Machine Learning - EITM 2024

Jason Anastasopoulos

Associate Professor of Public Policy and Statistics
University of Georgia

July 10, 2024

Outline

Introduction - Why Learn Machine Learning?

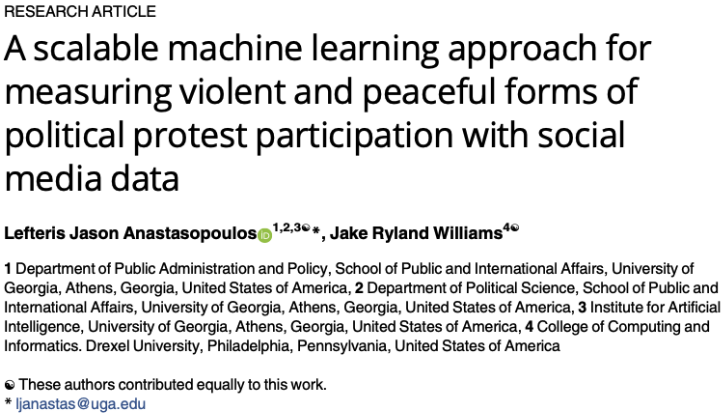
Machine Learning Basics and Applications

ML Foundations - Statistical Learning Theory

Application: Measuring Organization Reputation in Federal Agencies

R Example: Predicting Diabetes

Some recent research



RESEARCH ARTICLE

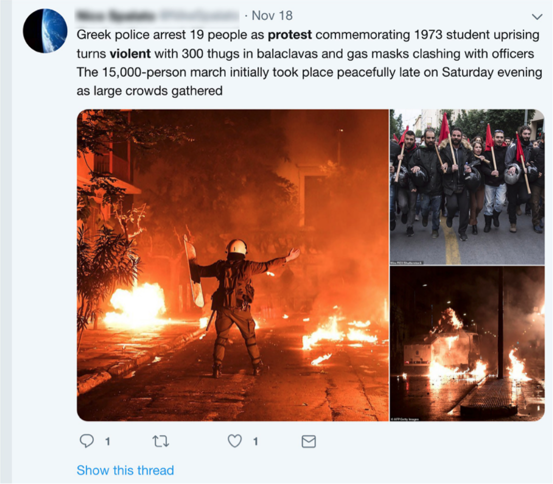
A scalable machine learning approach for measuring violent and peaceful forms of political protest participation with social media data

Lefteris Jason Anastasopoulos[1,2,3]◐*, **Jake Ryland Williams**[4]◐

1 Department of Public Administration and Policy, School of Public and International Affairs, University of Georgia, Athens, Georgia, United States of America, **2** Department of Political Science, School of Public and International Affairs, University of Georgia, Athens, Georgia, United States of America, **3** Institute for Artificial Intelligence, University of Georgia, Athens, Georgia, United States of America, **4** College of Computing and Informatics. Drexel University, Philadelphia, Pennsylvania, United States of America

◐ These authors contributed equally to this work.
* ljanastas@uga.edu

Some recent research



[redacted] · Nov 18

Greek police arrest 19 people as **protest** commemorating 1973 student uprising turns **violent** with 300 thugs in balaclavas and gas masks clashing with officers The 15,000-person march initially took place peacefully late on Saturday evening as large crowds gathered

Some recent research

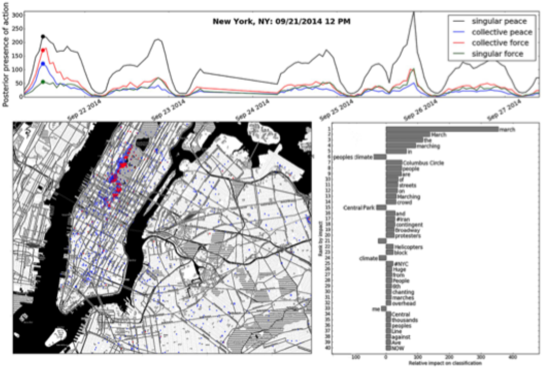
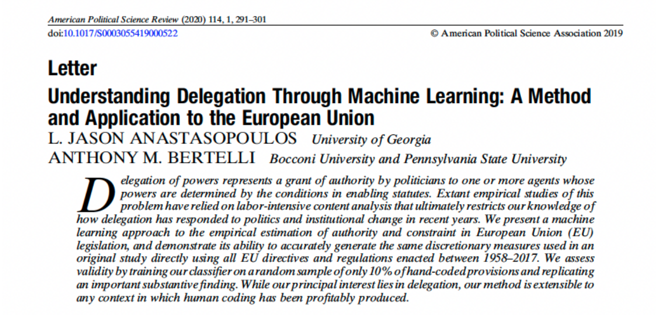


Fig 9. Above. A time series showing the total presence of social action types during the week of the People's Climate March which began on September 21, 2014. Left. Map of New York, NY depicting clusters of collective force and collective peace activity over one hour around 12 PM on September 21st during a climate change protest. The size of each cluster-circle represents the area from which tweets emerged (not the number of tweets contained), and the portion of each circle colored red indicates the portion of tweets classified to represent the collective force action. Right. A phrase shift showing the most impactful features present in all tweets classified as being representative of collective force. Phrases on the right pull the classifier toward a positive classification, and phrases on the left pull the classifier towards a negative classification.

https://doi.org/10.1371/journal.pone.0212834.g009

Some recent research



American Political Science Review (2020) 114, 1, 291–301
doi:10.1017/S0003055419000522

Letter

Understanding Delegation Through Machine Learning: A Method and Application to the European Union

L. JASON ANASTASOPOULOS *University of Georgia*
ANTHONY M. BERTELLI *Bocconi University and Pennsylvania State University*

Delegation of powers represents a grant of authority by politicians to one or more agents whose powers are determined by the conditions in enabling statutes. Extant empirical studies of this problem have relied on labor-intensive content analysis that ultimately restricts our knowledge of how delegation has responded to politics and institutional change in recent years. We present a machine learning approach to the empirical estimation of authority and constraint in European Union (EU) legislation, and demonstrate its ability to accurately generate the same discretionary measures used in an original study directly using all EU directives and regulations enacted between 1958–2017. We assess validity by training our classifier on a random sample of only 10% of hand-coded provisions and replicating an important substantive finding. While our principal interest lies in delegation, our method is extensible to any context in which human coding has been profitably produced.

Some recent research

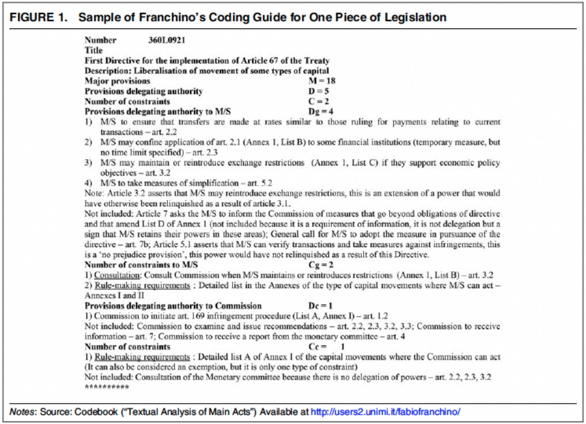


FIGURE 1. Sample of Franchino's Coding Guide for One Piece of Legislation

Number 360L0921
Title
First Directive for the implementation of Article 67 of the Treaty
Description: Liberalisation of movement of some types of capital
Major provisions M = 18
Provisions delegating authority D = 5
Number of constraints C = 2
Provisions delegating authority to M/S Dg = 4

1) M/S to ensure that transfers are made at rates similar to those ruling for payments relating to current transactions – art. 2.2
2) M/S may confine application of art. 2.1 (Annex 1, List B) to some financial institutions (temporary measure, but no time limit specified) – art. 2.3
3) M/S may maintain or reintroduce exchange restrictions (Annex 1, List C) if they support economic policy objectives – art. 3.2
4) M/S to take measures of simplification – art. 5.2

Note: Article 3.2 asserts that M/S may reintroduce exchange restrictions, this is an extension of a power that would have otherwise been relinquished as a result of article 3.1.

Not included: Article 7 asks the M/S to inform the Commission of measures that go beyond obligations of directive and that amend List D of Annex 1 (not included because it is a requirement of information, it is not delegation but a sign that M/S retains their powers in these areas); General call for M/S to adopt the measure in pursuance of the directive – art. 7b; Article 5.1 asserts that M/S can verify transactions and take measures against infringements, this is a 'no prejudice provision', this power would have not relinquished as a result of this Directive.

Number of constraints to M/S Cg = 2

1) <u>Consultation</u>: Consult Commission when M/S maintains or reintroduces restrictions (Annex 1, List B) – art. 3.2
2) <u>Rule-making requirements</u> : Detailed list in the Annexes of the type of capital movements where M/S can act – Annexes I and II

Provisions delegating authority to Commission Dc = 1

1) Commission to initiate art. 169 infringement procedure (List A, Annex I) – art. 1.2

Not included: Commission to examine and issue recommendations – art. 2.2, 2.3, 3.2, 3.3; Commission to receive information – art. 7; Commission to receive a report from the monetary committee – art. 4

Number of constraints Cc = 1

1) <u>Rule-making requirements</u> : Detailed list A of Annex I of the capital movements where the Commission can act (It can also be considered an exemption, but it is only one type of constraint)

Not included: Consultation of the Monetary committee because there is no delegation of powers – art. 2.2, 2.3, 3.2
****

Notes: Source: Codebook ("Textual Analysis of Main Acts") Available at http://users2.unimi.it/fabiofranchino/

Some recent research

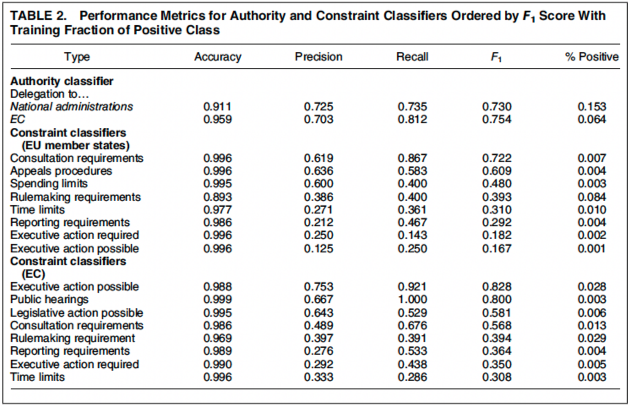


TABLE 2. Performance Metrics for Authority and Constraint Classifiers Ordered by F_1 Score With Training Fraction of Positive Class

Type	Accuracy	Precision	Recall	F_1	% Positive
Authority classifier					
Delegation to…					
National administrations	0.911	0.725	0.735	0.730	0.153
EC	0.959	0.703	0.812	0.754	0.064
Constraint classifiers (EU member states)					
Consultation requirements	0.996	0.619	0.867	0.722	0.007
Appeals procedures	0.996	0.636	0.583	0.609	0.004
Spending limits	0.995	0.600	0.400	0.480	0.003
Rulemaking requirements	0.893	0.386	0.400	0.393	0.084
Time limits	0.977	0.271	0.361	0.310	0.010
Reporting requirements	0.986	0.212	0.467	0.292	0.004
Executive action required	0.996	0.250	0.143	0.182	0.002
Executive action possible	0.996	0.125	0.250	0.167	0.001
Constraint classifiers (EC)					
Executive action possible	0.988	0.753	0.921	0.828	0.028
Public hearings	0.999	0.667	1.000	0.800	0.003
Legislative action possible	0.995	0.643	0.529	0.581	0.006
Consultation requirements	0.986	0.489	0.676	0.568	0.013
Rulemaking requirement	0.969	0.397	0.391	0.394	0.029
Reporting requirements	0.989	0.276	0.533	0.364	0.004
Executive action required	0.990	0.292	0.438	0.350	0.005
Time limits	0.996	0.333	0.286	0.308	0.003

Some recent research

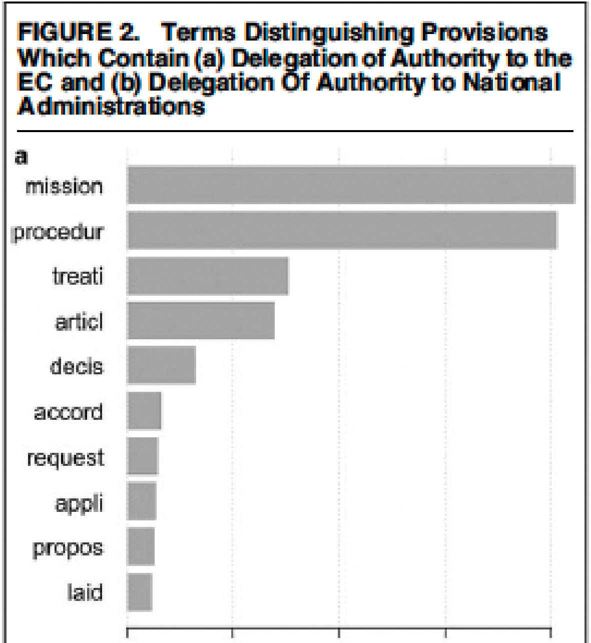


FIGURE 2. Terms Distinguishing Provisions Which Contain (a) Delegation of Authority to the EC and (b) Delegation Of Authority to National Administrations

Some recent research

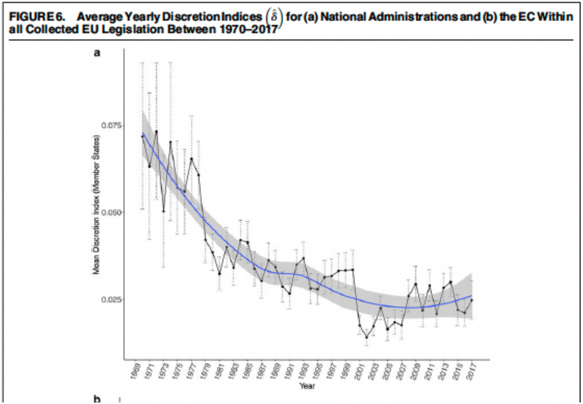
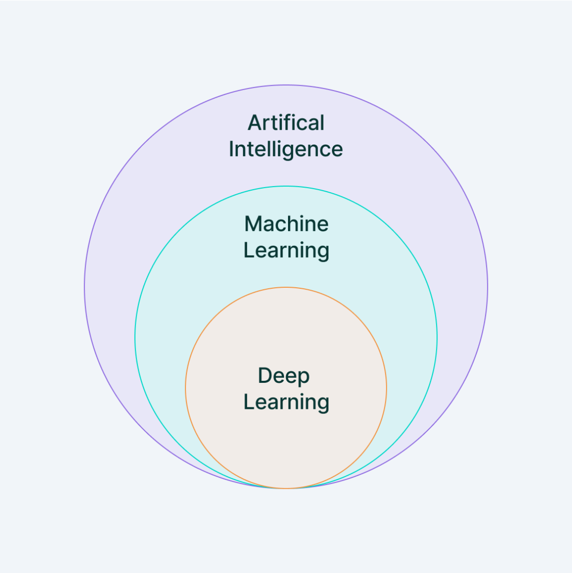


FIGURE 6. Average Yearly Discretion Indices $\left(\hat{\delta}\right)$ for (a) National Administrations and (b) the EC Within all Collected EU Legislation Between 1970–2017

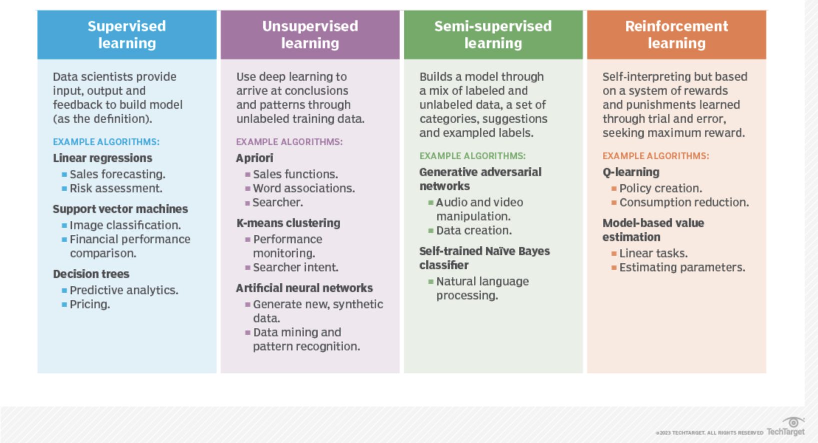
What is Machine Learning?

Machine learning is the art of making predictions with algorithms.
(my definition)

Machine learning in a broader context



Types of Machine Learning Algorithms



Supervised learning	Unsupervised learning	Semi-supervised learning	Reinforcement learning
Data scientists provide input, output and feedback to build model (as the definition).	Use deep learning to arrive at conclusions and patterns through unlabeled training data.	Builds a model through a mix of labeled and unlabeled data, a set of categories, suggestions and exampled labels.	Self-interpreting but based on a system of rewards and punishments learned through trial and error, seeking maximum reward.
EXAMPLE ALGORITHMS: **Linear regressions** ▪ Sales forecasting. ▪ Risk assessment. **Support vector machines** ▪ Image classification. ▪ Financial performance comparison. **Decision trees** ▪ Predictive analytics. ▪ Pricing.	EXAMPLE ALGORITHMS: **Apriori** ▪ Sales functions. ▪ Word associations. ▪ Searcher. **K-means clustering** ▪ Performance monitoring. ▪ Searcher intent. **Artificial neural networks** ▪ Generate new, synthetic data. ▪ Data mining and pattern recognition.	EXAMPLE ALGORITHMS: **Generative adversarial networks** ▪ Audio and video manipulation. ▪ Data creation. **Self-trained Naïve Bayes classifier** ▪ Natural language processing.	EXAMPLE ALGORITHMS: **Q-learning** ▪ Policy creation. ▪ Consumption reduction. **Model-based value estimation** ▪ Linear tasks. ▪ Estimating parameters.

Statistical learning theory

$$f : \mathcal{X} \to \mathcal{Y}$$
$$\mathcal{X} \in \mathbb{R}^{nxp}; \mathcal{Y} \in \mathbb{R}^{p}$$

- Abstractly is find a function f that accurately maps the inputs $\mathcal{X}$ to outputs $\mathcal{Y}$

Statistical learning theory

$$Y = f(X) + \epsilon$$

- More concretely, we are interested in finding a function $f(X)$ which can return values of an output Y.
- In introduction to regression courses, this is typically the equation you see.
- $f(X)$ is an unknown function of a matrix of predictors $X = (X_1, \cdots, X_p)$, an outcome Y and an error term ϵ.

Searching for $f(\cdot)$

$$Y = f(X) + \epsilon$$

- While X and Y are known, $f(\cdot)$ is unknown.
- The goal of statistical learning, then, is to utilize a set of approaches to estimate the "best" $f(\cdot)$ for the problem at hand.

$f(\cdot)$ in Social Science

$$
\begin{aligned}
f(X) &= \sum_{i=1}^{p} \beta_i x_i \\
\epsilon &\sim N(0, \sigma^2) \\
Y &= \sum_{i=1}^{p} \beta_i x_i + \epsilon
\end{aligned}
$$

- In social science, we often choose a linear function to estimate Y and assume that the error term is normally distributed with a zero mean.
- Parameters β are estimated by minimizing the sum of squared errors which form the normal equations $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y$

$f(\cdot)$ in Social Science: Causality

$$Y = \beta_0 + \beta_1 T + \sum_{i=i}^{p-1} \beta_i x_i + \epsilon$$

- Often we are interested in the values of one or two parameters and whether they are *causal* or not.
- There are many interpretations of statistical causality (ie Pearl (2009), Rubin (1974)).
- The general idea is that β_1 measures the extent to which ΔX_t will affect ΔY_{t+1}.

$f(\cdot)$ in Social Science: Causality

$$Y = \beta_0 + \beta_1 T + \sum_{i=i}^{p-1} \beta_i x_i + \epsilon$$

- Causal inference requires that $T \perp \epsilon$ or $T|X \perp \epsilon$.
- This often requires *randomization* of T under most circumstances.
- This implies that we are not really all that interested in choosing an optimal $f(\cdot)$.

$f(\cdot)$ in Social Science: Causality

$$\begin{aligned} \text{Choose design: } & \delta \subset \Delta \\ \text{s.t.: } & \exists x_i \in \mathbf{X} \\ \text{satisfying: } & x_i \perp \epsilon \end{aligned}$$

- Choose a subset of research designs δ from all possible designs Δ so that you have at least one treatment (variable) that is randomized.

$f(\cdot)$ in Machine Learning: Prediction

$$\hat{Y} = \hat{f}(X)$$

- Machine learning is primarily concerned with prediction.
- We are interested in finding the "best" $f(\cdot)$ and the "best" set of X's which give the best predictions, $\hat{Y}$.
- We want to find the function that minimize the difference between the *predicted* values and the *observed* values.

Reducible and irreducible error

$$\begin{aligned} \hat{f}(X) &= \hat{Y} && \text{estimated function} \\ f(X) + \epsilon &= Y && \text{true function} \end{aligned}$$

- Prediction of Y with $\hat{Y}$ can be broken down into two components: reducible and irreducible error.
- **reducible error** – $\hat{f}$ is used to estimate f but is not perfect. Improving the accuracy of $\hat{f}$ can be accomplished by adding more *observed* features (variables) to the model.
- **irreducible error** – ϵ represents all other features that can be used predict f. These are unobserved and thus are irreducible.

Reducible and irreducible error

$$\begin{aligned}
\mathbb{E}(Y - \hat{Y})^2 &= \mathbb{E}[f(X) + \epsilon - \hat{f}(X)]^2 \\
&= \mathbb{E}[(f(X) + \epsilon - \hat{f}(X))(f(X) + \epsilon - \hat{f}(X))] \\
&= [f(X) - \hat{f}(X)]^2 + Var(\epsilon)
\end{aligned}$$

Estimating $f(\cdot)$

- **Training data** – is required to “teach” our machine learning algorithm to predict outcomes.
- *Predicting presidential elections*
    - **outcome/Response**- presidential candidate vote share in each state for the Republican candidate.
    - **features** – state Republican vote share in last election, etc...

Estimating $f(\cdot)$ – example 1 – predicting elections

$$\begin{aligned} \text{Training data: } &\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\} \\ x_i = &(x_{i1}, x_{i2}, \cdots, x_{ip})^T \end{aligned}$$

- $n = 50$ states.
- $i = 1, \cdots, n$ observations (states), $j = 1, \cdots, p$ features (state-level variables).
- Training data: feature (or feature set) x_{ip} and outcome y_i (election results).

Estimating $f(\cdot)$ – example 2 – political sentiment in Tweets

$$\text{Training data: } \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$$
$$x_i = (x_{i1}, x_{i2}, \cdots, x_{ip})^T$$

- $n = 1,000$ Tweets.
- $i = 1, \cdots, n$ observations (Tweets), $j = 1, \cdots, p$ features (words, tweet length, etc).
- Training data: feature (or feature set) x_{ip} and outcome y_i (pro/anti Trump).

Estimating $f(\cdot)$ – parametric methods

$$\text{Step 1 – Functional form: } f(X) = \beta_0 + \sum_{i=1}^{p} \beta_i x_i$$

$$\text{Step 2 – Training: } Y = \beta_0 + \sum_{i=1}^{p} \beta_i x_i$$

- *parametric methods* are model-based approaches that involve two steps.
- **step 1** involves choosing a predefined functional form. Linear, quadratic, etc.
- **step 2** involves *training* or fitting the model using the training data.

Estimating $f(\cdot)$ – parametric methods – issues

$$
\begin{aligned}
Y = \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \sum_{i=1}^{p} + \\
\beta_i x_i^2 + \sum_{i=1}^{p} \beta_i x_i^3 + \cdots
\end{aligned}
$$

- Rigid models such as a strictly linear model may not fit the data well.
- More flexible models require more parameter estimation and may result in **overfitting** – a model that is only useful for the training data at hand.

Estimating $f(\cdot)$ – parametric methods – examples

- Linear regression.
- Logistic regression.
- Naive bayes.
- Neural networks.

Estimating f – non-parametric methods

- **non-parametric** methods do not assume anything about the functional form of f.
- Estimates a function only based on the data itself.

Estimating f – non-parametric methods – examples

- K-Nearest Neighbors.
- Support vector machines.
- Decision trees.

Accuracy and interpretability tradeoffs

- More accurate models often require estimating more parameters and/or having more flexible models.
- More models that are better at prediction generally are less interpretable.

Supervised v. unsupervised learning

- **Supervised learning** involves estimating functions with known observation and outcome data.
- **Unsupervised learning** involves estimating functions without the aid of outcome data.

Supervised learning – examples

- Naive bayes.
- Support vector machines.
- Neural networks.
- Linear regression.

Unsupervised learning – examples

- Topic models.
- K-Means clustering.
- Multidimensional scaling.
- Pagerank.

Unsupervised learning – examples

- Topic models.
- K-Means clustering.
- Multidimensional scaling.
- Pagerank.

Assessing model accuracy

- Machine learning is as much an art as it is a science.
- There is not best method, only a method that best fits a problem.

Measuring fit

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

- In the regression setting, the mean squared error is a metric of how well a model fits the data.
- To estimate model fit we need to partition the data:
    1. Training set – data that we will use to fit the model.
    2. Test set – data that we will use to test the fit of the model.

Measuring fit

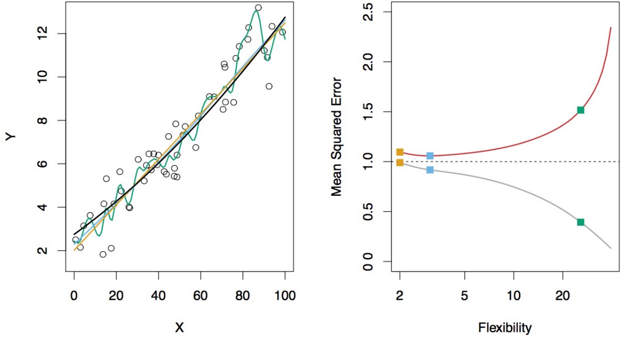
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

- **Training MSE** tells us how well our model fits the training data.
- **Test MSE** tells us how well our model fits new data.
- We are most concerned in minimizing *test MSE*.

How to choose training and test set?

- Divide labeled data randomly into two parts: training and test sets.
- **Cross-validation** involves randomly dividing the data into training and test sets several times and assessing the *average* model fit across each test set.

Training MSE, test MSE and model flexibility



- Increasing model flexibility tends to *decrease* training MSE but will eventually *increase* test MSE.

The bias-variance tradeoff

$$\mathbb{E}(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0)]^2 + Var(\epsilon)$$

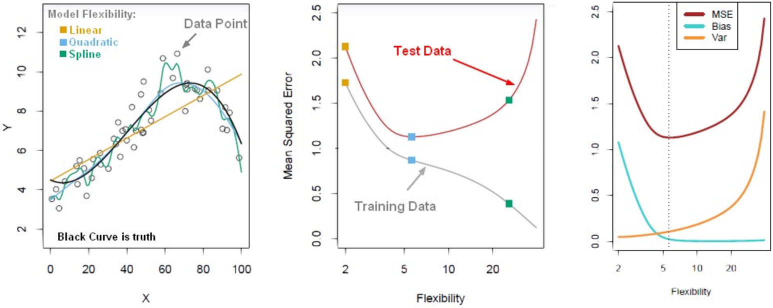
- It can be shown that the expected value for the test MSE can be decomposed into 3 components:
  1. $Var(\hat{f}(x_0))$ – Variance of the predictions.
  2. $[\text{Bias}(\hat{f}(x_0)]^2$ – Bias of the predictions.
  3. $Var(\epsilon)$ – Variance of the error terms.

The bias-variance tradeoff

$$\mathbb{E}(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [\text{Bias}(\hat{f}(x_0)]^2 + Var(\epsilon)$$

- It can be shown that the expected value for the test MSE can be decomposed into 3 components:
  1. $Var(\hat{f}(x_0))$ – how much would $\hat{f}$ change if we applied it to a different data set.
  2. $[\text{Bias}(\hat{f}(x_0)]^2$ – how well does the model fit the data?

The bias-variance tradeoff



- Simple models give consistent results across test sets (low variance) but don't predict well. (high bias).
- Very flexible (complex) models give inconsistent results across test sets (high variance), but do well at prediction (low bias).

Classification

$$\text{Error rate: } \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(y_i \neq \hat{y}_i)$$

- Our discussion of MSE previously was in the context of *regression* in which the outcome was a continuous predictor.
- There are some slight modifications that can be made in the setting in which we're interested in prediction *classes:*.
- $\{Democrat, Republican\}$, $\{Violent, Nonviolent\}$, $\{Protest, Non-protest\}$

Classification

$$\text{Error rate: } \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(y_i \neq \hat{y}_i)$$

- We are essentially interested in what % of classifications are correct.

Common measurements of classification error

- Accuracy
- Precision
- Recall
- F1 Score

Accuracy

- Definition: Ratio of correctly predicted instances to the total instances.
- Formula:
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$
- Use Case: Useful when classes are balanced.

Precision

- Definition: Ratio of correctly predicted positive observations to the total predicted positives.
- Formula:
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
- Use Case: Important when the cost of false positives is high.

Recall

- Definition: Ratio of correctly predicted positive observations to all the observations in the actual class.
- Formula:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- Use Case: Crucial when the cost of false negatives is high.

F1 Score

- Definition: Harmonic mean of precision and recall.
- Formula:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- Use Case: Useful for balancing precision and recall.

Example

Consider a binary classification problem with the following confusion matrix:

	Predicted Positive	**Predicted Negative**
Actual Positive	TP $=$ 50	FN $=$ 10
Actual Negative	FP $=$ 5	TN $=$ 100

Accuracy Example

- Accuracy:

$$\text{Accuracy} = \frac{50 + 100}{50 + 100 + 5 + 10} = \frac{150}{165} \approx 0.91$$

Precision Example

- Precision:

$$\text{Precision} = \frac{50}{50+5} = \frac{50}{55} \approx 0.91$$

Recall Example

- Recall:

$$\text{Recall} = \frac{50}{50+10} = \frac{50}{60} \approx 0.83$$

F1 Score Example

- F1 Score:

$$\text{F1 Score} = 2 \times \frac{0.91 \times 0.83}{0.91 + 0.83} \approx 0.87$$

Application Overview

- Use of ML to study organizational reputation in federal agencies a la Carpenter and Krause (2012).
- Data source: Tweets from 13 executive federal agencies

Reputation in Executive Agencies

Reputation Type	**Description**
Performative	“Can the agency do the job?”
Moral	“Does the agency protect the interests of its clients, constituency, and members?”
Procedural	“Does the agency follow accepted rules and norms...”
Technical	“Does the agency have the capacity and skill required for dealing in complex environments...”

Table: Reputation Types and Descriptions

Agencies that tweets were collected from

Agencies	
Health and Human Services	Education
Housing and Urban Development	Interior
Homeland Security	Treasury
Defense	Agriculture
Transportation	State
Commerce	Justice
Energy	

Table: List of Federal Agency Twitter Accounts

Supervised ML Pipeline

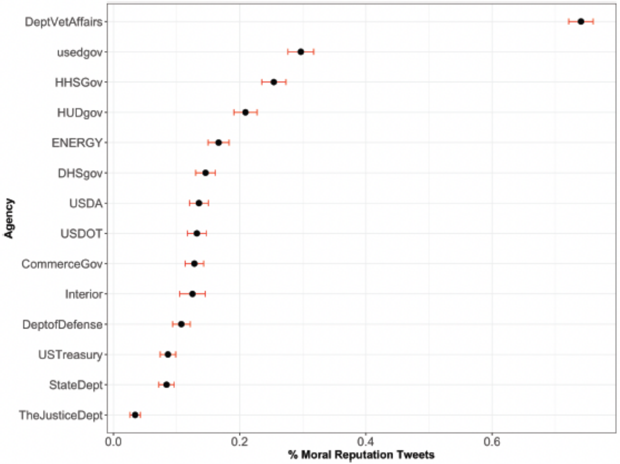
1. **Hand coding**: For a subset of tweets ($N = 200$), code by hand which tweets (for a given agency) are related to each of the four categories (or no category at all).
2. **Training**: Randomly parse the hand-coded data into two subsets. The training subset will be used to teach the algorithm the words and phrases that relate to each category. The other subset is a **testing** or **holdout** subset.
3. **Performance**: After training the algorithm, the final performance of the algorithm is assessed on the testing subset – fresh data on which it has been neither trained nor validated. In this case, performance is ultimately a measure of the classifier's ability to accurately classify a tweet into one of the four categories.

Results - Predicting Moral Reputation

	Accuracy	**No Info. Rate**	**Precision**	**Recall**	F_1 **(0–100)**
Expert	69.1%	51.7%	86.7%	44.8%	59
Nonexpert 1	46.7%	58.3%	28.6%	8.0%	12.5
Nonexpert 2	47.1%	86.7%	0.0%	0.0%	0

Table: Moral Reputation Classification Performance of Experts vs Nonexperts

Predicted Moral Reputation in 26,202 Tweets



Simple R Example - Predicting Diabetes

- Here we will use some common ML packages in R to predict diabetes among Pima Indians.
- Specifically we will use the random forest algorithm because it allows us to retreive variable importance.

Questions

Questions?