	0.050056
3	COHA_Fiction_1000	NB	23.493926	13	0.053656
4	COHA_Fiction_2000	NB	22.493363	12	0.054781
5	COHA_News_100	NB	19.384001	14	0.030845
6	COHA_News_500	NB	16.657565	12	0.034891
7	COHA_News_1000	NB	16.282261	12	0.035093
8	COHA_News_2000	NB	16.220065	12	0.035599

Naïve Bayes

Stratifying cases by topic and difficulty (length)

The Veil of Ignorance

A joke is not funny the second time because you already know the punchline.

Good performance on data you trained models on is very suspect, because models can easily be overfit.

Out of sample predictions are the key to being honest, if you have enough data/time for them.

Cross-Validation

Often we do not have enough data to separate training and evaluation data.

Train on *(k-1)/k* th of the data, evaluate on rest, then **repeat,** and average.

The win here is that you get a variance as to the accuracy of your model!

The limiting case is *leave one out validation.*

Amplifying Small Evaluation Sets

- **Create Negative Examples**: when positive examples are rare, all others are likely negative.
- **Perturb Real Examples**: This creates similar but synthetic ones by adding noise.
- **Give Partial Credit**: score by how far they are from the boundary, not just which side.

Blackbox vs. Descriptive Models

Ideally models are descriptive, meaning they explain why they are making their decisions.

Linear regression models are descriptive, because one can see which variables are weighed heaviest.

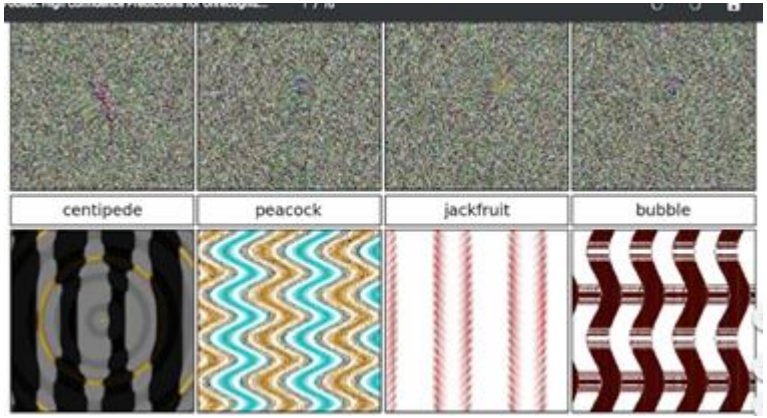
Neural network models are generally opaque.

War story: "Distinguishing cars from trucks."

Deep Learning Models are Blackbox

Deep learning models for computer vision are highly-effective, but opaque as to how they make decisions.

They can be badly fooled by images which would never confuse human observers.



Levels of Modeling

Interesting problems usually exist on several different levels, each of which require independent submodels.

Predicting the future price for a stock should involve submodels for analyzing (a) the general state of the economy, (b) its balance sheet, (c) the performance of its industrial sector, ...

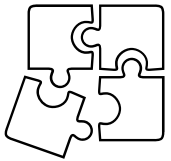
Hierarchical Decomposition

Imposing a hierarchical structure on the model permits it to be built and evaluated in a logical and transparent way, instead of as a black box.

Often subproblems lend themselves to theory-based, first-principle models, which can then be used as features in a data-driven general model.

Wrapup: Don‘t forget about Sophie‘s guest talk next week – please join!

Building & Evaluating Models



- Modelling is the process of using information to build a tool that lets you make predictions
- The key steps of modeling are building, fitting, and validating your model
- There are different measures for evaluating and comparing different models, such as accuracy, precision, recall, and F-score