Seeing the Forest through the Trees:

A Gentle Introduction to Machine Learning

David Muchlinski

Georgia Institute of Technology

Table of contents

Outline

First Things First

- To get the most out of this lecture, you will need:
    - At least one (preferably two or more) courses in probability theory or econometrics, ideally up to Maximum Likelihood Estimation
    - R or R Studio installed on your machine and at least some familiarity with the basics of R programming (i.e. building and analyzing regression models).
    - The Caret library installed in R or R Studio
    - Python is acceptable if you work in that language. I work exclusively in R, but the theory is the same for Python, and from what I understand, the syntax is largely similar.
    - The days of STATA or SPSS are over, we're well past those training wheels at this point.

First Things First

- After lunch, we will focus on some applications of machine learning to real world data. Visit my website here XXXXX to download the .csv files for the data as well as replication R code for our first application walk-through. You can access these slides there as well.
- After going through the replication, you will be asked to do your own analyzes in Caret, including running your own machine learning models, and analyzing their outputs.
- If you have not installed R, R Studio, or caret (with dependencies) yet, I suggest you do so now, as caret alone will take at least a hour to download with dependencies. Downloading it without dependencies is possible, but you may have questions when attempting to initialize certain models later on.
- Any initial questions?

Overview of Machine Learning

A Gentle Introduction to Machine Learning

- What is Machine (Statistical) Learning?
- What is it used for?
- Why is it suddenly everywhere?
- How is it different from traditional social science quantitative analysis (regression modeling)?

What is Statistical Learning?

- **Statistical learning** refers to a vast set of tools for understanding data[1].
- Statistical learning uses various non-parametric **algorithms** to estimate relationships among various outputs and inputs
- Statistical learning techniques, broadly, fall into two classes
    - **Supervised**: predictive target is specified in advance (i.e. fraud detection, predicting party identification, predicting civil war, sentence parsing for event data)
    - **Unsupervised**: no predictive target specified (i.e. topic modeling, clustering) - Not dealt with in this lecture

[1] James et al. (2013). An Introduction to Statistical Learning: with Applications in R. Springer Texts in Statistics

What is Statistical Learning?

- Think back to your first regression modeling class.
- In that class, you modeled the relationship between two variables X and Y according to the following function
  - $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ where
  - $y = \beta_0 + \beta_1 x1 + \epsilon$

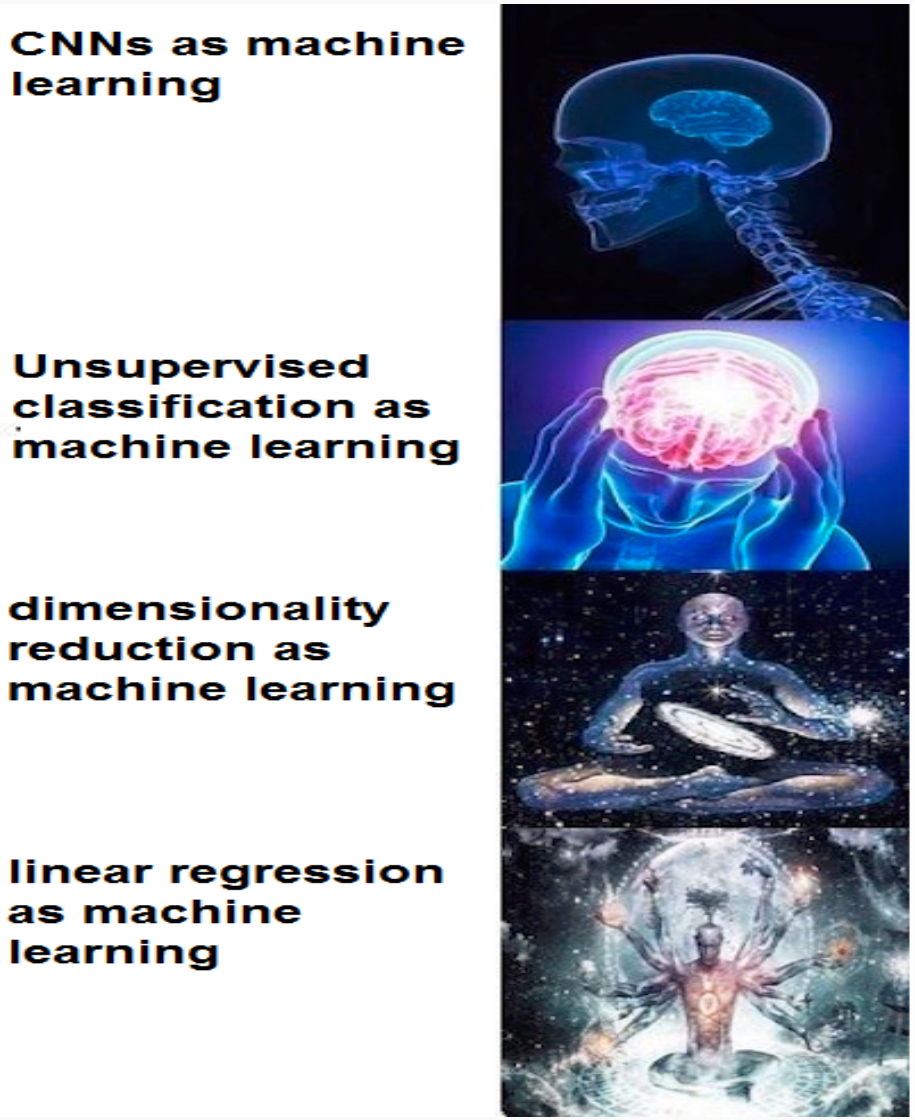
What is Statistical Learning?

- This model is a **parametric** model. It assumes a particular relationship between X and Y
- It measures the probability of observing a relationship between X and Y *at least as extreme* as that observed *assuming the null hypothesis represents the true state of the world*
- X variables have a statistically significant relationship with Y iff p is less than some certain threshold, usually 0.05.
- Goodness of fit tests are determined by R^2 to determine whether a linear relationship between X and Y is a useful approximation of the actual empirical relationship between those two variables in some larger population

What is Statistical Learning?

- By contrast, machine learning is **non-parametric**
- Instead of assuming a relationship from the data *a priori* and then testing to see if that relationship is true, machine learning uses various methods to estimate the relationship $f(x)$ directly from the data itself
- Machine learning methods run the gamut of "parametric-ness"
    - Some, like Ridge Regression or the Lasso are based on the linear model, but use non-parametric techniques including **regularization** to enhance predictive accuracy
    - Others, like deep neural networks are completely black boxes where it is almost impossible to determine the relationship between inputs and outputs

What is Statistical Learning Used For?



- Prediction - what predictors are associated with civil war onset?
- Inference - by how much does the probability of civil war increase if infant mortality increases by 10%?
- Flexible vs. restrictive approaches
- OLS is a tool of machine learning too

What is Statistical Learning Used For?

- Restrictive Methods
  - Linear Regression (OLS, logit, probit)
  - Lasso, Elastic Net, penalized regression models
  - Principal Components Analysis
  - **Use when the goal is inference**

- Flexible Methods
  - Support Vector Machines, splines
  - Boosting, bagging, tree-based methods
  - Neural Networks
  - **Use when the goal is prediction**

What is Statistical Learning Used For?

- Which goes to say, what makes statistical learning is not this or that algorithm
- It is a different say of understanding *how to do* statistical modeling
- Yes, some methods are vastly different from standard regression modeling, but standard models can be successfully incorporated into statistical learning too
- Hopefully by the end of today, you will get a basic understanding of how ML is philosophically different from regression modeling.

Why is Statistical Learning Suddenly Everywhere?

- 2 main factors
  - Increased computing power
  - Proliferation of data
- Both are necessary for statistical learning to be effective
- Sample size: $s = \frac{\sum_1^n (x_i - \bar{x})^2}{n-1}$
- Sample size: cross-validation, bias-variance trade-off

Why is Statistical Learning Suddenly Everywhere?

- Also the Zeitgeist - Facebook and election hacking, the Internet of Things, Quantum Computing, cyber attacks on Iranian C&C systems
- In short, the Digital Revolution would not be possible without machine learning
- We can also not understand the effects of this revolution on society without machine learning
- 2.5 quintillion bytes of data are currently created each day.
    - Over the last two year, humans created 90% of all the data ever produced in recorded history.
    - The increasing pace of human-produced data makes 19th century statistical methods obsolete

How is Statistical Learning Different from Regression Modeling?

- Differences in data (images, text, sound)
- $p >> n$ is no longer a problem
- Different focus, different questions
- But some problems still remain - "Big Data" will not eliminate fundamental statistical problems (sometimes it makes them worse)[2]
    - Causal inference
    - Omitted variable bias
    - Endogeneity
    - Interpretation

[2] Titiunik, R. (2015). Can big data solve the fundamental problem of causal inference?. PS: Political Science Politics, 48(1), 75-79.

How is Statistical Learning Different from Regression Modeling?

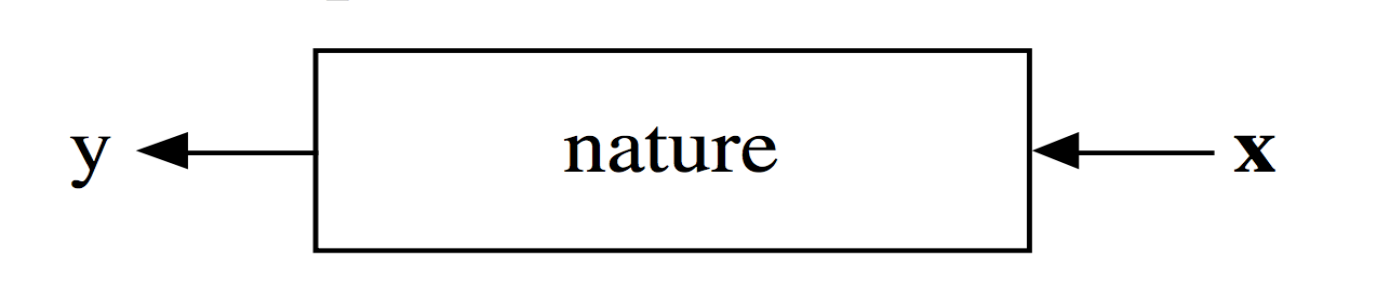
- Assuming we keep model $\mathcal{M}$ constant at $\mathcal{M}$=OLS

1. Focus is on prediction vs. explanation
2. **Cross-validation**
3. Exchange higher **variance** in parameter estimation across model subsets for lower **bias** in final model
4. Split-sample train and test sets
5. Model validation done by **out-of-sample** measures of fit vs. in-sample fit measurements
6. Focus is on the **model**, not the **parameters**
7. There are no p-values or standard errors

Prediction vs. Explanation, Regression Modeling vs. Machine Learning

Short Theoretical Exposition

- All statistics starts with data[3].
- Assume some data X and Y that are generated according to some DGP, we'll call $\mathcal{N}$ for now
- So $\mathcal{N}$ basically orders reality according to some process that is essentially unknown

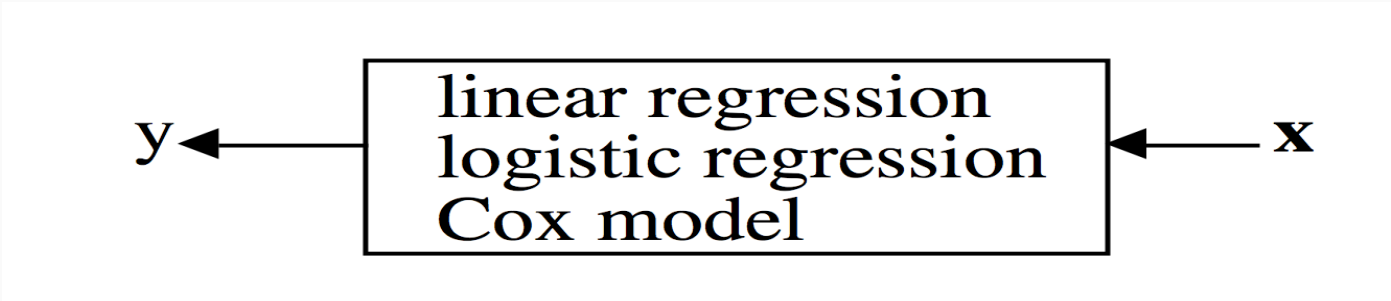


- What Leo Breiman calls "the two cultures" of statistical modeling understand the process of estimating $\mathcal{N}$ in different ways

[3] Breiman, Leo (2001). "Statistical Modeling: the Two Cultures. Statistical Science 16(3) 199-231

Short Theoretical Exposition

- The "Data Modeling" Culture[4]
- This culture starts by assuming $\mathcal{N} \sim$ i.i.d. Normal, Poisson, Weibull

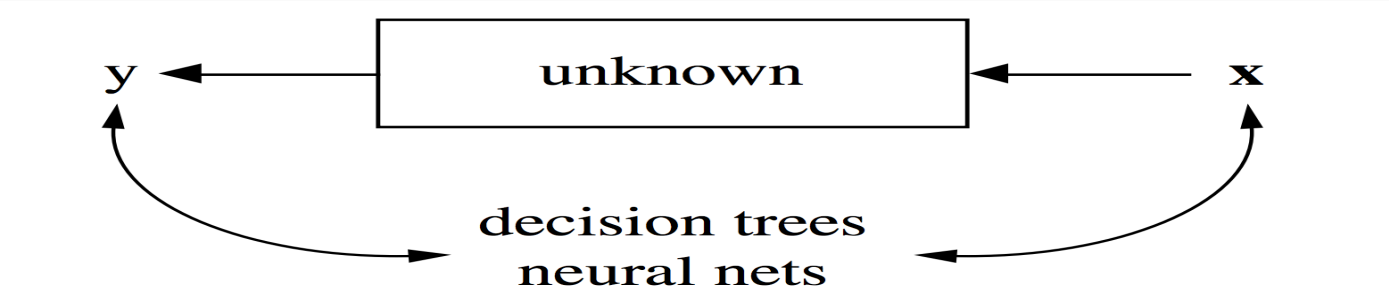


- Inputs influence Y according to known parametric DGP, which can be estimated using in-sample goodness of fit tests, combined with examination of residuals
- X affects Y iff $P_x < 0.05$

[4] Breiman, Leo (2001). "Statistical Modeling: the Two Cultures. Statistical Science 16(3) 199-231

Short Theoretical Exposition

- The “Algorithmic Modeling Culture”
- The nature of $\mathcal{N}$ is complex and unknown (unknowable?)
- Rather than assuming $\mathcal{N} \sim$ Normal, Negative Binomial, Beta, this “culture” simply wants to find a function f that does a good job of using X to predict the response Y



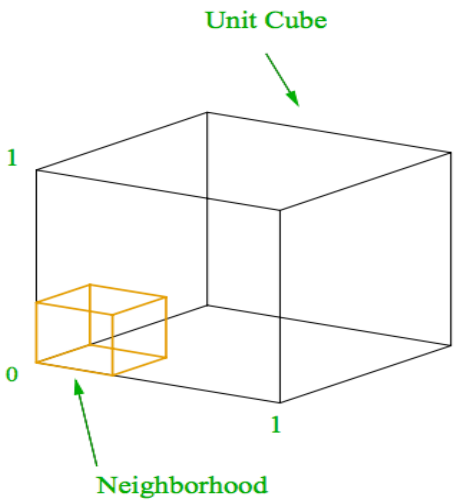
- Model validation rests on out-of-sample predictive accuracy

Short Theoretical Exposition

- Why do these two cultures adopt such different approaches to estimating $f(\mathcal{N})$?
- Each culture's response is an attempt to overcome the **Curse of Dimensionality**
- Simply put, it states that as a statistical model's parameter space increases linearly (1+2+3+4+5+6), data sparsity increases exponentially (1+2+4+8+16+32)
- Formally, as the number of parameters of a model increases, the total number of possible models is $\mathcal{M}(2^p)$

Short Theoretical Exposition

- Here is the CoD illustrated[5].



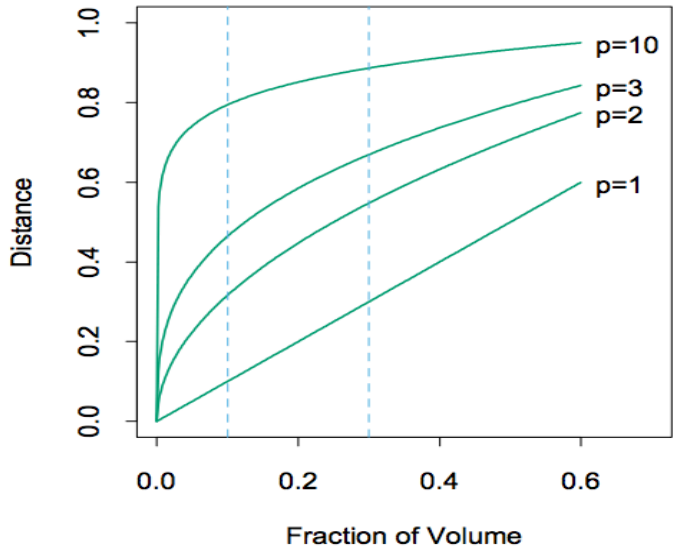


FIGURE 2.6. *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction r of the volume of the data, for different dimensions p. In ten dimensions we need to cover 80% of the range of each coordinate to capture 10% of the data.*

[5]Hastie, Trevor, Tibshirani, Robert, and Freidman, Jerome (2009). The Elements of Statistical Learning, 2nd Editon. Springer

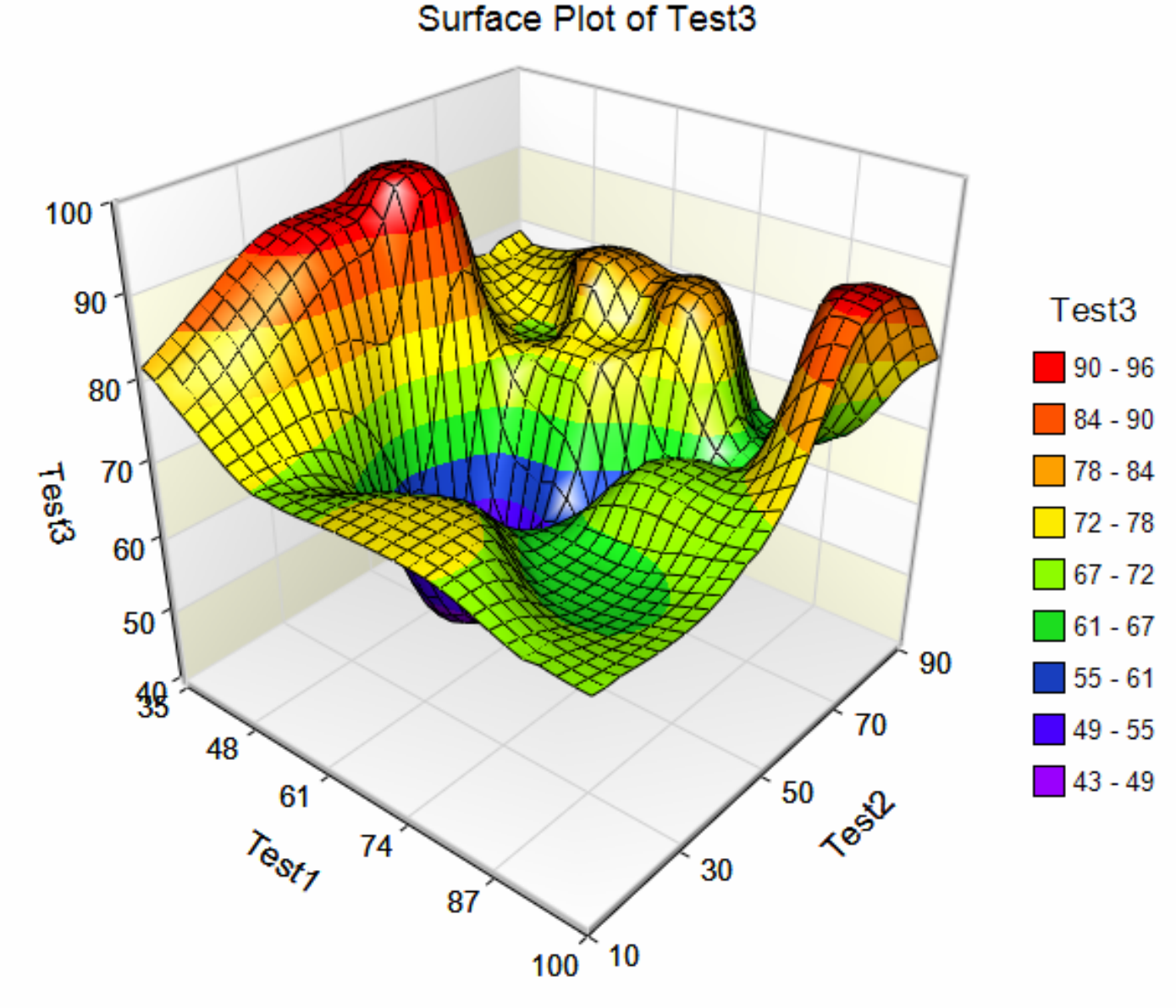
How to Overcome the CoD

- Data Modeling Culture:
  - Set restrictions on the shape of the relationship of $f(\mathcal{N})$
  - By considering only one possible relationship between X and Y, you are forcing the data to fit only a fraction of points in the data. Reduces amount of data needed to estimate a model.
  - Beware: Any linear model will do a poor job of predicting new Y if the true relationship between X and Y is non-linear

How to Overcome the CoD

- Algorithmic Modeling Culture
    - Use re-sampling methods (boosting, bagging, dropout), regularization/penalization, or other methods to estimate $f(\mathcal{N})$ directly from the data itself
    - Iteratively sample the data space T times until there is sufficient data to estimate $f(\mathcal{N})$ without **overfitting**
    - Beware: Re-sampling methods (and machine learning in general) require much more data to estimate $f(\mathcal{N})$ than parametric models because no restrictions are being placed on the form of f

Estimate Directly or Assume a Model?



Why Machine Learning?

- Statistical significance is a poor measure of predictive accuracy[6].
- Traditional regression modeling fits the model to the *population*[7]. This means most models are fitting noise rather than signal.

Table III. Number of correctly predicted onsets and false positives at varying cut-points

	Fearon & Laitin model		*Collier & Hoeffler model*	
Threshold	*Correctly predicted*	*False positives*	*Correctly predicted*	*False positives*
0.5	0/107	0	3/46	5
0.3	1/107	3	10/46	20
0.1	15/107	66	34/46	110

[6]Ward, M. D., Greenhill, B. D., & Bakke, K. M. (2010). The perils of policy by p-value: Predicting civil conflicts. Journal of peace research, 47(4), 363-375.

[7]This is dumb and wrong. Schrodt, P. A. (2014). Seven deadly sins of contemporary quantitative political analysis. Journal of peace research, 51(2), 287-300.

Why Machine Learning?

- Most significant variables fail to predict any events in out-of-sample data
- Implications for theory
    - Do we actually understand the causes of civil war if we can't predict where/when a new onset will occur?
- Implications for policy
    - Are we giving correct advice to policymakers?

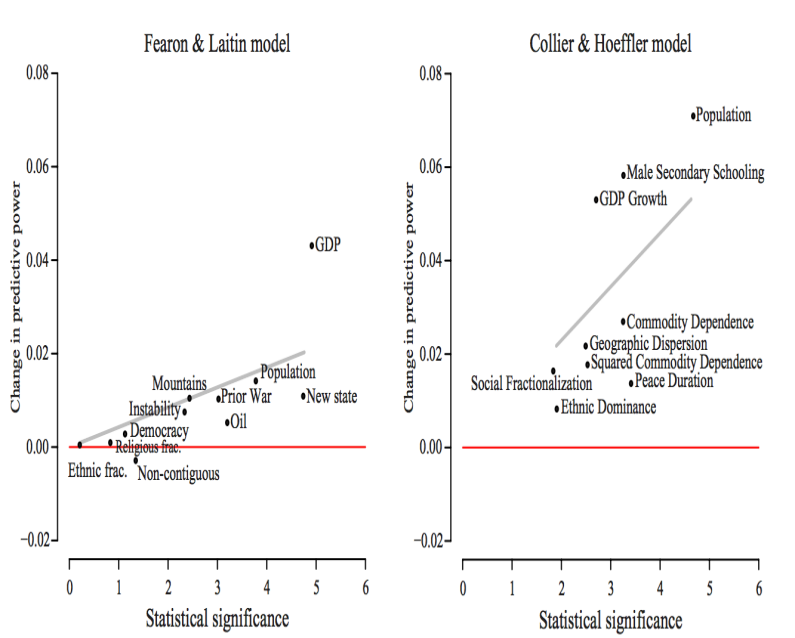


Figure 2. Comparison of predictive power and statistical significance

Superior Prediction for Machine Learning vs. Regression Modeling

- Due to the limitations of regression modeling, we want to discover if there are other methods that might be better able to predict events of interest (civil wars, regime changes, partisan realignments, etc...)
- Fundamentally, these limitations stem from the parametric assumptions we fit to our data. Most relationships between X and Y are not linear.
- Let's visualize why flexibility in estimating $f(\mathcal{N})$ may provide some benefit in prediction [8]

[8] Following slides are from Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction, Springer Series in Statistics.

Superior Prediction for Machine Learning vs. Regression Modeling: Simulated Data

- Linear Regression for Binary Predictor: i.e. Logistic Regression
    - Lots of misclassification
    - Restrictive decision boundary
    - Any linear boundary will induce high rate of error in this data

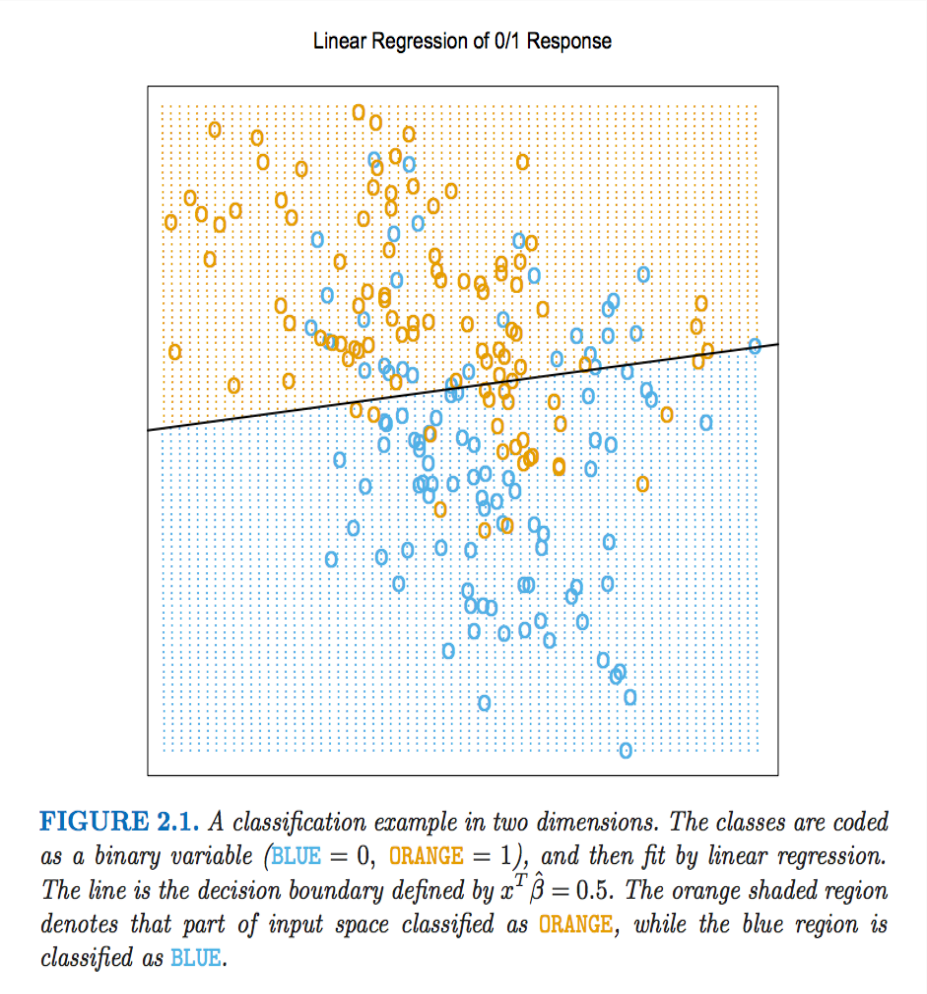


FIGURE 2.1. *A classification example in two dimensions. The classes are coded as a binary variable (BLUE = 0, ORANGE = 1), and then fit by linear regression. The line is the decision boundary defined by* $x^T\hat{\beta} = 0.5$. *The orange shaded region denotes that part of input space classified as ORANGE, while the blue region is classified as BLUE.*

Superior Prediction for Machine Learning vs. Regression Modeling: Simulated Data

- A Machine Learning Approach - k Nearest Neighbors (k=15)
    - Less misclassification/error
    - More “wiggly” decision boundary
    - Pretty good, but can we do better?

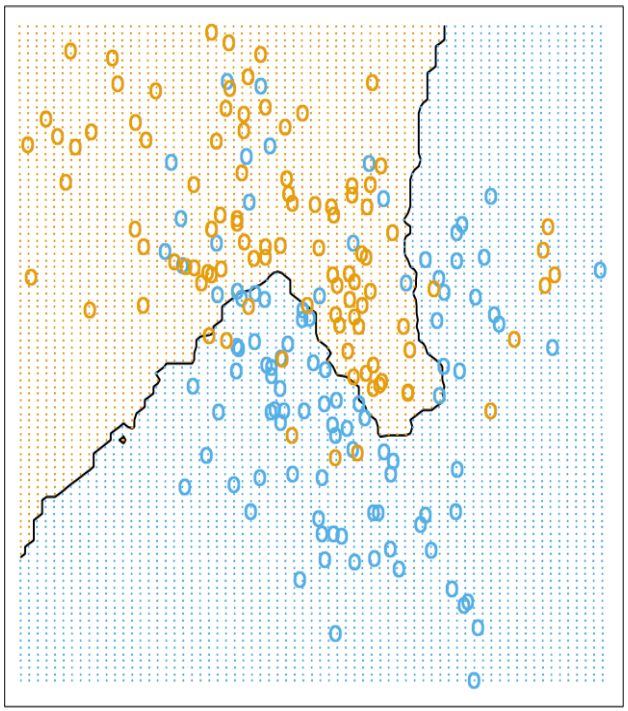


FIGURE 2.2. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$) *and then fit by* 15*-nearest-neighbor averaging as in (2.8). The predicted class is hence chosen by majority vote amongst the* 15*-nearest neighbors.*

Superior Prediction for Machine Learning vs. Regression Modeling: Simulated Data

- A Machine Learning Approach - k Nearest Neighbors (k=1)
    - Near perfect accuracy
    - Multiple decision boundaries
    - What's the possible danger here?

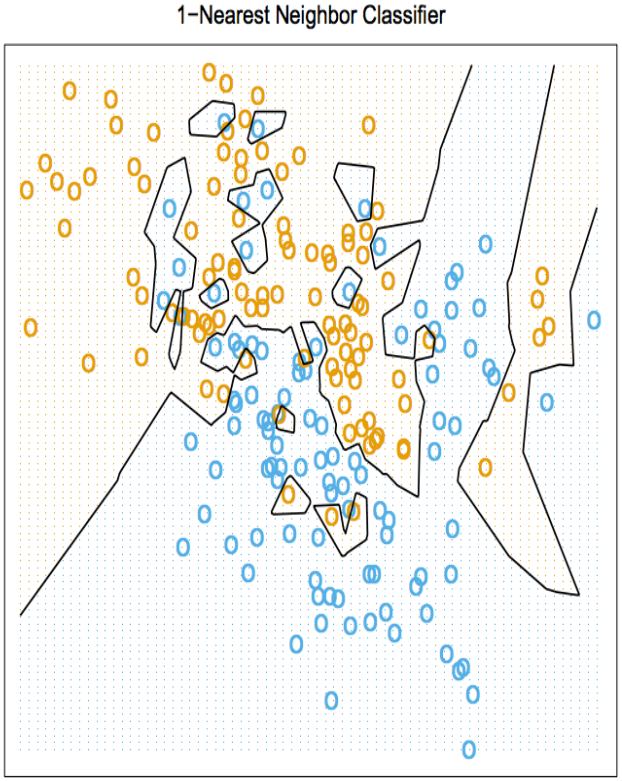


FIGURE 2.3. *The same classification example in two dimensions as in Figure 2.1. The classes are coded as a binary variable* (BLUE $= 0$, ORANGE $= 1$)*, and then predicted by 1-nearest-neighbor classification.*

Superior Prediction for Machine Learning vs. Regression Modeling: Simulated Data

- Error Rates for Regression and Nearest Neighbors
    - NN has far less error than linear regression model
    - Except as # neighbors approaches *N*
    - Optimal # of neighbors appears to be about 10

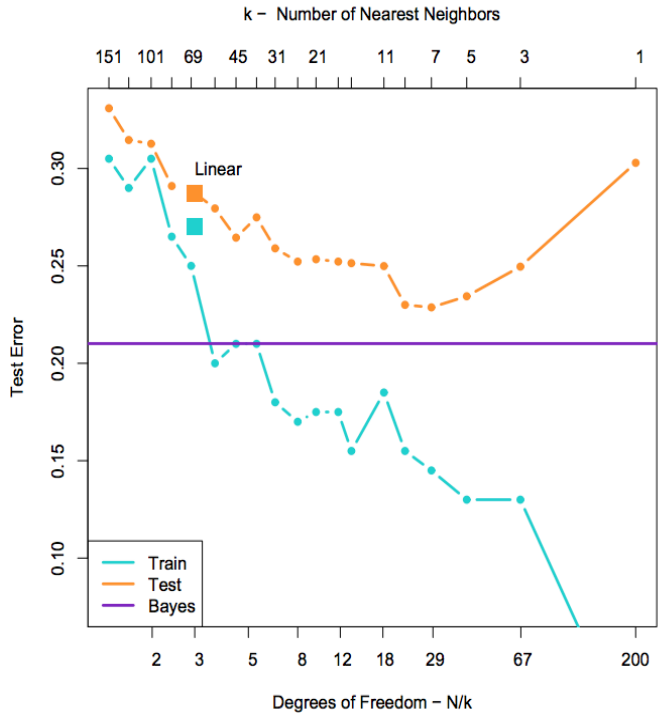


FIGURE 2.4. *Misclassification curves for the simulation example used in Figures 2.1, 2.2 and 2.3. A single training sample of size* 200 *was used, and a test sample of size* 10,000. *The orange curves are test and the blue are training error for k-nearest-neighbor classification. The results for linear regression are the bigger orange and blue squares at three degrees of freedom. The purple line is the optimal Bayes error rate.*

Superior Prediction for Machine Learning vs. Regression Modeling: Empirical Data

Table 1 Predicted probability of civil war onset: Logistic Regression and Random Forests

	Models and predicted probability of civil war onset			
Civil war onset	*Fearon and Laitin (2003)*	*Collier and Hoeffler (2004)*	*Hegre and Sambanis (2006)*	*Random Forests*
Afghanistan 2001	0.01	0.01	0.01	0.09
Angola 2001	0.04	0.01	0.01	0.13
Burundi 2001	0.00	0.00	0.00	0.05
Guinea 2001	0.00	0.00	0.01	0.22
Rwanda 2001	0.02	0.00	0.00	0.56
Uganda 2002	0.03	0.05	0.00	0.81
Liberia 2003	0.01	0.03	0.00	0.94
Iraq 2004	0.04	0.01	0.00	0.68
Uganda 2004	0.02	0.01	0.02	0.52
Afghanistan 2005	0.01	0.02	0.01	0.14
Chad 2006	0.01	0.07	0.02	0.21
Somalia 2007	0.00	0.00	0.00	0.52
Rwanda 2009	0.00	0.01	0.00	0.74
Libya 2011	0.00	0.01	0.00	0.34
Syria 2012	0.00	0.04	0.00	0.25
DR Congo 2013	0.00	0.00	0.00	0.76
Iraq 2013	0.01	0.00	0.00	0.25
Nigeria 2013	0.01	0.00	0.00	0.25
Somalia 2014	0.01	0.04	0.01	0.87

Prediction, Explanation, Theory, and Machine Learning: Integrating Machine Learning into Quantitative Social Science

So...What's to Be Done?

- Regression models lack predictive power, despite the presence of statistically significant variables which purport to "explain" why an event occurs.
- Yet if a variable explains shy something occurs, we should expect to find that "cause" when observing the same event in the future.
- But as Ward et al. (2010) show, this isn't the case with our regression models

Some Philosophy of Science

- Deductive Regression Modeling
    1. Build theory
    2. Collect data
    3. Test data against theory
    4. Make conclusions
    5. Repeat if necessary

Philosophy of Regression Modeling

- Chose variables identified by theory as causal
- Chose variables suggested by literature as conditional "controls"
- If causal variable is significant, given joint distribution of controls, then *X causes Y*
- Chuck in fixed/random FX, clustered standard errors, IVs, DID, etc...as causal identification strategies
- Examine residuals, conduct goodness of fit tests, examine effect sizes
- Conclude with causal story or policy advice

Problems with this Approach

- Theoretical problems[9]
    1. "Garbage Can Models" and the problem of multicollinearity in linear models[10]
    2. Misunderstanding model assumptions (perfectly measured predictors, error term uncorrelated with response, relationship between X and Y is linear)
    3. Incorrect utilization of NHST (i.e. we formulate our hypotheses as Bayesians, but test them as frequentists) and interpretation of p-values[11].

---

[9] Schrodt, P. A. (2014). Seven deadly sins of contemporary quantitative political analysis. Journal of peace research, 51(2), 287-300.

[10] Achen, C. H. (2005). Let's put garbage-can regressions and garbage-can probits where they belong. Conflict Management and Peace Science, 22(4), 327-339.

[11] Gill, J. (1999). The insignificance of null hypothesis significance testing. Political research quarterly, 52(3), 647-674.

So...What's to be Done?

- Theory:
  1. Reevaluate what theory is[12]
  2. Do we have to begin with theory? Exploratory data analysis.
  3. Identify large causal factors, move away from "proxies" and minutiae (i.e. "Greed" vs. "Grievance" debate is a good example of what factors to focus on)

[12] Ward, M. D. (2016). Can we predict politics? Toward what end?. Journal of Global Security Studies, 1(1), 80-91.

So...What's to be Done?

- For Policy:
  1. Focus on predictive accuracy - often what the client wants to know anyway
  2. Theory is probably irrelevant anyway
  3. If data collection is inexpensive, utilize as "big" a dataset as you can, with appropriate tools

More Philosophy of Science: Prediction vs. "Explanation"

- Conflation of prediction and explanation in statistics literature[13]
- Both are necessary for generating and testing theory, but in different ways

[13] Shmueli, G. (2010). To explain or to predict?. Statistical science, 25(3), 289-310.

Scientific Functions of Predictive ModelingShmueli, G. (2010). To explain or to predict?. Statistical science, 25(3), 289-310.

1. Uncovering new relationships and hypotheses (esp. in large datasets)
2. Discovering new/different measures or operationalizations of key variables
3. Enhancing current explanatory models by capturing more complex (non-linear, interactive) relationships
4. Greater external face validity. More immediate generalization.
5. **Straightforwrd way of assessing competing theories**.
6. **Gold standard of theory testing. Establishing baseline measures of theory validity**.

Are Explanation and Prediction Different?

- In one sense, explanation and prediction *should be* two sides of the same coin.
- For a model to explain something, it must be able to predict it at some level.
- The difference comes down to how models are *used* and why they are used that way
- But the original formulation is not wholly wrong. Model validation should be understood as a continuum where models are subjected to increasingly stringent *predictive* tests[14].

[14] Cranmer, S. J., Desmarais, B. A. (2017). What can we learn from predictive modeling?. Political Analysis, 25(2), 145-166.

Are Explanation and Prediction Different?

- Assume a model $F(x)$:
    - Assume further some relationship between Y and X represents F
    - The goal of statistical analysis is to estimate F, but there are the normal problems preventing us from doing so
    - Because of these problems, we cannot estimate F, so we instead settle for estimating a series of f's
    - But which f is the best representation of F?

Are Explanation and Prediction Different?

- Standard response: what is the goodness of fit of f?.
- Acceptable, but we now know this is steering between the rock of omitted variable bias the the hard place of an overfit model (i.e. Fearon and Laitin (2003)).
- Is our theory detailed enough to specify *every* variable that should belong in our regression equation, including controls?
- Addition of spurious variables causes regression coefficients to "jump around like a box of gerbils on methamphetamines"[15].

[15] Schrodt, P. A. (2014). Seven deadly sins of contemporary quantitative political analysis. Journal of peace research, 51(2), 287-300.

Are Explanation and Prediction Different?

- So, if we cannot reliably estimate F from a series of f's (at least not with standard regression techniques), why not attempt to estimate F in another way?
- Estimate F from the data directly
- Estimate how far from F a given f is by measuring different in predictive accuracy.
- At least we always know what F's predictive accuracy is. So it gives us a useful baseline against which to measure.

Introducing Machine Learning with Classification and Regression Trees (CART)

Binary Classification Using Logistic Regression

- Assume that some data were generated according to the following GDP:
- $\mathcal{D} \sim$ Binomial
- From logistic regression, we know:
- $p_i = \frac{1}{1+e^{-x_i\beta}}$, where $-x_i\beta = \mu_i$
- So the DGP is modeled as a function of the proportion of positive cases in the data (i.e. the mean of the DV)

Binary Classification Using Logistic Regression

- Logistic regression models a function $f(x)$ as a linear combination of independent variables to separate 0s from 1s in the data
- It attempts to do so by effectively creating a *hyperplane* through the data that does the best job of putting 0s on one side of the place and 1s on the other

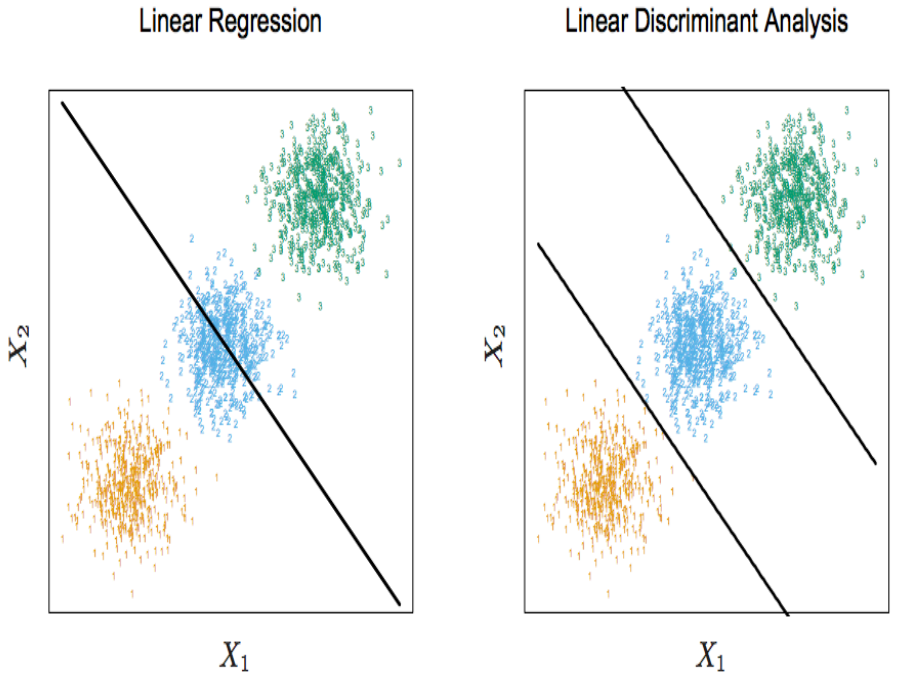


FIGURE 4.2. *The data come from three classes in* $\mathbb{R}^2$ *and are easily separated by linear decision boundaries. The right plot shows the boundaries found by linear discriminant analysis. The left plot shows the boundaries found by linear regression of the indicator response variables. The middle class is completely masked (never dominates).*

Binary Classification Using Logistic Regression

- While a linear fit does a good job of separating the green and orange observations, it does a poor job with the blue observations
- If this was just a simple binary classification problem (ignore blue), no further action would be required.
- However, little data in social science falls into such neatly ordered spaces
- Note than a multinomial logistic regression or linear discriminant analysis would be a good fit here

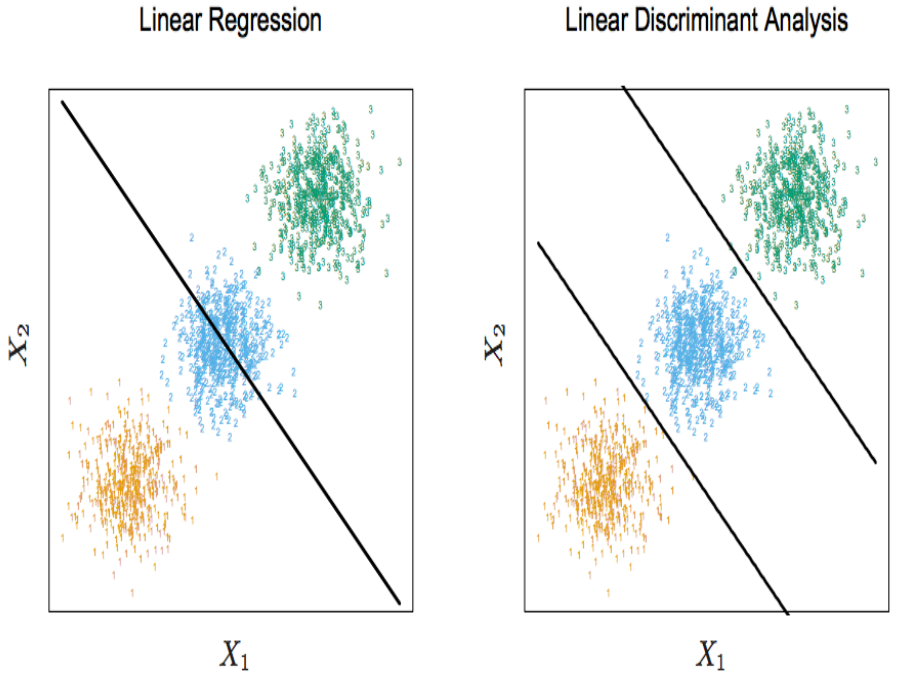


FIGURE 4.2. *The data come from three classes in $\mathbb{R}^2$ and are easily separated by linear decision boundaries. The right plot shows the boundaries found by linear discriminant analysis. The left plot shows the boundaries found by linear regression of the indicator response variables. The middle class is completely masked (never dominates).*

From Logistic Regression to Classification (Decision) Trees

- While logistic regression models the data space according to the joint distribution between the response and predictor variables, CART fits a classification tree to binary data according to different criteria
- The tree-growing algorithm is non-parametric, meaning it does not depend on the distribution of the data to make decisions regarding various cut points in the data.
- CART selects variables to partition the data on by maximizing data homogeneity (i.e. minimizing classification error)

Introducing Tree-Based Methods for Regression and Classification

- A regression (decision) tree is a recursive partitioning of the data space by some value (i.e. a constant) such that the data space becomes more homogeneous on Y according to each additional partition.
- For example, the full dataset exists at the top of the tree, and the algorithm selects the variable to partition the data such that the variable cut point optimizes different values of Y in the newly recreated regions $\mathcal{R}$.
- This process is repeated iteratively until some stopping criterion (i.e. depth, error rate) is achieved.
- The end result, displayed visually, looks something like this:

Introducing Tree-Based Methods for Regression and Classification[16]

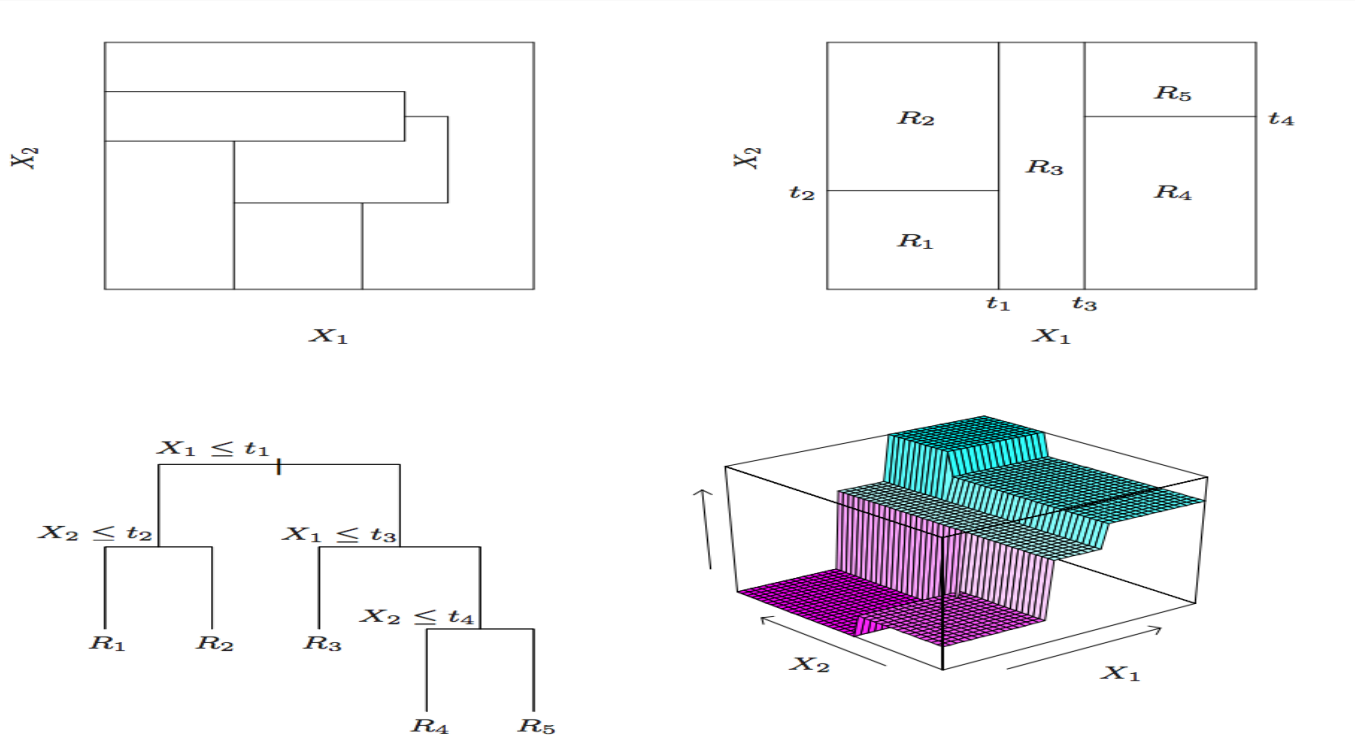


FIGURE 9.2. *Partitions and CART. Top right panel shows a partition of a two-dimensional feature space by recursive binary splitting, as used in CART, applied to some fake data. Top left panel shows a general partition that cannot be obtained from recursive binary splitting. Bottom left panel shows the tree corresponding to the partition in the top right panel, and a perspective plot of the prediction surface appears in the bottom right panel.*

[16] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction, Springer Series in Statistics

Introducing Tree-Based Methods for Regression and Classification

- Like normal trees, regression (classification) trees are “grown” to the data.
- Also like normal trees, their is an optimal size to grow each tree (i.e. minimization of error)
- The tree-growing algorithm selects variables (features) to split the data automatically.
- Tree complexity (i.e. size, topology) is determined by **cross-validation**

Cross-Validation

- **Cross-validation** is a means of reducing bias in the final model by exchanging greater variance in model prediction over k cross-validation "folds" of the original data
- Essentially, cross-validation takes your training data and splits it up into k different smaller datasets (called folds), applies the model to $k - 1$ folds k times, where $k - 1$ folds are used to train the model, and the $k - 1^{th}$ fold is used to validate the model. This is process is done k times such that each fold is used to train *and* validate the model

Cross-Validation[17]

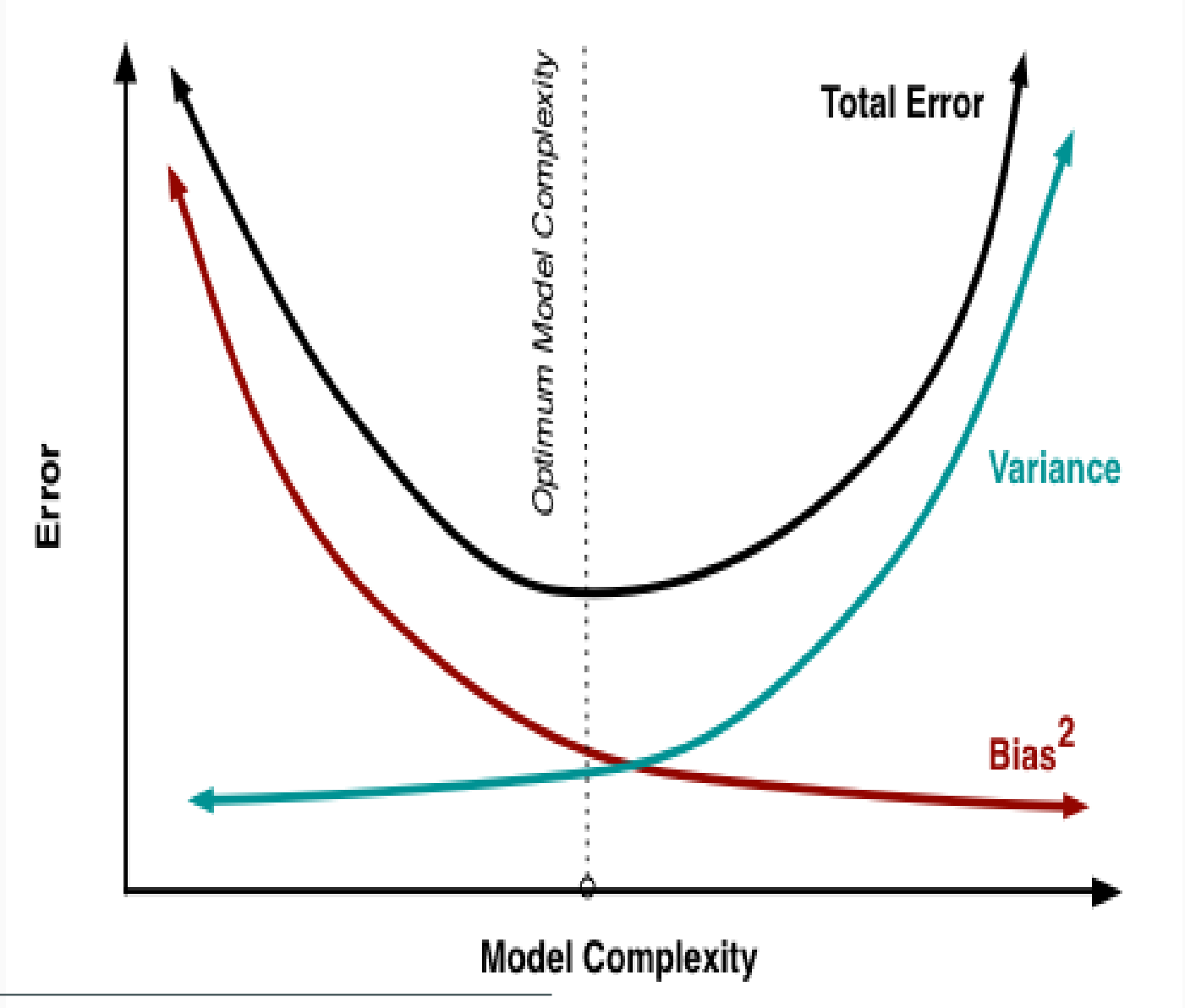
- Here's the process illustrated

[17] https://www.kaggle.com/dansbecker/cross-validation

Why Cross-Validation?

- Cross-validation ensures your model is not **overfit**, meaning that, while accurate (low bias), it will generalize poorly to new data (i.e. high variance)
- All models face a trade-off between minimizing bias and increasing variance.
- For example, Fearon and Laitin's (2003) and Collier and Hoeffler's (2004) models of civil war had very low bias, but could not predict new civil wars in out-of-sample data. The models were **overfit** to the data.
- Any model that is overfit to the training data will generalize poorly to out-of-sample data.

Bias-Variance Trade-off Visualized[18]

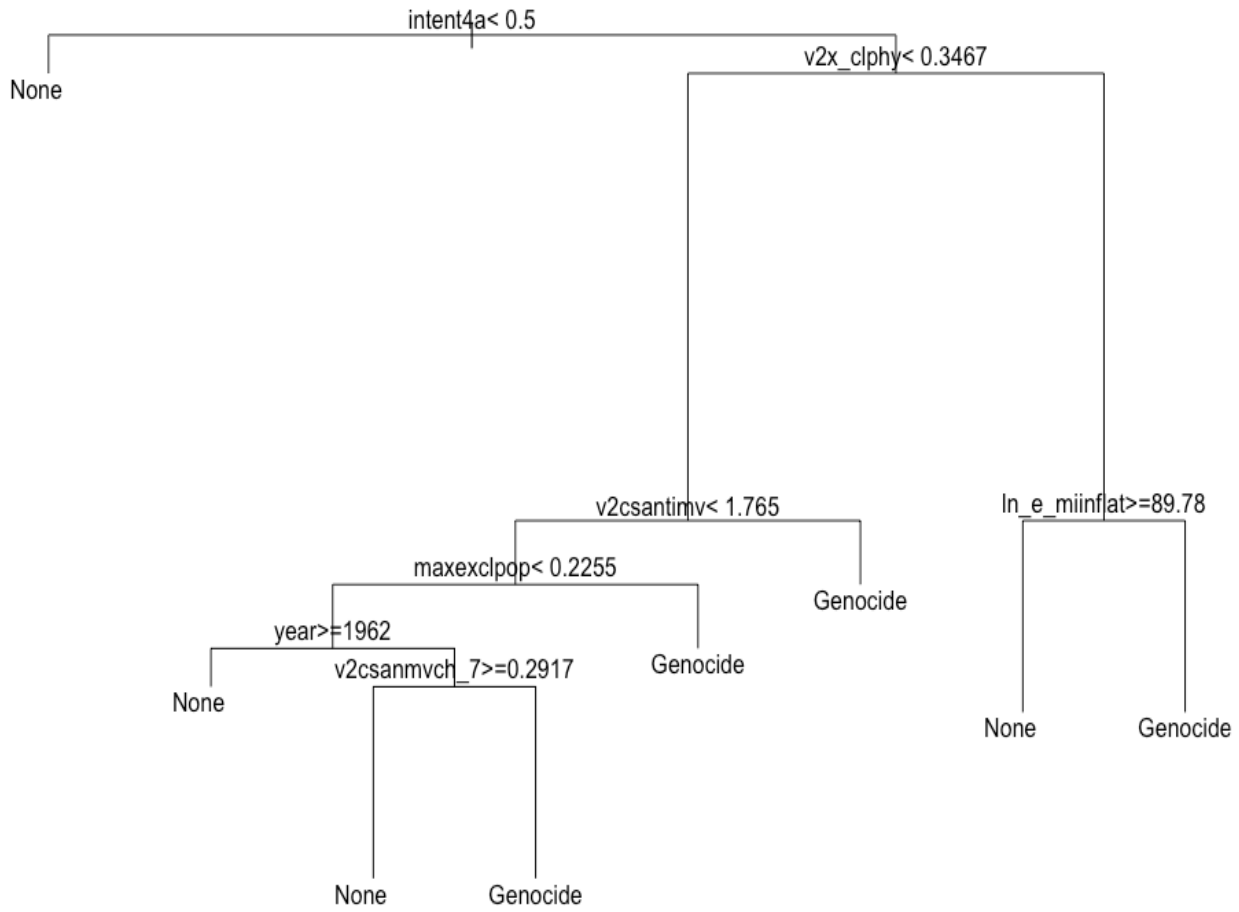


[18] http://scott.fortmann-roe.com/docs/BiasVariance.html

Cross-Validation and Decision Tree Complexity

- Let's say we wanted to grow a classification tree to predict the onset of genocide cross-nationally.
- We do the analysis (shown as follows) and get the following graphical output:

Cross-Validation and Decision Tree Complexity



Interpreting a Tree Visually

- The algorithm starts by inducing a single split on the data space, creating two daughter *nodes* at the end of each partition *branch*
- Splits are akin to a yes-no question. Is the value of the variable less than a certain threshold? If yes, then right, if no, then left.
- Additional partitions are made in the same way, leading to more breaches and nodes.
- This process continues until a stopping criterion is met (usually set by the researcher).
- The resulting structure shows the *nonlinear* and *interactive* structure of the tree. Should be interpreted as X_1 & X_2 & x_3 leads to $Y = 0$ or $Y = 1$

Cross-Validation and Decision Tree Complexity

- We use the **caret** (short for **C**lassification **A**nd **RE**gression **T**raining) library in **R** for the training of all machine learning models
- It has a built-in CV procedure, making CV easy
- It is a *wrapper* meaning it is a one-stop-shop for nearly every ML library in **R**. Nearly every ML algorithm is implementable with just a few lines of code
- Hopefully you have downloaded it, because of its size and dependencies, instillation may take a few hours.

Cross-Validation and Decision Tree Complexity

- First, we need to transform the coding of our DV into a factor variable which will ease interpretation of our results in caret
- If our DV is a binary (0,1) variable, we can use the following code to alter it
- The reason we want to transform our DV into a named factor (**classification only!!!**) is because our our evaluation metric which is the ROC curve. Caret will only give us ROC metrics if the DV is a named factor.

```
training$var.name<-factor(
  training$var.name,
  levels=c(0,1),
  labels=c("None", "Genocide"))
```

Cross-Validation and Decision Tree Complexity

- Next, we want to set up our cross-validation procedure by telling caret what type of cross-validation procedure we want to use, how many folds we want, how caret should present the summary of our models, and some ancillary information (i.e. compute class probabilities for hold-out samples, save model predictions, and allow parallel processing)
- There are many different types of cross-validation, some of which are relevant for various research designs (i.e. time-series forecasting), but generally 10-fold CV is sufficient for the vast majority of tasks. Become familiar with the different types of CV procedures on the caret website (http://topepo.github.io/caret/index.html)

```
fitControl<-trainControl(method="cv",
                         number=10,
                         summaryFunction=twoClassSummary,
                         classProb=T,
                         savePredictions = T,
                         allowParallel = T)
```

Cross-Validation and Decision Tree Complexity

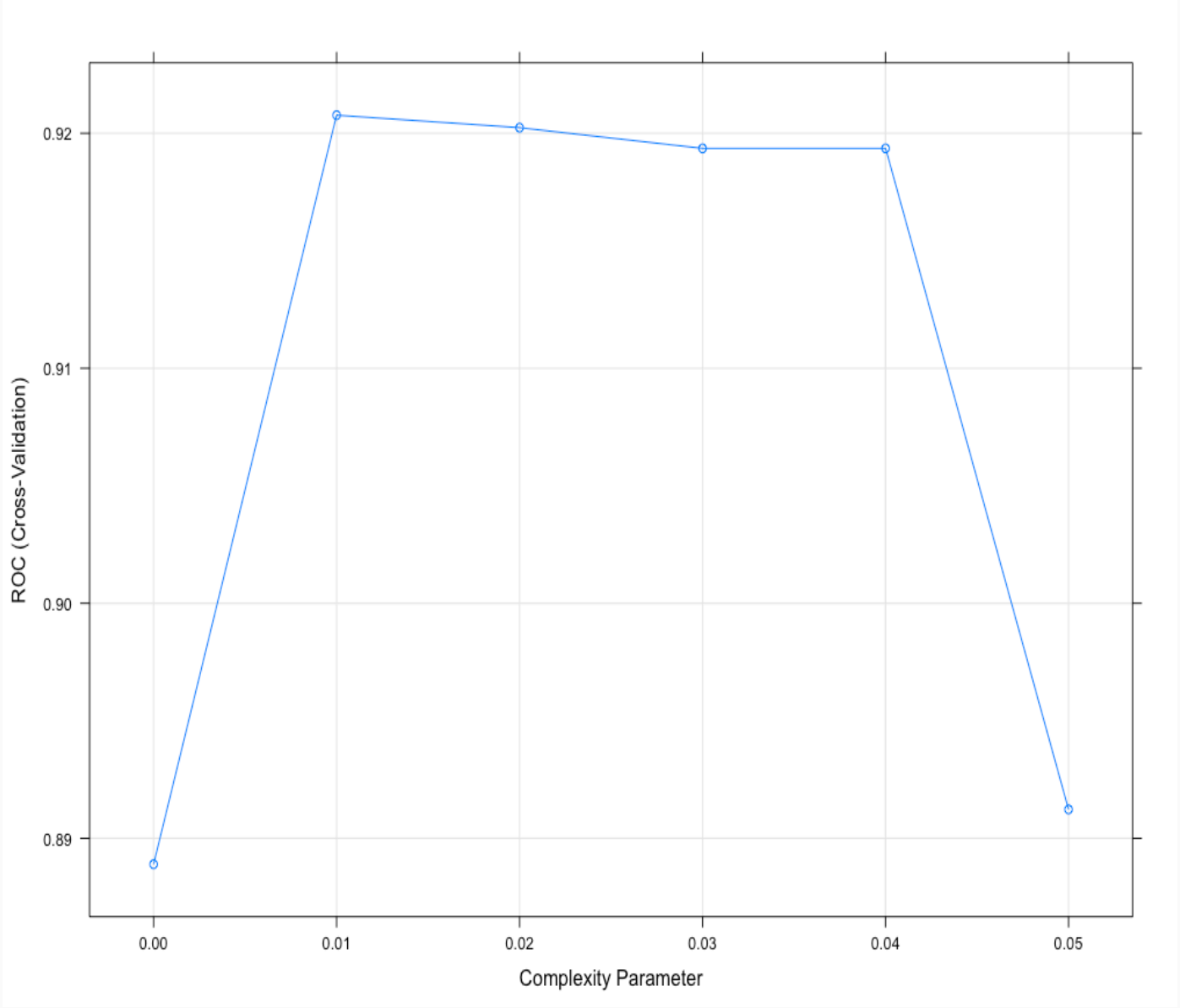
- Now we tell caret to grow a decision tree to our data to predict the onset of genocide/politicide

```
set.seed(99)

TMK.tree<-train(as.factor(genpol.onset)~.,
                method="rpart", # this is caret's regression/decision tree library
                trControl=fitControl,
                metric="ROC",
                tuneGrid=Grid,
                data=training[c(-1)])

TMK.tree
```

- And then tell caret to give us the summary of the various cross-validation runs with the complexity parameter of the tree (the only tunable parameter) varied according to the `Grid` command where `Grid` = (0, 0.05, 0.01)

```
> TMK.tree
CART

5067 samples
  58 predictor
   2 classes: 'None', 'Genocide'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 4560, 4561, 4561, 4560, 4561, 4561, ...
Resampling results across tuning parameters:

  cp    ROC        Sens       Spec
  0.00  0.8888991  0.9958132  0.3533333
  0.01  0.9207687  0.9958132  0.3533333
  0.02  0.9202370  0.9958132  0.3533333
  0.03  0.9193556  0.9974076  0.2400000
  0.04  0.9193556  0.9974076  0.2400000
  0.05  0.8912322  0.9982056  0.2200000

ROC was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.01.
```

Cross-Validation Accuracy Across Tuning Parameters



From Cross-Validation to Out-of-Sample Prediction

- Cross-validation accuracy can still be misleading. Though it helps avoid overfitting, it does not eliminate this risk, plus the model is still fit to the entirety of the training data. Remember Ward et al.'s critique of civil war studies.
- To assess model's predictive accuracy, we need to test the model on data the model **has not yet seen**.
- We do this in machine learning by passing the predictions of the model made on the test data to our training data.

```
### Getting Out of Sample Predictions

pred.tmk.tree<-predict(TMK.tree, newdata=testing, type="prob")
```

From Cross-Validation to Out-of-Sample Prediction

- What this creates is a matrix of predicted probabilities for each class per observation.

```
> pred.tmk.tree[1:10,]
          None     Genocide
44 0.9985585 0.001441516
45 0.9985585 0.001441516
46 0.9985585 0.001441516
47 0.9985585 0.001441516
48 0.9985585 0.001441516
49 0.9985585 0.001441516
50 0.9985585 0.001441516
51 0.9985585 0.001441516
52 0.9985585 0.001441516
53 0.9985585 0.001441516
```

From Cross-Validation to Out-of-Sample Prediction

- We can use these predicted probabilities to measure the accuracy of the model in a number of different ways
- Let's start with something called a "Confusion Matrix"

Confusion Matrix

```
> pred.TMK.tree<-predict(TMK.tree, newdata=testing, type="raw")
> confusionMatrix(pred.TMK.tree, testing$genpol.onset, positive="Genocide", mode="everything")
Confusion Matrix and Statistics

          Reference
Prediction None Genocide
  None     4512       16
  Genocide   36        4

               Accuracy : 0.9886
                 95% CI : (0.9851, 0.9915)
    No Information Rate : 0.9956
    P-Value [Acc > NIR] : 1.000000

                  Kappa : 0.1282
 Mcnemar's Test P-Value : 0.008418

            Sensitivity : 0.2000000
            Specificity : 0.9920844
         Pos Pred Value : 0.1000000
         Neg Pred Value : 0.9964664
              Precision : 0.1000000
                 Recall : 0.2000000
                     F1 : 0.1333333
             Prevalence : 0.0043783
         Detection Rate : 0.0008757
   Detection Prevalence : 0.0087566
      Balanced Accuracy : 0.5960422

       'Positive' Class : Genocide
```

Unpacking Machine Learning Metrics

- There are no p-values and standard errors in machine learning. Each model can be evaluated according to different metrics, and some metrics you may want to prioritize over others depending on your goal.
- *In general* the major accuracy metrics commonly reported in the literature as precision, recall, F-1, ROC-AUC, and for class-imbalanced data - PR-AUC
- What are these things measures of?

Unpacking Machine Learning Metrics

- Precision: the fraction of relevant instances among retrieved instances
  - $\frac{truepositives}{truepositives+falsepositives}$
- Recall: the fraction of relevant instances retrieved from the total number of possible positive instances
  - $\frac{truepositives}{truepositives+falsenegatives}$
- F-1 Score or F-measure: the harmonic mean of precision and recall
  - $2 \cdot \frac{precision \cdot recall}{precision+recall}$

Unpacking Machine Learning Metrics

- If your goal is simply accurate detection of events of interest (i.e. civil war onsets), maximize precision
- If your goal is to accurately predict civil war onsets in “at risk” countries, maximize recall
- If your goal is to build an accurate classifier that can effectively predict civil war onsets while minimizing false positives and false negatives, maximize F-1
- There is an inherent trade-off for precision and recall, you cannot maximize both

Unpacking Machine Learning Metrics

- Precision, recall, and F-1 are point estimates. You can bootstrap confidence intervals around these estimates, but since its simulated data, these CI's generally aren't reported.
- The number of instances predicted in the Confusion Martix, further, depends on a cut off of 0.50 for positive prediction.
- Often, however, we are interested in a range of predictions across a range of possible thresholds. Rarely is a country at 50% risk for civil war onset
- ROC curves are a way to visualize this range of predictions across all probability thresholds. The AUC (Area Under the Curve) gives the probability that a classifier will assign a higher predicted probability to a true positive than a true negative

ROC Curves

- AUC values for ROC range (theoretically) from 0-1
- Typically, values are only relevant from 0.5-1
- An AUC of 0.50 represents a random guess for each observation, while 1 represents perfect prediction of all observations
- Graphically, the closer to the upper-left corner the curve is, the better the model
- Use the aptly named **ROCR** library to draw your ROC plots

Unpacking Machine Learning Metrics

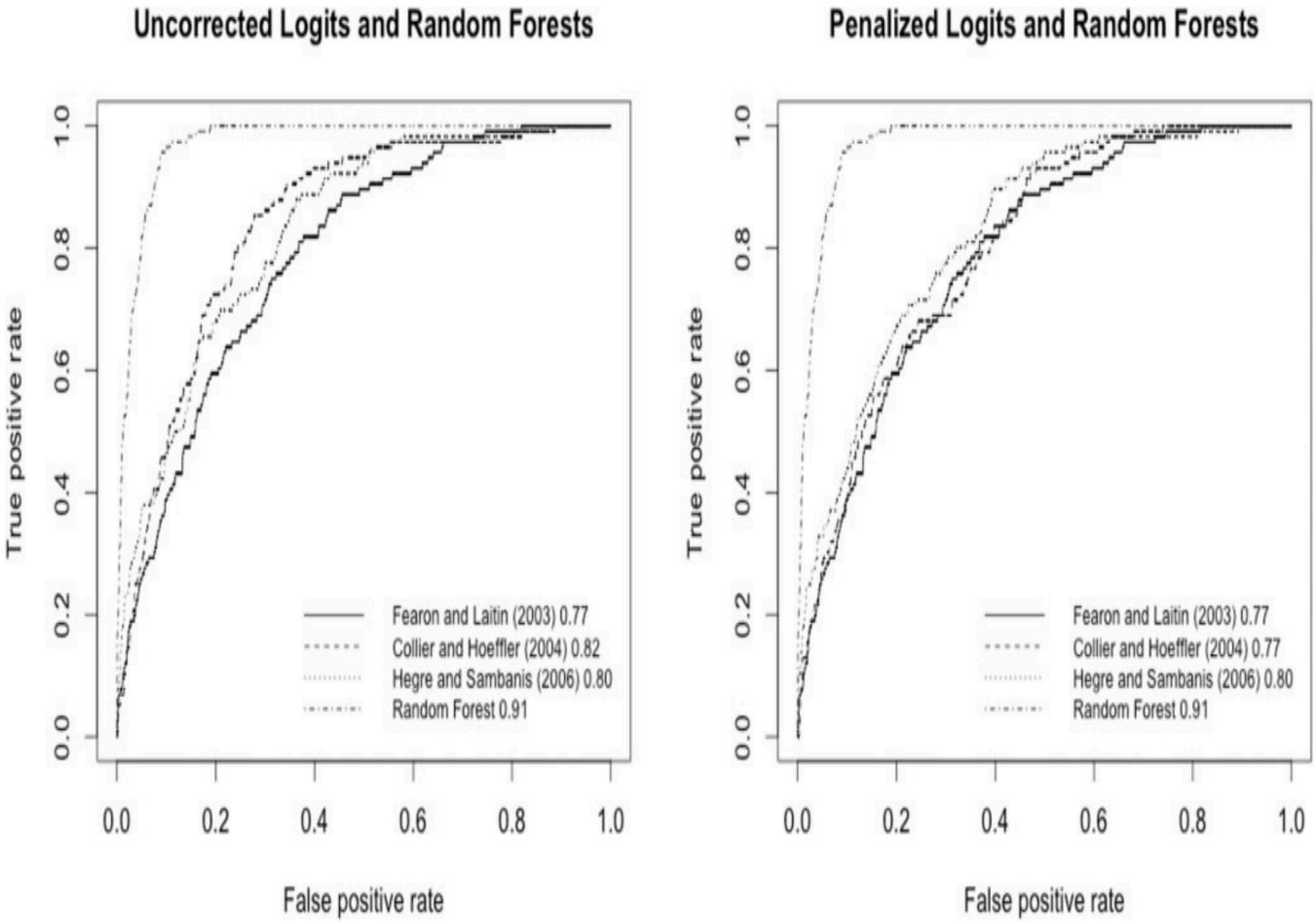


Fig. 2 ROC curves for all classifiers.

Class-Imbalanced Data and Accuracy Metrics

- Let's assume some data where the prevalence of negatives (0s) is 100 times that of positives (1s)
    - Not uncommon: fraud detection, most political violence problems
- You put a decision tree to the data, and *viola*! 98% accuracy!
- But looking at the data, 99% of all observations simply belong to one class. Your classifier is doing a good job of minimizing false positives, but it cannot predict a single true positive!
- Often, the majority class (0s) are substantively uninteresting. Who cares if you predict that France won't experience a civil war? We already knew that!
- This is a problem of rare events, and it plagues ML just as much as standard regression

Class-Imbalanced Data and Accuracy Metrics

- When dealing with class-imbalanced data, refrain from relying on ROC curves
- Use instead F-1 scores and the Precision-Recall curve
- Like the ROC curve, PR Curve has an AUC value
- AUC PR ranges from a baseline of $\frac{pos.\ \ cases}{N}$ to 1 and is interpreted as the percentage of positive cases correctly predicted by the model
- ROCR will draw PR Cuves too.

Summing Up CART

- CART grows a single tree to the data
- Variables are selected to partition the data by the algorithm in order to minimize misclassification error
- Optimal complexity of the tree is determined by cross-validation
- Accuracy of CART is assessed in out-of-sample data by various metrics

Comparing CART to Logistic Regression for Predicting Genocide

- Let's construct a logistic regression classifier in caret
- Syntax is still the same as a normal glm
- Is it more accurate than CART?

```
TMK.logit<-train(as.factor(genpol.onset)~.,
                 method="glm",
                 trControl=fitControl,
                 family="binomial",
                 metric="ROC",
                 data=training[c(-1)])
```

Comparing CART to Logistic Regression for Predicting Genocide

- AUC-ROC$_{Tree}$ = 0.866, AUC-ROC$_{logit}$ = .916 - all on testing data
- In this case, we would prefer the logistic regression over the tree. AUC is higher.

Introducing Ensembles of Trees: Random Forests

From CART to Random Forests

- Because of the way CART partitions data, it is an exceptionally low-bias algorithm: it tends to do well in most tasks.
- Due to the ability of the three to capture nonlinearities and interactions between variables
- Imagine now that the tree selected some other variable for inducing a split on the data
- If another variable was chosen to induce a split, the interpretation of the tree could easily be different.
- In high-dimensional data, there's lots of variables to choose from

From CART to Random Forests

- Because the structure of a tree depends heavily on the variables selected, trees are high-variance (noisy) predictors.
- Random Forests utilizes the inherent variance of trees to reduce model variance *even further*.
- Random Forests does this by growing an **ensemble** of N decision (regression) trees to the data instead of just one tree.
- The process by which Random Forests grows this ensemble of trees is known as **bagging**, short for bootstrap aggregation.

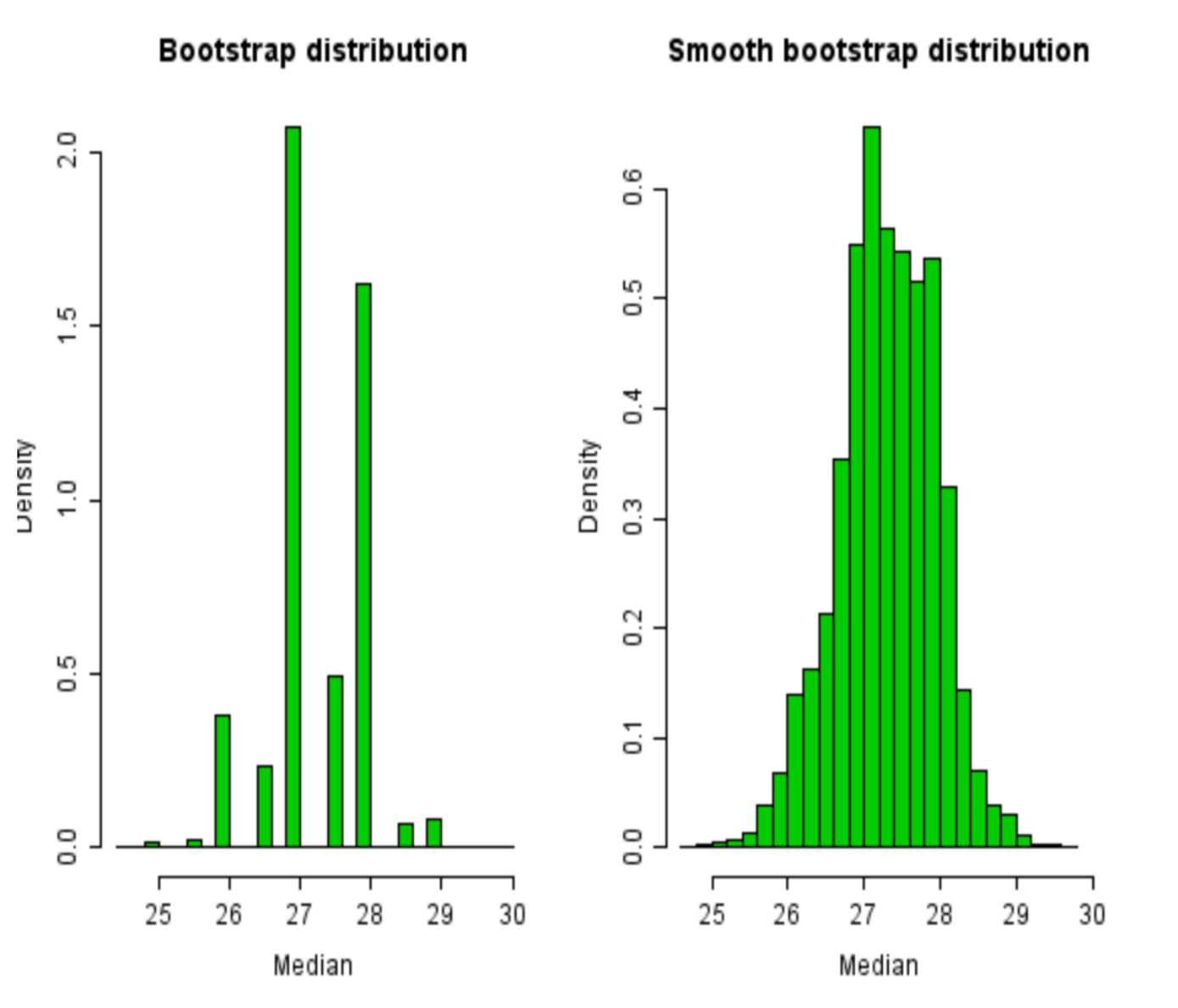
Introducing the Bootstrap

- The **bootstrap** is a method for estimating the properties of a statistical estimator (i.e. its predictive accuracy) by measuring those properties when sampling for an approximating distribution with replacement.
- One could bootstrap confidence intervals for a typical regression model by bootstrapping the model using repeated resamples of individual observations. Especially helpful in small samples, and in Bayesian statistical models
- The bootsrap is useful for estimating the distribution of a statistic since it does not rely on normality (i.e. z-scores, t-scores) to produce the distribution.

Introducing the Bootstrap

- Assume a simple coin flipping experiment.
- Assume further, we have forgotten the binomial theorem and need to estimate the probability of observing heads based on 10 tries
- Let the t-score of the mean equal $\bar{x} = \frac{1}{10}(x_1 + x_2 + \cdots + x_{10})$
- Instead, taking the bootstrap, we same (with replacement) from the data.
- $X_1^* = (x_2, x_1, x_5, x_8, x_2, x_4, x_9, x_3, x_6, x_{10})$
- Then take $\mu_{X_1^*}$ to compute the bootstrap estimated sample mean
- Repeat this process N times to repeatedly resample the sample mean from the data to arrive at a distribution of population mean.

Introducing the Bootstrap

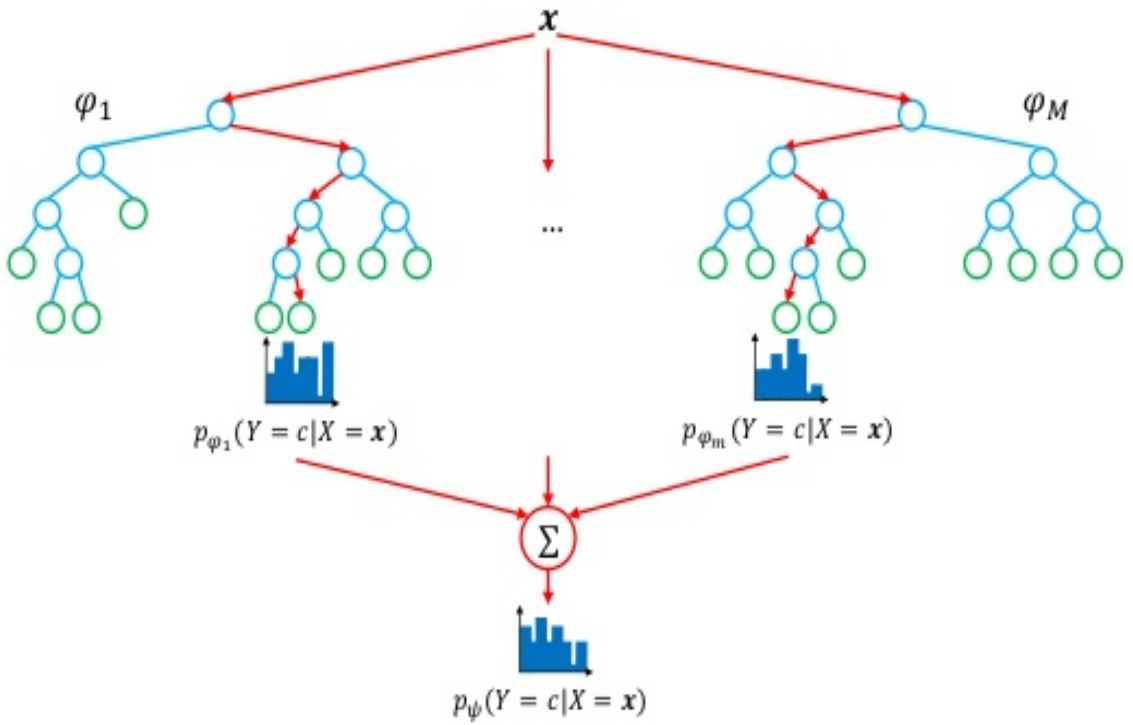


Bagging for Random Forests

- Bagging works by selecting a random sample of observations $i_1, i_2 \cdots, i_n$ and features $X_1, X_2, \cdots, X_n$
- The algorithm then grows a tree to the bootstrap selected observations and features.
- N trees are grown in this fashion, where N is a hyper-parameter set in advance. Trees are i.i.d. and uncorrelated.
- Then, observations **NOT** used to grow a tree are dropped down each tree and classified as they would be in CART (i.e. according to the partitions in the data space).
- These **out of bag** OOB observations are used to assess classification accuracy according to majority vote across all trees in the forest (classification) or simple arithmetic average (regression)

Random Forests Visualized

Random forests



Randomization

- Bootstrap samples
- Random selection of $K \leqslant p$ split variables
- Random selection of the threshold

} Random Forests

} Extra-Trees

14 / 39

Random Forests Algorithm

Algorithm 15.1 *Random Forest for Regression or Classification.*

1. For $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. Select m variables at random from the p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

Random Forest: Prediction

- Random Forests is one of the most consistently powerful machine learning models outside of deep-learning.
- Random Forests are particularly well suited to the analysis of comparative historical data common in CP and IR
    - Benefits most when data are highly non-linear and contain substantial interactions
- Two main prediction metrics: accuracy and the Gini Coefficient (node impurity, misclassification error)

Random Forests: Prediction

- Prediction is performed internally by Random Forest
- Because Random Forests grows a forest of uncorrelated decision (regression) trees, and uses OOB observations to generated predicted probabilities, both cross-validation and prediction are performed internally (on the training data)
- Out-of-sample predictions are performed as with CART (i.e. passing predictions in training data to new data)

Random Forest: Prediction

- Variable Importance and Partial Dependence Plots
- Variable importance is calculated according to the Gini Coefficient (traditionally)
- Gini is a measure of node impurity at end of forest growing procedure. Since classification is determined by majority vote among trees, obs. can be misclassified.
- Gini Coefficient ranges from [0,1] where 0 equals completely homogeneous and 1 a completely heterogeneous node. For this metric, lower Gini is better (i.e. more accurate).

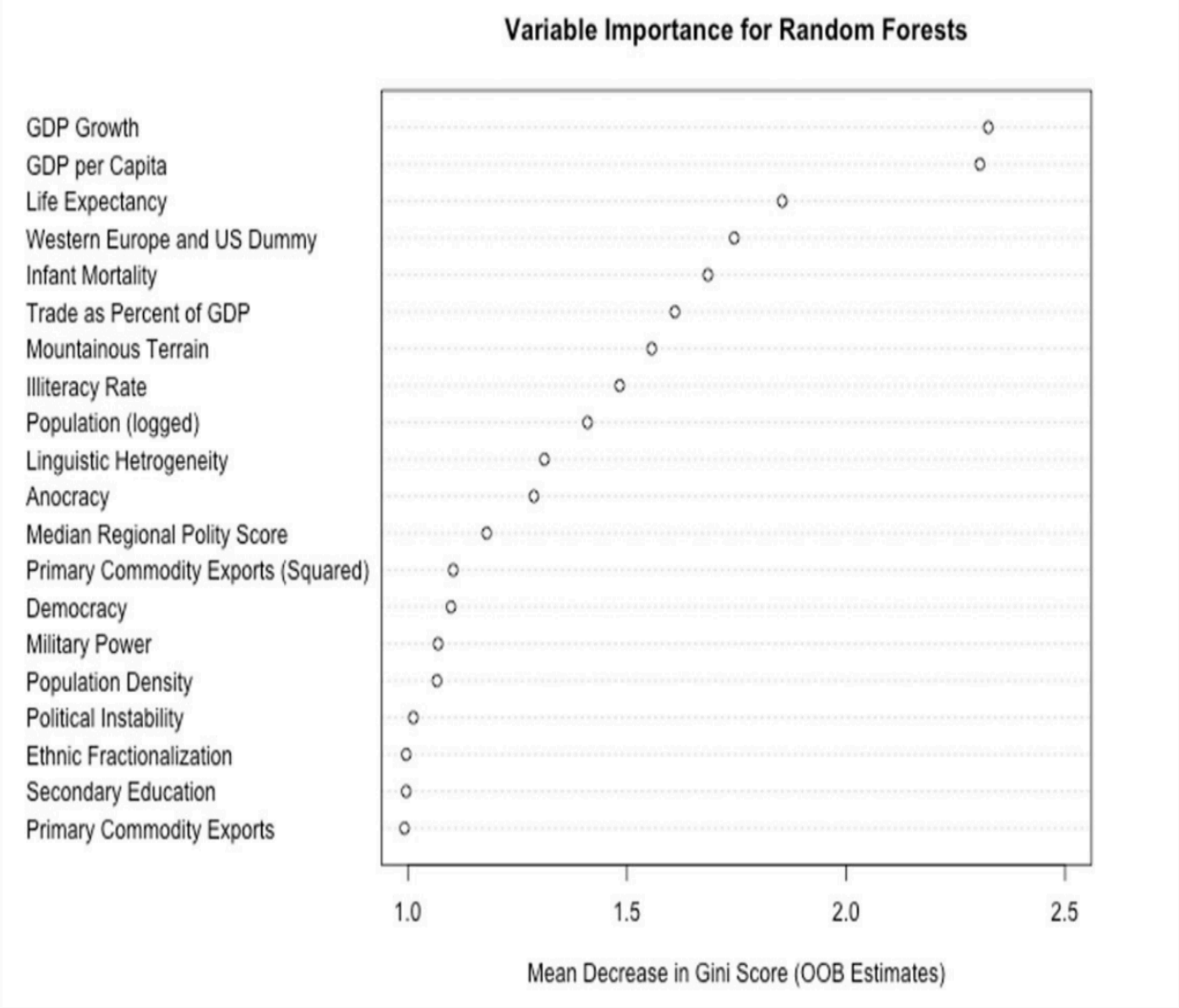
Random Forest: Prediction

- For Regression:
- Accuracy is measured by $\hat{y} = y$ (traditionally RMSE)
- For regression, the average $\hat{y}_i$ is taken for each y_i and used as the prediction of y_i.

Random Forests: Inference

- For inference, Random Forests calculates the mean decrease in Gini for each feacture X_i for all trees in which X_i was **NOT** selected to grow a classification tree.
- Measure the decrease in predictive accuracy when X_i is *ablated* from the overall model. If X_i is an important predictor, it will have a larger mean decrease in Gini score if removed from the model (since lower Gini is more accurate)
- Compute mean decrease in Gini for all features across all trees the the forest. Then plot the results.

Random Forests: Inference



Random Forests: Inference

- Partial Dependence Plots
- PD Plots show the estimated change in the fraction of votes among trees for class 0 or class 1 across the entire forest, across the range of each feature.
- Allows for a visual inspect of non-linearities, threshold effects, and other data characteristics which can aid in inferential analysis from Random Forest
- Keep in mind, PD plots are for the training data only.

Random Forests: Inference

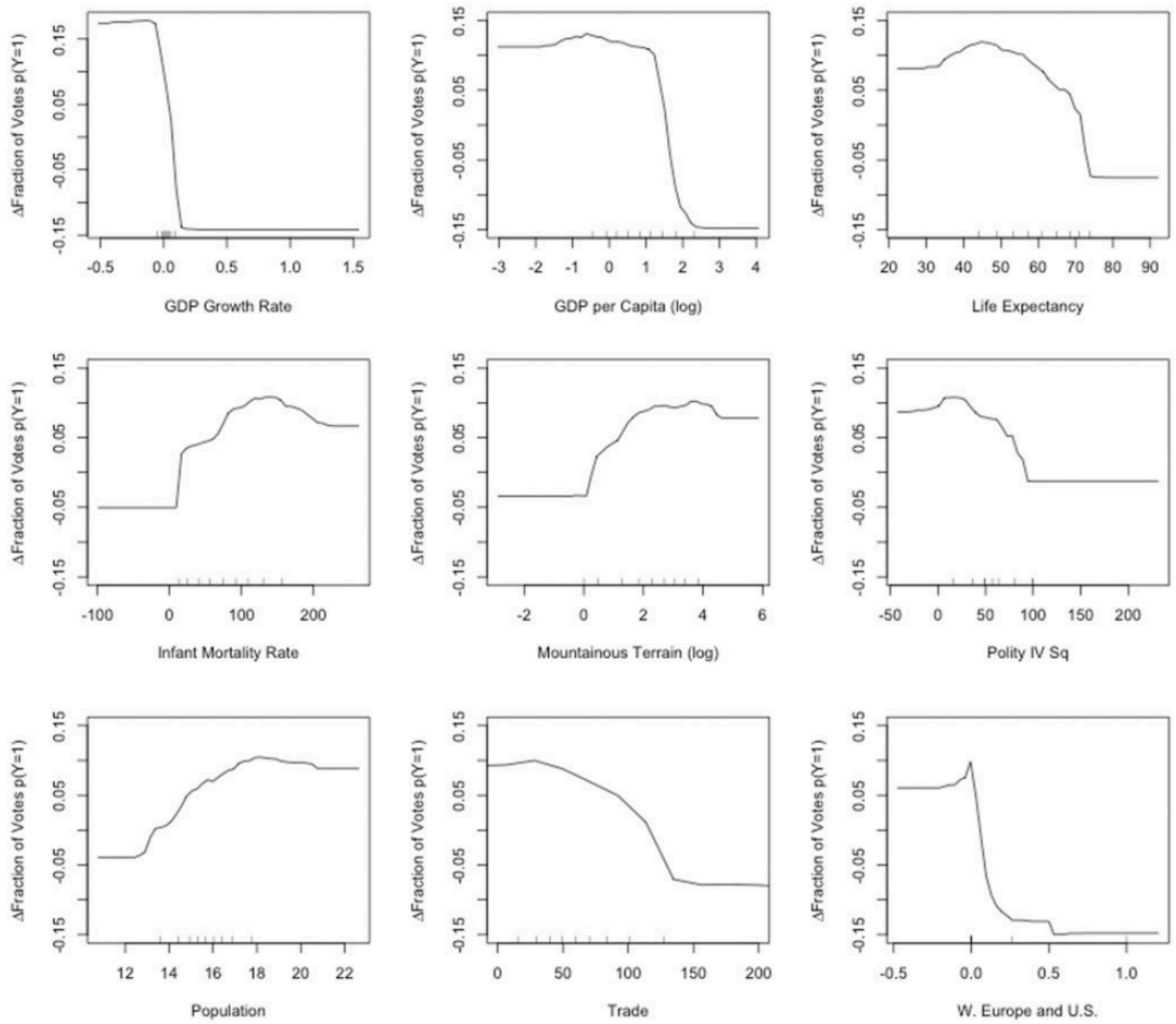


Fig. 5 Partial dependence plots.

Application: Predicting Civil War Onset

Predicting Civil War Onset

- Go to and download the R code for this walk-through
- Also download the two datasets we will be using: SambanisImp and data full
- Both should be saved in .csv format
- We will walk through this code together to predict civil war onset using logistic regression and Random Forests.
- We will produce various plots which can be used to calibrate the differences between models.

Application: Predicting Genocide

A Less-Guided Tutorial

- Using the`caret` library, I want you to implement the following models to predict civil war onset using the same data
  - The Elastic Net
  - K-nearest neighbors (k=5, 10, 15, 20)
  - Boosted decision trees
  - Random Forests
  - A single-layer feed-forward neural network
- Using ROC plots and Confusion Matrices, determine which model is the most accurate in predicting onsets of civil war in the test dataset
- Use the caret githib website `http://topepo.github.io/caret/index.html` to find instructions on hyperparameter tuning and how to run the various models.