



Norwegian School of Economics

Bergen, Fall 2019



Machine Learning for Resource Economics

A review of modern computational statistics with empirical applications in fisheries management

Ryan Jeffrey Potter

Supervisor: Lassi Ahlvik

Masters Thesis, Economics

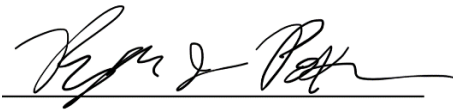
NORWEGIAN SCHOOL OF ECONOMICS

This thesis was written as a part of the Master of Science in Economics and Business Administration at NHH. Please note that neither the institution nor the examiners are responsible – through the approval of this thesis – for the theories and methods used, or results and conclusions drawn in this work.

Acknowledgements

I would like to express my gratitude to my supervisor, Assistant Professor Lassi Ahlvik, for his feedback and guidance through the process of writing this thesis. His generosity and expertise were invaluable over the semester. I would also like to thank all my professors from my time in Bergen for the tools and inspiration needed to write this thesis, and the university, NHH, for the resources it provided throughout the process.

NHH Norwegian School of Economics
December 2019

A handwritten signature in black ink, appearing to read 'Ryan J. Potter', written over a horizontal line.

Ryan Jeffrey Potter

Abstract

Machine learning increasingly permeates our everyday lives, from artificial intelligence suggesting how we complete a text message to big data selecting creepily relevant ads to show us as we browse the web. While science and technology researchers have pushed these methods forward and private companies have embraced their power in significant changes to their processes, the field of economics has largely watched them go by. Despite the credibility revolution and increased focus on estimating causal effects, the econometric techniques in use today are largely identical to the ones used three decades ago.

This thesis contributes to the growing field of literature at the intercept of machine learning and economics by exploring whether modern computational statistics methods can provide practical value to resource economists. I answer the following research question:

Can integrating machine learning methods into econometric models improve upon traditional methods and add value in solving resource economics problems?

To answer this question, I review the machine learning literature on causal analysis to find that machine learning methods solve certain types of problems in unique ways that traditional methods cannot. To test the benefit of these new methods in a resource economics setting, I apply machine learning to a fisheries problem based on the Costello, Gaines, & Lynham (2008a) article, *Can Catch Shares Prevent Fisheries Collapse?*, and analyse performance in a first-stage estimation task for propensity score matching.

The results show machine learning can improve performance for prediction-based econometrics tasks under certain conditions. Shrinkage-based methods like Lasso regression proved to substantially improve model fit for datasets with moderate variance, while performing in-line with traditional methods when this condition didn't hold. While more flexible methods like Random Forest performed extremely well fitting the data, they captured significant levels of noise by overfitting, challenging the external validity of their predictions.

Machine learning identified and modelled valid selection bias that traditional methods could not – demonstrating value in solving practical resource economics problems. The impact of first-stage overfitting on the final causal model was unclear and presents an important area for further research, but the overall findings support the application of machine learning methods for robustness analysis on prediction tasks in resource economics.

Contents

ACKNOWLEDGEMENTS.....	2
ABSTRACT.....	3
CONTENTS.....	4
ABBREVIATIONS	9
CHAPTER 1 BACKGROUND	10
MACHINE LEARNING DEFINED	10
SUPERVISED VS UNSUPERVISED MACHINE LEARNING.....	11
<i>Unsupervised Learning</i>	11
<i>Supervised Learning</i>	11
MACHINE LEARNING VS ECONOMETRICS	12
CHAPTER 2 SELECTED MACHINE LEARNING METHODS.....	14
ML FUNDAMENTALS	14
<i>Overfitting and Tuning Parameters</i>	15
<i>Resampling</i>	15
LINEAR ML ALGORITHMS	17
<i>Shrinkage Methods</i>	17
TREE-BASED ML ALGORITHMS.....	20
<i>Classification and Regression Tree (CART)</i>	20
ENSEMBLE METHODS	24
<i>Random Forest (RF)</i>	25
SUMMARY: ML METHOD COMPARISON	28
CHAPTER 3 ECONOMETRIC APPLICATIONS FOR MACHINE LEARNING	30
STATE OF MACHINE LEARNING IN ECONOMETRICS	30
ML FOR PREDICTION PROBLEMS (y)	31

<i>Treatment Effects under Potential Outcomes Framework</i>	31
<i>Synthetic Control</i>	32
ML FOR PARAMETER ESTIMATION PROBLEMS (β)	32
<i>Variable Selection</i>	33
<i>Instrumental Variables</i>	33
<i>Propensity Score Matching</i>	34
ML FOR RESOURCE ECONOMICS	34
CHAPTER 4 EMPIRICAL APPLICATION: FISHERIES	36
BACKGROUND	36
DATA.....	37
<i>Costello et al. Dataset</i>	37
<i>Recreated Dataset</i>	37
ECONOMETRIC METHODS BACKGROUND	39
<i>Difference-in-Differences</i>	39
<i>Logit Regression</i>	39
<i>Propensity Score Matching</i>	40
COSTELLO ET AL. MODELS	41
<i>First-Cut Model Specification</i>	41
<i>Base Model Specification</i>	42
<i>Advanced Model Specifications</i>	42
CHAPTER 5 EMPIRICAL ML TEST: PROPENSITY SCORE MATCHING	45
BACKGROUND	45
THE TEST	46
<i>Validation Method</i>	46

<i>Model-Fitting Procedure</i>	47
<i>Performance Evaluation</i>	48
TEST DATASET 1 (GLOBAL FISHERIES)	49
<i>Data</i>	49
<i>Model 1</i>	50
<i>Model 2</i>	53
TEST DATASET 2 (OECD FISHERIES)	55
<i>Data</i>	56
<i>Model 3</i>	57
ML IMPACT ON FINAL PARAMETER ESTIMATORS	61
<i>Overfitting Estimator and Causal Results</i>	61
<i>Improved Predictions and Causal Results</i>	63
<i>A Note on Causal Inference</i>	65
DISCUSSION	66
CHAPTER 6 CONCLUSION	68
REFERENCES	71
SOFTWARE PACKAGES USED:	77
LIST OF TABLES AND FIGURES	78
TABLES:	78
FIGURES:	79
APPENDIX 1: REPLICATION RESULTS	80
FIRST-CUT MODEL RESULTS	80
BASE MODEL RESULTS	80
PROPENSITY SCORE MODEL RESULTS	81
FIXED EFFECTS MODEL RESULTS	82

APPENDIX 2: P-SCORE APPLICATION REGRESSION RESULTS	83
MODEL 1 P-SCORES – CAUSAL REGRESSION RESULTS.....	83
<i>Propensity Score Base Model Results:</i>	<i>83</i>
<i>Propensity Score Fixed Effects Model Results</i>	<i>84</i>
MODEL 2 P-SCORES – CAUSAL REGRESSION RESULTS.....	85
<i>Propensity Score Base Model Results:</i>	<i>85</i>
<i>Propensity Score Fixed Effects Model Results</i>	<i>86</i>
MODEL 3 P-SCORES – CAUSAL REGRESSION RESULTS.....	88
<i>Propensity Score Base Model Results:</i>	<i>88</i>
<i>Propensity Score Fixed Effects Model Results</i>	<i>89</i>
APPENDIX 3: SUPPLEMENTARY BACKGROUND & ANALYSIS	91
BACKGROUND: POTENTIAL OUTCOMES (PO) FRAMEWORK	91
<i>Model Setup.....</i>	<i>91</i>
<i>Causal Effect</i>	<i>92</i>
<i>Assumptions.....</i>	<i>92</i>
<i>Randomization Levels.....</i>	<i>93</i>
BACKGROUND: SYNTHETIC CONTROL (SC) METHOD	95
<i>Set-up.....</i>	<i>95</i>
<i>Assumptions.....</i>	<i>95</i>
<i>Evaluating Results</i>	<i>97</i>
ANALYSIS: CAUSAL IDENTIFICATION ISSUES	98
<i>Issue 1: Bias from Logit Transformation.....</i>	<i>98</i>
<i>Issue 2: Propensity Score Problems</i>	<i>99</i>
APPENDIX 4: SUPPLEMENTARY TABLES & FIGURES	102
VALIDATION CURVES.....	102

<i>Model 1</i>	102
<i>Model 2</i>	103
<i>Model 3</i>	104
MODEL 3 COVARIATE DETAILS	105

Abbreviations

<i>Abbreviation</i>	<i>Description</i>
<i>ML</i>	Machine learning
<i>CV</i>	Cross-validation (machine learning process)
<i>MSE</i>	Mean-squared error (regression metric)
<i>OLS</i>	Ordinary least-squares regression (econometric method)
<i>CART</i>	Classification and Regression Trees (machine learning algorithm)
<i>RF</i>	Random Forest (machine learning algorithm)
<i>RSS</i>	Residual sum-of-squares (regression metric)
<i>PO</i>	Potential Outcomes (causal framework)
<i>SC</i>	Synthetic Control (econometric method)
<i>IV</i>	Instrumental variables (econometric method)
<i>2SLS</i>	Two-stage least-squares IV procedure (econometric method)
<i>PSM</i>	Propensity score matching (econometric method)
<i>P-score</i>	Propensity score (econometric metric)
<i>SUTVA</i>	Stable Unit Treatment Value Assumption (data assumption)
<i>DiD</i>	Difference-in-differences (econometric method)
<i>ITQ</i>	Individual Transferable Quota (fisheries regulation system)
<i>LME</i>	Large Marine Ecosystem (fisheries geographic classification)
<i>MLE</i>	Maximum Likelihood Estimation (econometric method)
<i>FB</i>	Fishbase (database)
<i>MHS</i>	Maddison Historical Statistics (database)
<i>SAU</i>	SeaAroundUs project (database)

Chapter 1 Background

This chapter provides a high-level introduction to machine learning and contextualizes it from an econometric perspective. I provide a formal definition for machine learning and an overview of the types of tasks it is designed for. Then I compare its goals, methods and strengths versus the traditional econometric tools used in practice today.

Machine Learning Defined

Machine Learning is a vast and rapidly growing field with an active body of literature coming out of academia and private companies such as Google and Microsoft Research. While initially developed in computer science departments, it has seeped into statistics and now touches most fields from social sciences to medicine. Over the past several decades increases in computational power and falling digital storage costs have contributed to a shift in computational statistics, sometimes dubbed the “data revolution,” (Einav & Levin, 2013).

Such a bold noun hints at the youth of the field which, along with its interdisciplinary nature, has resulted in a tangle of names used to describe its techniques: big data, artificial intelligence, data science, deep learning, neural networks, etc. There is significant overlap between all these classifications so the terminology disputes common to the field tend towards fruitless exercises in splitting hairs.

For the purpose of this thesis I will abstract from such semantic discussion and stick to machine learning as a broad label encapsulating all the recent data-driven statistical techniques. To formalize this in a single definition, machine learning (ML) is:

A set of techniques in which algorithms are applied to datasets to construct models, taking the data itself as an input determining model design. It is generally employed with the goal of prediction and characterized by tuning parameters fitted using the data in an iterative feedback process such as cross-validation.

This approach is at odds with traditional econometrics, where the model specification process and the data must be strictly divided - an expert pre-specifies the model design using their knowledge and intuition. This sharp methodological difference has advantages and disadvantages when it comes to the economist’s primary goal of causal analysis.

Supervised vs Unsupervised Machine Learning

While the primary definition of machine learning is accurate, it also is necessarily ambiguous due to the breadth of techniques in the field. At its core ML is a set of algorithms each designed for specific tasks, so field is commonly divided into two branches based on a classification of these tasks: supervised and unsupervised.

Unsupervised Learning

Unsupervised learning is used for grouping or clustering observations by splitting them into subgroups based on the similarity of their covariates. This sort of method takes *unlabelled* data - that is, data without a specified dependent variable - so its results are not testable for predictive accuracy (hence unsupervised). Unsupervised algorithms are most naturally applicable in pre-processing tasks called dimensionality reduction, where data that has many covariates is distilled into to a smaller set of new covariates that contain roughly the same information. Then traditional or supervised learning statistical methods can be applied to the new dataset. Some specific algorithms that fall in this category are k-means clustering, principal components analysis (PCA) and latent dirichlet allocation (LDA) models.

In a survey of ML applications for economics, Athey (2018a) finds unsupervised methods most useful as an intermediate step in empirical work as a data-driven way to create new variables. For example, given a textual product description, clustering algorithms could find and file products into subgroups based on similarity. Or taking Yelp data, an unsupervised algorithm could categorize the reviews into types (Athey, 2018a). While these are powerful tools for creating new and innovative datasets, this paper will focus on the other, supervised side of ML, which more closely parallels traditional econometric methods.

Supervised Learning

Supervised learning takes data with a label - or specified dependent variable - and uses data-driven models to find relationships between the label and covariates. Typical applications include prediction of continuous dependent variables (regression) and classification of categorical dependent variables. However, the methods are flexible and can also be used for dimensionality reduction and pre-processing work. The numerous supervised algorithms differ in their flexibility when fitting data, from simple linear specifications to highly non-linear and nonparametric models. Due to their development in computer science programs, these

methods tend to be computationally efficient and scalable to large sample sizes and many covariates (high-dimensional data). This thesis focuses only on supervised learning methods as this is the more naturally adapted branch. Specific supervised algorithms are discussed in Chapter 2 and then applied to a fisheries management problem in Chapter 5.

Machine Learning vs Econometrics

ML and econometrics have similarities from their shared roots in statistics, but they diverge in terms of goals, priorities and methods. The result is two sets of tools with distinct advantages and weaknesses, and potential to complement each other if combined wisely.

Since what Angrist and Pischke call the “credibility revolution in empirical economics,” econometric methods have been designed for causal inference, focusing on efficiency with relatively small data sets and a limited set of hand-selected covariates (2010). Ordinary least-squares (OLS) is such a popular method because it is easily interpretable, provides measures of marginal effects and has optimal asymptotic properties allowing easily calculated inference statistics. This is possible due to significant assumptions on the data structure and relationships which must be defended. Models are constrained in their flexibility by a requirement to be linear in parameters and the need to pre-specify relationships based on expert’s intuition rather than the data. For most economic applications this system has served well.

Machine learning was developed to solve these limitations of traditional statistical methods, sometimes at the cost of interpretability and statistical inference. The primary goal of machine learning is accurate predictions, so the methods are designed to have maximum flexibility lest any signal is missed in the fitting process. The result is models that can capture complex interrelationships in highly non-linear ways, with the data determining model design rather than any single person’s intuition. This flexibility necessitated new ways to test models, based less on asymptotic properties and more on proving performance on new data. The result was a set of computationally efficient methods that flexibly fit any kind of data – including high-dimensional and large datasets – with little loss in performance.

The below table provides a high-level comparison of the two fields:

Table 1: Comparison: Econometrics vs Machine Learning

	Econometrics	Machine Learning
Goals	Inference/causal analysis	Prediction
Strengths	Designed for causal inference (<i>interpretable, asymptotic properties, etc.</i>) Efficient with small data, low-dimensional data	Flexibility in fitting data (<i>capturing complex relationships, interactions, etc.</i>) Prediction accuracy
Weaknesses	Strong assumptions (<i>e.g. parametric form</i>) Limited flexibility in fitting data (<i>may not capture all the signal</i>)	Not designed for causal inference (<i>e.g. interpretability/black box challenges, lack of valid marginal effects/inference stats</i>) Too flexible in fitting data sometimes (<i>may capture too much noise, overfitting risk</i>)
External Validation Method	Inference statistics (<i>e.g. standard error, based on parametric assumptions</i>)	Validation sets (<i>e.g. cross-validation, sample-splitting</i>)
Flexibility	Moderate (<i>constrained to linear-in-parameters; some less-common exceptions – e.g. kernels, splines, sieves</i>)	High (<i>highly nonlinear and nonparametric modelling options</i>)
High-dimensional data capability	Medium (<i>computational limitations, least-squares “curse of dimensionality”</i>)	High (<i>many algorithms designed to deal with large number of covariates</i>)
Model design	Manual specification from intuition, ad-hoc experimentation (<i>e.g. if testing heterogeneity, must self-select groups in pre-analysis</i>)	Adaptive – model uses data itself to choose specification (“learns”) (<i>e.g. algo selects groups with heterogenous relationships using data</i>)
Honesty/Replication	Opacity in model construction process (<i>e.g. p-value hacking, multiple-hypothesis testing issues w/o validation process</i>)	Data-driven fitting process (<i>Systematic w/ validation process to prove generalizability</i>)

Chapter 2 Selected Machine Learning Methods

This chapter provides brief background on the most important concepts, methods and algorithms in Machine Learning. It covers the foundational off-the-shelf ML methods which are extensively covered in popular introductory ML textbooks (Hastie, Tibshirani, & Friedman, 2009; James, Witten, Hastie, & Tibshirani, 2013; Murphy, 2012; Efron, & Hastie, 2016; Géron, 2019). For deeper understanding beyond the scope of this thesis I refer the reader to these resources.

ML Fundamentals

All ML methods discussed in this thesis can be described by four components: a predictor model, loss function, tuning parameter and cross validation process. These parts are combined into an algorithm – the program that iterates over models and data until an optimized specification is discovered, constructed and output.

1) Predictor model

The predictor model is an algorithm that controls the underlying structure of the fitted model. It can be as simple as a linear function (e.g. Lasso regression) or more complex like a decision tree (e.g. CART). *Ensemble methods* extend these to combine multiple models into a single predictor. The best algorithm to use depends on data characteristics, output goals and preferences on bias-variance tradeoff.

2) Loss function

The loss-function acts as a measure of accuracy for the predictor model and is used in the fitting process to construct models. Common loss functions include mean squared error (MSE) for regression and cross-entropy for classification problems.

3) Tuning parameter

Tuning parameters control the flexibility of model fitting. They can be used to adjust penalty terms in the loss-function or as a constraint on some model feature. Typically cross validation is used to select an optimal value.

4) Cross validation (CV) process

Cross validation is a process that splits a dataset into a *training* set used to fit a model and a *test* set used to evaluate the model. It is standard process in ML to use CV to identify optimal tuning parameters and to measure external validity of the model. This

setup is a critical difference from econometrics as it allows ML algorithms to learn from the data without biasing performance (more on this in Resampling section below).

Overfitting and Tuning Parameters

ML algorithms' edge in predictive accuracy comes from their flexibility and ability to test many specifications on real data while fitting a model (i.e. use the data to decide model form). If left unchecked this feedback process tends to result in ungeneralizable models – models that do not perform well on new data (data unseen in the fitting process). This problem is called *overfitting* in ML literature and it occurs when the algorithm's learning process creates a model that internalizes both the signal and noise from the training data (*overtraining* is another term for this issue which makes the cause more explicit).

Because of this built-in tendency to overlearn from (overtrain on) the data, ML methods are characterized by adjustable *tuning parameters*. By limiting flexibility in the fitting processes, these regulate the complexity of the output model and implicitly control how much information (signal and noise) is captured in training. Tuning parameters are generally implemented as a penalty for complexity in the model's optimization criteria or constraints on output model form (e.g. number of levels on decision tree).

Tuning parameter values are a critical choice in constructing a good model that properly balances fit internally (prediction on the training data) and external validity (generalizability, prediction on new data). Because the best tuning parameter value is unique for each dataset, ML has developed ways to search for optimized values called cross-validation.

Resampling

Resampling methods are commonly used in training a model (*model selection*) and evaluating performance (*model assessment*). James et al. describe it as “drawing samples from a training set and refitting a model of interest on each sample in order to obtain additional information about the fitted model,” (2013). Practical implementations can be categorized as cross-validation and bootstrapping.

Cross-Validation

As discussed, overfitting is a primary concern in ML since the data is used as a direct input in model design. Cross validation (CV) aims to mitigate this issue by holding out observations

during the training step so that a valid test error can be calculated on data that was unseen during fitting. There are three common methods by which this sampling is applied:

Validation set method – The simplest form of CV entails randomly splitting the data into 2 sets (training set and test set). The model is fitted on the training set, then deployed to make predictions on the test set which are used to calculate the test-set error rate (often mean-squared error, MSE). While this is a simple method to implement, results can suffer from high variability (since it relies on a single sample) and test error may be overestimated since only a limited portion of the data (20-30%) is used in the calculation.

Leave One-Out method (LOOCV) – This method involves training/testing the model N times (where N is the number of observations), holding out a single observation as the test set each time. The resultant N approximately unbiased test error results are then averaged to create the overall test error estimate. While this method mitigates the main drawbacks of the simple validation set method (upward biased error, high sensitivity to sample), it is computationally intensive to retrain models N times.

k-Fold method – This method involves randomly¹ dividing the dataset into k different groups (or *folds*) of observations, then training/testing the model k times, each time holding out a different fold as the test set. The overall test error estimate is taken as the average error over all the runs. This is a compromise between the first two methods that balances robustness with computational feasibility.

All methods discussed above can be used in the *model selection* process of choosing tuning parameters. In practice they are deployed to calculate test errors in a *grid-search*, where the model is retrained across a set of different tuning parameter values. The parameter with the lowest test-error is taken as the optimal model specification. The term cross-validation in ML generally refers to this entire process (splitting, predicting, testing).

¹ *A note on random sampling for CV* – In validation-set and k-fold methods, it is critical for sampling to be random so that both test set and training set are representative of the full dataset. When there is some imbalance in the dataset – for example, if one group in a classification problem is very rare - a further *stratification* condition is often placed on the sampling so that the resultant test/training sets are comparable.

For testing the model’s external validity – the *model assessment* process – a holdout set is usually removed from the dataset before CV takes place. While this step is identical to the validation set method, ML literature generally reserves the term cross-validation for sample splitting and evaluation that occurs within the parameter-tuning context.

Bootstrap

Bootstrapping is a resampling method that differs from CV in that random samples are made *with replacement*. This results in many samples that are representative of - but not identical to - the original dataset, and thus can be used for evaluating the model. This method is particularly useful for estimating inference parameters when the model is non-linear and thereby standard errors are mathematically difficult to attain.

Certain ML algorithms also employ bootstrapping in their model training stages. For example, bagging (bootstrap aggregation), which is employed in the Random Forest algorithm, uses bootstrap samples to train many separate trees (see further discussion in Ensemble Methods section).

Linear ML Algorithms

Linear regularization algorithms are among the most popular ML methods, benefiting from relative simplicity and a functional form shared with the ubiquitous ordinary-least-squares (OLS) regression. In this section I focus on what Hastie, Tibshirani, and Friedman (2009) refer to as *shrinkage* methods: Lasso, Ridge and Elastic-Net regressions. These provide a good intro to ML and the power of tuning parameters. Other useful linear methods beyond the scope of this thesis include subset-selection and support vector machines (SVMs).

Shrinkage Methods

Shrinkage methods start with a traditional OLS model, then add a tuning-parameter-based regularization term to the optimization criteria that penalizes larger coefficients and shrinks their values towards 0. This tuning parameter is akin to a control knob for bias-variance tradeoff – higher values reduce variance at the cost of some bias in the coefficients. Regularization of this form results in “substantial reduction in the variance of the predictions, at the expense of a slight increase in bias,” usually a worthwhile tradeoff given that

performance metrics are a function of both bias and variance (James, Witten, Hastie, & Tibshirani, 2013).

By including a tuning parameter in the regularization term, shrinkage methods introduce flexibility in the model fitting and allow data to be used directly in the choosing the optimization criteria. This alternative fitting procedure can offer better predictive accuracy, particularly in the case of high-dimensional data (when number of covariates, p , is high relative to sample size, n). Under an assumption of sparsity (i.e. there are more covariates present than are relevant to the outcome variable), shrinkage algorithms can automate the *variable selection* process by removing the weak covariates from the model.

Ridge Regression

Ridge regression uses the least-squares model as a base but adds the ℓ_2 norm as a penalty term in the optimization criteria. This term serves to shrink the coefficient values towards 0 across the board. The fitting process minimizes²:

$$RSS + \lambda \sum_{j=1}^p \beta_j^2$$

A regularization parameter, λ , controls the weight given to the shrinkage criteria and is treated as a tuning parameter. Tuning is done by cross-validation to find an optimal regularization level that minimizes a scoring metric (usually MSE). As λ increases the coefficients will shift further towards 0 across the board, while $\lambda = 0$ results in a model with unconstrained coefficients identical to the OLS model.

Lasso Regression

Lasso regression uses the ℓ_1 norm as a penalty term in the optimization criteria, which will decrease as the coefficient values shrink towards 0. The fitting process minimizes:

$$RSS + \lambda \sum_{j=1}^p |\beta_j|$$

² Note: all shrinkage methods can be applied with a different base optimization criterion than least-squares – for example, to apply shrinkage to a maximum-likelihood estimator the penalty term is added to negative log-loss rather than RSS.

The regularization term in this model also shrinks coefficients, but unlike the Ridge criteria it will shrink some coefficients all the way down to 0 when λ is large enough. The resultant output is a *sparse model* which includes only a subset of the original covariates; a higher λ value shrink coefficients further towards 0 and increase sparsity of the output model. Because of this characteristic the Lasso regression can be employed as an automated means of *variable selection* to remove less useful variables and create more interpretable models. Like Ridge regression, $\lambda = 0$ results in a model identical to OLS.

Elastic-Net Regression

The Elastic-Net model includes penalty terms for both the ℓ_1 and ℓ_2 norms, so the fitting process minimizes:

$$RSS + \lambda_1 \sum_{j=1}^p |\beta_j| + \lambda_2 \sum_{j=1}^p \beta_j^2$$

The penalty terms are interpreted the same as Lasso and Ridge respectively. Depending on the two tuning parameters the model will be a combination of Ridge/Lasso ($\lambda_1 > 0$, $\lambda_2 > 0$) or one of three special cases: OLS ($\lambda_1 = 0$, $\lambda_2 = 0$), Ridge ($\lambda_1 = 0$, $\lambda_2 > 0$), or Lasso ($\lambda_1 > 0$, $\lambda_2 = 0$). This algorithm benefits by offering potentially sparse solutions using the ℓ_1 term, while the ℓ_2 term increases robustness in the case of highly-multicollinear variables.

Shrinkage Implementation Notes

To decide between Ridge, Lasso and Elastic-Net models, the literature recommends a heuristic approach to fit each using cross-validation and compare the results (James, Witten, Hastie, & Tibshirani, 2013). Ridge can perform better when many covariates are relevant and have smaller individual impacts, or when high multicollinearity in covariates is a concern. Lasso performs well when there are a small number of covariates with large impacts or when variable selection is a goal. Elastic-Net offers a balance between the two.

Since shrinkage penalizes coefficients for their size, the dataset should be *standardized* or *normalized* before implementing the regressions so that each covariate is on a common scale. Standardization involves transforming each covariate so that its mean is 0 and standard deviation is 1. Normalization rescales each covariate into a range of [0,1]³. Standardization is

³ Common practice is to also re-center normalized covariates to [-0.5,0.5] to help with convergence/optimization

more robust to outliers but makes implicit parametric assumptions on each covariate that are not always justifiable.

Tree-Based ML Algorithms

One benefit of ML methods is their ability to fit complex relationships in a non-parametric fashion. Regression trees offer a highly non-linear modeling strategy that exemplifies this flexibility gain when functional form assumptions are relaxed.

Classification and Regression Tree (CART)

Classification and Regression Tree (CART) is a non-parametric ML method that can be used to predict discrete values (classification) or continuous values (regression). Initially developed by Breiman et al. (1984), it is based on the hierarchical structure of a *decision tree* which divides the data into subsets by partitioning the covariate space. After creating these subsets – labelled as a leaf, l_m – an estimator is calculated for each leaf, generally a simple average value of the observations that lie within (or mode if dealing with categorical/classification problem). To make a prediction on new data, the algorithm identifies the leaf its' covariates fit into and use that leaf's estimate. Below is an example of the tree structure where the covariate space is two variables, x and z :

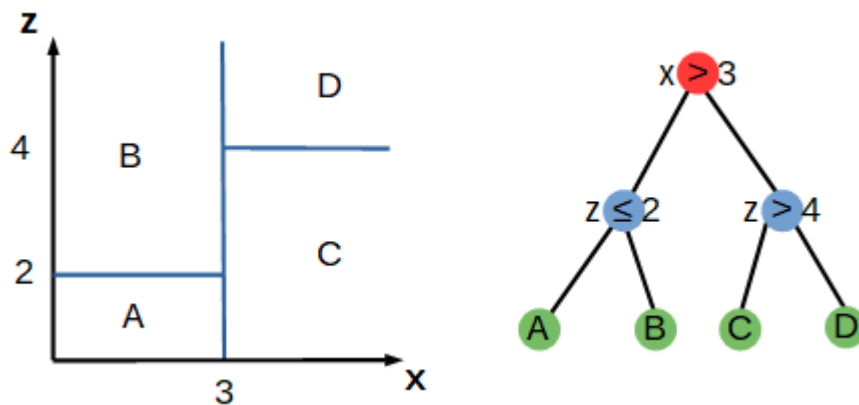


Figure 1: Example of Decision Tree covariate space partitioning

Terminology

While heavy on terminology, CART models use tree analogies to make it more intuitive. The structure is based around *nodes* (the dots in fig. 1), connected to one another by *branches* (the

lines). The *parent* of a node is the node immediately preceding it, while the *children* are the immediate successors beneath (e.g. blue nodes in fig. 1 are parents of green and children of the red).

The *root* node is on top of the tree and has no parents (red in fig. 1). It is a special-case of a *decision* node – which are nodes that take a set of observations, pass a condition splitting the data on some covariate, and output parted sets as children. Each condition on a decision node takes a *threshold value* and an inequality operator.

Leaf or *terminal* nodes (green in fig. 1) do not have any children and represent the final partitions the data is separated into, at the bottom of a tree. Each leaf in this example can be visualized in the 2D rendering of the covariate space on the left. The key requirement is no overlap in the partitions, so every observation lies in a single leaf.

The tree can be described by its *depth* – or the maximum length of a path from root node to leaf node (depth is 2 in the example). It can also be described by the number of leaves (4 in this example). Depth and number of leaves are commonly used as constraints on model complexity and tuned using cross-validation.

Setup

While intuitive and simple in structure, trees remain very flexible and require careful parameter tuning in the construction process. To formalize this method⁴, assume a dataset containing observation units i that each have a pair (X_i, Y_i) representing a vector of observable characteristics (covariates) and an outcome target (dependent variable) respectively. The tree is built to predict the outcome Y using the estimator $\hat{\mu}(X_i)$, which is the sample mean of dependent variable Y_i within leaf $l(X_i)$.

Tree Construction

The method for constructing – or *fitting/training* – a regression tree occurs in two stages: growing and pruning. In pre-processing the data is randomly split into a training sample and a validation sample using one of the cross-validation methods (e.g. validation set, k-folds). Below I cover the fitting process in detail for one potential implementation of CART in which

⁴ Note: formal CART model and notation in this section is based on (Athey, 2018b)

mean-squared error (MSE) is the base optimization criterion and cross-validation scoring metric, and the tuning parameter weights a penalty term on number of leaves in the tree.

Stage 1: Growing Tree

Goal: partition covariate space into a deep tree that maximizes Q^{crit} (–in-sample variance)

Recursive binary splitting is used to grow a deep tree model with the training sample data. The algorithm uses a top-down approach in that it starts with a single node containing all the observations and then progressively adds splits/decision nodes to partition the covariate space. It continues to add more partitions until each terminal node reaches some specified size. It is computationally infeasible to test every possible set of decision trees, so this approach uses a greedy algorithm to decide which splits to make. This means the split decisions occur in a bubble – they are not forward-looking and consider only the immediate/direct impact on optimization criterion. At each split the algorithm identifies which single partition will increase the optimization criterion (Q^{crit}) most and repeats this process.

For this stage the optimization criterion, Q^{crit} , is based on an in-sample goodness-of-fit function, $Q^{in-sample}$, and a regularization term with tuning parameter λ :

$$Q^{in-sample} = -MSE \text{ (Mean Squared Error)} = -\frac{1}{N} \sum_{i=1}^N (\hat{\mu}(X_i) - Y_i)^2$$

$$Q^{crit} = Q^{in-sample} - \lambda|L| \quad \text{where } |L| \text{ is total \# of leaves (terminal nodes)}$$

Overfitting is expected to occur when fitting using in-sample MSE estimates. Adding another split will always decrease the MSE, so by construction the criterion $Q^{in-sample}$ incentivizes the model to keep splitting until leaves have a single observation each (in which case MSE=0). The regularization term ($\lambda * \# \text{ leaves}$) is included in Q^{crit} to penalize the complexity of the model and restricts the depth of the tree depending on the tuning parameter value, λ . It is given a constant positive value in Stage 1 to encourage a deep tree that will be trimmed in Stage 2.

Stage 2: Pruning tree

Goal: identify regularization parameter (λ) that maximizes Q^{os} (–out-of-sample variance)

The second stage - *pruning* the tree - involves cross-validation to identify the optimal tuning parameter, λ , and thus specify the optimal size of the tree. The criterion, Q^{os} , is calculated using out-of-sample data (the validation-set held out from Stage 1).

$$Q^{os} = -MSE \quad \text{calculated on validation dataset}$$

Using separate data for growing and pruning makes variance estimates in this step unbiased – higher MSE values are a sign of overfit models and the lowest MSE option is the best model. This optimal model is typically identified using a grid-search over different λ levels, selecting the one that returns the highest criterion Q^{os} (the lowest out-of-sample MSE value). Higher λ values trim off more leaves (the so-called “pruning” of branches) and result in a simpler model.

Applying the Model

After identifying the optimal λ value in stage 2, estimator model can be constructed by simply taking the λ value from Stage 2 and plugging it into the Stage 1 algorithm to regrow the tree on training data. Given the greedy, algorithmic approach to fitting CART the specification on any single construction is unstable. Multiple different splits may have the same effect on Q^{crit} and the algorithm will be indifferent between them. The result is that 2 decision trees constructed on the same data could have different structures/splits. Random Forest methods attempt to reduce this variability by combining many different trees into a single estimator (see the Ensemble Methods section for further discussion).

Tuning Parameters and Scoring Criterion

An advantage of tree-methods is that they are compatible with a large variety of parameters, optimization criterion and scoring metrics. This allows them to fit nearly any data type but makes covering all varieties impractical in the scope of this thesis. The setup above (MSE plus a penalty term for number of leaves) is a standard off-the-shelf method that was included primarily for intuition on how tuning parameters are used in CART. Several other constraints can be substituted for or combined with the penalty term, such as depth of the tree or number of observations in each terminal leaf.

Likewise, in the pruning stage MSE is often substituted for different scoring metrics (e.g. cross-entropy for classification problems, negative log-loss for probability problems). In the empirical application section of this thesis I employ CART with tuning parameters for both maximum tree depth and number of leaves, using MSE as the splitting criterion in growing and log-loss as the scoring metric in the cross-validation stage.

Measuring Performance (External Validity)

Performance is typically measured using a held-out set of data (data not used in fitting or cross validation) to avoid bias from the feedback loop inherent in parameter tuning. While mean squared error (MSE) is the most popular metric for regression, classification tree methods often use accuracy scores or area-under the curve (AUC) analysis based on the ROC or precision-recall curves. Traditional inference values may also be calculated such as standard error but must be adjusted for the tree-structure by using leaf-level adjustments.

CART Fitting Process:

Preprocessing:

- (optional) Split off a validation set to test external validity on fitted model
- Make cross-validation splits (e.g. validation set method)
- Select optimization metric Q^{crit} :
 - $Q^{in-sample}$ metric (e.g. MSE)
 - Penalty terms/constraints (e.g. maximum tree depth, penalty for # leaves)
- Select a CV scoring metric Q^{os} (e.g. MSE, classification accuracy)

Stage 1: Growing Tree

- Implement algorithm on training set with weak penalty/constraint parameters
 - (will split to maximize Q^{crit} and make a deep tree)

Stage 2: Pruning Tree (cross-validation)

- Use tree from stage 1 and test set (from CV split) in grid-search to find optimal λ
 - (will choose λ that maximizes Q^{os})

Stage 3: Creating Estimator

- Plug optimal λ into algorithm from Stage 1 and train to get final model

Ensemble Methods

Ensemble methods are a popular ML strategy in which multiple base models are combined to create a single aggregated estimator. This can improve generalizability of the model by reducing sensitivity of the results to a single misspecified model and by combining diverse models that may capture different signal.

Averaging methods are a subclass of ensemble methods in which base model predictions are aggregated and then averaged. The base models can be of the same form (e.g. multiple CART regressions in a Random Forest) or based on different algorithms (e.g. combine CART with Lasso regression).

Boosting methods are a subclass of ensemble methods that use incremental learning – they train models sequentially with each subsequent fitting focusing on the residuals or misclassifications from the previous model. AdaBoost (adaptive boosting) is a popular implementation of this sort. While a promising field, boosting has not permeated the econometric field much so will not be further discussed in this thesis.

Random Forest (RF)

Random Forest (Breiman, 2001) is an averaging ensemble method that tries to improve upon the underlying CART model by combining predictions from many individual decision trees (hence the “forest”). A natural cost of their flexibility, individual decision trees tend to suffer from high variance - meaning they are very sensitive to the specific data sample on which they are trained. Forest methods reduce this variance by combining many trees fitted on different training samples, in the process mitigating bias from individual misspecified trees by blending them with the larger number of accurately specified trees. To achieve these better predictions, *bagged trees* are combined with a *decorrelating* mechanism.

Bagging

The *bootstrap* is a statistical method for resampling in which many separate samples are created from one dataset using random samples with replacement. The result is many representative samples that are unlikely to be identical to the original. *Bagging* (bootstrap aggregation) is when bootstrap samples are used to fit many individual models which are then combined (aggregated) to make a single prediction.

Bagged trees are grown deep and not pruned, so the individual trees have high variance but low bias. The aggregation of models reduces the overall variance and combats overfitting by using estimators constructed from multiple datasets.

Decorrelating Mechanism

While the randomness of bootstraps encourages lower correlation between trees versus training each on the same sample, bagged trees are still prone to high correlation. Running many trees on the same covariate space is likely to result in similar, correlated splits focused

on only the most important variables. If there is an especially strong covariate, it is likely to appear near the top of all individual trees fitted by CART regardless of the samples.

Since the benefits of aggregation are much higher with low-correlated trees, the Random Forest adds another level of decorrelation: when fitting the trees, every partition is restricted to a random subset of covariates. Under ordinary CART the decision trees search over the entire set of covariates (of size p), and partition at each stage on the one that increases the criterion most. Random Forest begins each partition by creating a new set of split candidates from a random sample of covariates (of size $m < p$). The rule-of-thumb approach in ML is to choose $m = \sqrt{p}$. This significantly reduces correlation from strong covariates, since the average fraction of splits that a given coefficient is excluded from is $\frac{(p-m)}{p}$. (James, Witten, Hastie, & Tibshirani, 2013).

Forest Construction Process:

The Random Forest construction process relies heavily on the CART algorithm, which is modified, applied many times, and the results averaged. Each tree is fitted on a different bootstrap sample from the training set and has an added layer of randomness in that each partition is restricted to a random sample of covariates of size $m < p$. The process can be broken down as follows⁵:

- 1) Specify model parameters for number of trees (B) and number of split candidates (m)
- 2) For each $b = 1, \dots, B$, draw a bootstrap sample $S_b \subseteq \{1, \dots, n\}$ from the training set
- 3) For each $b = 1, \dots, B$, grow a tree (deep, without pruning) on the sample S_b using recursive partitioning and restricting each partition's split candidates to a random sample of the covariates from X_i of size m . The result is a set of fitted trees, $\hat{f}^b(x)$.
- 4) Use the average predictions from all fitted trees as the estimator:

$$\hat{f}_{pred}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^b(x)$$

⁵ Note: formal notation based on that from (James, Witten, Hastie, & Tibshirani, 2013)

Cross-Validation Process:

Cross-validation is typically added to the above process to select optimal values of the tuning parameters, B and m . Drawing parallels to the CART process described above, the growing stage for RF involves constructing a large forest of many trees and the pruning stage removes trees (rather than branches) until the proper level of complexity is reached.

The bootstrap structure of the model-fitting process also enables an alternative to the traditional cross-validation methods that is useful if the sample size is small. To measure performance of a bagged estimator, one can employ the *out-of-bag (OOB)* error estimate. Breiman (1996) found that this method returns “nearly optimal estimates of generalization errors for bagged predictors” - and therefore it can be used in place of hold-out methods for evaluating external validity. On average each bootstrap sample will contain two-thirds of the observations from the training data, and the other one-third of observations is called the OOB sample. To construct the OOB error term, an algorithm makes predictions for each observation, i , by aggregating only the trees in which i falls in the OOB sample (i.e. only use predictions from trees that never saw i in training). These predictions are then used to calculate the error on each observation which are combined into a full OOB error estimate for the overall model.

Tuning Parameters and Scoring Criterion

The most critical parameters controlling complexity of the RF model are:

Number of trees (B) – This controls how many different bootstrap samples are taken and fitted to trees for use as a predictor. A higher value will be more flexible and can lead to overfitting past a point – particularly if the trees are highly correlated.

Number of split candidates (m) – This controls the number of covariates available to use at each partition of tree (common approach in ML is to choose $m = \sqrt{p}$). A high value will result in more highly correlated trees, so typically values are kept low to maximize the forest’s variance-reduction effect.

Tree-level parameters – Any tuning parameter available in the CART algorithm is available in RF too – such as depth of tree, number of leaves, etc. These are typically set at levels that encourage deep trees and no pruning/cross-validation is implemented. This makes the tree-level predictions low-bias, while the forest will mitigate the high variance.

As with CART models, the RF algorithm allows various scoring criterion to be used in the cross-validation/pruning step: MSE, accuracy score, R-squared, cross-entropy, log-loss, etc. Once the CV has identified the optimal tuning parameters, model implementation is the same as CART as well (refit algorithm on training set with optimal parameter value).

Interpreting RF Output

Random Forests tend to improve predictions and reduce variance versus a single decision tree, but it comes at the cost of interpretability. While a single CART estimator is easily visualized and the most important variables are salient on the tree diagram (higher up = more important), this is not practical when scaled to an entire forest. Therefore, RF algorithms include a measure of *variable importance* in their output which ranks each covariate by its impact on the optimization criteria (e.g. how much its splits reduce RSS). The rankings include scores that reveal relative importance – so it is clear when several of variables have outsized impact.

Measuring Performance

Performance is measured in same process as CART – using predictions from the final model on a held-out validation set for external validity checks.

Summary: ML Method Comparison

These five methods are a good representation of machine learning, covering many of the benefits over traditional statistical methods. Each has relative strengths and weaknesses (see table below), but the tuning parameters make each flexible enough for many applications and data types. To date, the bulk of ML econometrics literature sticks to these algorithms and as such they will be a good set of methods to test.

Table 2: Comparison of Selected ML Algorithms

ML Method	Strengths	Weaknesses
Ridge Regression	Variance reduction Highly multicollinear data robustness	Biased (shrunk) coefficients
Lasso Regression	Variance reduction Sparse model output (feature selection)	Biased (shrunk) coefficients
Elastic Net	Variance reduction Highly multicollinear data robustness Sparse model output (feature selection)	Biased (shrunk) coefficients
Trees (CART)	Flexible, non-linear model fitting (allows complex interactions) Computationally efficient	High variance tendency Stability issues (highly dependent on sample) No coefficients/marginal effects
Random Forest (RF)	Flexible, non-linear model fitting (allows complex interactions) Variance/stability improvements (over CART)	No coefficients/marginal effects

Chapter 3 Econometric Applications for Machine Learning

This chapter discusses machine learning tools in the context of economics applications. I review how econometric literature has integrated machine learning to date and overview several potential application areas. Finally, I frame the developments from a resource economics viewpoint to determine if ML has value to add to the field in practice.

State of Machine Learning in Econometrics

ML methods are increasingly finding their way into econometric literature as the field matures and its techniques become more salient. An example of the field's blend of academia and business, one of the early surveys to consider big data methods (trees, shrinkage methods) for causal questions was written by Google Chief Economist Hal Varian (2014), providing examples of macroeconomic applications. Belloni et al (2014) were early proponents of Lasso regressions for causal problems, exploring how they can help in high-dimensional settings through regularization and variable selection. In her survey, *The Impact of Machine Learning on Economics*, Susan Athey reviews the literature to conclude that ML “yields great improvements when the goal is semi-parametric estimation or when there are a large number of covariates relative to the number of observations,” (Athey, 2018a). Following up on this finding, Athey and Imbens (2019) identify several major classes of causal problems best suited to capitalize on these advantages, from heterogeneous treatment effects to adaptive experimental design.

Throughout these papers a common theme is that ML methods add the most value when applied for their natural strength: prediction. To identify prediction applications, Mullainathan and Spiess (2017) divide economics problems into two categories:

- 1) *parameter estimation*: concerned with finding $\hat{\beta}$ (estimate of β , the marginal effect of x on y)
- 2) *prediction*: concerned with finding $\hat{y}$ (estimate of y , the outcome based on x)

The paper goes on to declare “machine learning belongs in the part of the [economist's] toolbox marked $\hat{y}$ rather than in the more familiar $\hat{\beta}$ compartment,” since ML methods are not designed to create unbiased estimates of marginal effects (Mullainathan & Spiess, 2017).

ML for Prediction Problems ($\hat{y}$)

Supporting this analysis, I found most of the econometric ML literature to-date deals with $\hat{y}$ -problems. Prediction problems in economics are commonly implemented using the Potential Outcomes (PO) framework and, as such, the bulk of applied econometric research on ML focuses on this structure.

Treatment Effects under Potential Outcomes Framework

In simplified terms, the PO framework seeks to estimate the causal impact of a binary treatment (W) by taking the average difference in the control group's outcome ($y|W = 0$) and the treated group's outcome ($y|W = 1$). This difference is termed the average treatment effect (ATE) and it relies on the two groups being the same (e.g. totally random) aside from the treatment (see Appendix 3 for more in-depth discussion of this framework and assumptions). Since only the outcome y is of interest and no marginal effects are measured, this is a prime application for prediction methods. The canonical problem in ML applications of this sort estimates ATE under the uncounfoundness assumption, which relies on treatment being assigned as good as randomly conditional on observable characteristics of the observations (Athey & Imbens, 2019).

Since off-the-shelf ML algorithms do not calculate formal inference statistics with their predictions, they must be adapted to get valid confidence intervals for the treatment effect. Research in this area proposes adjustments in the algorithm settings (e.g. new fitting criterion) and then illustrate that the new specification meets asymptotic conditions necessary for inference. New cross-validation techniques are also proposed in order to avoid bias from the fitting process. For example, Athey and Imbens (2016) develop a *causal tree* method that substitutes a custom optimization criterion into the CART algorithm and prescribes a new, “honest” form of cross-validation based on sample-splitting to avoid bias. *Causal forest* techniques extend and adapt this methodology to a Random Forest-type ensemble algorithm (Athey, Tibshirani & Wager, 2019; Nie & Wager, 2017; Kunzel et al, 2019).

These methods are generally found to improve upon traditional semi-parametric methods (e.g. kernel, splines) through more flexible functional form and scalability to high-dimensional data without major losses in performance. However, beyond linearity-in-parameters, the methods do little to relax traditional assumptions and they require very specific conditions be met for

valid inference statistics (i.e. data must fulfil the strict PO and uncounfoundedness assumptions).

Synthetic Control

Synthetic control (SC) is a relatively new tool for estimating causal effects created by Abadie and Gardeazabal (2003) and then expanded on by Abadie, Diamond and Hainmueller (2010). Since then it has been deployed in numerous papers as a method of estimating better counterfactuals for comparative case studies. In their *Report on the State of Applied Econometrics*, Athey and Imbens (2017) go so far as to crown it “arguably the most important innovation in the policy evaluation literature in the last 15 years.”

In a case-study setting with panel data for a single treated unit and several control units, the method works by predicting the counterfactual outcome for the treated unit using some weighted mix of the control units’ observed outcomes in the post-treatment period. The difference between the post-treatment outcomes in the treated unit and the synthetic control unit is interpreted as the causal effect.

The critical step is calculating the optimal weight for each control unit to create a synthetic control unit with characteristics as close to the treatment group as possible. The standard method is to use linear regression for this task, but this limits the number of control units and matching criteria that can be used. ML methods have been proposed to improve the accuracy of the counterfactual prediction by allowing a flexible functional form in the matching, extending coverage to higher-dimensional datasets and allowing more control units to be used (Ben-Micheal, Feller, & Rothstein, 2018; Kinn, 2018).

ML for Parameter Estimation Problems ($\hat{\beta}$)

Even though ML methods do not work as the final model for a parameter estimation ($\hat{\beta}$) problem, there are several econometric processes in which prediction is a goal in pre-processing. In many multi-stage econometric processes, the preliminary steps implicitly use prediction to strengthen the causal interpretation of the final stages. Since these steps generally don’t require inference statistics, off-the-shelf ML algorithms can be applied to improve results, particularly when the data that is high dimensional, sparse or containing nonlinear underlying structures.

Variable Selection

While not a prediction application per se, the most basic application of off-the-shelf ML is for data-driven variable selection. Methods like Lasso and Elastic-Net, which return sparse models (i.e. remove some covariates), can be run on a high-dimensional model specification to identify which variables are least critical in the regression. Any variables whose coefficient is shrunk to 0 could be dropped and then the final model run using OLS to get unshrunk coefficients. For causal interpretations using this method an assumption of sparsity is required and there is a danger that regularization could remove a causally relevant variable while keeping its highly correlated pair. Mullainathan and Spiess (2017) show that there are some serious instabilities in variable selection using Lasso, so causal conclusions on the resultant sparse models are questionable. Furthermore, regardless of how sophisticated a model is used to choose variables, this method is vulnerable to bias from unobservables and the classical linear model assumptions must hold.

The Random Forest could also be helpful in variable selection due to its variable importance output. This measures which covariates reduce the optimization criterion (e.g. RSS) most across the many models and could give a researcher some insight to relationships present in a dataset even if the final model is not used for prediction.

Instrumental Variables

The two-stage least-squares (2SLS) instrumental variable procedure in econometrics involves an estimation step where a variable, x , is regressed on the instrument, z , to get a fitted value $\hat{x}$. In stage 2 the $\hat{x}$ is then included the regression for the causal model. With this set up, stage 1 is essentially a prediction problem in which overfitting must be avoided, a situation for which ML methods are designed. Belloni et al. (2014) propose using Lasso for the first-stage regression in order to perform variable selection on a set of potential instrumental variables and improve predictions. For cases of high-dimensional instruments in which the sparsity assumption cannot be made, Hansen and Kozbur (2014) offer a Ridge regression method argued to work with weak-instruments. Hartford et al. (2017) apply advanced, highly nonlinear ML algorithms called neural nets to a similar type of problem in what they refer to as *Deep IV*.

While each of these methods is designed to improve results on high-dimensional data, the exclusion requirement on instruments limits how often they are of practical use. Every

instrument and the dependent variable must only be related through the variable of interest, something that becomes difficult to support when there are many instruments used.

Propensity Score Matching

Propensity score matching (PSM) is very popular application for ML in causal analysis literature, used by a broad spectrum of researchers from epidemiology to economics. The method (covered in more detail in the Empirical Application chapter) aims to control for selection bias in treatment-assignment, often an unavoidable issue when using observational data. To control for the factors that influence whether an observation receives treatment, a two-stage process is implemented.

First, a binary indicator reflecting treatment implementation is regressed on observable variables to create an estimate for the probability each observation receives the treatment conditional on their observables. During stage 2 these estimates are integrated into the final regression – either by reweighting the outcome or by including the probability as a control variable.

Since predicting accurate probabilities of treatment is the main goal in stage one, it is common for practitioners to use all their available data to estimate it. The traditional implementation in this step (logit model) is not designed for high-dimensional data, so there is room for improvement by using off-the-shelf ML methods to form better probability estimates. Better probabilities theoretically result in better controls and a less biased causal model.

ML for Resource Economics

Due to the nature of assumptions in prediction ($\hat{y}$) type problems, ML methods based on the PO framework are not of much use in resource economics – a field that primarily deals with dynamic problems and time-series/panel data. In searching for applications, I found the PO assumption of strict Stable Unit Treatment Value Assumption (SUTVA) would often be violated. This contains the “no-interference” assumption that there are no externalities or network effects from treatment of a given unit. The method is designed for randomly or quasi-randomly sampled and unclustered data for which this is reasonable assumption. The panel data that resource problems use has two dimensions for interference to occur – time and panel

grouping – so the assumption is very difficult to justify and ML predictions of $\hat{y}$ will be rendered biased.

Synthetic control methods also rely on the no-interference assumption (Abadie et al., 2010), but in a case-study format interference becomes easier avoid. For example, if the treated unit is a country and there is concern over spillover effects within its region, the control donor pool can be restricted to countries outside that region. SC offers an alternative to the popular difference-in-difference identification strategy that can apply for policy evaluation in resource problems. For example, Smith (2015) uses synthetic control to analyse the resource curse using a panel of countries, Reimera and Haynie (2018) use it to explore economics impacts of marine reserves, and Sills et al (2015) apply the method to a tropical deforestation intervention.

Many resource economics problems are concerned with marginal effects, so the pre-processing ML methods designed for parameter estimation ($\hat{\beta}$ problems) may be of use. Instrumental variable problems are widely used in the field, so improvements to the 2SLS process have potential to add value. While propensity scores are often used within the PO model, they can also control for selection bias in common treatment-effect regression designs like difference-in-differences (DiD) models. In the next section I will test whether ML adds value in this propensity score DiD setting using an empirical example related to fisheries.

Chapter 4 Empirical Application: Fisheries

This chapter offers background information on the fisheries management article that I will use for an empirical test of machine learning's performance and value for resource economics. I introduce the original paper's dataset and my recreation of it, provide background on the paper's methodology, and discuss its core model specifications. The models and recreated dataset introduced in this chapter will be used in Chapter 5 where I apply ML methods to a propensity score estimation task.

Background

For evaluating machine learning methods on a practical resource economics problem, I will use a well-known fisheries management article titled *Can Catch Shares Prevent Fisheries Collapse?* (Costello, Gaines, & Lynham, 2008a). This paper examines a problem that is structurally representative of resource economics and uses a large dataset – particularly in terms of width (number of covariates) – making it a good test for ML.

Costello et al. examine the problem of fishery collapse, testing for a causal relationship with the implementation of catch-share fishery regulations. More concretely, the paper asks whether implementing ITQ⁶-type property rights in a fishery reduces the probability of fishery collapse thereafter. Collapsed fisheries are defined in line with prior research (Worm et al., 2006) as those in which annual harvest is less than 10% of the previous maximum harvest. The authors note that “this collapse metric may overestimate the frequency of collapsed fisheries,” but this errors on the side of conservative causal attribution (Costello, Gaines, & Lynham, 2008a). Several models are considered in the analysis, revolving around a binary treatment indicator for ITQ and a binary outcome indicator for fishery collapse.

⁶ Individual Transferable Quota

Data

Costello et al. Dataset

The original paper uses a panel dataset containing annual catch, taxonomic, and ITQ information for 11,135 fisheries from 1950-2003. To remain in line with the Worm et al. (2006) definition of collapse, Costello separates fisheries at the large marine ecosystem (LME) level and by species. Since this is a wide measure typically containing more than one regulatory body (multiple countries may fish in a single LME), ITQ treatment status for each LME is based on the primary commercial fishing country's system.

The species/catch time-series data was sourced from the *SeaAroundUs* public database (Pauly & Zeller, 2015). Costello et al. "searched published literature and government reports, interviewed experts on global fisheries," to manually identify the catch-share status of each fishery (2008a). For the study 121 fisheries were identified as under ITQ systems by 2003.

Recreated Dataset

To recreate the study, I likewise made use of the *SeaAroundUs* database for LME-level catch volume time-series and used Costello's list of ITQs (Pauly & Zeller, 2015; Costello, Gaines & Lynham, 2008b). The latest ITQ list has data through 2007 (148 fisheries) but I limited my recreation to the original 2003 cut-off to replicate the study more closely. This resulted in an initial database of 11,770 fisheries including 113 under catch-shares and catch data from 1950-2003.

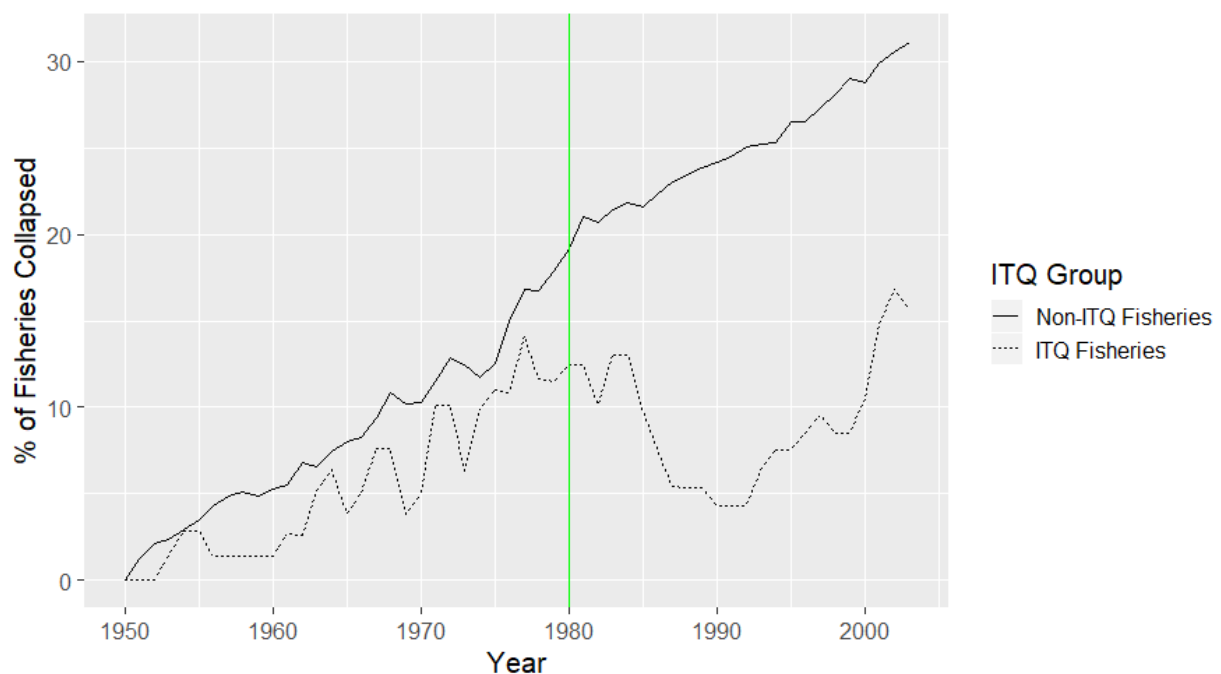
Due to inconsistencies in the taxonomic level at which fisheries were reported, I further trimmed the dataset so that only fisheries with both genus and species identified were included. Taxonomic information was cross-referenced using *FishBase*, *SeaLifeBase*, and *WoRMS* databases (Froese & Pauly, 2019; Palomares & Pauly, 2019; WoRMS Editorial Board, 2019). The cleaned and trimmed dataset contained 273,548 observations across 6,406 fisheries. This includes observations from 62 LMEs, 698 genera and 1,270 species.

The original paper doesn't report the exact data cleaning decisions, but as a point of comparison they end up with 302,852 observations over 64 LMEs, 687 genera and 1,179 species. Comparing over reported summary values from the original report, the recreated dataset looks to be reasonably close (conservative in collapse metric if anything):

Table 3: Comparison of Dataset Descriptive Statistics

Variable	Recreated	Costello
Percent Collapsed (2003)	30.1%	27%
Relative fraction collapsed, ITQ fisheries versus non-ITQ (2003)	50.2%	50%
Annual trend towards collapse (global)	0.58%	0.5%

Visualization of trends from the recreated dataset also resemble the trends from the original paper and show a divergence between ITQ and non-ITQ fisheries in prevalence of collapse after the implementation period begins (green line reflects year of first ITQ implementation):

**Figure 2:** Collapse Trends in Recreated Dataset

Replication Results

Using the recreated dataset, I attempted to replicate the primary models with generally close results (see Appendix 1 for regression results). This serves as a check on the base dataset and offers a point of comparison for several alternate data samples in future tests.

Econometric Methods Background

This section provides background on the theory and implementation of three econometric methods the Costello et al. paper relies on: difference-in-differences, logit regression and propensity score matching. Discussion is kept brief and focused within the context of this application, but Angrist and Pischke (2009) offer a useful reference for deeper background.

Difference-in-Differences

Costello et al. use a Difference-in-Differences (DiD) strategy for estimating the causal effect of the ITQ policy. The observations are separated into two groups – treatment and control – based on whether they received the intervention of interest (ITQ). This setup removes bias in the causal effect parameter stemming “from permanent differences between [treatment and control] groups, as well as biases from comparisons over time in the treatment group that could be the result of trends due to other causes of the outcome,” (“Difference-in-Difference Estimation,” n.d.). To implement DiD, a group indicator dummy and time trend are included in the regression along with the treatment variable (for Costello model treatment is number of years ITQ has been in place).

The critical assumption for DiD to result in unbiased estimates is the *parallel trends assumption*. This assumes that had the intervention never occurred, the difference in outcome between treatment and control groups would be constant over time. This means pre-treatment observed outcomes for both groups must follow the same trend and post-treatment the counterfactual outcome for the treatment group is assumed to be a parallel trend to the control group’s observed outcome.

Logit Regression

Due to the binary nature of the outcome variable (fishery collapse), Costello et al. employ a logit model to estimate the causal effect. This improves results versus an OLS regression by mitigating heteroskedasticity concerns and restricting probability estimates to the [0,1] range. To get these benefits the outcome variable is transformed using a logit link function, $\log\left(\frac{y_{it}}{1-y_{it}}\right)$, a binomial distribution is assumed, and least-squares optimization is replaced with maximum-likelihood estimation (MLE) – meaning the RSS criterion is replaced with log-loss.

These changes have consequences for the model output in that the estimated coefficients represent the rate of change in *log-odds* of the positive binary outcome rather than the more interpretable *probability*. To get the effect in terms of probability a non-linear transformation must be made ($\text{Pr} = \frac{1}{1+e^{-(\log\text{-odds})}}$). This means the marginal effect of interest is different at different levels of the covariates, so inference requires some form of averaging to get a single value (Costello uses average marginal effect). This also requires a new estimation of the standard errors for the marginal effect on probability, for which Costello uses the delta method.

Propensity Score Matching

Selection bias occurs in non-random studies when there is a systematic difference between control and treatment groups due to the way treatment is assigned (or *selected*). If this difference effects the trends over time, then the parallel trends assumption of the DiD model will be violated and causal estimates will be biased. Propensity score matching (PSM) is a technique designed to mitigate selection bias by conditioning each unit on the observable characteristics that effect its treatment. This is achieved using a *propensity score*, representing the probability of treatment (W_i) conditional on a set of observable variables (X_i):

$$\text{propensity score} = e(x) = P[W_i = 1|X_i = x]$$

Implementation

Implementation of PSM in a regression context occurs in two stages:

Stage 1: Estimating Propensity Scores

A propensity score value must be estimated for each observation – in econometrics this is traditionally done using a logit model by regressing the treatment indicator dummy on any covariates thought to impact it. Then probabilities are backed out from the log-odds fitted value:

$$\text{Pr}(W_i = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_i)}}$$

Since this is essentially a prediction problem, ML techniques offer an alternative method for estimating p-scores. By allowing a more flexible fit to the data these may result in more accurate estimated values. I test this on the Costello et al. models in Chapter 5.

Stage 2: Adding Propensity Scores to Causal Model

To realize the benefit of propensity scores in removing bias, they must be integrated into the final causal model by one of several ways, from directly matching up units with similar odds of treatment to reweighting outcomes based on the propensity score. In the DiD strategy propensity score estimates are generally added to the regression as a control variable that implicitly “matches” the observations.

Assumptions

For this process to fully eliminate the selection bias, two assumptions must hold:

Assumption A1 Unconfoundedness: $W_i | X_i \perp (Y_i(0), Y_i(1))$

Also called selection-on-observables or strongly ignorable treatment assignment, this assumes that, conditional on the observable variables, treatment assignment is as good as random.

Assumption A2 Overlap: $0 < e(x) < 1$

This assumption requires there to be overlap in covariate values of the treated group and control group. In practice this requires the distribution of propensity scores to be bounded away from 0 and 1.

Costello et al. Models

Costello et al. use several different models to estimate the causal effect of ITQ systems and check that results are robust. In this section I overview each of the specifications.

First-Cut Model Specification

As an initial baseline model, Costello et al. pool all the catch observations into two groups, ITQ and non-ITQ (just 2 observations per time period). They then run a regression of percent of fisheries within each group (indexed i) that are collapsed at time t ($PctCollapsed_{it}$) on a group dummy indicator (ITQ_i), the fraction of ITQ fisheries that have implemented their catch-shares by year t ($ITQimp_t$, trending towards 1), a linear time trend ($time_t$), and an interaction between the group dummy and implemented-ITQ fraction.

First-Cut Model:

$$PctCollapsed_{it} = \beta_0 + \beta_1 ITQ_i + \beta_2 ITQimp_t + \beta_3 ITQ_i * ITQimp_t + \beta_4 time_t$$

β_3 is the parameter of interest for causal analysis and represents how much implementing ITQs reduces the percentage of collapsed fisheries. This is used as a simple high-level model for checking that the data has reasonable trends. The main causal models in this article employ fishery-specific estimation methods.

Base Model Specification

To test the hypothesis that ITQ implementation reduces the probability of fishery collapse, Costello et al. use a difference-in-differences (DiD) framework with a logistic regression. The binary collapse indicator is transformed by the logit link function to create the outcome of interest. The base specification treats each annual catch observation (indexed by fishery, i , and time, t) as independent in a simple logit regression optimizing on maximum likelihood:

Base Model:

$$\Pr(\text{collapsed}_{it} = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 ITQ_i + \beta_2 \text{yearsofITQ}_{it} + \beta_3 \text{time}_t)}}$$

ITQ_i - is a dummy ITQ group indicator variable, takes value of 1 if fishery i implements a catch-share system by 2003

yearsofITQ_{it} - is the continuous treatment variable, reflects number of years a catch-share system has been in place at fishery i , time t

time_t - is a simple linear time-trend

Under this specification the parameter of focus is β_2 which represents the marginal effect of a year under ITQ management on the probability of collapse. The β_1 coefficient represents the general difference between collapse probability in the ITQ group versus the non-ITQ group. This term is critical for the DiD identification framework and controls for time-varying unobservables under the parallel trends assumption.

Advanced Model Specifications

Acknowledging the potential biases in the base model, Costello et al. then add a propensity score to the regression to control for selection bias and use a fixed-effects transformation to control for time-invariable fishery-specific unobservables.

Propensity Score Model

Costello et al. use propensity score matching to control for selection bias with the goal of creating “treated and control groups that are as similar as possible,” (2008a). To implement PSM in the model, they estimate propensity scores and add them to the base regression:

Base P-score Model:

$$\Pr(\text{collapsed}_{it} = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 ITQ_i + \beta_2 \text{yearsof} ITQ_{it} + \beta_3 \text{time}_t + \text{pscore}_{it}\lambda)}}$$

The researchers try several model specifications for estimating the propensity score, fitting a logit regression of the treatment indicator ($ITQimpl_{it}$, a year-specific dummy indicator for whether ITQ is in place) on dummy variables for the LMEs, Genera and Species. For example, the first specification used LME classification dummies (indexed $l \in L$) as the covariates:

LME P-score Predictor Model:

$$\text{pscore}_{it} = \Pr(ITQimpl_{it} = 1) = \frac{1}{1 + e^{-(\alpha_0 + \alpha_1 LME_1 + \dots + \alpha_L LME_L)}}$$

Costello then ran this regression for each year to get a probability of treatment for each fishery conditional on year and the observables included as covariates. Further specifications tested included genera dummies only, species dummies only, and an “all-in” model including all LME/genus/species dummies.

Given the number of levels in each categorical variable (64 LMEs, 687 genera, 1,179 species), Costello’s dummy approach results in large covariate sets. Since the estimators were refit for each year, they can be rewritten as a single model including year dummies (52) and full interaction terms. This results in very high-dimensional regressions... even the simplest specification (LME dummies) has 3,3961 covariates. The “all-in” specification (including LME, genus and species) has over 100,000 covariates. Since ML methods are designed for use with high-dimensional data in mind, this is an optimal area to apply them.

Fixed Effect Model

To control for potential bias from time-invariant unobservables, the authors rerun the base model after a Fixed-Effect transformation clustering on fisheries. This removes any time-constant variables so the ITQ_i indicator drops out of the regression and marginal effects on probability are not possible to calculate. Therefore reported coefficients for this regression

reflect rate of change in *log-odds*. Thus, while potential bias is removed, the interpretability is compromised.

Chapter 5 Empirical ML Test: Propensity Score Matching

This chapter provides a test on the value of off-the-shelf machine learning methods for first-stage propensity score estimation. Model specifications and data are inspired by the Costello et al. paper (2008a), but the goal is findings that are generalizable to parameter estimation problems across resource economics.

First, I explain the test design and implementation, which involves the application of logit, shrinkage (Lasso/Ridge/Elastic-Net), CART, and Random Forest regressions to three covariate sets to create propensity score predictors. Next, I compare the results of these first-stage regressions across fitting methods, discussing the quality of estimates with consideration for potential overfitting problems. Finally, I use each trained model to predict propensity scores and apply them to Costello’s second stage causal models to evaluate robustness of the main results to the new methods.

Background

As discussed in Chapter 3, propensity score estimation is a popular application for machine learning methods in causal analysis literature because it is a prediction problem for which inference statistics are not generally required. In practice, this causal indifference in stage one results in a tendency to include many covariates to improve the model fit. Zigler and Dominici refer to this as the “kitchen sink” approach – in which researchers throw all their available data fields into the matching estimator and ignore the risk of confounders (2014).

When applying high-dimensional data, the risk is that resulting estimates will include both true signal (the variables that are causally impacting treatment) and noise from extra variables (confounders) – which could result in more variance in stage two without any improvement on bias. Schuster et al. test this with high-dimensional logistic estimators and conclude “overfitting of propensity score models should be avoided to obtain reliable estimates of treatment or exposure effects” (2016).

While the final impact of overfitting propensity scores is still an area of active debate, the root issue is one that ML procedures have been designed to account for. The standard process of cross-validation is tailored to improve external validity and prevent noise from overpowering the signal in fitted models, while still allowing for a flexible fit on the data. To test if this adds

value in a propensity score estimator, I apply several algorithms to a high-dimensional dataset side by side with traditional logistic regression, using a hold-out validation set to measure the results.

The Test

I test the claim that off-the-shelf ML methods produce better out-of-sample predictions than traditional econometric methods in a propensity score estimation scenario. The expected result is better prediction through ML's ability "to fit complex and very flexible functional forms to the data without simply overfitting," (Mullainathan & Spiess, 2017). To support this hypothesis, ML methods must show two results:

- 1) More accurate predictions (lower error) than the traditional alternative
- 2) Levels of overfitting in-line with or below the traditional alternative

Result 1 serves as evidence that the flexible fitting process does improve the model fit and prediction accuracy. Result 2 is a requirement for the model to be generalizable and for the findings in Result 1 to be meaningful, since overfit models result in poor out-of-sample predictions and produce misleading (overstated) fit metrics.

Validation Method

Since ML tuning parameters are chosen with feedback from the training dataset, model fit must be evaluated on data unseen during the model construction process. Before fitting the estimators, a hold-out validation set containing 30% of the observations is randomly selected and removed, leaving 70% of the data for training. While this separation of data is unnatural for the logit models used by Costello in the original paper (no tuning parameters), the process provides a better point of comparison between methods and allows a test for overfitting. The same training set is used for each method, so they are on the same playing field.

Since the outcome is unbalanced (only ~6% of the observations have ITQs implemented), I use an endogenous stratified splitting method when creating the validation set. The result is that each set has similar fraction of positive binary outcomes, which helps ensure the two samples are representative of the full dataset.

Model-Fitting Procedure

Each model is fit on the training set (70% of full dataset), using its method's standard off-the-shelf implementation. This means the ML methods add an extra step for parameter tuning (cross-validation), while the traditional method (logit model) is fitted in one step on the full training set.

Cross-Validation

The specific cross-validation process used for each model is dependent on the algorithm. Validation-set CV was employed for the shrinkage models (Lasso, Ridge, Elastic-Net), holding out a stratified random sample of 30% of the training set for use as the test sample in scoring different tuning parameters. This decision was made primarily on computational grounds, but k-fold was also tested and found to have similar performance results.

The tree methods (CART, Random Forest) make use of stratified k-fold CV with 5 folds. This was possible due to the computational efficiency of their fitting method. Testing on several specifications, 5 folds had results better than a simple validation-set and in-line with higher fold-counts.

Parameter-Tuning

Each algorithm was implemented using the standard tuning parameters that are covered in most introductory literature:

Table 4: *Parameters Tuned by Cross Validation*

Algorithm	Tuning Parameters
Ridge	λ (weight on l2 regularization penalty)
Lasso	λ (weight on l1 regularization penalty)
Elastic-Net	λ_1 (weight on l1 / Lasso penalty) λ_2 (weight on l2 / Ridge penalty)
CART	Maximum Depth Maximum Leaves (terminal nodes)
Random Forest	Number of Trees

	Maximum Features (m , size of split candidate pool as percent of parameter count p)
	Maximum Depth (of each individual tree)

For all methods the final tuning parameters were selected from a range of values using a grid-search with a log-loss scoring criteria calculated on the CV test set(s). Since the tree methods have several parameters, I added a step to narrow down the parameter ranges included in the grid-search and make it more computationally tractable.

This parameter-search step calculated test scores for a range of each parameter while holding the others constant. The results are best visualized in a validation curve like the one below:

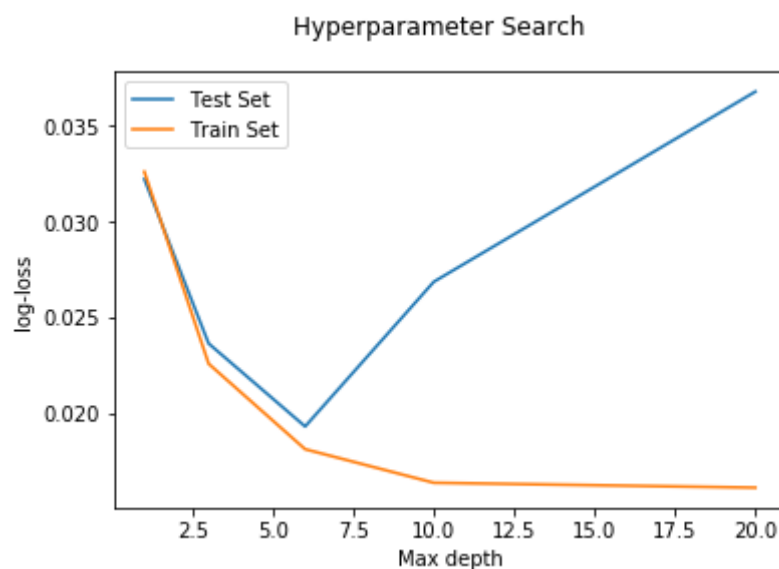


Figure 3: Validation Curve Example (Model 1, CART, Max-Depth)

The curve provides a visual example of overfitting – increasing model complexity (max depth) always improves the training fit, but past a certain point (6) it results in worse performance on the CV set and harms generalizability. For the grid search I focused on values around this optimal depth of 6 (see Appendix 4 for reporting of curves for other parameters).

Performance Evaluation

The fitted models were used to calculate performance metrics (measures of fit) on both the training and validation sets created in the initial split. To test overall model fit (Result 1), the performance scores are directly compared using the validation set. To check for overfitting (Result 2) a ratio of the performance score on the validation set to score on the training set is calculated and may be compared across models. A very high validation error relative to the

training set error is a sign that noise in the training set is overfitted into the model and the predictions on new data may be questionable.

Performance Metric

Performance is compared using the log-loss and mean-squared-error (MSE) metrics, since they are both meaningful for the prediction type (probability) and robust to unbalanced outcomes. Log-loss is the fitting criterion in the logistic model (maximum-likelihood optimization) – it penalizes bad probability estimates exponentially for their degree of wrongness and gives a value in range $[0, \infty)$ where 0 is perfect predictions.

Mean squared error (MSE) was chosen as a secondary metric since it is the standard measure of error in ML for most regression functions (and the splitting criterion in tree methods). In this case MSE is equivalent to the Brier score which is designed as a loss-function for probability regressions. It takes the average squared difference between the predicted probability and the actual outcome, resulting in a value between 0 and 1 with lower values representing better fit.

Test Dataset 1 (Global Fisheries)

Data

Starting with the recreated dataset described in Chapter 4, I dropped all observations before 1970 to remove the unnecessarily long pre-treatment time-series. This has little effect on the overall results (see “first-cut” results below) and serves to create a more balanced dataset in terms of ITQ-implementations versus total observations. This improves convergence on the solution at no significant cost on the empirical results or causal identification. It also increases the ratio of number of parameters to observations, which increases the overfitting risk and creates a good test for ML.

Table 5: Estimated Coefficients from First-Cut Model (Dataset 1)

Variable	Dataset 1	Costello
ITQ _i	-0.0460*** (0.00560)	-0.0428*** (0.00505)
ITQ_imp _t	0.0387*** (0.0141)	0.0090 (0.01255)
ITQ _i *ITQ_imp _t	-0.175*** (0.0135)	-0.1367*** (0.01188)
time _t	0.00484*** (0.000274)	0.0049*** (0.00025)
Intercept	0.0199*** (0.00627)	0.0096* (0.00627)
R-squared	0.922	0.92
N	108	108

Standard errors in parentheses. Coefficients reflect rate of change for percent of fisheries collapsed

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Model 1

Specification (Model 1)

All propensity score estimator models tested in this thesis differ from the Costello et al. (2008a) version by removing the full interaction term between year dummy and the categorical variable dummies. That specification resulted in a high-dimensional covariate space that would not converge to a solution under the logit model given my limited computational resources. Instead I included full year dummies to capture any year-specific shocks and a linear time trend to capture constant changes in treatment implementation over time. The resultant models are still relatively high-dimensional and remain a valid test with potential for overfitting.

As a base specification, Model 1 includes all LME dummies and time trends (95 covariates):

Model 1:

$$pscore_{it} = \Pr(ITQimpl_{it} = 1) = \frac{1}{1 + e^{-(\alpha_0 + \alpha_1 LME_1 + \dots + \alpha_L LME_L + \theta_2 YR_2 + \dots + \theta_T YR_T + time_t)}}$$

Where LME_l is indicator dummy for fishery LME, YR_t is time dummy indicator, $time_t$ is linear time trend

Results (Model 1)

ML was widely ineffective on Model 1, with the Logit model outperforming all others on both validation-set scoring metrics (MSE and log-loss):

Table 6: Propensity Score Fitting Results (Model 1)

Model	Optimal Tuning Parameters	MSE (valid. set)	Log-loss (valid. set)	Overfit Ratio (MSE)	Overfit Ratio (Log-Loss)
Logit	NA	0.00503	0.01755	1	1
Lasso	λ : 0.1	0.00504	0.01773	1.002	1.010
Ridge	λ : 0.1	0.00504	0.01773	1.002	1.007
Elastic-Net	$\lambda_1 = 0.1$ (Lasso) $\lambda_2 = 0$ (Ridge)	0.00504	0.01773	1.002	1.009
CART	Max depth: 10 Max leaves: 45	0.00505	0.01822	1.031	1.076
Random Forest	Num of trees: 35 Max features: 50% Max depth: 10	0.00505	0.01779	1.031	1.056

Performance metrics (MSE, Log-loss) calculated on the held-out validation set (lower score is better fit). The overfit ratio represents performance score calculated on validation set divided by score calculated on the training set (large values are sign of overfitting).

Under this specification the Logit model is unanimously the best model, both on data fit and overfitting – of which there is none since performance is equal in training and validation sets (overfit ratio = 1).

All three shrinkage methods optimized at the lowest tested tuning parameter ($\lambda = 0.1$), setting a small weight on the penalty term and a low level of regularization. This result implies that reducing bias is relatively more important than reducing variance for improving the performance metric (log-loss). Under Lasso and Elastic-Net the tuning parameter was large enough to engage variable selection – shrinking coefficients on 33 of the 95 covariates to down 0.

Non-linear ML methods beat logit performance on the training set but scored worse on the validation set. This illustrates their ability to capture more information from hidden and

nonlinear interactions, but in this case the additional information was largely noise (overfitting). Random Forest performance is very close to the single CART with a slight reduction in overfitting (overfit ratio on log-loss 1.056 vs 1.076). The similar scoring between the two methods is a sign that the data is relatively stable and therefore doesn't benefit much from the extra layers of randomization (bootstrapping and splitting criterion).

A potentially useful output from the Random Forest algorithm is *feature importance* rankings. This improves model interpretability by ranking all covariates based on their contribution towards the estimates (calculated as the total reduction in error, RSS, from all the covariate's splits across all trees, divided by the total number of trees). A covariate is more important if it is used more often and if it has a bigger impact on the error term. Scores are then scaled so that the total importance metrics sum to 1. For this regression the top 10 most important covariates are:

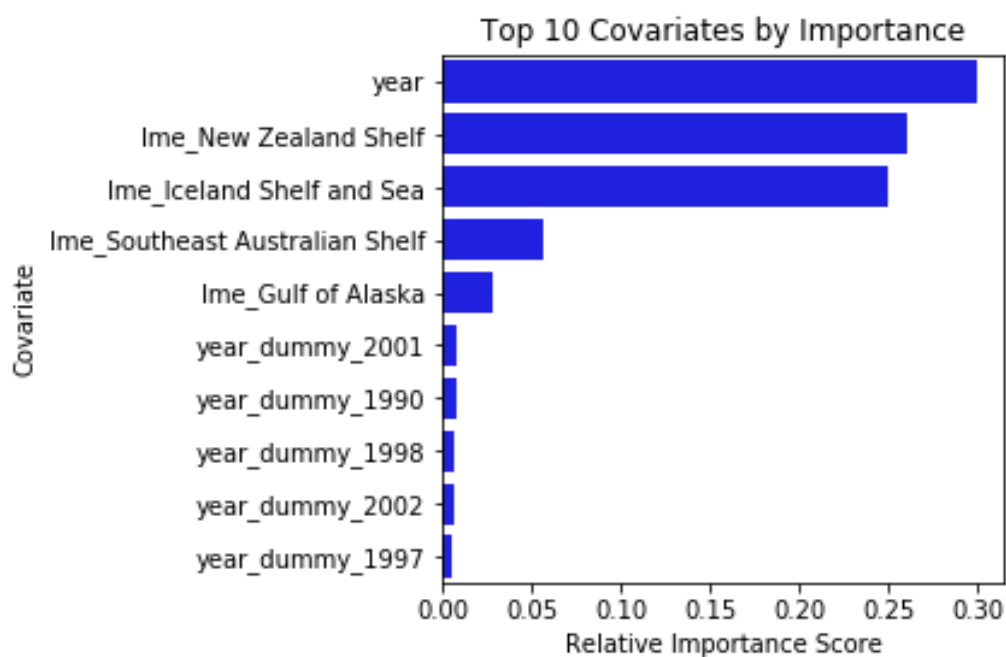


Figure 4: Random Forest Covariate Importance (Model 1)

The results on this test do not support ML methods for p-score estimations, but they do show a potential use as a data exploration tool for high-dimensional datasets. The variable selection from shrinkage methods can direct researchers towards variables that they may want to reconsider including in the analysis, while the covariate importance metrics from the random forest can help identify some of the most important drivers of variance.

There is no clear explanation for ML's underperformance on this test, but I have two theories on potential causes:

- 1) the dimensionality of the dataset/specification
- 2) the variance in the underlying data

While 95 covariates represent a high-dimensional problem by economics standards, the model did not result in much overfitting which suggests that it was not high-dimensional enough to benefit from ML's strength. Likewise, the variance in these covariates is limited since they are sparse dummies in which each observation takes only three non-zero values out of 95. This does not give the ML models, particularly the non-linear ones, much to work with to find highly complex relationships. To test each of these theories I ran tests on two more models.

Model 2

Specification (Model 2)

To test a higher-dimensional specification I initially set out to recreate Costello's "all-in" specification (including LME, genus, and species dummies) in my analysis. However, even excluding the time-interaction variables, convergence failed due to high multicollinearity between species and genus dummy variables. 461 genera in the dataset include only a single observed species and over three-quarters of the genera contain only two. Since the two fields largely capture the same information, I ran the second model with genus dummies but not species.

The second test specification includes LME, genus and time dummies/trend (792 covariates):

Model 2:

$$pscore_{it} = \Pr(ITQimpl_{it} = 1) = \frac{1}{1 + e^{-(\alpha_0 + \alpha_1 LME_1 + \dots + \alpha_L LME_L + \gamma_1 Genus_1 + \dots + \gamma_G Genus_G + \theta_1 YR_1 + \dots + \theta_T YR_T + time_t)}}$$

Where LME_1 is indicator dummy for fishery LME, $Genus_g$ is indicator dummy for fishery genus, YR_t is time dummy, $time_t$ is linear time trend

Results (Model 2)

Again, ML methods proved largely ineffective, with logit outperforming all but the nonlinear models, which suffered from overfitting:

Table 7: Propensity Score Fitting Results (Model 2)

Model	Optimal Tuning Parameters	MSE (valid. set)	Log-loss (valid. set)	Overfit Ratio (MSE)	Overfit Ratio (Log-Loss)
Logit	NA	0.00249	0.00883	1.13	1.11
Lasso	λ : 0.1	0.00249	0.00887	1.13	1.11
Ridge	λ : 0.1	0.00251	0.00900	1.12	1.10
Elastic-Net	$\lambda_1 = 0.1$ (Lasso) $\lambda_2 = 0$ (Ridge)	0.00249	0.00887	1.13	1.11
CART	Max depth: 20 Max number leaves: 75	0.00190	0.01201	1.13	1.20
Random Forest	Num of trees: 100 Max features: 0.28% Max depth: 40	0.00128	0.00609	3.05	3.88

Performance metrics (MSE, Log-loss) calculated on the held-out validation set (lower score is better fit). The overfit ratio represents performance score calculated on validation set divided by score calculated on the training set (large values are sign of overfitting).

Logit slightly outperformed its linear ML alternatives in both performance metrics and, while the higher-dimensional setting increases the overfitting ratio, the results are within reasonable levels (validation set only $\sim 10\%$ higher error). Regularization tuning parameters in the shrinkage models again optimized at the lowest penalty weightings, implying that variance is already low. Lasso/Elastic-Net models performed significant variable-selection, shrinking 446 of 792 coefficients down to 0.

CART performed very well on the larger covariate space by the MSE metric but underperformed on log-loss. This might be explained by the difference in criterion for choosing tree partitions (MSE) versus the criterion for fitting logit regressions (log-loss). The grid search was based on log-loss as the scoring metric, but splitting decisions look to have a bigger performance impact in fitting. Given the higher overfitting score (20% increase in log-

loss from training to validation) and poor log-loss score, logit remains the preferred model.

Random Forest highly outperformed all other methods on both metrics, but there is evidence of serious overfitting: log-loss jumps nearly 4x and MSE increases 3x from training to validation. Caution is advised when using this model's predictions as it contains significant noise. The variable importance list may still provide some value in identifying strong predictors:

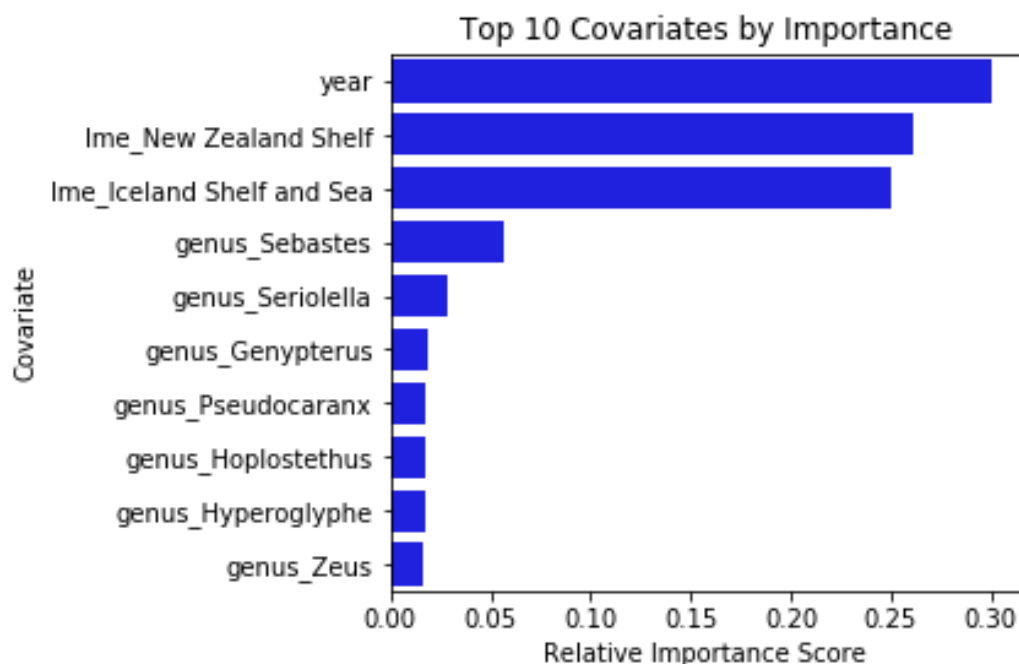


Figure 5: Random Forest Covariate Importance (Model 2)

The top 3 covariates are the same as Model 1, with time trend the most important variable (by far in this case). Still the ML methods on whole are underwhelming even under the higher-dimensional setting. To check the theory that higher variance in the covariate set is necessary to unlock ML potential I ran one more model on a newly constructed dataset.

Test Dataset 2 (OECD Fisheries)

The covariate sets tested in Models 1 and 2 had limited variance because they were all sparse dummies (plus a linear trend). Since ML methods are best for picking up complex and hidden variance in high dimensions this may not be a fair test for their potential. To test the theory that ML adds more value in the presence of more continuous and time-varying covariates, I

constructed a new dataset: instead of area and taxonomic dummy variables the new fields capture similar characteristics using continuous biological, commercial and macro data.

Data

Sample Restrictions

Due to limited data availability, several changes were made to the data sample for the final test. The time-series date range has been changed to 1970-2008, trimming away some pre-treatment observations and adding 5 years to the tail end to counter reduction in sample size. To allow the use of biological data, species have been restricted to those in the *Fishbase* database (Froese & Pauly, 2019). In effect this removes just three groups which differ from fish in significant ways: crustaceans, molluscs, and cephalopods. The bulk of fisheries remain covered after this restriction, so the results are still generalizable to a wide range of species and ecosystems. The final, most restrictive change is a constraint on countries to facilitate the use of macro data. Only OECD fisheries are considered – defined as fisheries whose catch reporting entities since 1950 have all been in the current list of 36 OECD countries.

The resulting dataset contains 59,969 observations and 2,074 fisheries, covering 54 different LMEs, 383 genera and 631 species. There are 30 fisheries with ITQs – a ratio in-line with the original dataset in the ~1% range. As a sanity check on general trends for the new sample, I compare results on the “first-cut” model versus the full Costello dataset:

Table 8: *Estimated Coefficients from First-Cut Model (OECD sample)*

Variable	OECD Sample	Costello
ITQ _i	-0.113*** (0.0172)	-0.0428*** (0.00505)
ITQ_imp _t	0.107** (0.0449)	0.0090 (0.01255)
ITQ _i *ITQ_imp _t	-0.0799** (0.0135)	-0.1367*** (0.01188)
time _t	0.00279* (0.00153)	0.0049*** (0.00025)
Intercept	0.113** (0.0465)	0.0096* (0.00627)
R-squared	0.75	0.92

N	78	108
---	----	-----

Standard errors in parentheses. Coefficients reflect rate of change for percent of fisheries collapsed

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Trends are reasonably close to the original findings – the effect of implementing ITQs in the new sample is an 8% reduction in percentage of collapsed fisheries versus 13% in the original data.

New Data Fields

Using this sample of fisheries, I gathered additional data to create a broad set of covariates with more variance than Costello’s dummy fields, while covering a similar range of biological, commercial, and regional information that could impact ITQ implementation. Biological fields were primarily pulled from the *FishBase* (FB) database (Froese & Pauly, 2019). The *SeaAroundUs* (SAU) database was tapped for species classification fields and catch composition details such as country, gear-type and end-use type (Pauly & Zeller, 2015). *Maddison Historical Statistics* (MHS) was used for historical macro data on GDP and population (Maddison Project, 2018; Bolt et al., 2018). A detailed list of these new fields with sources is included in the model description below.

Model 3

Specification (Model 3)

To test the ML methods on a higher variance “kitchen sink” specification, I substituted the full set of newly created covariates for the dummy variables and kept the time dummies/trend. The resulting model uses the following 124 covariates to estimate probability of ITQ implementation (see Appendix 4 for breakdown of dummy variable classes):

Table 9: Description of Covariates in Model 3

Variable	Type	Description	Source
Biological Covariates			
Commercial group	Dummies (9)	Broadest species grouping, classifies species along general commercial lines (e.g. anchovies, tuna/billfishes)	SAU
Functional group	Dummies (23)	Secondary species grouping metric, classifies on taxonomic traits, ecosystem, diet and size	SAU

Common depth	Continuous (2)	Variables for species' common high and low depth, measure of ecosystem	FB
Trophic level	Continuous	Measure of the species' diet and level in the food chain	FB
Vulnerability	Continuous	Measure 0-100 of fish vulnerability to extinction, est. based on ecological and life-history characteristics	FB*
Years-to-maturity	Discrete, 4 levels	Estimate of species' time to sexual maturity, based on FB resilience classifications (range 0.5-15 years), measure of species' productivity	FB
Commercial Covariates			
Gear category	Continuous, fractions (7)	Ratio of year's tonnage caught on different fishing methods (e.g. trawls/dragged gear, gillnet), fishery-level	SAU
End use type	Continuous, fractions (4)	Ratio of year's catch earmarked for 4 different uses (e.g. human consumption, discard), fishery-level	SAU
Fishing sector	Continuous, fractions (4)	Ratio of year's tonnage caught by type of fishing entities (e.g. commercial, subsistence), fishery-level	SAU
Price category	Discrete, 4 levels	Species price category (ordinal values 1-4), model-based estimates pulled from FB categories low-very high	FB**
Macro Covariates			
Real GDP per capita	Continuous	GDP per capita, in 2011 US dollars, annual measure, weighted by country historical catch	MHS
Population	Continuous	Annual measure, weighted by country historical catch	MHS
Historical country weight	Continuous, fractions (30)	Time-constant covariate indicating ratio of total tonnage the country has accounted for in given fishery (1950-2008)	SAU
Time covariates			
Linear time-trend	Discrete, 39 levels	Annual time trend – to capture long-term shifts towards ITQ implementation	N/A

Time dummies	Dummy (38)	Dummy for each year (excluding base), capturing any year-specific shocks in implementation	N/A
--------------	------------	--	-----

* *vulnerability score based on model (Cheung, Pitcher, & Pauly, 2005)*

** *price category estimate based on model (Sumaila et al., 2007)*

This set of variables is designed to capture most of the factors from Costello's high-dimensional specification while including more continuous, discrete and time-varying metrics. The specification remains reasonably high in dimensions – 484 observations per parameter – so should provide a good test for ML methods.

Results (Model 3)

ML methods beat the base Logit model across the board with largely tempered overfit levels:

Table 10: Propensity Score Fitting Results (Model 3)

Model	Optimal Tuning Parameters	MSE (valid. set)	Log-loss (valid. set)	Overfit Ratio (MSE)	Overfit Ratio (Log-Loss)
Logit	NA	0.00729	0.04041	1.00	0.99
Lasso	λ : 0.2	0.00504	0.02077	1.02	1.04
Ridge	λ : 0.1	0.00505	0.02084	1.02	1.04
Elastic-Net	$\lambda_1 = 0.19$ (Lasso) $\lambda_2 = 0.57$ (Ridge)	0.00504	0.02078	1.02	1.04
CART	Max depth: 4 Max number leaves: 15	0.00516	0.03084	1.01	1.17
Random Forest	Num of trees: 50 Max features: 15% Max depth: 15	0.00270	0.00927	3.29	2.45

Performance metrics (MSE, Log-loss) calculated on the held-out validation set (lower score is better fit). The overfit ratio represents performance score calculated on validation set divided by score calculated on the training set (large values are sign of overfitting).

The logit model again shows no overfitting, with an overfit ratio below 1 – evidence of underfitting if anything. ML results support this finding that the logit left signal uncaptured by

greatly improving performance. Shrinkage methods provide the most value as they boost performance without raising overfitting concerns.

Shrinkage methods resulted in a nearly two-fold improvement in performance versus the base logit model. Both Elastic-Net and Lasso tuning parameters optimized above the minimum tested level and thus engaged stronger regularization (traded more bias for variance) than in Models 1 and 2. Variable selection from the regularization resulted in 41 coefficients zeroed-out using Lasso and 31 using Elastic-Net – out of a total 124.

The single CART method results in a relatively shallow tree (4 layers, 15 nodes), which underperforms shrinkage and but improves upon logit performance to the tune of ~25%. Overfitting risk is lower on this dataset than the previous models (no overfit on MSE, 1.17 overfit ratio on log-odds), but it remains higher than the linear models.

Random Forest greatly outperforms all other methods, even reducing the error metrics versus shrinkage methods by ~50%. However, there is again a high degree of overfitting present since performance in validation was around 3x worse than the training results. The model's feature importance metrics show a more balanced set of covariates than previous models, but again Iceland is a key influence in predicting ITQ implementations:

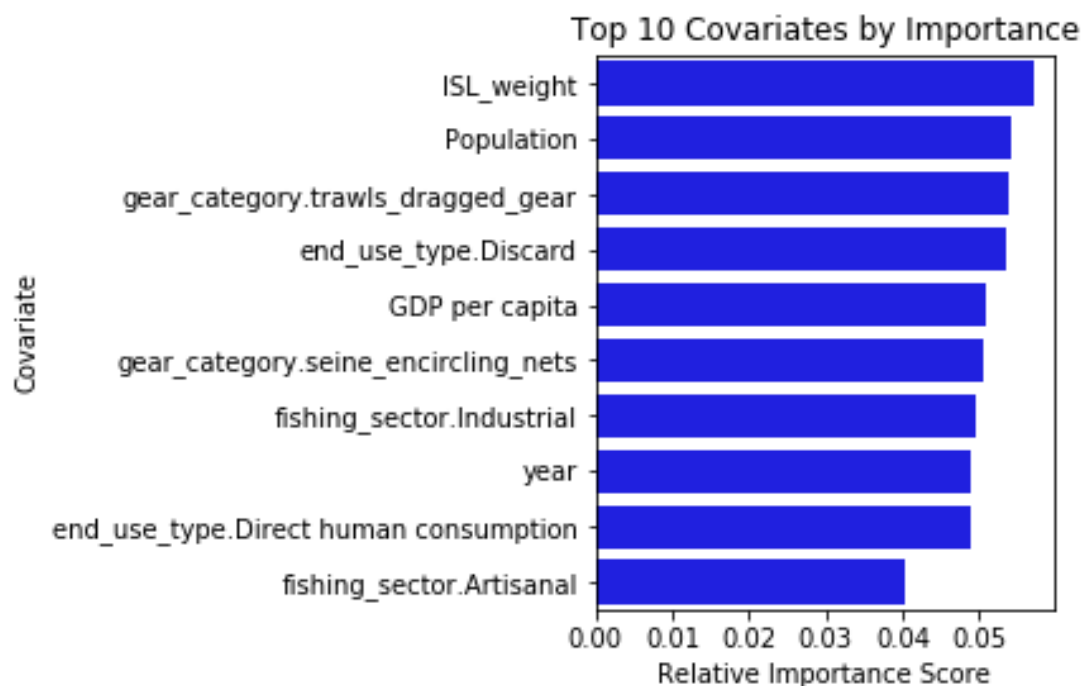


Figure 6: Random Forest Covariate Importance (Model 3)

These results suggest that the underlying data has sizeable variance, sparsity and no particularly strong nonlinear relationships. Lasso regression results in the best balance of performance and overfitting, proof that off-the-shelf ML can provide value in pre-processing prediction tasks for econometrics. Again, overfitting in the more advanced/nonlinear ML methods resulted in less generalizable models that may have suspect results. Different economics datasets may very well benefit from the tree methods with generalizable results, but in this application researchers are better off adhering to Occam's razor and using the less complication method.

ML Impact on Final Parameter Estimators

While ML methods have shown to improve accuracy of propensity score estimations under some conditions, it is unclear what effect this has on the final causal model. To test if the new methods resulted in any notable changes to the coefficient of interest (the causal effect of ITQs), I used each fitted model to predict propensity scores and then included these values in the final regressions according to the Costello et al. models (see Appendix 2 for full results of regressions).

ML estimates resulted in considerably different causal parameters versus the logit model estimates in some cases. For a sense of these effects, this section presents two examples aimed at answering two of the most important questions to economists:

- 1) Does overfitting in stage 1 impact the causal parameter estimates in stage 2?
- 2) Do improvements to propensity score estimates have a notable impact on causal parameter results?

Tests for question 1 serve as a check for whether practitioners should be concerned with overfitting in the pre-processing task. Tests for question 2 check if practitioners should care about the fit improvement ML offers in stage 1 (i.e. is there a value-add for the ultimate goal of causal inference).

Overfitting Estimator and Causal Results

The non-linear estimators from Model 1 provide an ideal test for whether overfitting in stage one of the propensity score process impacts causal parameters in stage two. Performance metrics on the validation set are similar across all fitting methods, so the level of overfitting is

the primary difference between estimators: none in Logit model, low levels in shrinkage models and moderately high levels in CART/RF models. Using Costello's base p-score causal model, there is no significant impact from overfitting on the final model. All predictors result in a causal effect of just below 0.009% reduction in collapse probability for each year of ITQ, significant at the 1% level:

Table 11: Regression Results for P-Score Base Model (Model 1 estimates)

	Logit Pcores (Model 1)	CART Pcores (Model 1)	Rand Forest Pcores (Model 1)
itq	-0.155*** (0.0126)	-0.157*** (0.0126)	-0.157*** (0.0126)
years_of_itq	-0.00895*** (0.00213)	-0.00867*** (0.00214)	-0.00869*** (0.00214)
time	0.00583*** (0.0000982)	0.00583*** (0.0000982)	0.00583*** (0.0000982)
p_score	0.164*** (0.0272)	0.165*** (0.0267)	0.166*** (0.0269)
Pseudo R2	0.0205	0.0205	0.0205
AIC	192749.4	192747.8	192747.7
BIC	192800.1	192798.4	192798.4
Observations	186843	186843	186843

P-scores included in regression were estimated using test Model 1 and method noted in column title

Coefficients reflect average marginal effect on probability of collapse

Standard errors in parentheses below coefficients, calculated using the delta method

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

The fixed-effect model tells a different story. Again, the logit and shrinkage methods result in similar causal effect coefficients (around -0.011 log-odds effect and not significantly different from 0 at 10%). However, CART/RF p-scores result in a causal coefficient (*years_of_itq*) around 4x larger than the other regressions and significant at the 5% level. The overfit models reduce standard error on the causal parameter, which is a potential indicator that the overfit model improved the matching quality.

Table 12: Regression Results for P-Score FE Model (Model 1 estimates)

	Logit Pcores (Model 1)	CART Pcores (Model 1)	Rand Forest Pcores (Model 1)
itq	0 (.)	0 (.)	0 (.)
years_of_itq	-0.0114 (0.0202)	-0.0438** (0.0195)	-0.0417** (0.0195)
time	0.0763*** (0.00106)	0.0755*** (0.00105)	0.0757*** (0.00105)
p_score	-5.563*** (0.448)	-4.010*** (0.405)	-4.204*** (0.409)
Fixed Effects	Yes	Yes	Yes
Pseudo R2	0.0801	0.0794	0.0795
AIC	68013.8	68069.6	68062.2
BIC	68042.0	68097.8	68090.4
Observations	89848	89848	89848

Fixed effects clustering by fishery

P-scores included in regression were estimated using test Model 1 and method noted in column title

Coefficients reflect log-odds of collapse (marginal effects not possible with fixed-effects model)

Standard errors in parentheses below coefficients

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

This result runs contrary to the Schuster et al. study that found “considerably inflated standard errors of effect estimates when using overfitted propensity score models,” (2016). It is possible that more severe overfitting levels have a different effect than that recorded in this test. The lightly supportable, though cautious, finding here is that overfitting in the first stage does not negatively impact the causal parameter estimation when restricted to moderately high levels.

Improved Predictions and Causal Results

Model 3 estimators serve as a good test for the impact of improved first-stage model fit on the final causal parameters using p-scores. Shrinkage methods significantly improved prediction accuracy over the logit model without any major changes in overfitting levels so comparing the two can identify if there is a parameter estimation impact. Using Costello’s base p-score model, the causal parameter is 4x larger using shrinkage estimates:

Table 13: Regression Results for P-Score Base Model (Model 3 estimates)

	Logit Pscores (Model 3)	Lasso Pscores (Model 3)	Ridge Pscores (Model 3)	Elast Net Pscores (Model 3)
itq	-0.237*** (0.0244)	-0.207*** (0.0254)	-0.205*** (0.0254)	-0.206*** (0.0254)
years_of_itq	0.00811*** (0.00271)	0.0318*** (0.00339)	0.0324*** (0.00340)	0.0320*** (0.00339)
time	0.00716*** (0.000159)	0.00766*** (0.000160)	0.00770*** (0.000160)	0.00767*** (0.000160)
p_score	- (dropped, lack of variance)	-1.654*** (0.106)	-1.801*** (0.111)	-1.705*** (0.108)
Pseudo R2	0.0307	0.0360	0.0365	0.0362
AIC	68169.2	67795.7	67763.6	67784.7
BIC	68205.2	67840.7	67808.6	67829.7
Observations	59969	59969	59969	59969

P-scores included in regression were estimated using test Model 3 and method noted in column title

Coefficients reflect average marginal effect on probability of collapse

Standard errors in parentheses below coefficients, calculated using the delta method

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

An interesting result in this case is that the propensity score completely dropped out when using the logit-estimated values due to a lack of variance in the scores. According to the logit estimator the observations are already properly balanced in terms of their likelihood of ITQ implementation (~50% odds for all):

Table 14: Propensity Score Estimates Distribution (Model 3)

	Logit	Lasso	Ridge	Elastic Net	CART	R. Forest
Min.	0.4996	0	0	0	0	0
1st Qu.	0.4997	0.0000002	0.0000007	0.0000003	0.0005410	0
Median	0.4999	0.0000097	0.000021	0.0000135	0.0005410	0
Mean	0.5	0.007381	0.007384	0.0073842	0.007372	0.007241
3rd Qu.	0.5002	0.000414	0.000570	0.0004668	0.006044	0
Max.	0.5004	0.992405	0.991068	0.9919689	1	1
SD	0.0002515	0.047630	0.046920	0.0474256	0.046689	0.067285

Distribution statistics of p-score estimates on the full OECD dataset, method of p-score prediction in column title

The better-fit models have a much broader range of conditional treatment probabilities, meaning they detect observable selection bias where the logit model does not. Since there is not overfitting, this difference is attributable to valid signal – meaning that the traditional method suffers from selection bias which ML is able to capture and fix. Ramifications on the

causal parameters for the fixed effects model are even more significant, changing the coefficient from negative to positive and highly significant:

Table 15: Regression Results for P-Score FE Model (Model 3 estimates)

	Logit Pcores (Model 3)	Lasso Pcores (Model 3)	Ridge Pcores (Model 3)	Elast Net Pcores (Model 3)
itq	0 (.)	0 (.)	0 (.)	0 (.)
years_of_itq	-0.0368* (0.0195)	0.283*** (0.0356)	0.291*** (0.0362)	0.286*** (0.0358)
time	0.0702*** (0.00156)	0.0746*** (0.00160)	0.0748*** (0.00160)	0.0747*** (0.00160)
p_score	0 (dropped, lack of variance)	-10.02*** (0.780)	-10.51*** (0.760)	-10.16*** (0.760)
Fixed Effects	Yes	Yes	Yes	Yes
Pseudo R2	0.0815	0.0910	0.0914	0.0911
AIC	26320.3	26050.3	26039.3	26047.3
BIC	26337.2	26075.6	26064.7	26072.6
Observations	34756	34756	34756	34756

Fixed effects clustering by fishery

P-scores included in regression were estimated using test Model 3 and method noted in column title

Coefficients reflect log-odds of collapse (marginal effects not possible with fixed-effects model)

Standard errors in parentheses below coefficients

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

The key finding here is that improved fit in stage 1 models can have major impacts on the causal parameter. When there is no overfitting, a better-fit model results in propensity score estimations that capture more valid selection bias and improve the causal interpretability of the final parameters. This supports the conclusion that by improving first-stage regression fit in a parameter estimation scenario, ML methods can improve final causal analysis and provide value in resource economics.

A Note on Causal Inference

There are two issues that challenge the causal interpretation of the methodologies used both in the original paper by Costello et al. (2008a) and the empirical application in this thesis. First, Lechner (2010) warns of bias in the causal effect estimation when using difference-in-differences with a logit regression, showing that the non-linear transformation results in a

violation of the parallel trends assumption. Second, there is evidence of the overlap assumption being violated in implementation of the propensity scores. Given the unbalanced nature of the treatment (ITQ implementation is rare), p-score estimates in model specifications – both from the Costello models and the test models – tend to have values at or near 0 (and 1 in some cases). Each of these issues is discussed further in Appendix 3.

Discussion

While results from the causal parameter tests in this chapter may be threatened by the aforementioned causal model issues, the primary finding of this empirical analysis remains valid: there is potential for ML to improve non-inference based econometric tasks under certain circumstances. ML methods were shown to improve the fit of prediction-based pre-processing tasks, conditional the underlying data having a moderate level of variance. In the case of low-variance data, increasing dimensionality from moderate to high was shown to have little impact on the effectiveness of ML. Further research is necessary to see if finding would hold for a higher-variance dataset.

Despite procedures designed around creating externally valid models, highly flexible ML algorithms proved to result in substantial overfitting. Though Random Forest resulted in significantly better model fits in-sample, it captured significant levels of noise in the training making out-of-sample predictions questionable. Shrinkage methods proved to offer an adequate level of flexibility in the resource economics application, performing in-line with or better than traditional methods in all cases, and resulting in preferable outcomes to the tree-based alternatives.

While an impact on causal results from using ML in pre-processing was detected, its exact nature and sensitivity to overfitting remains uncertain. Due to causal shortcomings in model design, the end-model results of these tests must be taken with caution. Practitioners should be careful to make sure results from ML models in pre-processing adhere to the underlying assumptions on the identification strategies used.

At the very least these results support the use of ML in economics for robustness analysis whenever there are prediction tasks that do not require formal marginal parameters or confidence intervals. Given the prediction abilities of ML illustrated in the test of Model 3,

applications involving prediction like synthetic control and matching look to be great areas for further research.

Chapter 6 Conclusion

This thesis aims to answer two questions: whether traditional econometric tools can be improved by integrating machine learning methods, and whether these improvements have practical value for the goal of solving resource economics problems. To answer these questions, an implicit step was required to identify the econometrics tasks that machine learning is most likely to benefit. In reviewing existing literature, I found that ML outperforms traditional methods in prediction tasks and, since off-the-shelf ML is not well suited for inference, I determined the best application for resource economics would be in pre-processing (non-inference) prediction tasks. To gather evidence in response the two primary questions, I used machine learning on one such task – propensity score estimation – in an empirical application to fisheries using a paper by Costello et al. (2008a).

Empirical test results give strong evidence that ML does improve upon traditional methods' prediction performance when there is moderate variance in the underlying data and a reasonably high number of covariates. This result proved to be of practical value for resource economics, as there is direct evidence that ML models identified and fitted valid selection bias that was otherwise missed by the traditional logit model.

ML methods were not without limitations however – empirical results show that for low-variance data they do not significantly improve predictions and have a tendency towards overfitting when there are high dimensions. When higher variance covariates were used, overfitting levels fell. One limitation of this study is that moderate-variance data was available for only 124 covariates; testing on a higher-dimensional dataset would improve the robustness of the results.

As theory would predict, overfitting was shown to be highest for the most flexible ML algorithms tested (CART and Random Forest) and under the highest-dimensional dataset tested. Test results show signs that these methods fit a significant amount of noise from the training set into their models, which threatens generalizability of their results. The analysis was unable to reveal the consequences of this overfitting for the final causal model. Since overfitting is the biggest threat to ML results in practice, this represents an important area for future research. Still a higher-level conclusion from the results is that more complexity in the model is not always better.

The main limitation of this empirical test was an inability to reliably measure how improved predictions in stage 1 translated to changes in causal results in stage 2. This was the unfortunate result of likely violations in propensity score assumptions and issues in the final causal model that challenge a causal interpretation of stage 2 results. While this leaves some important questions unanswered, it serves as a good reminder for practitioners that regardless of the complexity or performance of models plugged into econometric tasks, the foundational assumptions must still hold. ML methods should in no way be treated as a silver bullet allowing researchers to be lazy elsewhere.

Findings from this thesis support the use of off-the-shelf ML methods to improve pre-processing prediction tasks for resource economics, for example propensity score matching and two-stage instrumental variable tasks. Since the downstream effects of overfitting are not yet fully understood, however, immediate application of these methods is best restricted to use for robustness analysis. The empirical application in this thesis provides a potential best practice method for checking the robustness of these first-stage regressions: fit the five machine learning models and the traditional one, calculating model fit and overfit metrics to evaluate the credibility of each result.

An alternate approach that requires less steps for researchers would be to simply implement the Elastic-Net regression. A grid search using this model can test specifications for Lasso, Ridge, Lasso/Ridge mixes, and traditional OLS/logit regressions all in a single run. While shrinkage methods do not have the extreme flexibility in fitting that tree methods do, the empirical test proved them more resilient to overfitting and they performed roughly as well or better than traditional methods in every test. They also provide a simple, intuitive introduction to ML methods due to their similarity to OLS. This implementation offers an efficient solution to robustness analysis and would be relatively painless for practitioners.

This thesis contributes to econometrics literature by providing an empirical test for machine learning in non-inference tasks and, to my knowledge, provides the first general discussion of machine learning for resource economics problems. The results of this thesis encourage further research on machine learning both for resource economics and economics more generally. In particular, the synthetic control model is one area that is underexplored for machine learning applications and potentially offers significant value for empirical resource economics. Research on the effect of overfitting in the first-stage predictions on second-

stage causal model results would fill a critical gap in this thesis' findings and could foster more confident implementation of machine learning methods in practice.

While machine learning integration for resource economics may not have the “revolutionary” impact that it has made on the technology and business world, this thesis illustrates its serious potential. The field has been underexplored to date, but causal analysis using ML is gaining attention every day and the literature is growing. Economics practitioners should pay attention and keep an open mind, so the next three decades may see even more progress than those previous.

References

- Abadie, A., & Gardeazabal, J. (2003). The Economic Costs of Conflict: A Case Study of the Basque Country. *American Economic Review*, 93(1), 113-132. doi:10.1257/000282803321455188
- Abadie, A., Diamond, A., & Hainmueller, J. (2010, June). Synthetic Control Methods for Comparative Case Studies: Estimating the Effect of California's Tobacco Control Program. *Journal of the American Statistical Association*, 105(490), 493-505. doi:10.1198/jasa.2009.ap08746
- Angrist, J. D., & Pischke, J.-S. (2010). The Credibility Revolution in Empirical Economics: How Better Research Design is Taking the Con out of Econometrics. *Journal of Economic Perspectives*, 24(2), 3-30. doi:10.1257/jep.24.2.3
- Angrist, J., & Pischke, S. (2009). *Mostly Harmless Econometrics*. Princeton, NJ: Princeton University Press. doi:10.2307/j.ctvc4j72
- Athey, S. (2018a). The impact of machine learning on economics. In A. Agrawal, J. Gans, & A. Goldfarb (Eds.), *The economics of artificial intelligence: an agenda* (pp. 507-547). University of Chicago Press.
- Athey, S. (2018b). An Introduction to Regression Trees (CART) [PowerPoint slides]. Retrieved from https://drive.google.com/drive/folders/1_h0gepokalfe-dXzoSsS7scc9EE3-DTl
- Athey, S., & Imbens, G. W. (2016). Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27), 7353-7360. doi:10.1073/pnas.1510489113
- Athey, S., & Imbens, G. W. (2017). The state of applied econometrics: causality and policy evaluation. *The Journal of Economic Perspectives*, 31(2), 3-32. doi:10.1257/jep.31.2.3
- Athey, S., & Imbens, G. W. (2019). *Machine learning methods economists should know about*. Stanford GSB Working Paper 3776. Retrieved from <https://arxiv.org/pdf/1903.10075.pdf>

- Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized Random Forests. *The Annals of Statistics*, 1148-1178.
- Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized Random Forests. *The Annals of Statistics*, 47(2), 1148-1178. doi:<https://doi.org/10.1214/18-AOS1709>
- Athey, S., Wager, S., & Muhlbach, N. N. (2019, April). Exploring Causal Inference in Experimental and Observational Studies - Part 1. Retrieved from <https://drive.google.com/drive/folders/10hj9yZ8tmO9VpkkwzUdliXntDFnsWpsZ>
- Belloni, A., Chernozhukov, V., & Hansen, C. (2014). High-dimensional methods and inference on structural and treatment effects. *Journal of Economic Perspectives*, 28(2), 29-50. doi:10.1257/jep.28.2.29
- Ben-Michael, E., Feller, A., & Rothstein, J. (2018). The Augmented Synthetic Control Method. Retrieved from <https://arxiv.org/abs/1811.04170>
- Bolt, J., Inklaar, R., de Jong, H., & van Zanden, J. (2018). *Rebasing 'Maddison': new income comparisons and the shape of long-run economic development*. Maddison Project Working paper 10.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123-140. doi:10.1023/A:1018054314350
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.
- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and Regression Trees*. Boca Raton, FL: Chapman & Hall/CRC.
- Chamberlain, G. (1980). Analysis of Covariance with Qualitative Data. *Review of Economic Studies*, 47, 225-238. doi:10.3386/w0325
- Cheung, W. W., Pitcher, T. J., & Pauly, D. (2005). A fuzzy logic expert system to estimate intrinsic extinction vulnerabilities of marine fishes to fishing. *Biological Conservation*, 124(1), 97-111. doi:10.1016/j.biocon.2005.01.017
- Costello, C., Gaines, S. D., & Lynham, J. (2008a). Can catch shares prevent fisheries collapse? *Science*, 321, 1678-1680. doi:10.1126/science.1159478

-
- Costello, C., Gaines, S. D., & Lynham, J. (2008b). *Fisheries Managed by Individual Transferable Quotas (ITQs) [PDF file]*. Retrieved from http://fiesta.bren.ucsb.edu/~costello/research/CatchShares/ITQ_Managed_Fisheries_List_Map.pdf
- Coupé, T. (2005). Bias in Conditional and Unconditional Fixed Effects Logit Estimation: A Correction. *Political Analysis*, 13(3), 292-295. doi:10.1093/pan/mpi019
- Difference-in-Difference Estimation*. (n.d.). Retrieved from Columbia Mailman School of Public Health: <https://www.mailman.columbia.edu/research/population-health-methods/difference-difference-estimation>
- Einav, L., & Levin, J. D. (2013). The data revolution and economic analysis. *NBER Working Paper 19035*. doi:10.3386/w19035
- Elfron, B., & Hastie, T. (2016). *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science*. Cambridge University Press.
- Froese, R., & Pauly, D. (2019). FishBase [worldwide web electronic publication]. Retrieved from <https://www.fishbase.org>.
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. Sebastopol, CA: O'Reilly Media, Inc.
- Guo, R., Cheng, L., Li, J., Hahn, P. R., & Liu, H. (2010, March). A survey of learning causality with data: problems and methods. *Transactions on the Web*, 9(4). Retrieved from <https://arxiv.org/abs/1809.09337>
- Hajage, D., Tubach, F., Steg, P. G., D. L., & Rycke, Y. D. (2016, March). On the use of propensity scores in case of rare exposure. *BMC Medical Research Methodology*, 16. doi:10.1186/s12874-016-0135-1
- Hansen, C., & Kozbur, D. (2014, October). Instrumental variables estimation with many weak instruments using regularized JIVE. *Journal of Econometrics*, 182(2), 290-308. doi:10.1016/j.jeconom.2014.04.022
- Hartford, J., Lewis, G., Leyton-Brown, K., & Taddy, M. (2017). Deep IV: A Flexible Approach for Counterfactual Prediction. In D. Precup, & Y. W. Teh (Ed.), *Proceedings*

-
- of the 34th International Conference on Machine Learning* (pp. 1414-1423). Sydney: PMLR. Retrieved from <http://proceedings.mlr.press/v70/hartford17a.html>
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer.
- Holland, P. W. (1986). Statistics and Causal Inference. *Journal of the American Statistical Association*, 81(396), 945-960. doi:10.1080/01621459.1986.10478354
- Imbens, G. W. (2019). *Potential Outcome and Directed Acyclic Graph Approaches to Causality: Relevance for Empirical Practice in Economics*. NBER Working Paper No. 26104. doi:10.3386/w26104
- Imbens, G., & Rubin, D. (2015). *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge: Cambridge University Press. doi:10.1017/CBO9781139025751
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *Introduction to Statistical Learning: With Applications in R*. New York, NY: Springer.
- Kinn, D. (2018). Synthetic Control Methods and Big Data. Retrieved from <https://arxiv.org/abs/1803.00096>
- Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10), 4156–4165. doi:10.1073/pnas.1804597116
- Lechner, M. (2010). *The Estimation of Causal Effects by Difference-in-Difference Methods*. University of St. Gallen Department of Economics working paper series 2010 2010-28, Department of Economics, University of St. Gallen.
- Maddison Project. (2018). Maddison Project Database [database]. Retrieved from <https://www.rug.nl/ggdc/historicaldevelopment/maddison>.
- Mullainathan, S., & Spiess, J. (2017). Machine learning: an applied econometric approach. *Journal of Economic Perspectives*, 31(2), 87-106. doi:10.1257/jep.31.2.87

-
- Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: The MIT Press.
- Nie, X., & Wager, S. (2017). *Quasi-Oracle Estimation of Heterogeneous Treatment Effects*. Retrieved from <https://arxiv.org/abs/1712.04912>
- Palomares, M., & Pauly, D. (2019). SeaLifeBase [worldwide web electronic publication]. Retrieved from <https://www.sealifebase.org>.
- Pauly, D., & Zeller, D. (2015). Sea Around Us [concepts, design and data]. Retrieved from <http://www.seaaroundus.org/>
- Pearl, J. (2009). *Causality*. Cambridge University Press. doi:10.1017/CBO9780511803161
- Reimera, M. N., & Haynieb, A. C. (2018, March). Mechanisms matter for evaluating the economic impacts of marine reserves. *Journal of Environmental Economics and Management*, 88, 427-446. doi:10.1016/j.jeem.2018.01.009
- Rosenbaum, P. R. (2007). Interference Between Units in Randomized Experiments. *Journal of the American Statistical Association*, 101(470), 191-199. doi:10.1198/016214506000001112
- Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 688-701. doi:10.1037/h0037350
- Rubin, D. B. (1978). Bayesian Inference for Causal Effects: The Role of Randomization. *Annals of Statistics*, 6(1), 34-58.
- Schuster, T., Lowe, W. K., & Platt, R. W. (2016). Propensity score model overfitting led to inflated variance of estimated odds ratios. *Journal of Clinical Epidemiology*, 80, 97-106. doi:10.1016/j.jclinepi.2016.05.017
- Sills, E. O., Herrera, D., Kirkpatrick, A. J., Amintas Brandão, J., Dickson, R., Hall, S., . . . Pfaff, A. (2015). Estimating the Impacts of Local Policy Innovation: The Synthetic Control Method Applied to Tropical Deforestation. *PLoS One*, 10(7). doi:10.1371/journal.pone.0132590

- Smith, B. (2015). The resource curse exorcised: Evidence from a panel of countries. *Journal of Development Economics*, 116, 57-73. doi:10.1016/j.jdeveco.2015.04.001
- Sumaila, R. U., Marsden, D. A., Watson, R., & Pauly, D. (2007). A global ex-vessel fish price database: construction and applications. *Journal of Bioeconomics*, 9(1), 38-51. doi:10.1007/s10818-007-9015-4
- Varian, H. R. (2014). Big data: new tricks for econometrics. *Journal of Economic Perspectives*, 28(2), 3-28. doi:10.1257/jep.28.2.3
- Worm, B., Barbier, E. B., Beaumont, N., Duffy, J. E., Folke, C., Halpern, B. S., . . . Watson, R. (2006). Impacts of Biodiversity Loss on Ocean Ecosystem Services. *Science*, 314(5800), 787-790. doi:10.1126/science.1132294
- WoRMS Editorial Board. (2019). World Register of Marine Species. doi:10.14284/170
- Zigler, C. M., & Dominici, F. (2014). Uncertainty in Propensity Score Estimation: Bayesian Methods for Variable Selection and Model Averaged Causal Effects. *J Am Stat Assoc*, 109(505), 95-107. doi:10.1080/01621459.2013.869498

Software Packages Used⁷:

Language	Packages	Use
R	Tidyverse	Data cleaning/wrangling Data exploration/ visualizations
	rfishbase	Data gathering (API)
	Haven	Exporting to Stata format
Python	scikit-learn	Machine learning modeling and analysis
	Pandas	Data cleaning/wrangling
	Seaborn	Data visualizations
	Pickle	Model result storage
Stata	margins	Econometrics modeling (causal model regressions)
	estout	Constructing regression tables
Julia	DataFrames	Data wrangling
	GLM	Fitting logit p-score models (for replication task only)

⁷ Copy of code used in analysis for this thesis available on request

List of Tables and Figures

Tables:

Table 1: Comparison: Econometrics vs Machine Learning	13
Table 2: Comparison of Selected ML Algorithms	29
Table 3: Comparison of Dataset Descriptive Statistics	38
Table 4: Parameters Tuned by Cross Validation	47
Table 5: Estimated Coefficients from First-Cut Model (Dataset 1)	50
Table 6: Propensity Score Fitting Results (Model 1)	51
Table 7: Propensity Score Fitting Results (Model 2)	54
Table 8: Estimated Coefficients from First-Cut Model (OECD sample)	56
Table 9: Description of Covariates in Model 3	57
Table 10: Propensity Score Fitting Results (Model 3)	59
Table 11: Regression Results for P-Score Base Model (Model 1 estimates)	62
Table 12: Regression Results for P-Score FE Model (Model 1 estimates)	63
Table 13: Regression Results for P-Score Base Model (Model 3 estimates)	64
Table 14: Propensity Score Estimates Distribution (Model 3)	64
Table 15: Regression Results for P-Score FE Model (Model 3 estimates)	65
Table 16: Regression Results for First-Cut Model Replication	80
Table 17: Regression Results for Base Model Replication	80
Table 18: Regression Results for P-Score Base Model Replication	81
Table 19: Regression Results for Fixed Effects Model Replication	82
Table 20: Regression Results for P-Score Base Model (Model 1 estimates, pt. 1)	83
Table 21: Regression Results for P-Score Base Model (Model 1 estimates, pt. 2)	83
Table 22: Regression Results for P-Score FE Model (Model 1 estimates, pt. 1)	84
Table 23: Regression Results for P-Score FE Model (Model 1 estimates, pt. 2)	85
Table 24: Regression Results for P-Score Base Model (Model 2 estimates, pt. 1)	85
Table 25: Regression Results for P-Score Base Model (Model 2 estimates, pt. 2)	86
Table 26: Regression Results for P-Score FE Model (Model 2 estimates, pt. 1)	86
Table 27: Regression Results for P-Score FE Model (Model 2 estimates, pt. 2)	87
Table 28: Regression Results for P-Score Base Model (Model 3 estimates, pt. 1)	88
Table 29: Regression Results for P-Score Base Model (Model 3 estimates, pt. 2)	88
Table 30: Regression Results for P-Score FE Model (Model 3 estimates, pt. 1)	89

Table 31: Regression Results for P-Score FE Model (Model 3 estimates, pt. 2).....	89
Table 32: Model 3 Covariate Details.....	105

Figures:

Figure 1: Example of Decision Tree covariate space partitioning	20
Figure 2: Collapse Trends in Recreated Dataset	38
Figure 3: Validation Curve Example (Model 1, CART, Max-Depth)	48
Figure 4: Random Forest Covariate Importance (Model 1).....	52
Figure 5: Random Forest Covariate Importance (Model 2).....	55
Figure 6: Random Forest Covariate Importance (Model 3).....	60
Figure 7: Propensity Score Distributions – Full Recreated Dataset.....	100
Figure 8: Propensity Score Distributions – Recr. Dataset (Treated Group).....	101
Figure 9: Validation curves from parameter search (Model 1, CART)	102
Figure 10: Validation curves from parameter search (Model 1, Rand. Forest)	102
Figure 11: Validation curves from parameter search (Model 2, CART)	103
Figure 12: Validation curves from parameter search (Model 2, Rand. Forest)	103
Figure 13: Validation curves from parameter search (Model 3, CART)	104
Figure 14: Validation curves from parameter search (Model 3, Rand. Forest)	104

Appendix 1: Replication Results

This appendix reports regression results from my replication of the Costello et al. causal models using the recreated dataset side by side with reported results from the original study (Costello, Gaines, & Lynham, 2008a). Results show similar trends and significance levels in general.

First-Cut Model Results

Table 16: Regression Results for First-Cut Model Replication

Variable	Recreated	Costello
ITQ _i	-0.0460*** (0.00560)	-0.0428*** (0.00505)
ITQ_imp _t	0.0387*** (0.0141)	0.0090 (0.01255)
ITQ _i * ITQ_imp _t	-0.175*** (0.0135)	-0.1367*** (0.01188)
time _t	0.00484*** (0.000274)	0.0049*** (0.00025)
Intercept	0.0199*** (0.00627)	0.0096* (0.00627)
R-squared	0.92	0.92
N	108	108

Standard errors in parentheses below coefficients

Dependent variable was percent of fisheries collapsed.

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Base Model Results

Table 17: Regression Results for Base Model Replication

Variable	Recreated Logit	Costello Logit
ITQ _i	-0.103*** (0.00875)	-0.0706*** (0.00490)
years_of_itq _{it}	-0.00682*** (0.00159)	-0.0049*** (0.00136)
time _t	0.00623*** (0.0000490)	0.0054*** (0.00004)

Pseudo R2	0.0715	0.06
AIC	229,880.6	251,696.6
BIC	229,922.7	251,739.1
Observations	273,548	302,852

Coefficients reflect average marginal effect on probability of collapse

Standard errors in parentheses below coefficients, calculated using the delta method

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Propensity Score Model Results

Table 18: Regression Results for P-Score Base Model Replication

Variable	Recr. Pscore (LME)	Costello Pscore (LME)	Recr. Pscore (Genus)	Costello Pscore (Genus)	Recr. Pscore (species)	Costello Pscore (species)
ITQ_i	-0.103*** (0.00875)	-0.0741*** (0.00428)	-0.103*** (0.00875)	-0.0679*** (0.00443)	-0.103*** (0.00875)	- 0.0687*** (0.00441)
$years_of_itq_{it}$	-0.00682*** (0.00159)	-0.0037*** (0.00137)	-0.00682*** (0.00159)	-0.0054*** (0.00136)	- 0.00682*** (0.00159)	- 0.0051*** (0.00139)
$time_t$	0.00623*** (0.0000490)	0.0054*** (0.00004)	0.00623*** (0.0000490)	0.0054*** (0.00004)	0.00623*** (0.0000490)	0.0054*** (0.00004)
$Pscore$	-0.0261 (0.0236)	Not Reported	0.0154 (0.0262)	Not Reported	0.0198 (0.0214)	Not Reported
<i>Intercept</i>	NA	NA	NA	NA	NA	NA
Pseudo R2	0.0715	0.06	0.0715	0.06	0.0715	0.06
AIC	229881.4	251,580.60	229882.3	251,575.60	229881.8	251,494.6
BIC	229934.0	251,931.1	229934.9	251,926.1	229934.4	251,845.1
Observations	273548	302,852	273548	302,852	273548	302,852

P-scores included in regression were estimated using logit regression on categorical variable in column title

Coefficients reflect average marginal effect on probability of collapse

Standard errors in parentheses below coefficients, calculated using the delta method

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Fixed Effects Model Results

Table 19: Regression Results for Fixed Effects Model Replication

	Recr. FE (full sample)	Costello FE (full sample)	Recr FE (ITQ only sample)	Costello FE (ITQ only sample)
ITQ_i	-	-	-	-
$years_of_itq_{it}$	-0.119*** (0.0179)	-0.1206*** (0.01363)	-0.0259 (0.0230)	-0.0123*** (0.00184)
$time_t$	0.0763*** (0.000657)	0.0888*** (0.00063)	0.0336*** (0.00663)	0.00788*** (0.00000304)
<i>intercept</i>	NA	NA	NA	NA
Fixed Effects	Yes	Yes	Yes	Yes
Pseudo R2	0.157	0.18	0.0291	0.10
AIC	93,739.0	123,430.1	1225.8	1,448.551
BIC	93,758.7	123,450.4	1236.7	1,460.052
Observations	134,338	186,554	1760	2,322

Fixed effects clustering by fishery

Coefficients reflect log-odds of collapse (marginal effects not possible with fixed-effects model)

Standard errors in parentheses below coefficients

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Appendix 2: P-Score Application Regression Results

This appendix reports full regression results from the empirical tests in Chapter 5. Causal model specifications are based on Costello et al. (2018a), fitted using the datasets and propensity scores estimates discussed in Chapter 5.

Model 1 P-Scores – Causal Regression Results

Propensity Score Base Model Results:

Table 20: Regression Results for P-Score Base Model (Model 1 estimates, pt. 1)

	Logit Pcores (Model 1)	Lasso Pcores (Model 1)	Ridge Pcores (Model 1)	Elast Net Pcores (Model 1)
itq	-0.155*** (0.0126)	-0.155*** (0.0126)	-0.155*** (0.0126)	-0.155*** (0.0126)
years_of_itq	-0.00895*** (0.00213)	-0.00895*** (0.00213)	-0.00895*** (0.00213)	-0.00895*** (0.00213)
time	0.00583*** (0.0000982)	0.00583*** (0.0000982)	0.00583*** (0.0000982)	0.00583*** (0.0000982)
p_score	0.164*** (0.0272)	0.164*** (0.0273)	0.165*** (0.0274)	0.164*** (0.0273)
Pseudo R2	0.0205	0.0205	0.0205	0.0205
AIC	192749.4	192749.6	192749.4	192749.6
BIC	192800.1	192800.3	192800.1	192800.3
Observations	186843	186843	186843	186843

P-scores included in regression were estimated using test Model 1 and method noted in column title

Coefficients reflect average marginal effect on probability of collapse

Standard errors in parentheses below coefficients, calculated using the delta method

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Table 21: Regression Results for P-Score Base Model (Model 1 estimates, pt. 2)

	Logit Pcores (Model 1)	CART Pcores (Model 1)	Rand Forest Pcores (Model 1)
itq	-0.155*** (0.0126)	-0.157*** (0.0126)	-0.157*** (0.0126)
years_of_itq	-0.00895*** (0.00213)	-0.00867*** (0.00214)	-0.00869*** (0.00214)

time	0.00583*** (0.0000982)	0.00583*** (0.0000982)	0.00583*** (0.0000982)
p_score	0.164*** (0.0272)	0.165*** (0.0267)	0.166*** (0.0269)
Pseudo R2	0.0205	0.0205	0.0205
AIC	192749.4	192747.8	192747.7
BIC	192800.1	192798.4	192798.4
Observations	186843	186843	186843

P-scores included in regression were estimated using test Model 1 and method noted in column title
Coefficients reflect average marginal effect on probability of collapse
Standard errors in parentheses below coefficients, calculated using the delta method
* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Propensity Score Fixed Effects Model Results

Table 22: Regression Results for P-Score FE Model (Model 1 estimates, pt. 1)

	Logit Pcores (Model 1)	Lasso Pcores (Model 1)	Ridge Pcores (Model 1)	Elast Net Pcores (Model 1)
itq	0 (.)	0 (.)	0 (.)	0 (.)
years_of_itq	-0.0114 (0.0202)	-0.0110 (0.0202)	-0.0110 (0.0202)	-0.0110 (0.0202)
time	0.0763*** (0.00106)	0.0763*** (0.00106)	0.0763*** (0.00106)	0.0763*** (0.00106)
p_score	-5.563*** (0.448)	-5.585*** (0.448)	-5.621*** (0.451)	-5.585*** (0.448)
Fixed Effects	Yes	Yes	Yes	Yes
AIC	68013.8	68013.1	68013.1	68013.1
BIC	68042.0	68041.3	68041.3	68041.3
Observations	89848	89848	89848	89848

Fixed effects clustering by fishery

P-scores included in regression were estimated using test Model 1 and method noted in column title

Coefficients reflect log-odds of collapse (marginal effects not possible with fixed-effects model)

Standard errors in parentheses below coefficients

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Table 23: Regression Results for P-Score FE Model (Model 1 estimates, pt. 2)

	Logit Pcores (Model 1)	CART Pcores (Model 1)	Rand Forest Pcores (Model 1)
itq	0 (.)	0 (.)	0 (.)
years_of_itq	-0.0114 (0.0202)	-0.0438** (0.0195)	-0.0417** (0.0195)
time	0.0763*** (0.00106)	0.0755*** (0.00105)	0.0757*** (0.00105)
p_score	-5.563*** (0.448)	-4.010*** (0.405)	-4.204*** (0.409)
Fixed Effects	Yes	Yes	Yes
Pseudo R2	0.0801	0.0794	0.0795
AIC	68013.8	68069.6	68062.2
BIC	68042.0	68097.8	68090.4
Observations	89848	89848	89848

Fixed effects clustering by fishery

P-scores included in regression were estimated using test Model 1 and method noted in column title

Coefficients reflect log-odds of collapse (marginal effects not possible with fixed-effects model)

Standard errors in parentheses below coefficients

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Model 2 P-Scores – Causal Regression Results

Propensity Score Base Model Results:

Table 24: Regression Results for P-Score Base Model (Model 2 estimates, pt. 1)

	Logit Pcores (Model 2)	Lasso Pcores (Model 2)	Ridge Pcores (Model 2)	Elast Net Pcores (Model 2)
itq	-0.153*** (0.0131)	-0.153*** (0.0131)	-0.153*** (0.0131)	-0.153*** (0.0131)
years_of_itq	-0.00947*** (0.00253)	-0.00934*** (0.00254)	-0.00947*** (0.00253)	-0.00934*** (0.00254)
time	0.00590*** (0.0000975)	0.00590*** (0.0000975)	0.00590*** (0.0000975)	0.00590*** (0.0000975)
p_score	0.0584**	0.0563**	0.0608**	0.0563**

	(0.0267)	(0.0270)	(0.0278)	(0.0270)
Pseudo R2	0.0204	0.0204	0.0204	0.0204
AIC	192780.0	192780.4	192780.0	192780.4
BIC	192830.7	192831.1	192830.7	192831.1
Observations	186843	186843	186843	186843

P-scores included in regression were estimated using test Model 2 and method noted in column title
Coefficients reflect average marginal effect on probability of collapse
Standard errors in parentheses below coefficients, calculated using the delta method
* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Table 25: Regression Results for P-Score Base Model (Model 2 estimates, pt. 2)

	Logit Pcores (Model 2)	CART Pcores (Model 2)	Rand Forest Pcores (Model 2)
itq	-0.153*** (0.0131)	-0.131*** (0.0126)	-0.153*** (0.0136)
years_of_itq	-0.00947*** (0.00253)	0.00154 (0.00257)	-0.00936*** (0.00296)
time	0.00590*** (0.0000975)	0.00593*** (0.0000972)	0.00591*** (0.0000973)
p_score	0.0584** (0.0267)	-0.145*** (0.0313)	0.0444 (0.0307)
Pseudo R2	0.0204	0.0205	0.0204
AIC	192780.0	192762.0	192782.6
BIC	192830.7	192812.7	192833.3
Observations	186843	186843	186843

P-scores included in regression were estimated using test Model 2 and method noted in column title
Coefficients reflect average marginal effect on probability of collapse
Standard errors in parentheses below coefficients, calculated using the delta method
* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Propensity Score Fixed Effects Model Results

Table 26: Regression Results for P-Score FE Model (Model 2 estimates, pt. 1)

	Logit Pcores (Model 2)	Lasso Pcores (Model 2)	Ridge Pcores (Model 2)	Elast Net Pcores (Model 2)
itq	0 (.)	0 (.)	0 (.)	0 (.)
years_of_itq	-0.00946 (0.0253)	-0.00747 (0.0253)	-0.00285 (0.0253)	-0.00747 (0.0253)

time	0.0741*** (0.00103)	0.0741*** (0.00103)	0.0742*** (0.00104)	0.0741*** (0.00103)
p_score	-1.726*** (0.337)	-1.790*** (0.342)	-1.986*** (0.359)	-1.790*** (0.342)
Fixed Effects	Yes	Yes	Yes	Yes
Pseudo R2	0.0784	0.0784	0.0785	0.0784
AIC	68140.8	68139.8	68136.6	68139.8
BIC	68169.0	68168.1	68164.8	68168.1
Observations	89848	89848	89848	89848

Fixed effects clustering by fishery

P-scores included in regression were estimated using test Model 2 and method noted in column title

Coefficients reflect log-odds of collapse (marginal effects not possible with fixed-effects model)

Standard errors in parentheses below coefficients

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Table 27: Regression Results for P-Score FE Model (Model 2 estimates, pt. 2)

	Logit Pcores (Model 2)	CART Pcores (Model 2)	Rand Forest Pcores (Model 2)
itq	0 (.)	0 (.)	0 (.)
years_of_itq	-0.00946 (0.0253)	0.0108 (0.0248)	0.00401 (0.0276)
time	0.0741*** (0.00103)	0.0738*** (0.00103)	0.0738*** (0.00103)
p_score	-1.726*** (0.337)	-2.088*** (0.308)	-1.456*** (0.299)
Fixed Effects	No	Yes	Yes
Pseudo R2	0.0784	0.0787	0.0784
AIC	68140.8	68116.2	68141.9
BIC	68169.0	68144.4	68170.2
Observations	89848	89848	89848

Fixed effects clustering by fishery

P-scores included in regression were estimated using test Model 2 and method noted in column title

Coefficients reflect log-odds of collapse (marginal effects not possible with fixed-effects model)

Standard errors in parentheses below coefficients

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Model 3 P-Scores – Causal Regression Results

Propensity Score Base Model Results:

Table 28: Regression Results for P-Score Base Model (Model 3 estimates, pt. 1)

	Logit Pcores (Model 3)	Lasso Pcores (Model 3)	Ridge Pcores (Model 3)	Elast Net Pcores (Model 3)
itq	-0.237*** (0.0244)	-0.207*** (0.0254)	-0.205*** (0.0254)	-0.206*** (0.0254)
years_of_itq	0.00811*** (0.00271)	0.0318*** (0.00339)	0.0324*** (0.00340)	0.0320*** (0.00339)
time	0.00716*** (0.000159)	0.00766*** (0.000160)	0.00770*** (0.000160)	0.00767*** (0.000160)
p_score	- (dropped, lack of variance)	-1.654*** (0.106)	-1.801*** (0.111)	-1.705*** (0.108)
Pseudo R2	0.0307	0.0360	0.0365	0.0362
AIC	68169.2	67795.7	67763.6	67784.7
BIC	68205.2	67840.7	67808.6	67829.7
Observations	59969	59969	59969	59969

P-scores included in regression were estimated using test Model 3 and method noted in column title

Coefficients reflect average marginal effect on probability of collapse

Standard errors in parentheses below coefficients, calculated using the delta method

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Table 29: Regression Results for P-Score Base Model (Model 3 estimates, pt. 2)

	Logit Pcores (Model 3)	CART Pcores (Model 3)	Rand Forest Pcores (Model 3)
itq	-0.237*** (0.0244)	-0.225*** (0.0266)	-0.188*** (0.0245)
years_of_itq	0.00811*** (0.00271)	0.03174*** (0.00365)	0.0241*** (0.00385)
time	0.00716*** (0.000159)	0.00779*** (0.000160)	0.00722*** (0.000159)
p_score	- (dropped, lack of variance)	-4.328*** (0.21190)	-0.367*** (0.0647)

Pseudo R2	0.0307	0.0393	0.0312
AIC	68169.2	67550.69.	68134.1
BIC	68205.2	67605.7	68179.1
Observations	59969	59969	59969

P-scores included in regression were estimated using test Model 3 and method noted in column title

Coefficients reflect average marginal effect on probability of collapse

Standard errors in parentheses below coefficients, calculated using the delta method

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Propensity Score Fixed Effects Model Results

Table 30: Regression Results for P-Score FE Model (Model 3 estimates, pt. 1)

	Logit Pcores (Model 3)	Lasso Pcores (Model 3)	Ridge Pcores (Model 3)	Elast Net Pcores (Model 3)
itq	0 (.)	0 (.)	0 (.)	0 (.)
years_of_itq	-0.0368* (0.0195)	0.283*** (0.0356)	0.291*** (0.0362)	0.286*** (0.0358)
time	0.0702*** (0.00156)	0.0746*** (0.00160)	0.0748*** (0.00160)	0.0747*** (0.00160)
p_score	0 (dropped, lack of variance)	-10.02*** (0.780)	-10.51*** (0.760)	-10.16*** (0.760)
Fixed Effects	Yes	Yes	Yes	Yes
Pseudo R2	0.0815	0.0910	0.0914	0.0911
AIC	26320.3	26050.3	26039.3	26047.3
BIC	26337.2	26075.6	26064.7	26072.6
Observations	34756	34756	34756	34756

Fixed effects clustering by fishery

P-scores included in regression were estimated using test Model 3 and method noted in column title

Coefficients reflect log-odds of collapse (marginal effects not possible with fixed-effects model)

Standard errors in parentheses below coefficients

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Table 31: Regression Results for P-Score FE Model (Model 3 estimates, pt. 2)

	Logit Pcores (Model 3)	CART Pcores (Model 3)	Rand Forest Pcores (Model 3)
itq	0 (.)	0 (.)	0 (.)
years_of_itq	-0.0368*	0.510***	0.0802***

	(0.0195)	(0.0596)	(0.0297)
time	0.0702*** (0.00156)	0.0749*** (0.00161)	0.0710*** (0.00157)
p_score	0 (<i>dropped, lack of variance</i>)	-20.29*** (1.474)	-2.320*** (0.429)
Fixed Effects	Yes	Yes	Yes
Pseudo R2	0.0815	0.0956	0.0826
AIC	26320.3	25918.4	26290.0
BIC	26337.2	25943.7	26315.4
Observations	34756	34756	34756

Fixed effects clustering by fishery

P-scores included in regression were estimated using test Model 3 and method noted in column title

Coefficients reflect log-odds of collapse (marginal effects not possible with fixed-effects model)

Standard errors in parentheses below coefficients

* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$

Appendix 3: Supplementary Background & Analysis

Background: Potential Outcomes (PO) Framework

The Potential Outcomes (PO) framework is a widely used model for causal analysis across many disciplines. Jerzey Neyman laid early groundwork for this model in the 1920s, with its modern form developed by Donald Rubin in the 1970s (Rubin, 1974; Rubin, 1978). The model is alternately referred to as the Rubin Causal Model or Neyman-Rubin Model. While different causal frameworks have been proposed since – most notably Judea Pearl’s graphical DAG approach (Pearl, 2009) which has gained traction in ML causal identification literature – the Rubin Causal Model remains a standard in treatment-effect work for economics today.

At a high level, the PO framework is a causal analysis tool where a treatment (or intervention, manipulation) is applied to a unit to discover how it effects some outcome of interest. Imbens (2019) highlights three critical components of the model. First is the treatment/cause, which takes on different values for individuals depending on what intervention is applied to them. Second is the presence of multiple units – a requirement so that outcomes may be observed from units receiving different levels of treatment. And finally, an assignment mechanism which selects which units receive which treatment. The nature of each of these components determines which data, assumptions, and estimands are proper for causal inference. To make this more concrete I cover some formal implementations below⁸.

Model Setup

The formal PO model setup contains N units indexed $i = 1, \dots, N$. Units can have various forms but typically represent a person, object or group at a certain point in time. The treatment variable, W , indicates the different levels of intervention. This thesis focuses on a binary treatment where indicator $W_i \in \{0,1\}$ takes a value of 1 when unit i is exposed to the treatment and 0 when unit i is not exposed to the treatment (is in control group). Each unit is assigned to a single treatment.

⁸ Notation and formulas are based on Athey, Wager and Muhlbach’s tutorial (2019)

Potential outcomes are denoted $Y_i(W)$, representing the outcome variable of interest for unit i when exposed to treatment level W . Under the binary treatment scenario, every unit has two potential outcomes $(Y_i(1), Y_i(0))$ – one for each unit-treatment pair. Since every unit is assigned to a single treatment group, only one of the potential outcomes is observable, a distinction highlighted by the observed outcome variable, $Y_i^{obs} = W_i Y_i(1) - (1 - W_i) Y_i(0)$. The data for each unit is (Y_i^{obs}, W_i, X_i) , where Y_i^{obs} and W_i are the single observed outcome and treatment indicator variables respectively and X_i is a vector containing all other observable variables in the dataset. X_i is sometimes referred to as the pre-treatment variables, suggestive of the importance in considering their timing and potential for reverse causality.

Causal Effect

The value of interest in this framework is the unit-level causal effect: $\tau_i = Y_i(1) - Y_i(0)$. Defined as the difference between potential outcomes for unit i under treatment and control groups, this metric is not directly observable – a fact known as the “fundamental problem of causal inference” (Holland, 1986). Since every unit is assigned to a single treatment group only one of the potential outcomes is observable, the other, unobserved outcome is referred to as the counterfactual and must be estimated.

This is an important point and worth repeating: the causal effect is *defined* by potential outcomes but *estimated* using observed outcomes. Since only one treatment is observed for each unit, this estimation can be thought of as a missing data problem in which the counterfactuals must be imputed. The method of imputation is dependent on characteristics of the data, with a few baseline assumptions required across the board.

Assumptions

There are a few critical assumptions that must hold for unbiased causal inference using the PO model:

Assumption PO.1: Stable Unit Treatment Value Assumption (SUTVA)

“The potential outcomes for any unit do not vary with the treatments assigned to other units, and, for each unit, there are no different forms or versions of each treatment level, which lead to different potential outcomes.” - (Imbens & Rubin, 2015)

This assumption can be decomposed into two different parts:

-
- A. the *no-interference assumption* that there are no externalities or network effects from treatment of a given unit
 - B. the *no hidden variations of treatment assumption* that a treatment is applied consistently across units in a specific treatment assignment group.

Assumption PO.2: The data must be an as-good-as-random sample drawn from a large population.

The conditions upon which this assumption can be supported determine which estimand to use and depend on the treatment assignment mechanism in play.

Randomization Levels

Randomized Experiments

Randomized experiments are the gold standard for causal inference and are characterized by an assignment mechanism where treatment assignment is completely random for each unit (i.e. probability of every unit-treatment pair is even). This design ensures that treatment is independent of the potential outcomes (independence assumption):

$$W_i \perp (Y_i(0), Y_i(1)) \quad \textbf{Assumption: Independence}$$

With the above assumption met from random assignment, a naïve approach can be made for treatment effect estimation and the ATE estimator may be specified as the simple average of each observed group:

$$ATE_{Naive} = E[Y_i(1)|W_i = 1] - E[Y_i(0)|W_i = 0]$$

The implied naïve ATE estimator:

$$\hat{\tau}_{Naive} = \frac{1}{N_T} \sum_{\{i: W_i=1\}} Y_i - \frac{1}{N_C} \sum_{\{i: W_i=0\}} Y_i$$

Observational Data Under Unconfoundedness:

In practice, economists rarely have the luxury of experimental conditions and must make do with observational data. This means that the assignment mechanism cannot be controlled and directly randomized, rather the units must be split into treatment/control groups based on some previous record, observable datapoint or identification strategy. The nonrandomness of this mechanism violates the original independence assumption, but may fulfill the weaker unconfoundedness (ignorability) assumption:

$$W_i \perp (Y_i(0), Y_i(1)) \mid X_i \quad \textbf{Assumption: Unconfoundedness}$$

This assumption is referred to “selection on observables,” since it implies that treatment assignment is randomly assigned conditional on the observable control variables X_i . For conditioning to work an overlap assumption must also hold:

$$0 < \Pr(W_i = 1 \mid X_i) < 1 \quad \textbf{Assumption: Overlap}$$

When these properties hold, causal inference is possible, but the ATE estimator must be adjusted – conditioned on X_i – to find the conditional ATE (CATE):

$$\begin{aligned} CATE(x) &= E[\tau_i \mid X_i = x] \\ &= E[Y_i(1) - Y_i(0) \mid X_i = x] \\ &= E[Y_i(1) \mid X_i = x] - E[Y_i(0) \mid X_i = x] \quad \because \textit{linearity of expectations} \\ &= E[Y_i(1) \mid W_i = 1, X_i = x] \\ &\quad - E[Y_i(0) \mid W_i = 1, X_i = x] \quad \because \textit{unconfoundedness} \\ &= \mu(1, x) - \mu(0, x) \end{aligned}$$

The CATE estimator can be estimated by several different methods including propensity score weighting and rebalancing on covariates (for discussion on how propensity score matching may be implemented in practice see Econometric Methods Background section in Chapter 4).

Observational Data with Unobserved Confounders

Observational data that doesn’t fulfill the unconfoundedness assumption is subject to selection bias – the assignment mechanism cannot be made independent of the potential outcomes due to unobserved confounders. Therefore, the treatment variable cannot be used for causal analysis. An instrumental variable (IV) approach may open such a dataset up for causal analysis (Angrist & Pischke, 2009) but this is beyond the scope of this paper.

Based on these randomization techniques the PO framework estimates the treatment effect by fitting models to the proper estimator (ATE, CATE, etc.). This tends to abstract away from marginal effects of control variables, but there are some additional steps that may be taken to estimate heterogeneity in treatment effect (HTE). However, when coefficient values on a regression are of primary importance the classical linear model is more appropriate.

Background: Synthetic Control (SC) Method

Synthetic control (SC) is a relatively new econometric tool first created by Abadie and Gardeazabal (2003) and then expanded on by Abadie, Diamond and Hainmueller (2010). Formal implementation of the model discussed below is based on the 2010 paper, using potential outcome notation from the previous section.

Set-up

SC may be applied when there is a single treatment unit of interest and several control units from which a researcher would like to construct a counterfactual. The set up can be described using potential outcomes notation where there are $J+1$ observed units, indexed by i . A single unit ($i=1$) receives the intervention and the remaining units ($i = 2, \dots, J+1$) make up the control units – what Abadie et al. dub the *donor pool* for counterfactual estimation. The outcome of interest is observed over periods $t = 1, \dots, T$, and T_0 represents the final pre-intervention period. There must be at least one pre-intervention period in which the treatment unit is observed in the control state ($1 \leq T_0 < T$). Once the intervention is made in the treatment unit it remains in place and is indicated by the binary treatment variable⁹, D_{it} taking a value of 1. Since only region 1 received treatment and only after period T_0 :

$$D_{it} = \begin{cases} 1 & \text{if } i = 1 \text{ and } t > T_0 \\ 0 & \text{otherwise} \end{cases}$$

The relevant data format for SC is visualized a matrix with columns representing the time-series and a row for each individual observation:

$$D_{J+1 \times T} = \begin{pmatrix} 0 & \dots & 0 & 1 & 1 & i = 1, \text{treated all } t > T_0 \\ 0 & \dots & 0 & 0 & 0 & i = 2, \text{Never treated} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \text{Never treated} \\ 0 & \dots & 0 & 0 & 0 & i = J + 1, \text{Never treated} \end{pmatrix}$$

Assumptions

To interpret SC results as causal there are two key assumptions that must hold:

⁹ Note: treatment variable receives slightly different notation versus the PO model above, replacing W with D to avoid confusion with the weights, w_j , used in SC

Assumption SC.1 No interference between units

Assumes outcomes of untreated units (donor units) are unaffected by the intervention in treatment unit. Further discussion of this assumption can be found in Rosenbaum's paper (2007).

Assumption SC.2 No anticipation of intervention

Assumes that the treated unit is unaffected by the intervention until after the implementation date (only affected when $t > T_0$). If this assumption doesn't hold but the period when anticipation begins is known then T_0 can be adjusted to reflect the first period that there may be a reaction so that there is an estimated counterfactual for all periods where the intervention has an effect (anticipatory and post-treatment).

Model Estimator:

Treatment effect takes its form from the PO framework:

$$\tau_{it} = Y_{it}(1) - Y_{it}(0)$$

Observed outcome takes its form from the PO framework:

$$Y_{it}^{obs} = D_{it}Y_{it}(1) + (1 - D_{it})Y_{it}(0)$$

Our value of interest is the treatment effect for unit $i=1$ in treatment periods $T_0 < t \leq T$:

$$\tau_{1t} = Y_{1t}(1) - Y_{1t}(0)$$

Because the value of $Y_{1t}(0)$ is unobserved for these periods, SC uses the donor pool's observed outcomes to estimate its value. This is done by assigning weights to each control unit so that the estimator is:

$$\hat{\tau}_{1t} = Y_{1t}(1) - \sum_{j=2}^{J+1} w_j^* * Y(0)_{jt}$$

for $t \in \{T_0, \dots, T\}$

where w_j^ are the optimized weights output by SC*

The critical function of SC is to calculate this optimal weight, w_j^* , for each control unit.

Suppose W is a $(J \times 1)$ vector of weights = $(w_2, \dots, w_{j+1})$ such that:

$$w_j \geq 0, \forall j \in 2, \dots, J + 1$$

$$\text{and } \sum_{j=2}^{J+1} w_j = 1$$

The vector W represents a synthetic control – the weighted average of all units in the donor pool. To fit these weights to be the best possible counterfactual, the SC method algorithmically calculates the optimal weights for each unit (represent them as w_j^*) to match values from a factor model for the control outcome:

$$Y_{it}(0) = \delta_t + Z_i \theta_t + \lambda_t \mu_i + \epsilon_{it}$$

δ_t	unobserved common time factor
Z_i	(r x 1) vector of observed covariates (unaffected by intervention)
θ_t	(1 x r) vector of unknown parameters
μ_i	(1 x F) vector of unknown common factors
ϵ_{it}	unobserved transitory shocks at unit-level, has zero mean

The $\lambda_t \mu_i$ term captures heterogeneous responses to multiple unobserved factors. By reweighting the controls (donor pool) so that the SC unit matches Z_i and the pre-treatment Y_{1t} (for the treated unit), this ensures that the common factors μ_i are matched and the SC will provide an unbiased counterfactual. Put in simpler terms, this method identifies the control unit weights that create a synthetic control unit with the same observed covariates and outcome values in the pre-intervention stage. The weights can then be used to impute the counterfactual over the intervention period in an unbiased way.

Evaluating Results

A visual analysis is common for SC methods, in which the outcome of interest is plotted over time for the treated unit and the synthetic control unit. The quality of the synthetic control can be estimated by looking how close the pre-treatment trend in the graph is, while the causal effect is represented by the difference between the plots after treatment date. While the model is restrictive in that only a single treatment unit is analysed at once, this setup allows longer term effects and heterogeneity in effect over time to be seen.

Analysis: Causal Identification Issues

This section covers several challenges to a causal interpretation of the Costello et al. (2008a) model specifications. The focus is on possible bias from the logit implementation and violations of critical propensity score matching assumptions.

Issue 1: Bias from Logit Transformation

Due to the binary outcome variable, $collapsed_{it}$, an OLS regression would fail the homoskedasticity and normality assumptions. Since the simple linear model doesn't restrict the range of possible fitted values it could also result in probability estimates outside of the 0-1 range. To account for this, Costello et al. transform the outcome using the logit link function, $\log\left(\frac{collapsed_{it}}{1-collapsed_{it}}\right)$ and use a binomial distribution. This solves the issues associated with OLS but introduces new problems since the regression is based on *log-odds* rather than the interpretable value of interest, *probability*.

To back out the probability using the fitted model, there must be a non-linear transformation ($\Pr = \frac{1}{1+e^{-(\log-odds)}}$). This means the marginal effect of interest is different at different levels of the covariate... inference requires some form of averaging to get a single value. This also means that standard errors must be backed out using non-traditional ways like the delta method or bootstrapping. Such transformations may be unbiased under the strictest OLS assumptions (i.i.d. error term $N(0)$), but the Difference-in-Differences (DiD) and Fixed-Effects (FE) models Costello employs are designed to relax this very assumption. In the non-linear transformation their algebraic properties cancelling out portions of systematic error break down.

The DiD framework allows the error term to contain time-varying unobservables that effect all units the same (parallel trend). Lechner shows that logit-type models generally violate this parallel trend assumption because the linear trend does not carry over to the non-linear transformation, and “[t]herefore, estimation based on this model does not identify the causal effect,” (2010). He proves that the results will only be unbiased if there are no group-specific differences, which is exactly the parameter of interest.

By using conditional fixed effects with a logit transformation, unbiased estimates of the *log-odds* coefficients are possible (Chamberlain, 1980). However, Coupé discusses that this method does not provide individual estimates for the fixed-effect value “which are needed if

one wants to compute statistics like marginal effects,” (2005). This means probability-based marginal effects backed out of a FE logit model will be biased. For this reason, Costello only reports the FE models using log-odds coefficients – a big sacrifice in the interpretability of results.

Issue 2: Propensity Score Problems

Overlap Assumption Violation

The overlap assumption in propensity score matching states that for unbiased causal effect in the face of selection bias from observables, there must be common support over the observables for both treatment and control units. That is, for any stratification of observables $X_i = x$ there must observations for each group present, so a proper match can be used in calculating the counterfactual. Formally this condition is:

$$0 < \Pr(W_i = 1 | X_i) < 1 \quad \textbf{Overlap Assumption}$$

This means that the p-scores must be bounded away from 0 and 1 – something that can be tested by looking at the distribution of p-score estimates.

Since Costello runs the p-score regressions on each year individually, this clearly isn’t the case – from 1950 until 1975 there are no treated units so these p-scores will be bounded on 0. Likewise, splitting the covariate space on such high-dimensional dummy variables creates very narrow strata that are unlikely to contain the needed overlap. To check the extent of this overlap violation, I estimated propensity score using Costello’s models on the recreated dataset – first running it on each year individually (*_yr* suffix) and then running the models on the full dataset (all years at once) including just a linear time trend (*_notr* suffix). A boxplot of the results shows that the propensity scores for all specifications are highly skewed towards 0 – and often equal to 0, indicating that the overlap assumption is violated (diamond represents the mean):

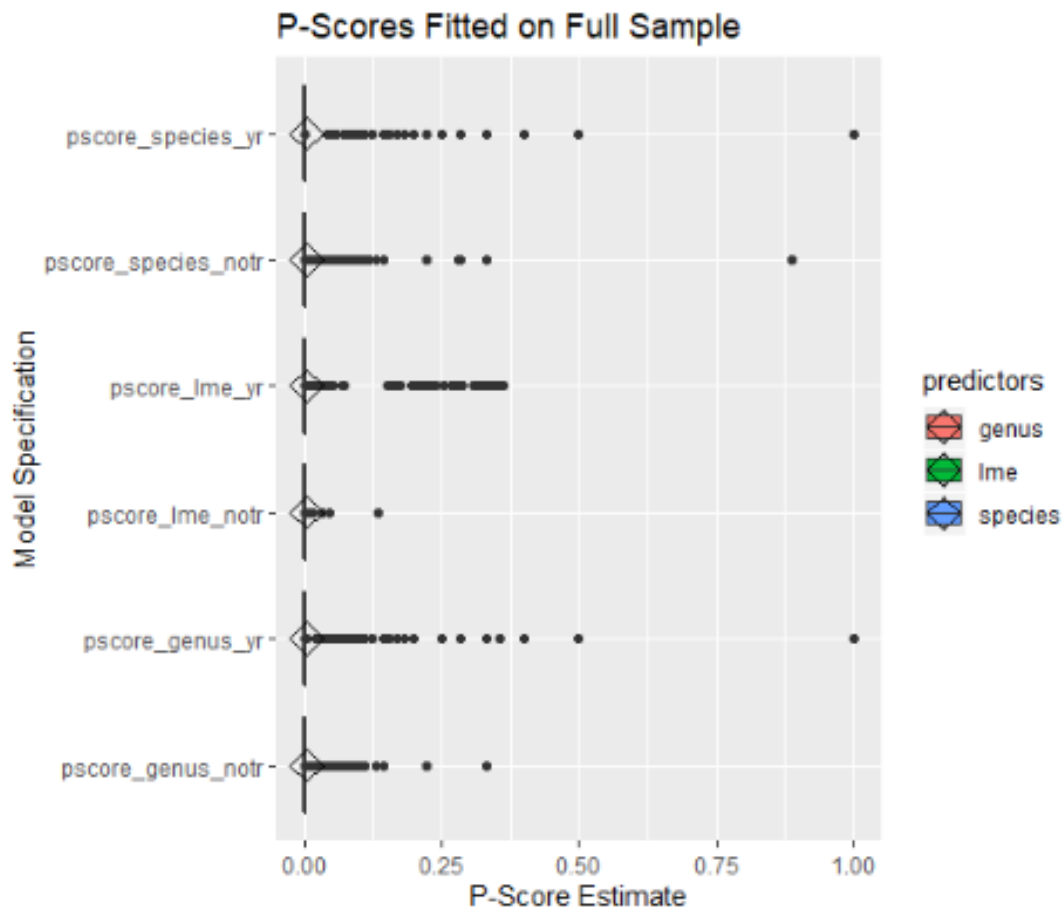


Figure 7: Propensity Score Distributions – Full Recreated Dataset

A primary factor resulting in these skewed propensity score estimates is that ITQ implementation is rare. Only 1% of fisheries have ITQ systems by 2003 and, even then, each ITQ fishery has at least a 25 year period of pre-implementation observations. To get overlap there must be much wider strata (e.g. use Order dummies instead of Species for taxonomic strata) or the sample must be restricted. Hajage et al (2016) find that, in the case of rare exposure, propensity score matching performs better when run on only the treatment group. This removes the difference-in-difference benefits from the control group but increases the quality of propensity score in controlling selection bias. To test this, I reran the p-score predictor models on a sample of the recreated model including only the treated units:

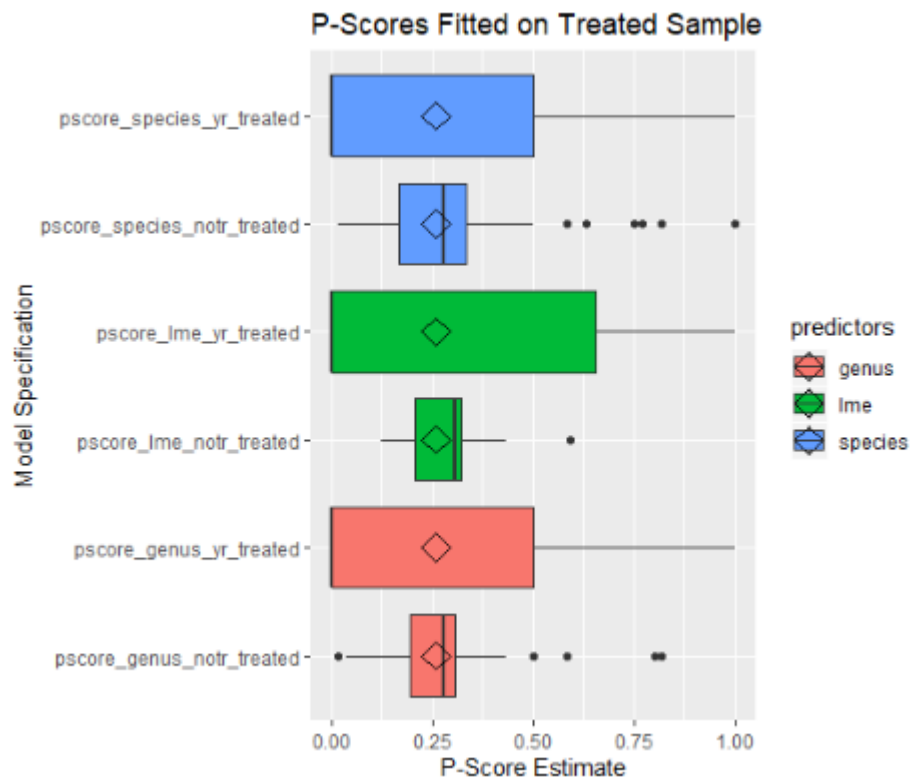


Figure 8: Propensity Score Distributions – Recr. Dataset (Treated Group)

Under this restriction the overlap issue is much reduced, however there are still some values at the extremes (0 and 1). Further limitations on the dataset yet may be required for strict overlap conditions to hold. This illustrates that the use of propensity scores by Costello et al. on the full sample is likely to result in biased causal effects.

Appendix 4: Supplementary Tables & Figures

Validation Curves

Model 1

CART

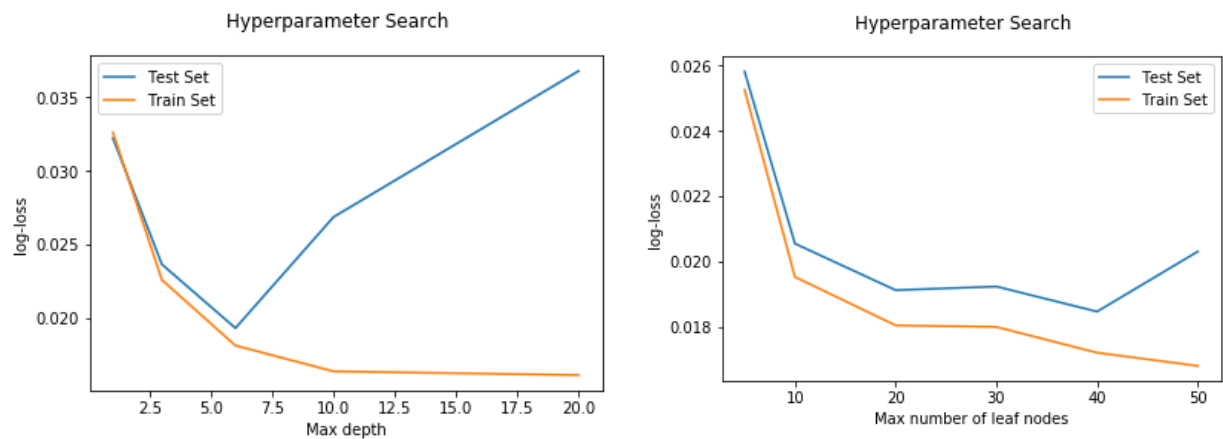


Figure 9: Validation curves from parameter search (Model 1, CART)

Random Forest

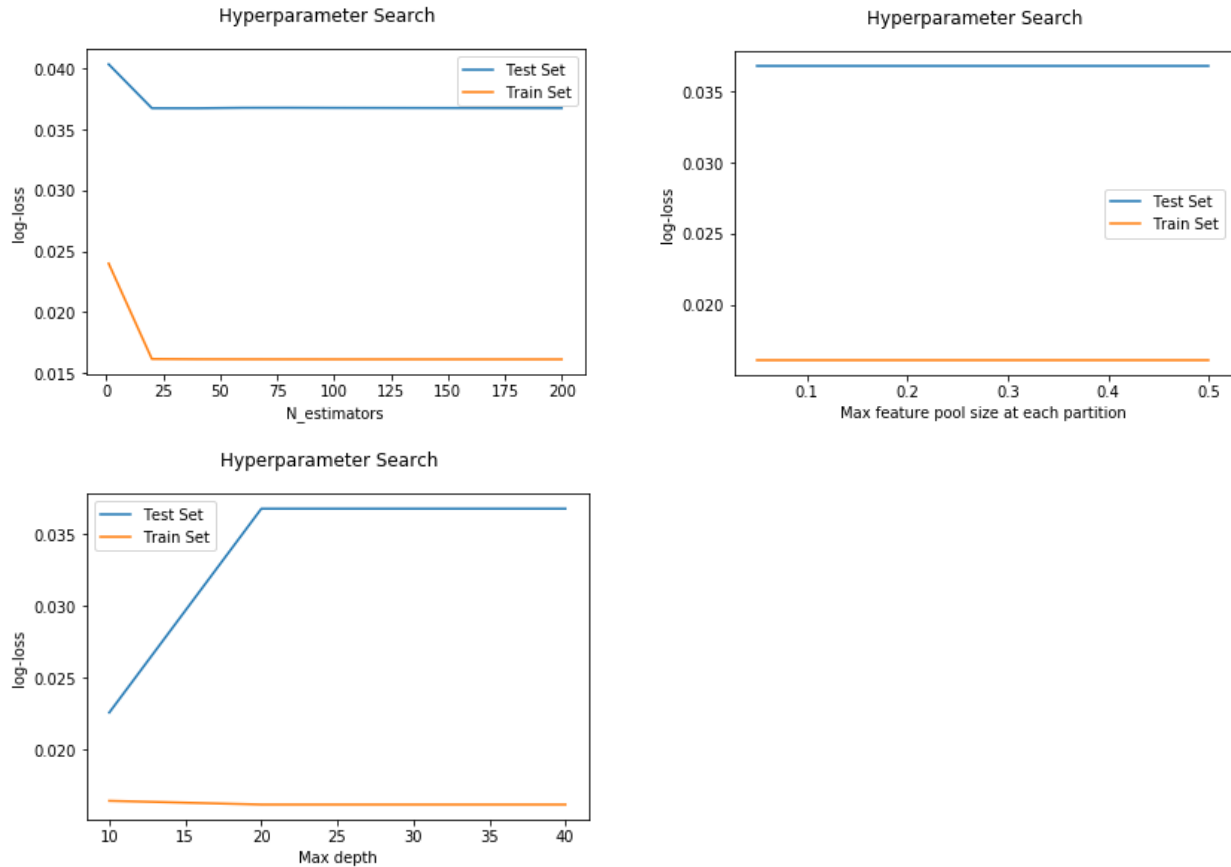


Figure 10: Validation curves from parameter search (Model 1, Rand. Forest)

Model 2

CART

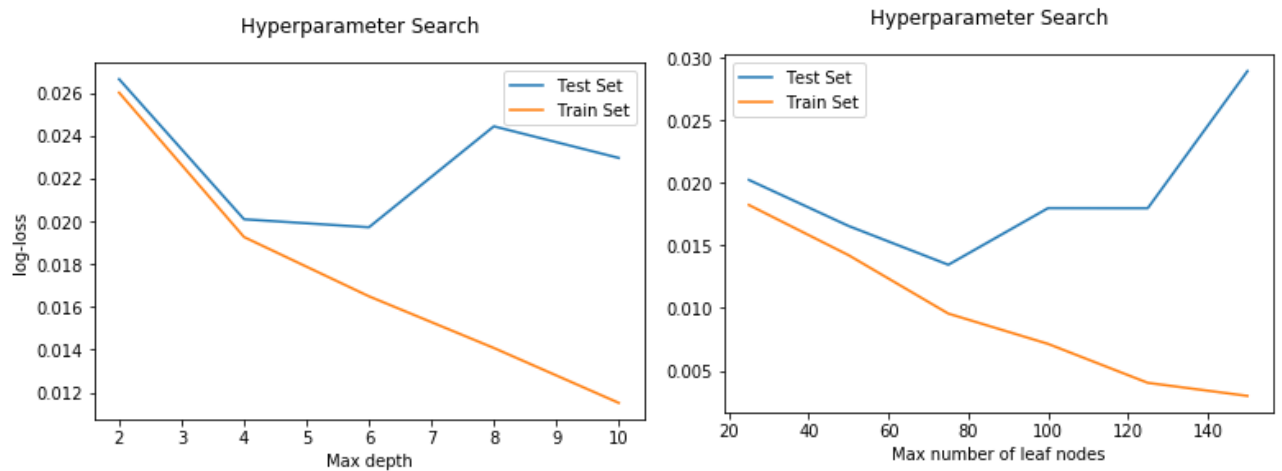


Figure 11: Validation curves from parameter search (Model 2, CART)

Random Forest

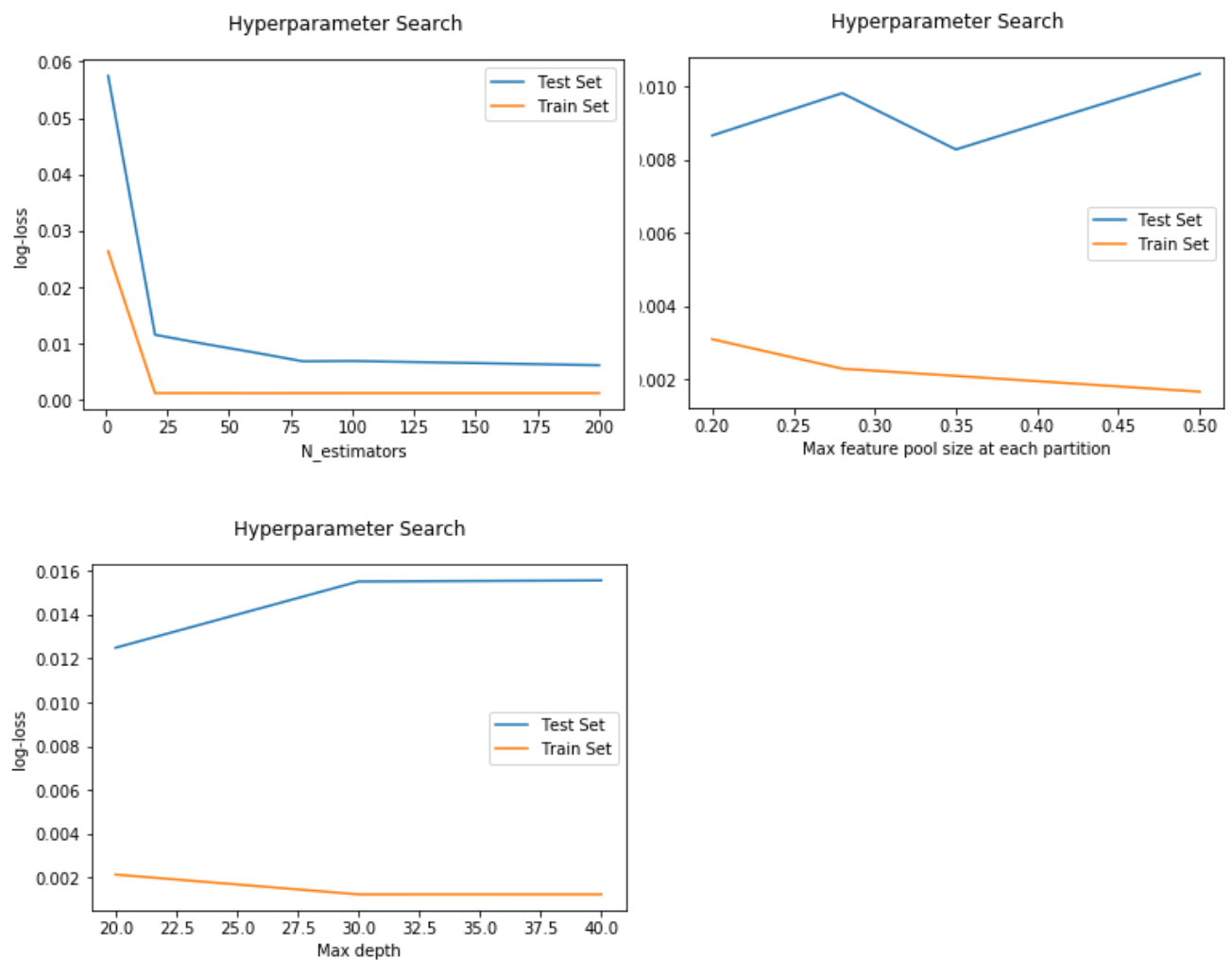


Figure 12: Validation curves from parameter search (Model 2, Rand. Forest)

Model 3

CART

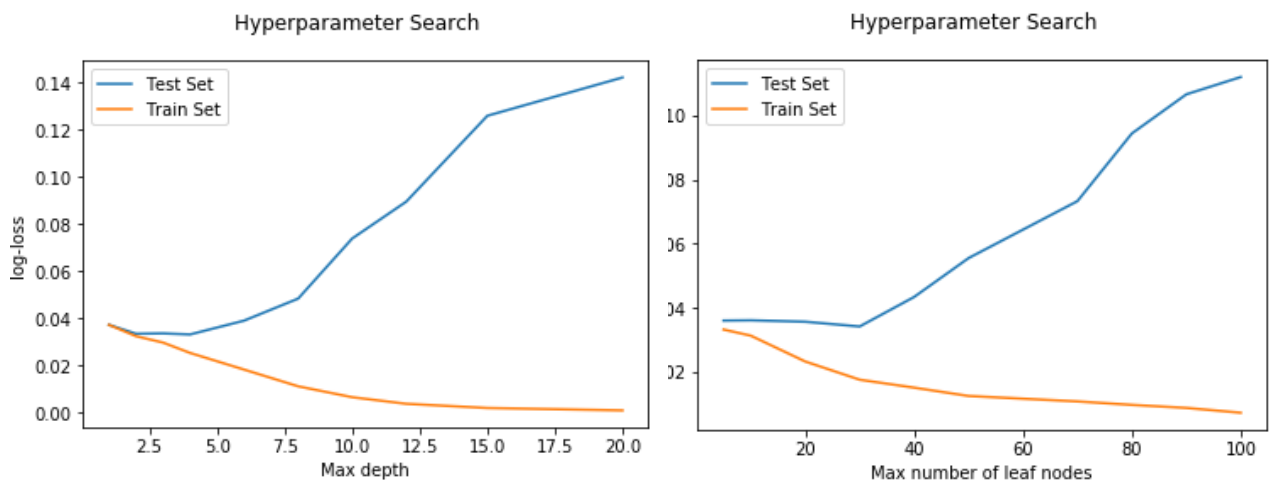


Figure 13: Validation curves from parameter search (Model 3, CART)

Random Forest

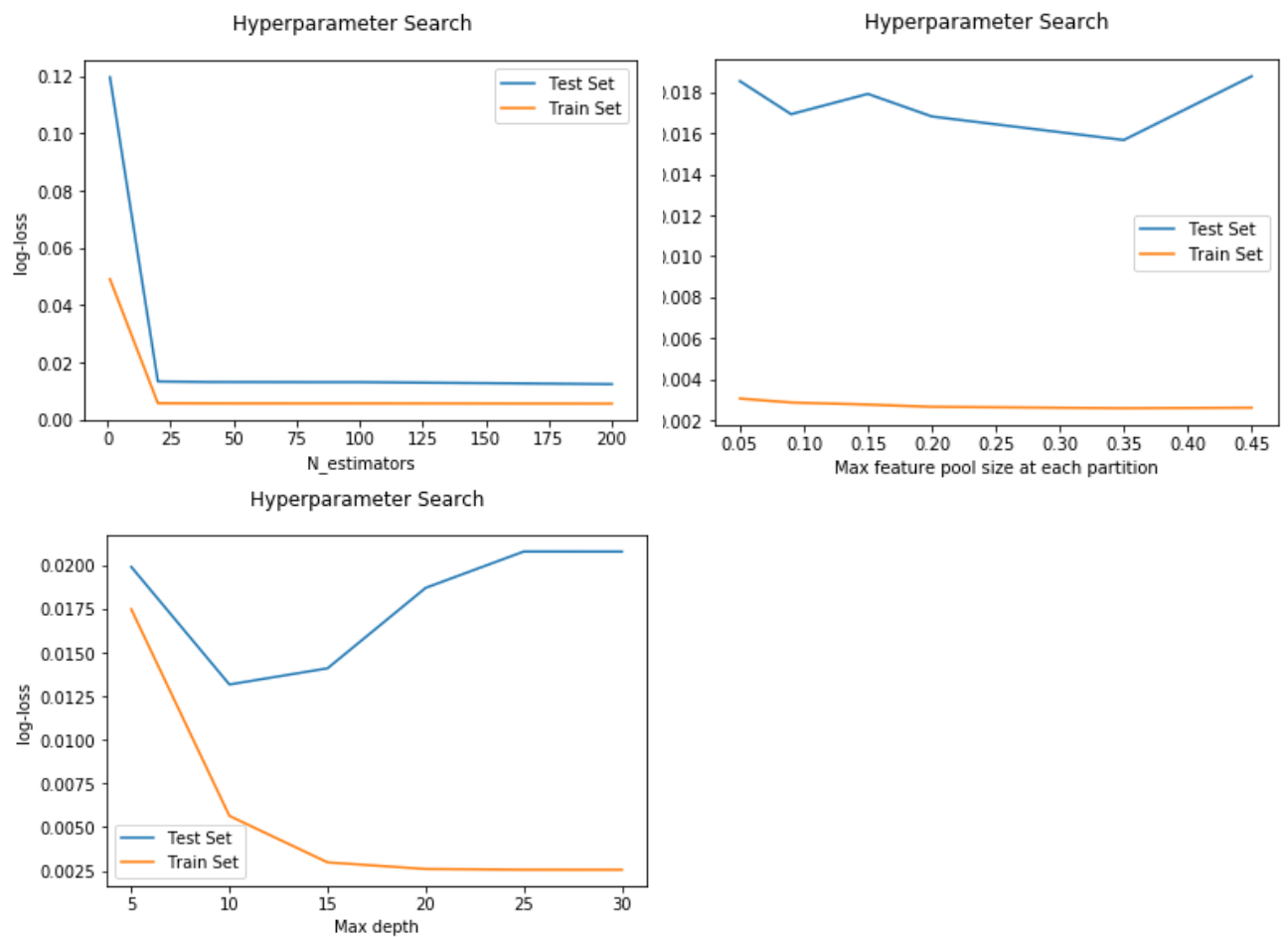


Figure 14: Validation curves from parameter search (Model 3, Rand. Forest)

Model 3 Covariate Details

This table provides a list of the different possible values for each categorical variable included in propensity score Model 3. It also provides reference data for two continuous variables that are estimated values based on models.

Table 32: Model 3 Covariate Details

Variable	Note	
Commercial Groups	Cod-likes Flatfishes Herring-likes Perch-likes Salmon, smelts, etc Scorpionfishes Sharks & rays Tuna & billfishes Other fishes & inverts	
Functional Groups	Large bathypelagics (≥ 90 cm) Large benthopelagics (≥ 90 cm) Large demersals (≥ 90 cm) Large flatfishes (≥ 90 cm) Large pelagics (≥ 90 cm) Large rays (≥ 90 cm) Large reef assoc. fish (≥ 90 cm) Large sharks (≥ 90 cm) Medium bathydemersals (30 - 89 cm) Medium bathypelagics (30 - 89 cm) Medium benthopelagics (30 - 89 cm) Medium demersals (30 - 89 cm)	Medium pelagics (30 - 89 cm) Medium reef assoc. fish (30 - 89 cm) Small bathydemersals (< 30 cm) Small bathypelagics (< 30 cm) Small benthopelagics (< 30 cm) Small demersals (< 30 cm) Small pelagics (< 30 cm) Small reef assoc. fish (< 30 cm) Small to medium flatfishes (< 90 cm) Small to medium rays (< 90 cm) Small to medium sharks (< 90 cm)
Gear Category	gillnet_gear line_gear pot_trap_gear	

	seine_encircling_nets subsistence_artisanal_recreational_gear trawls_dragged_gear unkown_gear
End use types	Direct human consumption Discard Fishmeal and fish oil Other
Fishing sector	Artisanal Industrial Recreational Subsistence
Price category	Pulled from FishBase, estimator model based on ex-vessel price paper (<i>Sumaila et al, 2007</i>)
Vulnerability	Pulled from FishBase, estimator model based on paper (<i>Cheung, Pitcher & Pauly, 2005</i>)