Methods: Mind the Gap

Webinar Series

A Primer in Machine Learning in Epidemiology and Health Outcomes Research

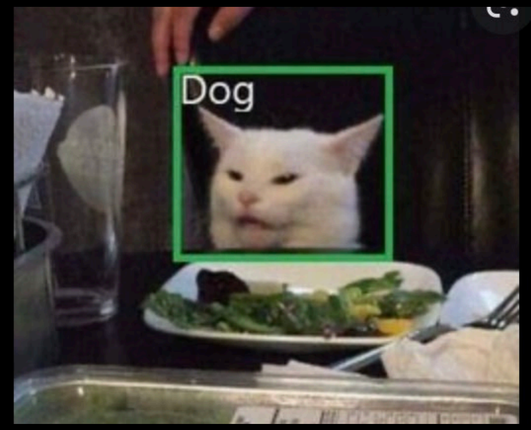
Presented by:

Timothy Wiemken, Ph.D., M.P.H.

Pfizer

Saint Louis University

National Institutes of Health
Office of Disease Prevention



Conflicts and Notes

I am an employee and shareholder of Pfizer Inc.

This presentation is not sponsored or affiliated with my employer and is solely my personal opinion and thoughts on the subject matter

I am not a theoretical statistician or mathematician

This is a story in three acts.

For the purposes of this presentation, we will focus on one type of machine learning: supervised machine learning

Act One

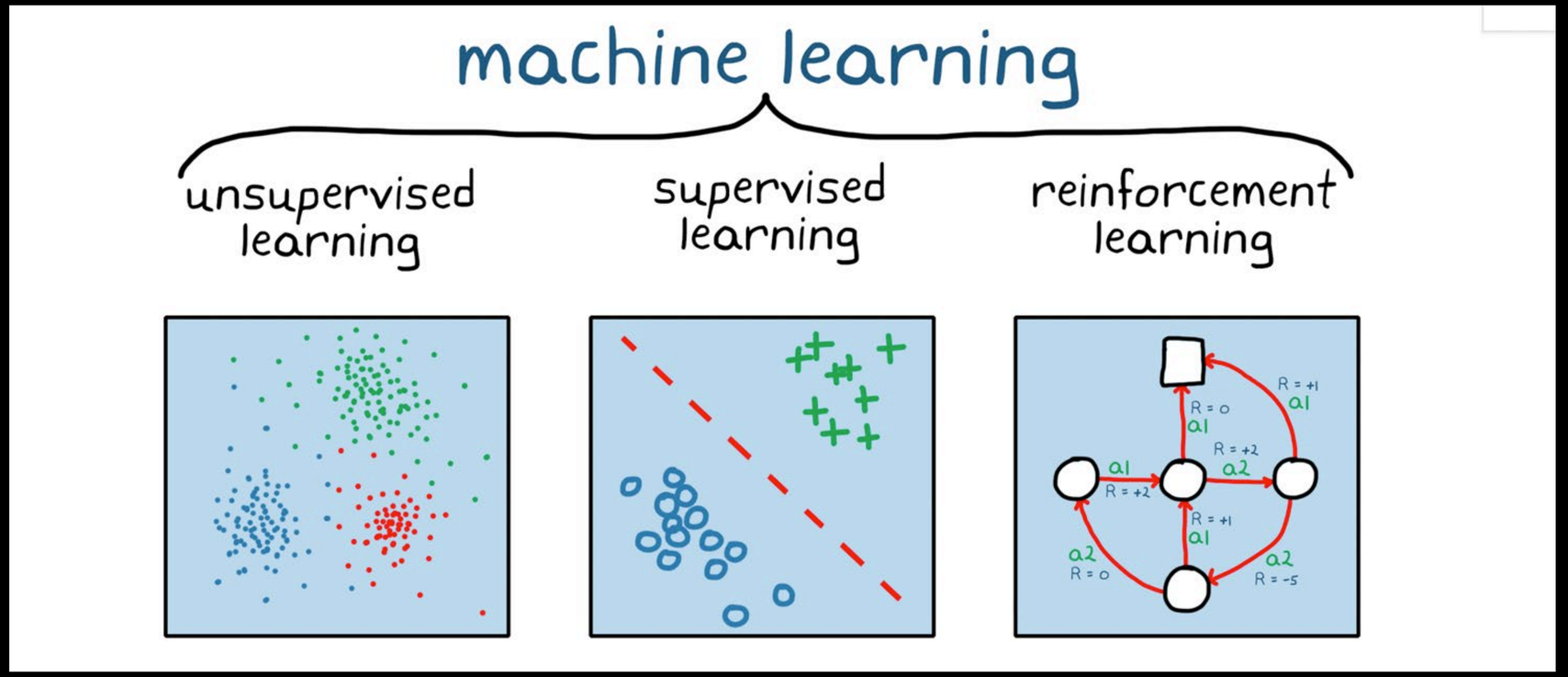
Source: https://stevetobak.com/2018/12/15/its-alive/

(but not really)

What is machine learning?

An umbrella term for several different *algorithms* that allow a *model* to be created - typically used to *predict* outcomes.

ML is not AI, but a single means to achieve AI



Source: https://www.mathworks.com/discovery/reinforcement-learning.html

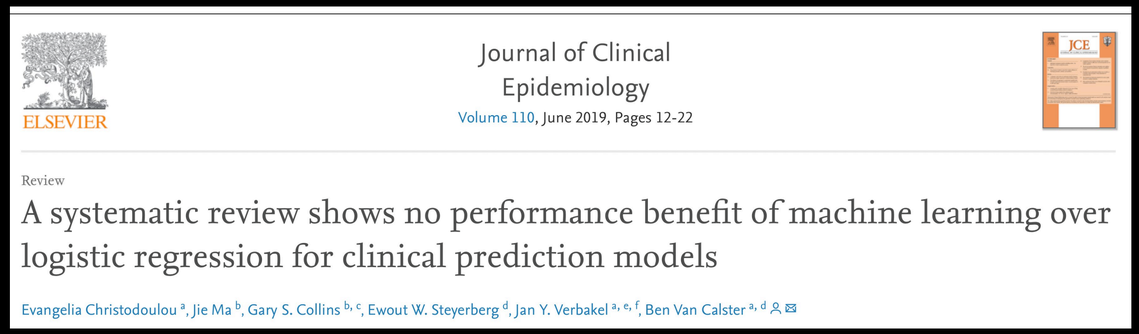
Term in Biostatistics	Term in Machine Learning
Outcome/Dependent/Response	Label
Independent/Predictor/Exposure/Covariate	Feature
Estimation/Fitting	Learning/Training
Regression	Regression or Classification (Supervised Learning)
Clustering/Principal Components/Data Reduction	Unsupervised Learning
Model Options	Hyperparameters

Is machine learning better than traditional biostatistics?

For typical use cases where you need to predict something…

Traditional methods work just as well as methods that we often call machine learning

…even though all ‘traditional’ regression models can be ‘machine learning’


ELSEVIER

Journal of Clinical Epidemiology

Volume 110, June 2019, Pages 12-22

Review

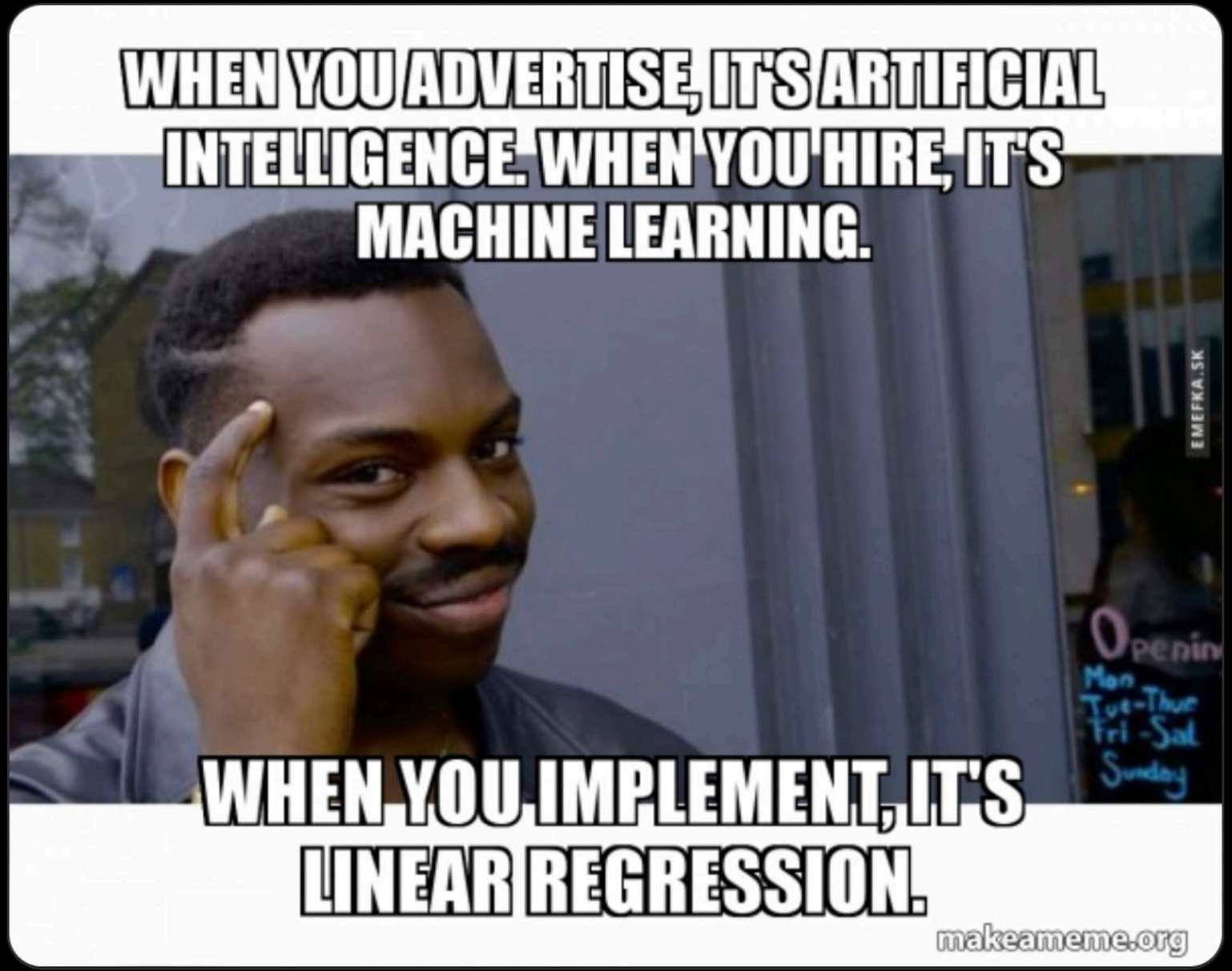
A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models

Evangelia Christodoulou [a], Jie Ma [b], Gary S. Collins [b, c], Ewout W. Steyerberg [d], Jan Y. Verbakel [a, e, f], Ben Van Calster [a, d]

A poor rule:

Inference = traditional methods

Predictions = ML (when nonlinear relationships or many more columns than rows)



Source: https://twitter.com/deeplearningai_/status/1282736549155799041?lang=da

How does it work?

Models are *trained* on a subset of data (training data), so the algorithm can recognize *patterns* related to the outcome of interest

New data are supplied to the trained model to get a prediction based on the recognition of similar patterns in the *testing* dataset

Full Dataset, 3107 rows, 10 columns

label	adi	region	division	length_of_life_rank	quality_of_life_rank	health_behaviors_rank	clinical_care_rank	social_economic_factors_rank	physical_environment_rank
0	91.73004	South	East South Central	4	8	9	11	5	54
0	82.78481	South	East South Central	5	2	6	4	3	2
0	132.54901	South	East South Central	52	60	65	36	60	34
0	115.78544	South	East South Central	31	35	55	44	50	50
0	104.74045	South	East South Central	24	13	15	54	18	51
0	132.47416	South	East South Central	34	63	63	32	62	38
0	111.32265	South	East South Central	51	54	57	51	57	25
0	107.24071	South	East South Central	40	21	22	25	28	29
0	112.29772	South	East South Central	29	47	42	29	39	48
1	102.14360	South	East South Central	33	23	29	43	22	64
0	110.95679	South	East South Central	16	29	41	47	27	57
0	118.72958	South	East South Central	44	49	58	13	29	40
0	121.57026	South	East South Central	57	58	60	45	55	65
0	113.04658	South	East South Central	48	39	53	50	40	4
0	103.40949	South	East South Central	20	17	26	60	21	41
0	101.34288	South	East South Central	10	7	13	34	8	12
0	103.27002	South	East South Central	30	22	16	20	16	5
0	121.47571	South	East South Central	63	59	47	62	54	22

Training: 80%

Testing: 20%

Training: 2485 rows, 10 columns

label	adi	region	division	length_of_life_rank	quality_of_life_rank	health_behaviors_rank	clinical_care_rank	social_economic_factors_rank	physical_environment_rank
0	91.73004	South	East South Central	4	8	9	11	5	54
0	82.78481	South	East South Central	5	2	6	4	3	2
0	132.54901	South	East South Central	52	60	65	36	60	34
0	104.74045	South	East South Central	24	13	15	54	18	51
0	132.47416	South	East South Central	34	63	63	32	62	38
0	111.32265	South	East South Central	51	54	57	51	57	25
0	107.24071	South	East South Central	40	21	22	25	28	29
0	112.29772	South	East South Central	29	47	42	29	39	48
1	102.14360	South	East South Central	33	23	29	43	22	64
0	110.95679	South	East South Central	16	29	41	47	27	57

Testing: 622 rows, 10 columns

label	adi	region	division	length_of_life_rank	quality_of_life_rank	health_behaviors_rank	clinical_care_rank	social_economic_factors_rank	physical_environment_rank
0	115.78544	South	East South Central	31	35	55	44	50	50
0	121.57026	South	East South Central	57	58	60	45	55	65
0	101.34288	South	East South Central	10	7	13	34	8	12
0	104.86499	South	East South Central	55	53	46	15	46	10
0	112.33730	South	East South Central	17	14	21	31	19	14
0	124.43065	South	East South Central	53	46	54	61	53	63
0	119.04478	South	East South Central	27	30	28	66	41	9
0	146.01690	South	East South Central	59	67	67	55	64	19
0	104.01849	South	East South Central	11	24	27	8	14	8

Full Dataset, 3107 rows, 10 columns

label	adi	region	division	length_of_life_rank	quality_of_life_rank	health_behaviors_rank	clinical_care_rank	social_economic_factors_rank	physical_environment_rank
0	91.73004	South	East South Central	4	8	9	11	5	54
0	82.78481	South	East South Central	5	2	6	4	3	2
0	132.54901	South	East South Central	52	60	65	36	60	34
0	115.78544	South	East South Central	31	35	55	44	50	50
0	104.74045	South	East South Central	24	13	15	54	18	51
0	132.47416	South	East South Central	34	63	63	32	62	38
0	111.32265	South	East South Central	51	54	57	51	57	25
0	107.24071	South	East South Central	40	21	22	25	28	29
0	112.29772	South	East South Central	29	47	42	29	39	48
1	102.14360	South	East South Central	33	23	29	43	22	64
0	110.95679	South	East South Central	16	29	41	47	27	57
0	118.72958	South	East South Central	44	49	58	13	29	40
0	121.57026	South	East South Central	57	58	60	45	55	65
0	113.04658	South	East South Central	48	39	53	50	40	4
0	103.40949	South	East South Central	20	17	26	60	21	41
0	101.34288	South	East South Central	10	7	13	34	8	12
0	103.27002	South	East South Central	30	22	16	20	16	5
0	121.47571	South	East South Central	63	59	47	62	54	22

Training: 80%

SET SEED!

Testing: 20%

SET SEED!

Training: 2485 rows, 10 columns

label	adi	region	division	length_of_life_rank	quality_of_life_rank	health_behaviors_rank	clinical_care_rank	social_economic_factors_rank	physical_environment_rank
0	91.73004	South	East South Central	4	8	9	11	5	54
0	82.78481	South	East South Central	5	2	6	4	3	2
0	132.54901	South	East South Central	52	60	65	36	60	34
0	104.74045	South	East South Central	24	13	15	54	18	51
0	132.47416	South	East South Central	34	63	63	32	62	38
0	111.32265	South	East South Central	51	54	57	51	57	25
0	107.24071	South	East South Central	40	21	22	25	28	29
0	112.29772	South	East South Central	29	47	42	29	39	48
1	102.14360	South	East South Central	33	23	29	43	22	64
0	110.95679	South	East South Central	16	29	41	47	27	57

Testing: 622 rows, 10 columns

label	adi	region	division	length_of_life_rank	quality_of_life_rank	health_behaviors_rank	clinical_care_rank	social_economic_factors_rank	physical_environment_rank
0	115.78544	South	East South Central	31	35	55	44	50	50
0	121.57026	South	East South Central	57	58	60	45	55	65
0	101.34288	South	East South Central	10	7	13	34	8	12
0	104.86499	South	East South Central	55	53	46	15	46	10
0	112.33730	South	East South Central	17	14	21	31	19	14
0	124.43065	South	East South Central	53	46	54	61	53	63
0	119.04478	South	East South Central	27	30	28	66	41	9
0	146.01690	South	East South Central	59	67	67	55	64	19
0	104.01849	South	East South Central	11	24	27	8	14	8

Train!

During training, need to tune hyperparameters for the model you select - They vary for each model.
Use cross validation
This is the art of ML

```
nnet_grid <- expand.grid(
  decay = c(0.5, 1e-2, 1e-3),
  size = c(3,5,10,20))

nn <- train( label ~ .,
             method     = "nnet",
             trControl  = trainControl(method="cv",10
             data       = train,
             tuneGrid = nnet_grid,
             verbose = FALSE)
```

Full Dataset, 3107 rows, 10 columns

label	adi	region	division	length_of_life_rank	quality_of_life_rank	health_behaviors_rank	clinical_care_rank	social_economic_factors_rank	physical_environment_rank
0	91.73004	South	East South Central	4	8	9	11	5	54
0	82.78481	South	East South Central	5	2	6	4	3	2
0	132.54901	South	East South Central	52	60	65	36	60	34
0	115.78544	South	East South Central	31	35	55	44	50	50
0	104.74045	South	East South Central	24	13	15	54	18	51
0	132.47416	South	East South Central	34	63	63	32	62	38
0	111.32265	South	East South Central	51	54	57	51	57	25
0	107.24071	South	East South Central	40	21	22	25	28	29
0	112.29772	South	East South Central	29	47	42	29	39	48
1	102.14360	South	East South Central	33	23	29	43	22	64
0	110.95679	South	East South Central	16	29	41	47	27	57
0	118.72958	South	East South Central	44	49	58	13	29	40
0	121.57026	South	East South Central	57	58	60	45	55	65
0	113.04658	South	East South Central	48	39	53	50	40	4
0	103.40949	South	East South Central	20	17	26	60	21	41
0	101.34288	South	East South Central	10	7	13	34	8	12
0	103.27002	South	East South Central	30	22	16	20	16	5
0	121.47571	South	East South Central	63	59	47	62	54	22

Training: 80%

SET SEED!

Testing: 20%

SET SEED!

Training: 2485 rows, 10 columns

label	adi	region	division	length_of_life_rank	quality_of_life_rank	health_behaviors_rank	clinical_care_rank	social_economic_factors_rank	physical_environment_rank
0	91.73004	South	East South Central	4	8	9	11	5	54
0	82.78481	South	East South Central	5	2	6	4	3	2
0	132.54901	South	East South Central	52	60	65	36	60	34
0	104.74045	South	East South Central	24	13	15	54	18	51
0	132.47416	South	East South Central	34	63	63	32	62	38
0	111.32265	South	East South Central	51	54	57	51	57	25
0	107.24071	South	East South Central	40	21	22	25	28	29
0	112.29772	South	East South Central	29	47	42	29	39	48
1	102.14360	South	East South Central	33	23	29	43	22	64
0	110.95679	South	East South Central	16	29	41	47	27	57

Testing: 622 rows, 10 columns

label	adi	region	division	length_of_life_rank	quality_of_life_rank	health_behaviors_rank	clinical_care_rank	social_economic_factors_rank	physical_environment_rank
0	115.78544	South	East South Central	31	35	55	44	50	50
0	121.57026	South	East South Central	57	58	60	45	55	65
0	101.34288	South	East South Central	10	7	13	34	8	12
0	104.86499	South	East South Central	55	53	46	15	46	10
0	112.33730	South	East South Central	17	14	21	31	19	14
0	124.43065	South	East South Central	53	46	54	61	53	63
0	119.04478	South	East South Central	27	30	28	66	41	9
0	146.01690	South	East South Central	59	67	67	55	64	19
0	104.01849	South	East South Central	11	24	27	8	14	8

Train!

During training, need to tune hyperparameters for the model you select - They vary for each model.
Use cross validation
This is the art of ML

Find your parameters and make a new ‘final’ model

```
  decay  size  Accuracy   Kappa
  0.001    3   0.7474843  0.2676984
  0.001    5   0.7766014  0.4303814
  0.001   10   0.7585349  0.3545977
  0.001   20   0.7646942  0.3834182
  0.010    3   0.7901363  0.4667154
  0.010    5   0.7712601  0.4144783
  0.010   10   0.7856364  0.4514401
  0.010   20   0.7621965  0.3701082
  0.500    3   0.7815195  0.4268364
  0.500    5   0.7901295  0.4524565
  0.500   10   0.7819462  0.4325758
  0.500   20   0.7593629  0.3727550

Accuracy was used to select the optimal model using the largest value.
The final values used for the model were size = 3 and decay = 0.01.
```

```
nnet_grid <- expand.grid(
  decay = c(0.5, 1e-2, 1e-3),
  size = c(3,5,10,20))

nn <- train( label ~ .,
             method     = "nnet",
             trControl  = trainControl(method="cv",10
             data       = train,
             tuneGrid = nnet_grid,
             verbose = FALSE)
```

Pass testing data to model to get predictions for each test case and check accuracy of your model

yo <- predict(nn, newdata = test, type = "prob")

Row in test data

```
> yo
           Low       High
4   0.90707279 0.09292721
13  0.90710474 0.09289526
16  0.75055419 0.24944581
19  0.59300836 0.40699164
23  0.89825851 0.10174149
27  0.91580503 0.08419497
30  0.91518679 0.08481321
32  0.91726343 0.08273657
35  0.69077196 0.30922804
37  0.27883309 0.72116691
41  0.13079429 0.86920571
```

Predicted Probability of
“Low Uptake” (<50%)
vs
“High Uptake (>=50%)

Compare probability at some cutoff (usually 50%) to actual outcome
(this is why it is supervised! It learned from the known outcome in training)

```
> confusionMatrix(test$pred, test$label, positive = "High")
Confusion Matrix and Statistics

          Reference
Prediction Low High
      Low   33   86
      High 390  107

               Accuracy : 0.2273
                 95% CI : (0.1947, 0.2624)
    No Information Rate : 0.6867
    P-Value [Acc > NIR] : 1

                  Kappa : -0.2574

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.55440
            Specificity : 0.07801
         Pos Pred Value : 0.21529
         Neg Pred Value : 0.27731
             Prevalence : 0.31331
         Detection Rate : 0.17370
   Detection Prevalence : 0.80682
      Balanced Accuracy : 0.31621

       'Positive' Class : High
```

No balance in outcome
easier to predict ‘no’ than look for patterns.

```
> table(train$label)

 Low High
1713  722
```

~30% ‘High’

Downsample/SMOTE and re-train

Compare probability at some cutoff (usually 50%) to actual outcome
(this is why it is supervised! It learned from the known outcome in training)

```
> confusionMatrix(test$pred, test$label, positive = "High")
Confusion Matrix and Statistics

          Reference
Prediction Low High
      Low   33   86
      High 390  107

               Accuracy : 0.2273
                 95% CI : (0.1947, 0.2624)
    No Information Rate : 0.6867
    P-Value [Acc > NIR] : 1

                  Kappa : -0.2574

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.55440
            Specificity : 0.07801
         Pos Pred Value : 0.21529
         Neg Pred Value : 0.27731
             Prevalence : 0.31331
         Detection Rate : 0.17370
   Detection Prevalence : 0.80682
      Balanced Accuracy : 0.31621

       'Positive' Class : High
```

No balance in outcome
easier to predict ‘no’ than look for patterns.

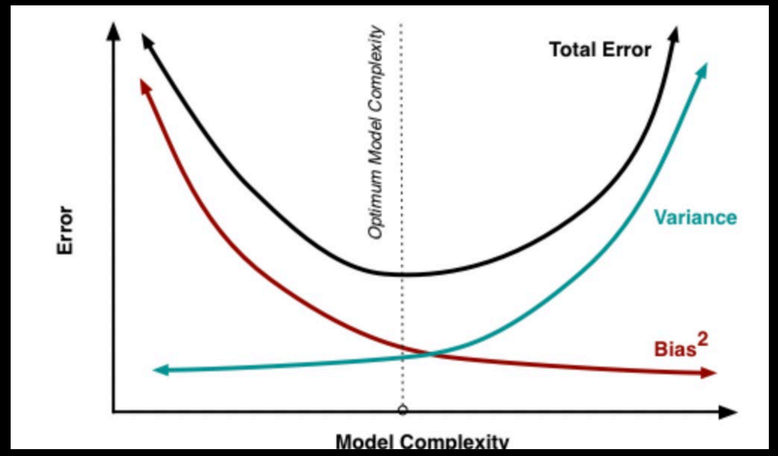
```
> table(train$label)

 Low High
1713  722
```

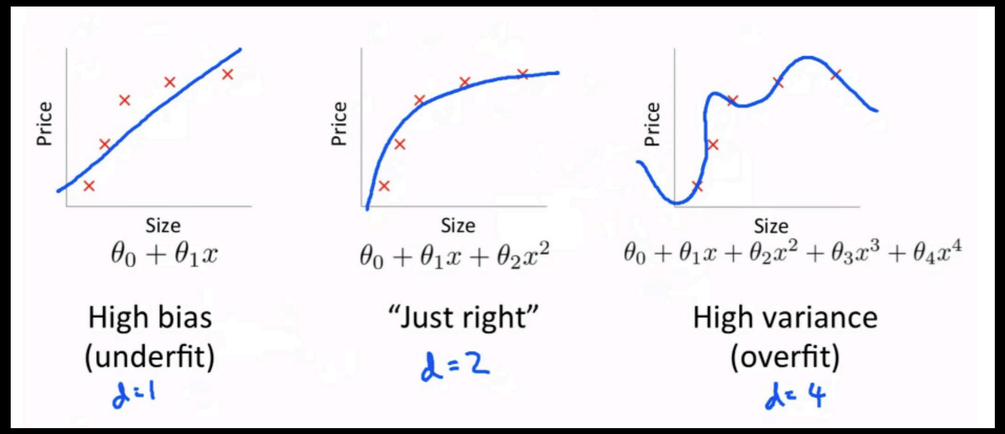
~30% ‘High’

Downsampled and re-trained - tested on same test set…

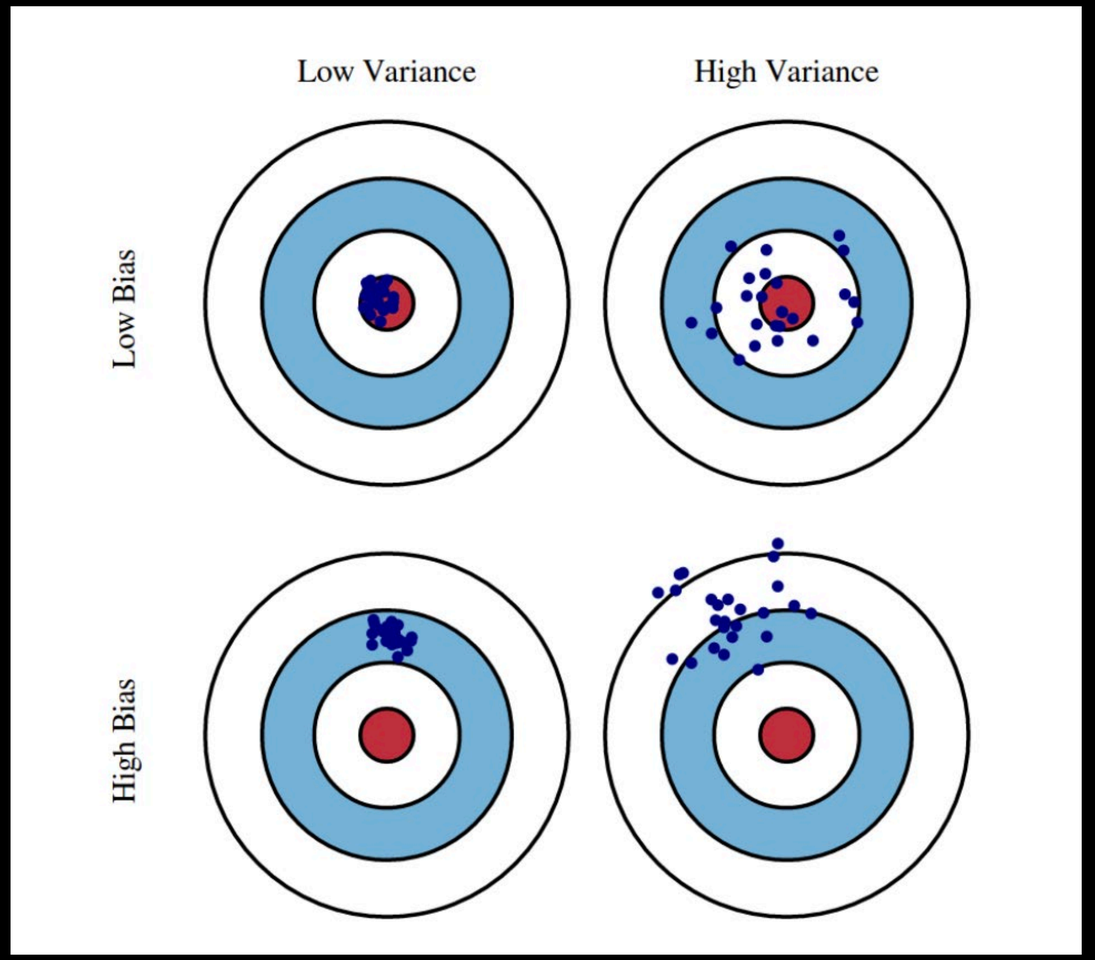
```
Confusion Matrix and Statistics

          Reference
Prediction Low High
      Low   18   20
      High  47    9

               Accuracy : 0.2872
                 95% CI : (0.1986, 0.3898)
    No Information Rate : 0.6915
    P-Value [Acc > NIR] : 1.000000

                  Kappa : -0.3281

 Mcnemar's Test P-Value : 0.001491

            Sensitivity : 0.31034
            Specificity : 0.27692
         Pos Pred Value : 0.16071
         Neg Pred Value : 0.47368
             Prevalence : 0.30851
         Detection Rate : 0.09574
   Detection Prevalence : 0.59574
      Balanced Accuracy : 0.29363

       'Positive' Class : High
```

Downsample/SMOTE and re-train





Source: https://towardsdatascience.com/this-thing-called-weight-decay-a7cd4bcfccab



Source: http://scott.fortmann-roe.com/docs/BiasVariance.html

Bias = Don't predict what you want (maybe not enough data to recognize patterns)
Variance = Too much noise (maybe too much noisy data)

Too complex of a model with too much data = tough to explain anyway

Act Two

Source: https://www.pinterest.com/pin/525373112778996159/

Feature Engineering

Source: https://www.simpo.io/blog/simpo-engineering-yes-we-code-and-much-more

With ML - the more data, the better - but don’t overfit.

Feature engineering for dates

Day of week
Week of year
Month, Day, Quarter, Year
Beginning of month/end of month
Frequency of each item above
Proportion of time through the month (since different n days in a month!)
Time (H/M/S)
Season of year
Time since some index

Use this

To Create These

date	month	month.freq	year	year.freq	dow	dow.freq	quarter	quarter.freq	season	days.since.pandemic	prop.through.month
2021-01-01	1	62	2021	365	Friday	105	1	180	winter	289 days	3.225806
2021-01-02	1	62	2021	365	Saturday	105	1	180	winter	290 days	6.451613
2021-01-03	1	62	2021	365	Sunday	104	1	180	winter	291 days	9.677419
2021-01-04	1	62	2021	365	Monday	104	1	180	winter	292 days	12.903226
2021-01-05	1	62	2021	365	Tuesday	104	1	180	winter	293 days	16.129032
2021-01-06	1	62	2021	365	Wednesday	104	1	180	winter	294 days	19.354839
2021-01-07	1	62	2021	365	Thursday	104	1	180	winter	295 days	22.580645
2021-01-08	1	62	2021	365	Friday	105	1	180	winter	296 days	25.806452
2021-01-09	1	62	2021	365	Saturday	105	1	180	winter	297 days	29.032258
2021-01-10	1	62	2021	365	Sunday	104	1	180	winter	298 days	32.258065
2021-01-11	1	62	2021	365	Monday	104	1	180	winter	299 days	35.483871
2021-01-12	1	62	2021	365	Tuesday	104	1	180	winter	300 days	38.709677
2021-01-13	1	62	2021	365	Wednesday	104	1	180	winter	301 days	41.935484
2021-01-14	1	62	2021	365	Thursday	104	1	180	winter	302 days	45.161290
2021-01-15	1	62	2021	365	Friday	105	1	180	winter	303 days	48.387097
2021-01-16	1	62	2021	365	Saturday	105	1	180	winter	304 days	51.612903
2021-01-17	1	62	2021	365	Sunday	104	1	180	winter	305 days	54.838710

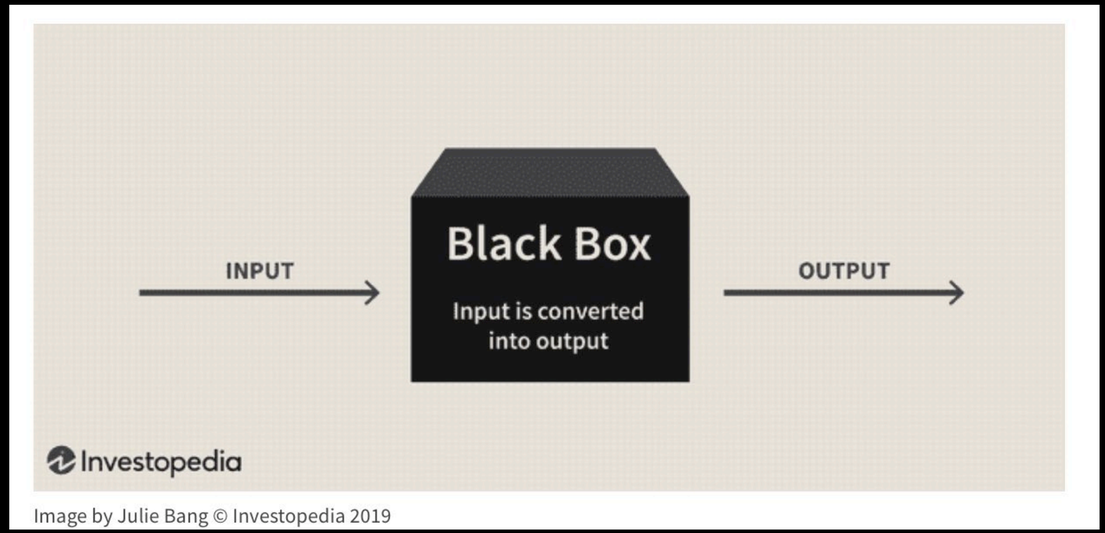
Feature engineering for missing data

label	adi	region	division	length_of_life_rank	quality_of_life_rank	health_behaviors_rank	clinical_care_rank	social_economic_factors_rank	physical_environment_rank
0	91.73004	South	East South Centra	*NA*	8	9	11	5	54
NA	*NA*	South	East South Centra	*NA*	2	6	4	3	*NA*
NA	*NA*	South	East South Centra	52	60	65	36	60	*NA*
NA	*NA*	South	*NA*	*NA*	*NA*	55	*NA*	*NA*	50
0	*NA*	South	*NA*	24	13	15	54	18	*NA*
0	*NA*	South	*NA*	34	*NA*	*NA*	*NA*	*NA*	38
NA	*NA*	*NA*	East South Centra	51	54	57	51	57	*NA*
0	107.24071	South	*NA*	40	21	22	*NA*	28	*NA*
NA	112.29772	South	East South Centra	29	47	*NA*	29	39	*NA*
NA	102.14360	*NA*	East South Centra	33	23	29	*NA*	22	64
NA	110.95679	*NA*	*NA*	16	29	*NA*	47	27	57
0	118.72958	South	East South Centra	44	*NA*	58	13	29	40
NA	*NA*	South	East South Centra	57	*NA*	*NA*	45	55	65
0	113.04658	*NA*	East South Centra	48	39	53	*NA*	40	*NA*
0	103.40949	*NA*	East South Centra	20	17	26	*NA*	21	41

label	length_of_life_rank	lol.indicator	lol_mean.imp	lol_freq_miss
0	*NA*	1	48.1523	33.06370
NA	*NA*	1	48.1523	33.06370
NA	52	0	52.0000	33.06370
NA	*NA*	1	48.1523	33.06370
0	24	0	24.0000	33.06370
0	34	0	34.0000	33.06370
NA	51	0	51.0000	29.50644
0	40	0	40.0000	33.06370
NA	29	0	29.0000	33.06370
NA	33	0	33.0000	29.50644
NA	16	0	16.0000	29.50644
0	44	0	44.0000	33.06370
NA	57	0	57.0000	33.06370
0	48	0	48.0000	29.50644
0	20	0	20.0000	29.50644
0	*NA*	1	48.1523	33.06370

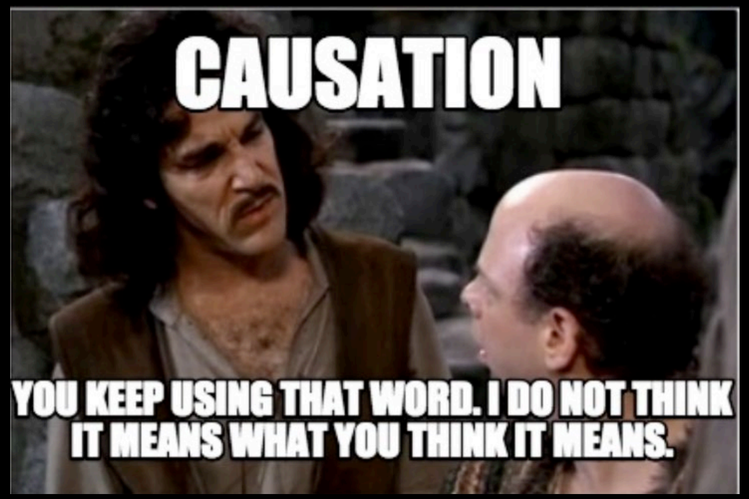
- **Binary indicator of missingness**
- **Imputation of missing data with some function**
- **Impute with frequency of each unique group if categorical**
- **Regression/random forest/etc imputation**
- **Clusters, PCA**
- **Very complex functions!**

1) Reduce the data to a usable set.
2) Use a model that reduces data for you



Source: Julie Bang from Investopedia 2019.

Act Three



Source: https://stevetobak.com/2018/12/15/its-alive/

Causal inference in machine learning

“Causal” is a colloquial use here.

ML models such as causal forests allow you to extract treatment effects (think absolute risk differences), even across a multitude of subgroups.

This is generally not possible in traditional statistics since we are limited by sample size and multiple comparisons false positive rate inflations

A Randomized Study Evaluating the Effectiveness of Oseltamivir Initiated at the Time of Hospital Admission in Adults Hospitalized With Influenza-Associated Lower Respiratory Tract Infections

Julio Ramirez [1], Paula Peyrani [1], Timothy Wiemken [2], Sandra S Chaves [3], Alicia M Fry [3]

Affiliations + expand

PMID: 29659754 DOI: 10.1093/cid/ciy163

Standard of care = 24% clinical failure
Oseltamivir + SoC = 15% clinical failure

P = 0.414

Conclusions: Initiation of oseltamivir more than 5 days after illness onset did not reduce clinical failures among hospitalized patients with I- LRTIs. However, we did not enroll our projected sample size of I-LRTI.

A Randomized Study Evaluating the Effectiveness of Oseltamivir Initiated at the Time of Hospital Admission in Adults Hospitalized With Influenza-Associated Lower Respiratory Tract Infections

Julio Ramirez [1], Paula Peyrani [1], Timothy Wiemken [2], Sandra S Chaves [3], Alicia M Fry [3]

Affiliations + expand

PMID: 29659754 DOI: 10.1093/cid/ciy163

Standard of care = 24% clinical failure
Oseltamivir + SoC = 15% clinical failure

P = 0.414

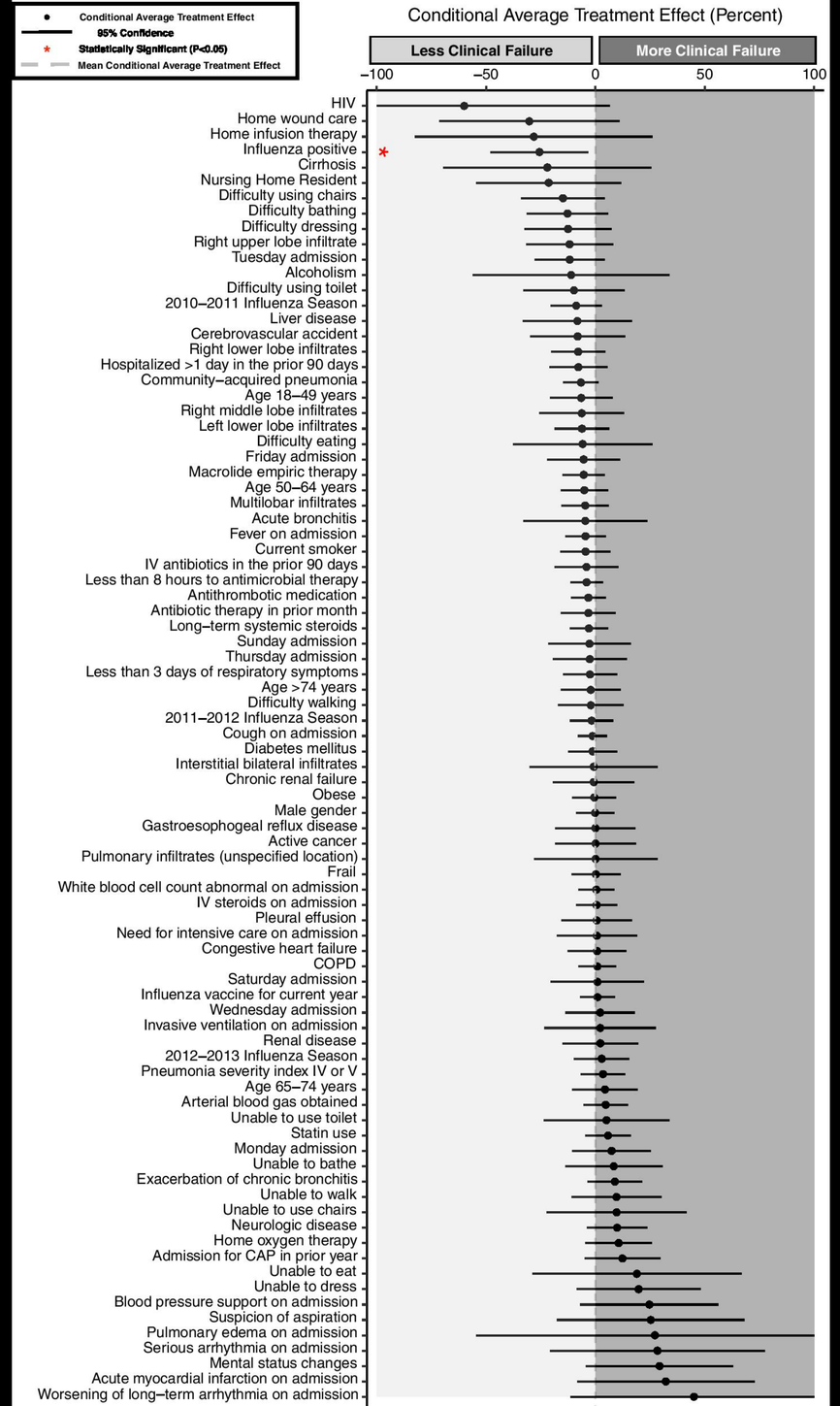
Conclusions: Initiation of oseltamivir more than 5 days after illness onset did not reduce clinical failures among hospitalized patients with I- LRTIs. However, we did not enroll our projected sample size of I-LRTI.

Issue = enrolled all patients with LRTI, not just influenza (18% of LRTI in hospital during flu season)

Effectiveness of oseltamivir treatment on clinical failure in hospitalized patients with lower respiratory tract infection

Timothy L. Wiemken ✉, Stephen P. Furmanek, Ruth M. Carrico, Paula Peyrani, Daniel Hoft, Alicia M. Fry & Julio A. Ramirez

BMC Infectious Diseases **21**, Article number: 1106 (2021) | Cite this article



Epilogue

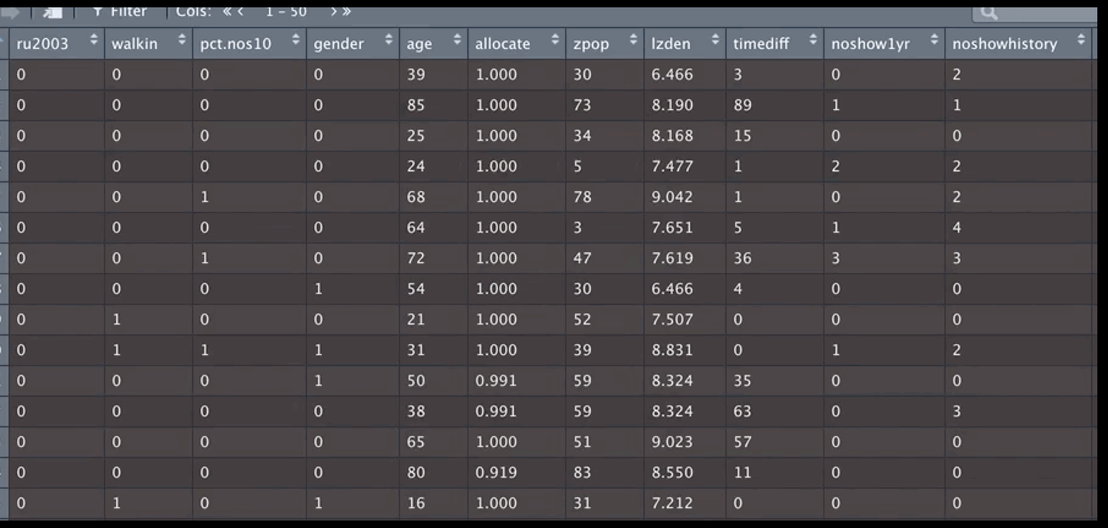
Source: https://tenor.com/view/boy-that-escalated-quickly-ron-burgundy-will-ferrell-meme-gif-17035028

Automation in ML

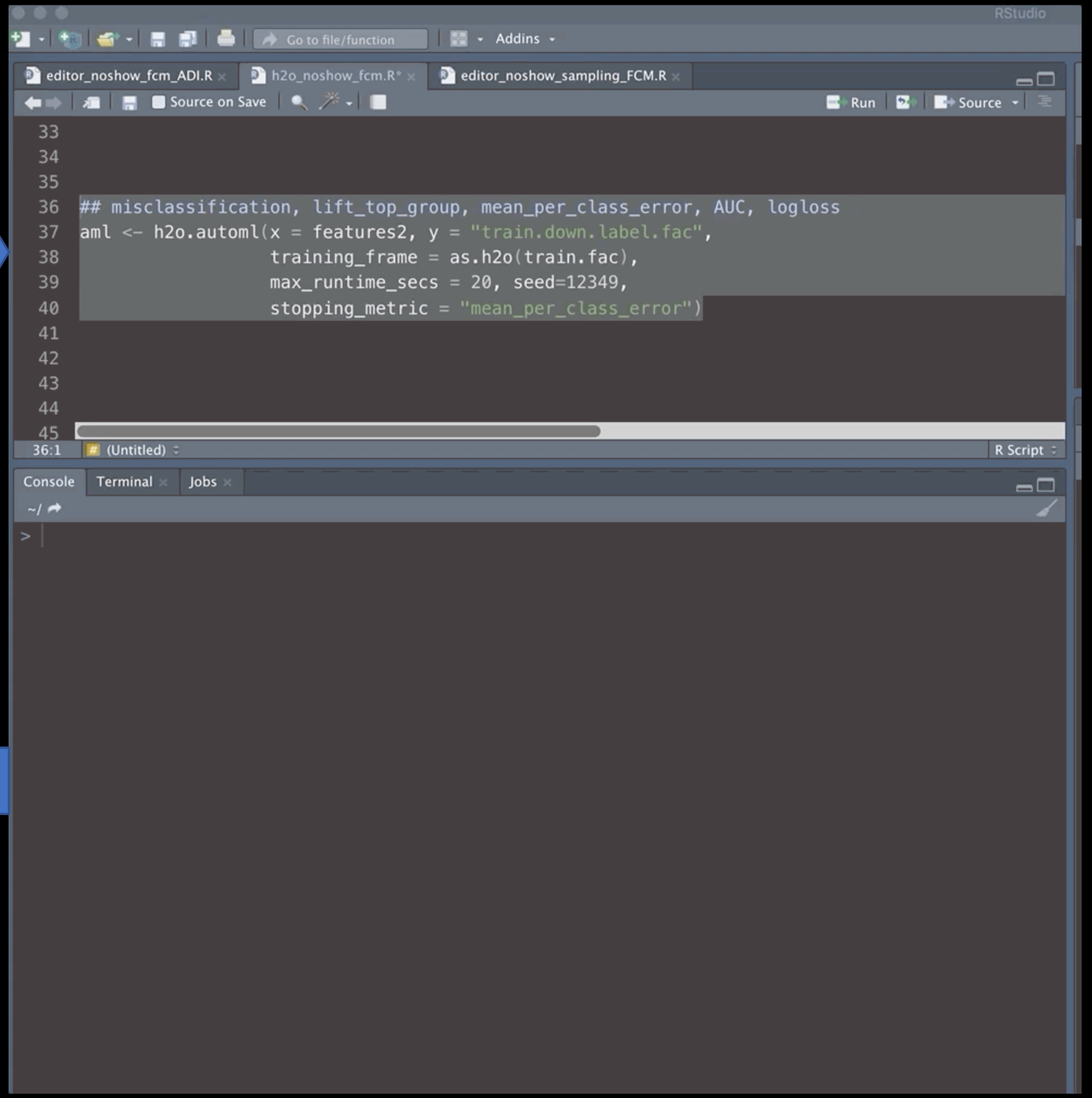
Selecting and building models has become essentially automated for typical use cases.

H2O, Keras/Tensorflow, and many other open-source frameworks will automate training, testing, validation, and even feature engineering.

Feature engineering is still probably the most important piece of ML. Creativity is key.



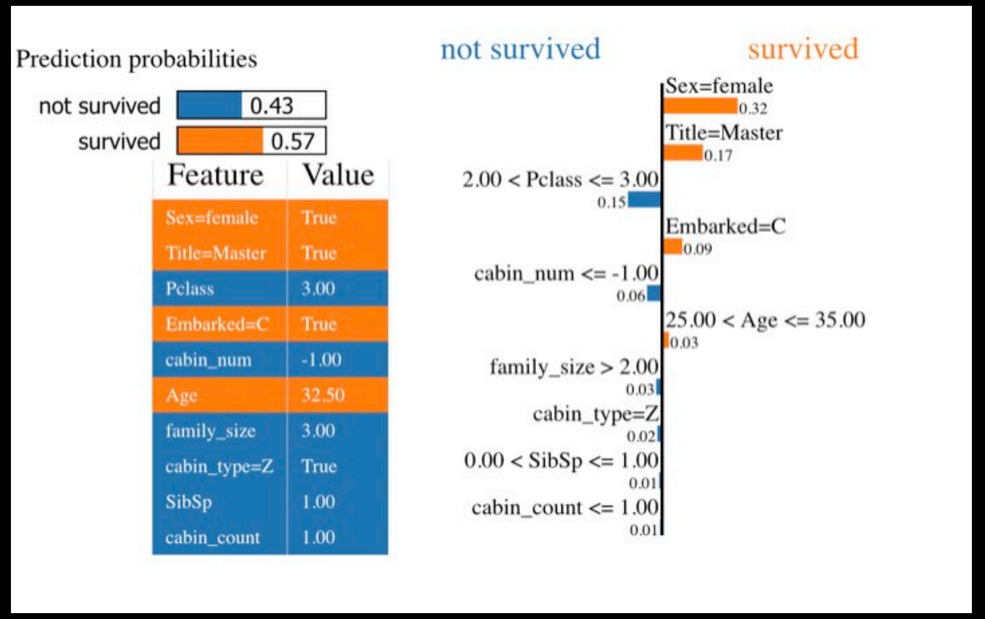
ru2003	walkin	pct.nos10	gender	age	allocate	zpop	lzden	timediff	noshow1yr	noshowhistory
0	0	0	0	39	1.000	30	6.466	3	0	2
0	0	0	0	85	1.000	73	8.190	89	1	1
0	0	0	0	25	1.000	34	8.168	15	0	0
0	0	0	0	24	1.000	5	7.477	1	2	2
0	0	1	0	68	1.000	78	9.042	1	0	2
0	0	0	0	64	1.000	3	7.651	5	1	4
0	0	1	0	72	1.000	47	7.619	36	3	3
0	0	0	1	54	1.000	30	6.466	4	0	0
0	1	0	0	21	1.000	52	7.507	0	0	0
0	1	1	1	31	1.000	39	8.831	0	1	2
0	0	0	1	50	0.991	59	8.324	35	0	0
0	0	0	0	38	0.991	59	8.324	63	0	3
0	0	0	0	65	1.000	51	9.023	57	0	0
0	0	0	0	80	0.919	83	8.550	11	0	0
0	1	0	1	16	1.000	31	7.212	0	0	0




```
          Sensitivity : 0.83088
          Specificity : 0.77869
       Pos Pred Value : 0.29504
       Neg Pred Value : 0.97636
            Prevalence : 0.10029
        Detection Rate : 0.08333
  Detection Prevalence : 0.28245
      Balanced Accuracy : 0.80479

       'Positive' Class : noshow
```

De-black box with Explainers

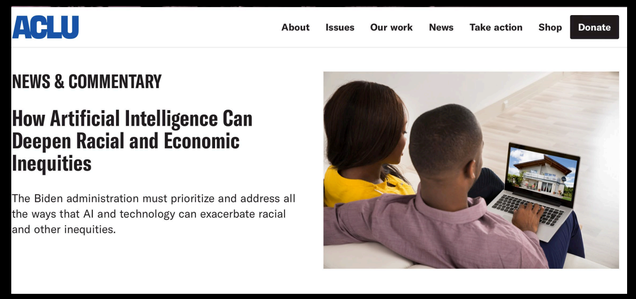
1. *survival*: 0 = No, 1 = Yes
2. *pclass*: Ticket class (1 = 1st, 2 = 2nd, 3 = 3rd)
3. *sex*: Sex
4. *Age*: Age in years
5. *sibsp*: number of siblings / spouses aboard the Titanic
6. *parch*: number of parents / children aboard the Titanic
7. *ticket*: Ticket number
8. *fare*: Passenger fare
9. *cabin*: Cabin number
10. *embarked*: Port of Embarkation (C = Cherbourg, Q = Queenstown, S = Southampton)

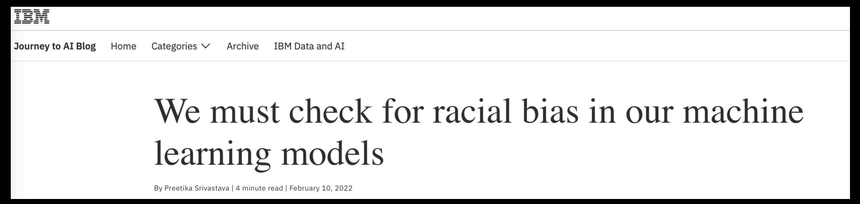


Source: https://teemukanstren.com/2020/06/10/explaining-machine-learning-classifiers-with-lime/

Biggest issue: Unmeasured data

Models can rapidly become extremely biased as it is just an algorithm looking at patterns in data the scientist included in the training dataset





Additional Reading

Wiemken TL, Kelley RR. Machine Learning in Epidemiology and Health Outcomes Research. *Annual Reviews in Public Health. 2019, 41:21-36.* https://www.annualreviews.org/doi/abs/10.1146/annurev-publhealth-040119-094437

Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning Second Edition, 2017: https://hastie.su.domains/ElemStatLearn/printings/ESLII_print12_toc.pdf

James G, Witten D, Hastie T, Tibshirani R. An Introduction to Statistical Learning. Second Edition. https://www.statlearning.com