Machine Learning in the Organizational Sciences: From There to Here and What's Next

Andrew Speer

February 2024

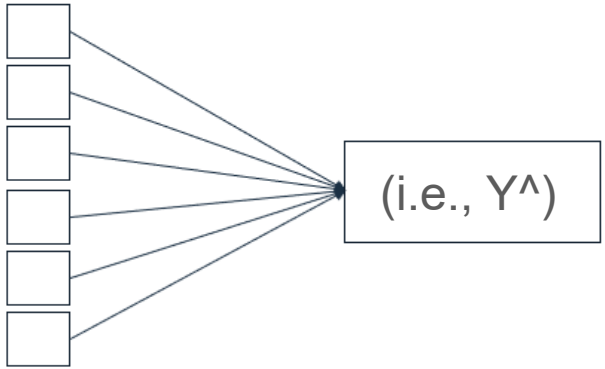
Today's Agenda

- Intro to modern prediction methods and how we got where we are today
- In what settings are modern prediction methods most advantageous in the organizational sciences
- Implications of new tools like large language models

Quick Overview of Machine Learning

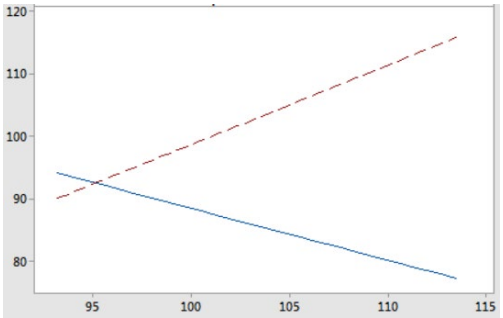
Let's Predict Something

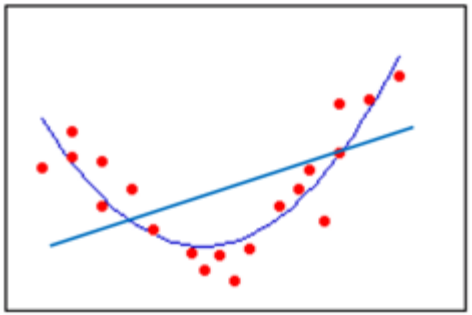
- We often want to predict some outcome (i.e., DV, target)
  - E.g., Job performance, turnover
- Prior to the whole "ML" craze, how was this accomplished?
  - Regression (e.g., ordinary least squares or logistic)
  - Unit-weighting
  - Etc.
- But there are limitations to methods like these…



Some Difficulties with Ordinary Least Squares (OLS)

- Underfitting: If nonlinear or complex relationship between X_i and Y, OLS will be suboptimal

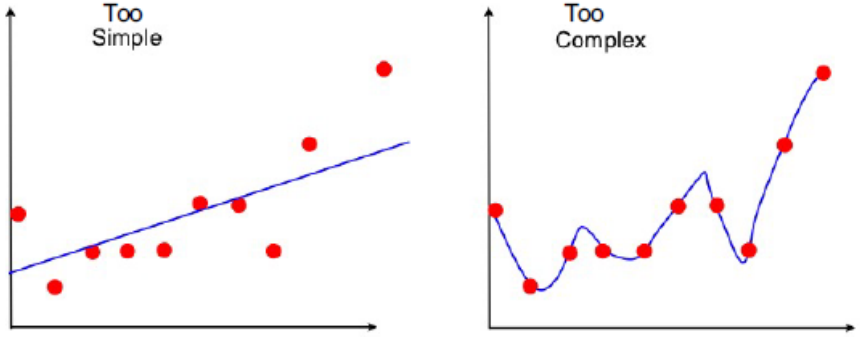




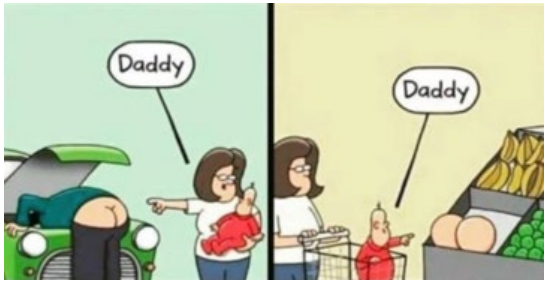
  - But can't you can add polynomials and interactions to OLS….. Yes, but…

Some Difficulties with OLS

- Overfitting: When a model describes the training data better than the population of interest (i.e., model is too complex)
  - E.g., too many variables, small sample size, multicollinearity, too much model complexity



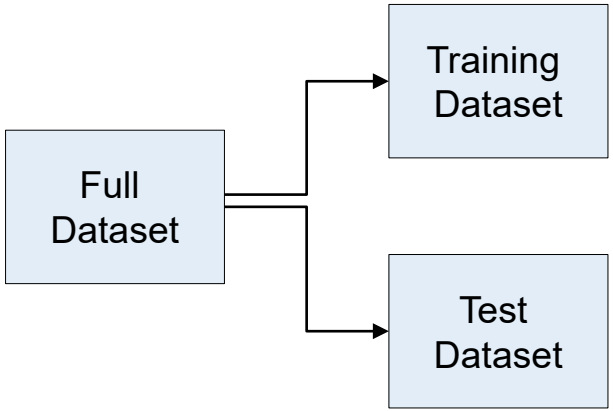
Not necessarily specific to OLS, but we want to avoid capturing chance fluctuations in data



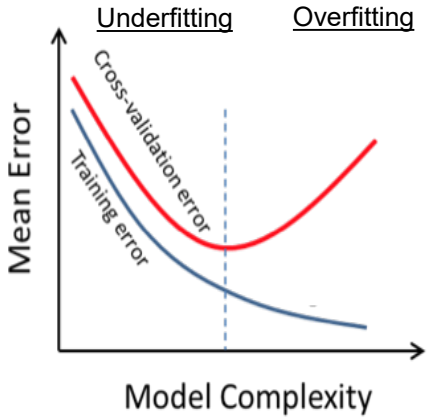
Example using OLS: With 125 predictors and N=400 for training, the multiple R = .70. When tested on a holdout sample though, the average holdout correlation (*N* = 2000) with job performance is just .27

Modern Prediction Methods

- Goal of supervised machine learning solutions is to produce generalizable predictions
  - Traditional methods generally use the same sample to develop and test the model. This is misleading if the goal is to generalize to new samples
  - ML develops a model on a “training set” of data, and the goal is to minimize prediction errors in new “test set” of data that were not used to develop the model
  - Modern modeling methods attend to the tradeoff between overfitting and underfitting data, doing so with different types of algorithms…



(i.e., cross-validation)



Modern Prediction Methods: Many Algorithms

- The number of modern prediction methods available is continuously expanding
  - Below are some available via caret (https://rdrr.io/cran/caret/man/models.html)

```
  [1] "ada"                 "AdaBag"              "AdaBoost.M1"         "adaboost"            "amdai"               "ANFIS"               "avNNet"
  [8] "awnb"                "awtan"               "bag"                 "bagEarth"            "bagEarthGCV"         "bagFDA"              "bagFDAGCV"
 [15] "bam"                 "bartMachine"         "bayesglm"            "binda"               "blackboost"          "blasso"              "blassoAveraged"
 [22] "bridge"              "brnn"                "BstLm"               "bstSm"               "bstTree"             "C5.0"                "C5.0Cost"
 [29] "C5.0Rules"           "C5.0Tree"            "cforest"             "chaid"               "CSimca"              "ctree"               "ctree2"
 [36] "cubist"              "dda"                 "deepboost"           "DENFIS"              "dnn"                 "dwdLinear"           "dwdPoly"
 [43] "dwdRadial"           "earth"               "elm"                 "enet"                "evtree"              "extraTrees"          "fda"
 [50] "FH.GBML"             "FIR.DM"              "foba"                "FRBCS.CHI"           "FRBCS.W"             "FS.HGD"              "gam"
 [57] "gamboost"            "gamLoess"            "gamSpline"           "gaussprLinear"       "gaussprPoly"         "gaussprRadial"       "gbm_h2o"
 [64] "gbm"                 "gcvEarth"            "GFS.FR.MOGUL"        "GFS.LT.RS"           "GFS.THRIFT"          "glm.nb"              "glm"
 [71] "glmboost"            "glmnet_h2o"          "glmnet"              "glmStepAIC"          "gpls"                "hda"                 "hdda"
 [78] "hdrda"               "HYFIS"               "icr"                 "J48"                 "JRip"                "kernelpls"           "kknn"
 [85] "knn"                 "krlsPoly"            "krlsRadial"          "lars"                "lars2"               "lasso"               "lda"
 [92] "lda2"                "leapBackward"        "leapForward"         "leapSeq"             "Linda"               "lm"                  "lmStepAIC"
 [99] "LMT"                 "loclda"              "logicBag"            "LogitBoost"          "logreg"              "lssvmLinear"         "lssvmPoly"
[106] "lssvmRadial"         "lvq"                 "M5"                  "M5Rules"             "manb"                "mda"                 "Mlda"
[113] "mlp"                 "mlpKerasDecay"       "mlpKerasDecayCost"   "mlpKerasDropout"     "mlpKerasDropoutCost" "mlpML"               "mlpSGD"
[120] "mlpWeightDecay"      "mlpWeightDecayML"    "monmlp"              "msaenet"             "multinom"            "mxnet"               "mxnetAdam"
[127] "naive_bayes"         "nb"                  "nbDiscrete"          "nbSearch"            "neuralnet"           "nnet"                "nnls"
[134] "nodeHarvest"         "null"                "OneR"                "ordinalNet"          "ordinalRF"           "ORFlog"              "ORFpls"
[141] "ORFridge"            "ORFsvm"              "ownn"                "pam"                 "parRF"               "PART"                "partDSA"
[148] "pcaNNet"             "pcr"                 "pda"                 "pda2"                "penalized"           "PenalizedLDA"        "plr"
[155] "pls"                 "plsRglm"             "polr"                "ppr"                 "pre"                 "PRIM"                "protoclass"
[162] "qda"                 "QdaCov"              "qrf"                 "qrnn"                "randomGLM"           "ranger"              "rbf"
[169] "rbfDDA"              "Rborist"             "rda"                 "regLogistic"         "relaxo"              "rf"                  "rFerns"
[176] "RFlda"               "rfRules"             "ridge"               "rlda"                "rlm"                 "rmda"                "rocc"
[183] "rotationForest"      "rotationForestCp"    "rpart"               "rpart1SE"            "rpart2"              "rpartCost"           "rpartScore"
[190] "rqlasso"             "rqnc"                "RRF"                 "RRFglobal"           "rrlda"               "RSimca"              "rvmLinear"
[197] "rvmPoly"             "rvmRadial"           "SBC"                 "sda"                 "sdwd"                "simpls"              "SLAVE"
[204] "slda"                "smda"                "snn"                 "sparseLDA"           "spikeslab"           "spls"                "stepLDA"
[211] "stepQDA"             "superpc"             "svmBoundrangeString" "svmExpoString"       "svmLinear"           "svmLinear2"          "svmLinear3"
[218] "svmLinearWeights"    "svmLinearWeights2"   "svmPoly"             "svmRadial"           "svmRadialCost"       "svmRadialSigma"      "svmRadialWeights"
[225] "svmSpectrumString"   "tan"                 "tanSearch"           "treebag"             "vbmpRadial"          "vglmAdjCat"          "vglmContRatio"
[232] "vglmCumulative"      "widekernelpls"       "WM"                  "wsrf"                "xgbDART"             "xgbLinear"           "xgbTree"
[239] "xyf"
```

Modern Prediction Methods: Example 1 (penalized regression)

- Penalized regression helps prevent overfitting when N/k ratio is small
  - OLS weights are unstable unless very large sample sizes
  - Penalized regression intentionally biases training sample prediction by penalizing large weights
  - Helpful to prevent overfitting. However, not really designed to model high complexity…

$$\sum_{i=1}^{n}(y_i-\hat{y}_i)^2+\lambda\sum_{j=1}^{p}\beta_j^2$$

Ridge Regression Formula

Speer, Christiansen, Robie, & Jacobs (2022)

Operationalization	*k*	OLS (N=500)	Penalized (N=500)
Personality Items	224	.10	.19
FFM dimensions	5	.12	.12

Shown are holdout validity coefficients

Modern Prediction Methods: Example 1 (penalized regression)

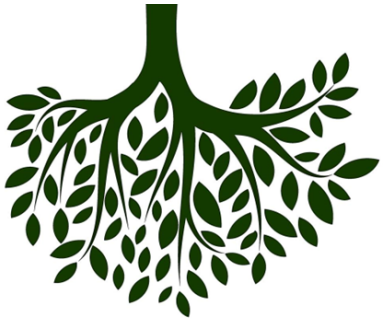
- 15k+ simulations across varied conditions (e.g., criterion-related validity, number of predictors, item or scale scoring, inter-correlations)

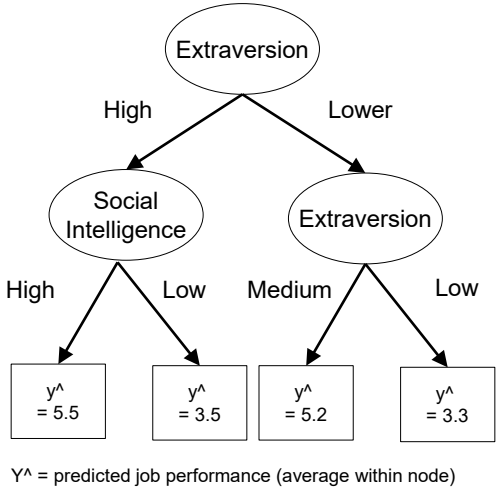
Operationalization	OLS (N=250)	Penalized (N=250)
50 predictors (i.e., items)	.21	.24
200 predictors (i.e., items)	.30	.45

Shown are holdout validity coefficients

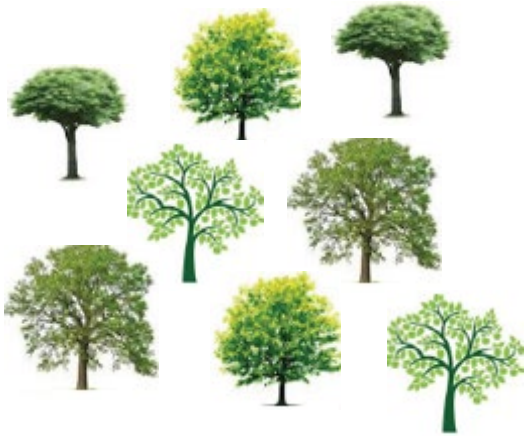
Modern Prediction Methods: Example 2 (random forests)

- Random forests (and similar tree models, e.g., boosted trees) can handle substantial data complexity such as interactions and curvilinear effects
  - Also designed to help prevent overfitting (e.g., randomization of variables and cases)
  - Tree-based methods frequently used in winning ML competitions in previous years





Y^ = predicted job performance (average within node)



Some General Observations

Hyper-Parameter Tuning

- Most ML algorithms have numerous hyper-parameters that influence how the algorithm works. Proper “tuning” leads to improved performance

How do we determine this hyper-parameter (λ)? = Grid search varying parameter, and look at holdout results

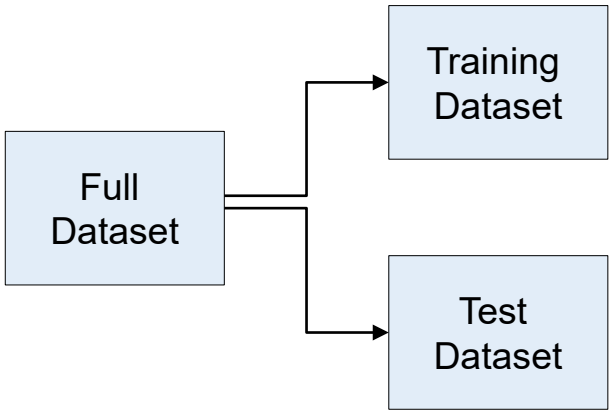
	Lambda (λ)	MSE
Trial 1	.00	.41
Trial 2	.10	.42
Trial 3	.20	.38
Trial k	… λ_i	?

$$\sum_{i=1}^{n}(y_i-\hat{y}_i)^2+\lambda\sum_{j=1}^{p}\beta_j^2$$

  - For example…
    - Random forests can tune # predictors sampled, node size or max depth, # trees (robust to overfitting)
    - Xgboost can tune many (e.g., # trees, learning rate, tree depth, k sampling rate)

Hyper-Parameter Tuning: Problem 1 – Not Separate Tuning & Testing

- Avoid data leakage that convolutes hyper-parameter identification and model testing



These should be completely separated for all stages of the research

Hyper-Parameter Tuning: Problem 1 – Not Separate Tuning & Testing

- Avoid data leakage that convolutes hyper-parameter identification and model testing

Ideally, hyper-parameters are determined in a sample completely different from the test set

Training Dataset

Example 5-fold

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Train	Train	Train	Train	Validate
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Train	Train	Train	Validate	Train
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Train	Train	Validate	Train	Train
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Train	Validate	Train	Train	Train
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Validate	Train	Train	Train	Train

Resampling results across tuning parameters:

eta	max_depth	gamma	colsample_bytree	subsample	nrounds	RMSE	Rsquared	MAE
0.05	2	0	0.7	0.7	50	0.9324276	0.14351605	0.7503665
0.05	2	0	0.7	0.7	75	0.9272552	0.14857164	0.7401798
0.05	2	0	0.7	0.7	100	0.9272354	0.14868763	0.7378531
0.05	2	0	0.7	0.7	150	0.9389821	0.13397814	0.7421551
0.05	2	0	0.7	0.9	50	0.9279723	0.15454227	0.7455958
0.05	2	0	0.7	0.9	75	0.9255584	0.15213681	0.7371857
0.05	2	0	0.7	0.9	100	0.9293426	0.14681638	0.7383274
0.05	2	0	0.7	0.9	150	0.9319005	0.14572223	0.7361623
0.05	2	0	0.9	0.7	50	0.9276502	0.15114414	0.7457827
0.05	2	0	0.9	0.7	75	0.9289406	0.14419052	0.7403321
0.05	2	0	0.9	0.7	100	0.9315856	0.14117339	0.7391523
0.05	2	0	0.9	0.7	150	0.9428218	0.12809921	0.7476214
0.05	2	0	0.9	0.9	50	0.9280909	0.15292444	0.7468656
0.05	2	0	0.9	0.9	75	0.9245379	0.15264339	0.7380022
0.05	2	0	0.9	0.9	100	0.9291582	0.14645873	0.7364559
0.05	2	0	0.9	0.9	150	0.9400374	0.13293514	0.7411245
0.05	2	5	0.7	0.7	50	0.9358790	0.13358298	0.7532730
0.05	2	5	0.7	0.7	75	0.9356041	0.13113795	0.7458368
0.05	2	5	0.7	0.7	100	0.9353196	0.13266124	0.7433627

The hyper-parameter configuration with the best performance in validation holdout folds is chosen to form algorithm

Final algorithm is formed

Model is tested on completely independent sample

Test Dataset

Hyper-Parameter Tuning: Problem 1 – Not Separate Tuning & Testing

- Avoid data leakage that convolutes hyper-parameter identification and model testing

Instead, sometimes people rely on the average “validate” folds for model evaluation after tuning

Full Dataset

Example 5-fold

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Train	Train	Train	Train	Validate
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Train	Train	Train	Validate	Train
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Train	Train	Validate	Train	Train
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Train	Validate	Train	Train	Train
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Validate	Train	Train	Train	Train

Resampling results across tuning parameters:

eta	max_depth	gamma	colsample_bytree	subsample	nrounds	RMSE	Rsquared	MAE
0.05	2	0	0.7	0.7	50	0.9324276	0.14351605	0.7503665
0.05	2	0	0.7	0.7	75	0.9272552	0.14857164	0.7401798
0.05	2	0	0.7	0.7	100	0.9272354	0.14868763	0.7378531
0.05	2	0	0.7	0.7	150	0.9389821	0.13397814	0.7421551
0.05	2	0	0.7	0.9	50	0.9279723	0.15454227	0.7455958
0.05	2	0	0.7	0.9	75	0.9255584	0.15213681	0.7371857
0.05	2	0	0.7	0.9	100	0.9293426	0.14681638	0.7383274
0.05	2	0	0.7	0.9	150	0.9319005	0.14572223	0.7361623
0.05	2	0	0.9	0.7	50	0.9276502	0.15114414	0.7457827
0.05	2	0	0.9	0.7	75	0.9289406	0.14419052	0.7403321
0.05	2	0	0.9	0.7	100	0.9315856	0.14117339	0.7391523
0.05	2	0	0.9	0.7	150	0.9428218	0.12809921	0.7476214
0.05	2	0	0.9	0.9	50	0.9280909	0.15292444	0.7468656
0.05	2	0	0.9	0.9	75	0.9245379	0.15264339	0.7380022
0.05	2	0	0.9	0.9	100	0.9291582	0.14645873	0.7364559
0.05	2	0	0.9	0.9	150	0.9400374	0.13293514	0.7411245
0.05	2	5	0.7	0.7	50	0.9358790	0.13358298	0.7532730
0.05	2	5	0.7	0.7	75	0.9356041	0.13113795	0.7458368
0.05	2	5	0.7	0.7	100	0.9353196	0.13266124	0.7433627

How to avoid this?

- Use a fully independent test dataset
  - However, don’t forget sampling error impacts the test dataset just like any research!!!!!!!!
- Use nested k-folds cross-validation

Hyper-Parameter Tuning: Problem 2 - Not Fully Tuning

- Important parameters are sometimes not tuned well, making cross-algorithm comparisons unhelpful
  - Simple methods like ridge regression are straightforward and there isn't massive need to grid-search your data "to death"
  - Other methods have core parameters that researchers often ignore or treat coarsely and need more thorough tuning

Let's say we're using xgboost with R caret…

```
xgbGrid <- expand.grid(nrounds = c(50, 75, 100, 125, 150, 200, 250),  #ntrees (default = 100)
                       max_depth = c(2, 4, 6), #default 6
                       eta = c(.05, .1, .2, .3), #learning rate (default = .3)
                       colsample_bytree = c(.7, .9), #default is 1
                       gamma= c(0, 1, 2, 3, 4), #default regularization = 0
                       subsample = c(.7, .8, .9))
```

Xgboost has many hyper-parameters…

What if we only train the number of trees and use otherwise default settings?

Tuning	Test r_{xy}
Tune all hyper-parameters	.39
Only tune # trees	.28

Construct Validation: It's not just about r_{xy} with the target….

- Sometimes we simply care about predicting an event….
- However, what do our predicted scores actually represent…
- Construct validation requires a preponderance of evidence to support scores…

Considerations

- Exert strong construct validation design (e.g., convergent and discriminant measures)
- Content validity evidence
  - Garbage-in-garbage out… construct-relevant in-construct-relevant out
  - Can SMEs perform content validity judgments of the inputs?
  - Irreducible error may limit a model from fully reflecting construct (e.g., text to infer personality)…

Situations Where ML is Most Advantageous

Thoughts on ML Efficacy

- Tabular Data
  - Small-sized datasets: penalized regression will be helpful. As datasets get larger, possibly helpful
  - Medium-sized datasets: random forests and gradient boosted machines become more feasible, and these are both powerful (particularly if data are complex)
  - Large and complex datasets: neural networks are often the leader in prediction as data complexity increases, though with tabular data, alternative methods often work just as well

A	B	C	D	E	F	G	H	I	J	K	L
CarSales	Education	Work_Exp	Work_Exp	Leader_Ex	GMA	Adaptive1	Adaptive2	Adaptive3	Adaptive4	Adaptive5	Adaptive6
1.666667	3	3	4	4	17	-0.51046	0.887176	0.704088	-1.4103	-0.67094	-2.70052
3.5	5	0	5	4	20	-0.42303	-1.61984	-1.43523	0.550836	-0.78653	1.455685
7.571429	5	0	6	7	11	0.866362	-1.57333	-1.24801	0.234875	1.010886	0.492639
10.41667	4	3	6	7	18	0.287595	0.42792	0.502463	-0.29899	0.221272	-1.64092
8.833333	3	3	7	7	17	-0.7589	-0.35543	0.095285	0.323593	-0.85016	-0.46035
8.25	2	1	6	6	13	0.348664	0.43228	-0.10634	1.392881	-1.0125	0.175519
7.333333	2	1	5	6	14	-0.68396	1.09791	1.477858	-0.60717	-0.0073	1.349691
9.166667	5	0	5	5	17	0.404181	0.727308	0.656955	0.806096	0.65504	1.043455
6.5	2	0	5	4	13	0.381974	0.311653	0.622914	1.62635	0.336857	1.097724
8.545455	4	2	6	5	15	2.556862	0.230266	1.124359	3.058293	1.377122	0.492833
10.2	3	3	6	6	22	0.387526	-0.25079	0.389867	-0.52156	0.712183	0.310596
10.5	2	0	6	7	14	-0.32448	-1.17366	-0.97307	-0.82819	-0.49562	-0.05017
2	3	0	5	7	15	-0.42719	0.557268	0.134562	0.557062	-1.44887	-0.52041

Thoughts on ML Efficacy

- Unstructured and Complex Data
  - In general, this is where ML shines!
  - Text Data and Natural Language Processing (NLP)
    - Where the largest gains have been seen in our field thus far
  - Big, complicated data (e.g., digital breadcrumbs, visual data)

Pros

Great place to work at, great people, and always something to do

Cons

Limited benefits, management did not honor schedule sometimes

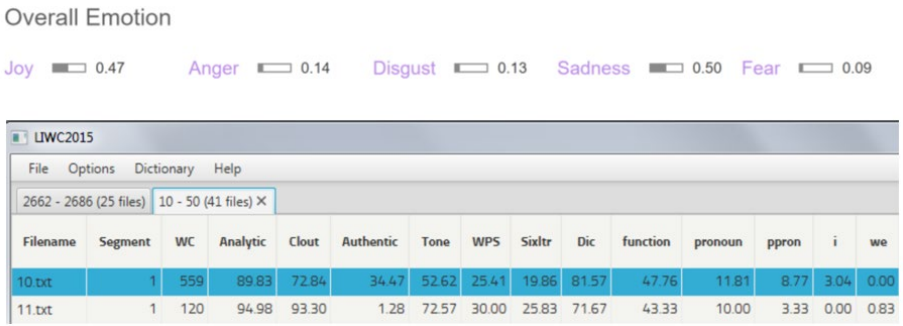
Natural Language Processing: Frequently Used in the Org Sciences

Some recent organizational sciences papers using NLP…

- Screening & Assessment (Campion et al., 2016; Fan et al., 2023; Hickman et al., 2022; Thompson et al., 2023; Yankov & Speer, 2023)
- Analyze Open-Ended Survey Responses (Speer et al., 2023)
- Measuring Culture (Pandey & Pandey, 2017)
- Measuring Job Performance (Speer, 2018; Speer, 2020)
- These are just a few, and these are just academic applications…

Natural Language Processing: Early Approaches

- Simple dictionary counts
  - Can measure text themes (what is the text about) or valence (how positive is the construct)



text <- *'I hate that cat'*
syuzhet::get_sentiment (text, method = "afinn") ### = -3

text <- *'I do not hate that cat'*
syuzhet::get_sentiment (text, method = "afinn") ### = -3

"*There is simply no option for growth in my current company*"

growth…

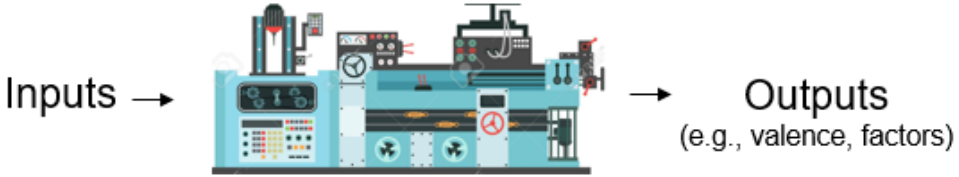
development… ladder…

advancement…

Natural Language Processing: Early Approaches

- **Supervised Bag of words (BOW-ML)**: words or word combination are only analysis attributes
  - Ignores word information order (semantic structure), but generally performs similarly to more advanced methods (Kobayashi et al., 2017)
  - Word vectors trained to recreate target score (e.g., performance rating, SME rating) via ML
  - Most commonly used by social science researchers in the past
  - Semantic meaning can still be captured somewhat using n-grams (“White”, “White House”)

ID	DV	Word1	Word2	Work3	Work_k
1	5.5	0	0	0	1
2	4.3	0	1	0	0
3	5.7	1	0	0	1
4	4.1	0	0	1	1
5	5.9	0	0	0	1
i	6	1	0	0	0



Natural Language Processing: Early Approaches

- BOW-ML is reasonable, though oftentimes performance is lower than desired
  - These are just a few examples…

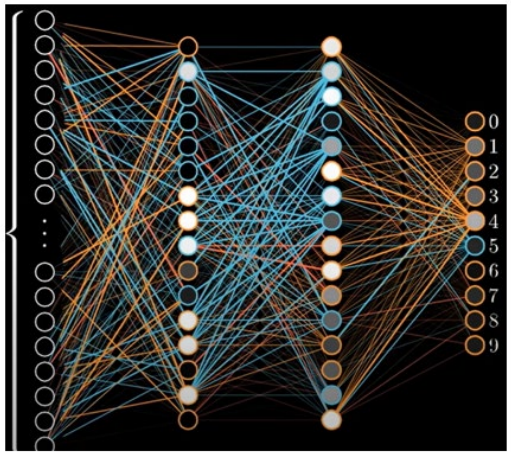
BOW results

Study	Constructs	BOW - Convergent Correlation
Speer et al. (2023)	Attitude constructs *BOW = dictionary	.30
Thompson et al. (2023)	Virtual AC ratings	.70
Yankov & Speer (2023)	Virtual AC ratings	.65
Speer et al. (2023)	Job performance valence	.55

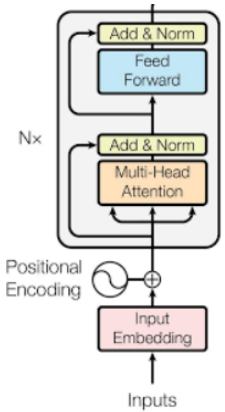
DV = SME ratings

Natural Language Processing: Transformer Neural Networks

- Deep Neural Network Transformers: Transformer models are neural network architectures that learn context from text and therefore better reflect relationships between data
  - Just like BOW-ML, can train to predict a target score
  - Org sciences was late to the party, but usage increased around 2020 onward (later in academia)



From 3 blue one brown



"This job is great if you like working for Satan"

Transfer learning!

Natural Language Processing: Transformer Neural Networks

- Drastic improvements seen by using transformers
  - See below. Other examples with favorable transformer properties: e.g., Koenig et al. 2023; Liu et al., 2023)

Study	Constructs	BOW - Convergent Correlation	Transformer - Convergent Correlation
Speer et al. (2023) / unpublished	Attitude constructs *BOW = dictionary	.30	.73
Thompson et al. (2023)	Virtual AC ratings	.70	.84
Yankov & Speer (2023)	Virtual AC ratings	.65	.69
Speer et al. (2023)	Job performance valence	.55	.84

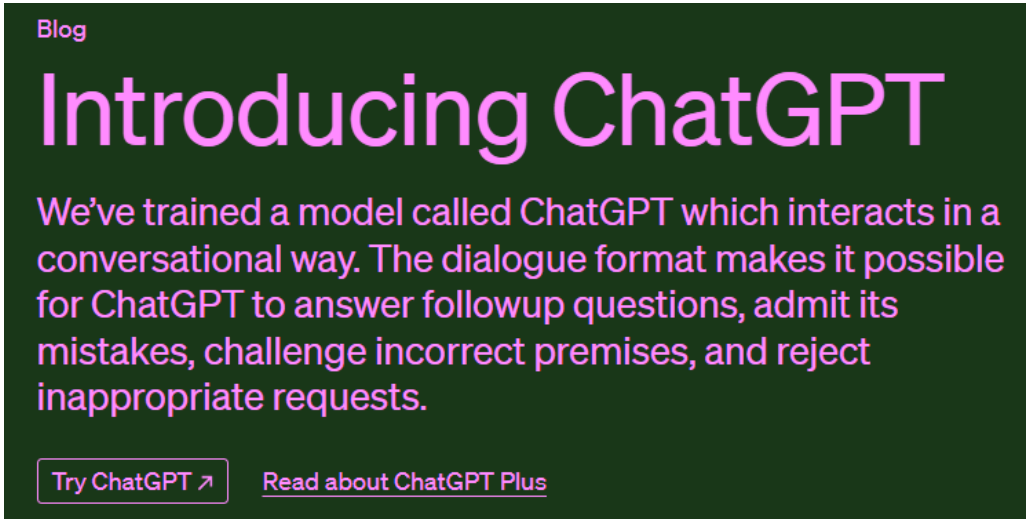
DV = SME ratings

Little reason to use BOW going forward

Large Language Models (LLMs)

We’ve Barely Scratched the Surface: Transformers on steroids

LLMs: Connecting Everyone to Artificial Intelligence



LLMs: Connecting Everyone to Artificial Intelligence

- Most of us know it can converse in a human-like way

You

I want to practice my interviewing skills. You are playing the role of an interviewer for the job of salesperson. You are to handle this by pretending we are just beginning the interview. Start with the usual niceties and early conversation, and then ask me questions. Make sure you sound realistic and not too robotic. Once you get to the core interview questions, I'll respond to your first interview question. Then, pause, and give me feedback on my response.

ChatGPT

Absolutely, let's get started.

Good morning! It's a pleasure to have you here. How are you today? I hope you found our office easily. Before we begin, would you like anything to drink—water, coffee, tea?

Great, now that we're settled, let's start with a bit about our company. As you might already know, we pride ourselves on not just the products we sell but also the relationships we build with our customers. Our sales team is crucial in maintaining these relationships and expanding our reach.

With that in mind, I'd love to learn more about you. Could you tell me a little about your background and what drew you to this salesperson position with us?

LLMs: Connecting Everyone to Artificial Intelligence

- More powerful models possess all the previous advantages of earlier transformers while having many more capabilities at large for the organizational sciences
  - Develop teaching plans
  - Content generation (e.g., items, SJTs, assessment center plans)
  - Generate reports
  - Interactive assessments
  - Measurement (i.e., automated)
    - Identify themes
    - Measure valence
  - Fake assessments
  - Serve as coach
  - Perform work
  - Etc.

LLMs: A Few Examples of “off-the-shelf” capabilities

- Replace resource intensive supervised modeling?…

Performance Appraisal Comment

“*There are times when you seem perturbed during meetings, both verbally and non-verbally (e.g., short in communication, arms defensively folded). This can come across as hostile; it’s important to focus on appearing more controlled with others*”

Trained on 10k+ cases

BOW - Convergent Correlation	Supervised Transformer - Convergent Correlation
.55	.84

Raises the question: In what contexts can GPT judgments be used to replace SMEs? When would we feel comfortable doing this?

LLMs: A Few Examples of “off-the-shelf” capabilities

- Use as a SME for content analysis

Task 1: Unstructured Theme Identification

I recently collected several hundred open-ended responses on why applicants choose to decline a job

Used GPT4 API to identify 25-30 common themes across all texts.

Manual review by me to clean up, reduce, and provide definitions after reviewing text

Task 2: Assign Comments to Theme Categories

Had GPT review each comment and assign it to whichever themes were relevant.

Simultaneously, I independently coded 70 comments myself. Agreement was high (agreement = 95%; kappa = .83)

Given this, I used GPT to code the remaining comments

In Conclusion

In Conclusion

- There's a lot to machine-learning, and this talk only covered a very small portion
- The world of ML is shifting rapidly, which will impact the field of organizational sciences
- Organizational scientists would benefit from learning these methods

Thank you!

Appendix

Hyper-Parameter Tuning: Problem 1 – Not Separate Tuning & Testing

10-fold nested cross-validation

Selection = validation sample

Hyper-Parameter Tuning

	Training	Selection	Training	Selection	Training	Selection		Training	Selection
Data cut 1									
Data cut 2	X		X		X				X
Data cut 3	X		X		X			X	
Data cut 4	X		X		X			X	
Data cut 5	X		X		X			X	
Data cut 6	X		X		X			X	
Data cut 7	X		X		X			X	
Data cut 8	X		X			X		X	
Data cut 9	X			X	X		...	X	
Data cut 10		X	X		X			X	

	Training	Selection	Training	Selection	Training	Selection		Training	Selection
Data cut 1	X		X		X				X
Data cut 2									
Data cut 3	X		X		X			X	
Data cut 4	X		X		X			X	
Data cut 5	X		X		X			X	
Data cut 6	X		X		X			X	
Data cut 7	X		X		X			X	
Data cut 8	X		X			X		X	
Data cut 9	X			X	X		...	X	
Data cut 10		X	X		X			X	

...

	Training	Selection	Training	Selection	Training	Selection		Training	Selection
Data cut 1	X		X		X				X
Data cut 2	X		X		X			X	
Data cut 3	X		X		X			X	
Data cut 4	X		X		X			X	
Data cut 5	X		X		X			X	
Data cut 6	X		X		X			X	
Data cut 7	X		X			X		X	
Data cut 8	X			X	X			X	
Data cut 9		X	X		X		...	X	
Data cut 10									

Model Evaluation

Training	Test
	X
X	
X	
X	
X	
X	
X	
X	
X	

Training	Test
X	
	X
X	
X	
X	
X	
X	
X	
X	

Training	Test
X	
X	
X	
X	
X	
X	
X	
X	
X	
	X