Aalto University
School of Business

MACHINE LEARNING IN APPLIED ECONOMETRICS

Deriving personal income drivers with randomized decision forests

Economics

Master's thesis

Henri Ikonen

2016

Department of Economics
Aalto University
School of Business

ABSTRACT

In this paper I explore a modern field of research in applied econometrics: machine learning and the estimation of synthetic treatment effects.

Data generation is currently on an exponential growth path: smart phones, social media and networks of interconnected devices are generating information at an unprecedented pace. The size, structure and velocity of these information streams vary to a great extent. The field of econometrics is also evolving: classic econometric models can lead to biased results with big data and will not scale up to modern data sets.

I propose the well- performing Random Forests algorithm for use in econometrics. To adjust this method for causal analysis, recent theory on causal decision trees is explored. The proposed framework is then tested by estimating personal income drivers for the top 1% in U.S. population. The data used is the American Community Survey 5- year sample consisting of approximately 20 million rows.

It appears that high income is in fact driven by four core factors: education, experience, working hours and gender. To rank these predictors, a synthetic treatment effect simulation is run. I find that investing in education after a master's degree has a significant positive effect in the likelihood of high income. Additionally, it appears that the negative gender income effect for females can be undone with a combination of work experience and exceptional work- ethic.

KEYWORDS: ECONOMETRICS, MACHINE LEARNING, DECISION TREES, CAUSALITY, BIG DATA, RANDOM FORESTS, INCOME, AMERICAN COMMUNITY SURVEY

Table of Contents

LIST OF FIGURES AND TABLES

1 INTRODUCTION

Economic systems are known for their complexity. Changes in these systems are often hard to anticipate due to noise, complex relations and an unintelligible amount of potential influencing factors. Examples of such systems include drivers behind a country's GDP growth rate, factors that influence workforce participation and changes in consumer behaviour resulting from a tax change or public policy decision. Economists strive to understand these systems by simplifying them into mathematical models where different parameter effects can be isolated and analysed separately. 'Structural' approaches emphasize theory and create algebraic representations of phenomena while 'non- structural' approaches refer to more data- driven studies where there is little or no economic theory behind the variable selection and model fitting process. Econometrics stands for empirical analysis and statistical modelling for economic phenomena. Both theory- (structural) and data- driven (non- structural) approaches are accepted and widely applied in the field (Reiss & Wolak, 2007).

Data scarcity has been commonplace in many econometric studies in the past (Einav & Levin, 2014). The availability and quality of data has had drastic variations between countries and different economic phenomena. Macroeconomic problems illustrate this example: Consider regressing annual GDP growth against a selected set of variables in an exploratory fashion. The observations for the dependent variable are limited by the amount of years with concise data collection, observations for independent variables may have different limitations and data quality varies between countries and regions. Building a robust statistical model and considering all relevant information becomes increasingly hard. To understand the forces at work the focus shifts from prediction to causal inference. Instead of predicting a Y given a set of some variables $X_{i\ldots j}$, econometrics looks to identify a change in Y, given a change in some variable X_i. Classic statistical methods like ordinary least squares estimation (linear regression) tend to work well with small to medium sized data sets and basic data structures, which is why they've become the go-to tools for most applied economists (Einav & Levin, 2014).

However, in today's digitalized world where information flows from multiple streams and comes in different structures and sizes, the amount of data related to a single problem could range from gigabytes to terabytes and more. The biggest sources for economic data today

include the Internet and social media, smart devices and scanners (Einav & Levin, 2014). The World Bank, US open data platform 'www.data.gov' and the US Census Bureau are among some of the richest and easy-to-use sources for econometric analysis. These developments bring both new possibilities and challenges for econometrics. The amount of problems that can be answered is greatly increased, forecasting becomes more efficient and a wider range of potential variables can be addressed in modelling (Varian, 2014). There are certain challenges that arise: (1) New data comes with various structures and sizes that differ greatly from classic panel- or time series data sets. (2) Several situations arise where classic econometric tools like linear regression lead to biased results. These include high-dimensions, nonlinear effects and high correlation between independent variables. (3) The modern algorithms required to work with large data add a level of complexity to inference and require different approaches to model fitting. However, a successful combination of prior knowledge in statistics and econometrics with acquired expertise in machine learning can enable econometricians to achieve new fruitful results, and researchers expect the field of econometrics to evolve in the coming years (Varian, 2014; Einav & Levin, 2014).

The aim in this thesis is to review a family of machine learning methods, or decision trees, in the context of econometrics. The goal is to present a framework for working with modern data sources and to incorporate treatment effect estimation into machine learning applications. The American Community Survey 5- year data is used to find drivers and justification behind earnings for the top percentile of U.S. population. The income effects of these drivers are then simulated by applying the 'Two Trees' algorithm as proposed by Athey & Imbens (2015).

The literature review starts by reviewing the possibilities and challenges offered by big data in econometrics. The challenges are evaluated with examples of linear regression. The concept of machine learning is then introduced along with the types of problems that learning methods are commonly used for. The second section takes a deeper look into decision tree algorithms and explains how Classification- and Regression Trees work. Econometric literature on decision trees is then extended by introducing the concept of ensemble models and a specific learning algorithm called 'Random Forests'. Random Forests is a combination of multiple decision trees (or an ensemble) with random variable selection. It offers an unbiased estimate of model generalization error through bootstrapping and testing on out-of-bag observations (Breiman, 2001). The concept of cross- validation for model performance

evaluation and parameter optimization is also introduced. The literature review ends with a review of methods for estimating treatment effects with decision trees, as proposed by Athey & Imbens in their 2015 paper 'Machine Learning Methods for Estimating Heterogenous Causal Effects'.

Using a 5-year aggregate of the American Community Survey Public Use Microsample Data, a Random Forest model is built to predict if an individual belongs to the top percentile of the population in terms of income. The optimal degree of randomization for the forest is found by implementing a grid search function with cross- validation (Appendix A). The most influential independent variables are selected by a decision- tree specific importance measure called 'Gini impurity'. Bivariate 'ceteris paribus' analysis is applied in the tree model through 'partial plots' to assess the effects captured. Finally, the 'two trees' approach to synthetic treatment effects is adopted from Athey & Imbens (2015) to rank the estimated 'prospensity effects' of different independent variables and draw conclusions.

The model fit was very efficient, classifying more than 94% of cases correctly in a hold- out validation set. The most important predictors in high income are namely the highest level of education achieved, weekly working hours, age and gender. In simulating treatment effects, investing in high education (MBA or PhD) is found to have the largest positive impact in the likelihood of high earnings. The gender gap is prevalent here as well as female gender treatment is found to decrease likelihood. According to the simulation results however, an individual can overcome the negative gender effect with seniority and a strong work ethic combined.

The causal trees and synthetic treatment estimation approach is an interesting way to adjust machine learning for econometrics. Research here is still very new, but interesting applications are easy to come up with. As many practical econometric models and forecasts suffer from the fact that experiments can be hard to set up, there could be significant benefits in experimenting with causal tree models and treatment effect estimators.

2 ECONOMETRICS AND 'BIG DATA'

2.1 A new data landscape

Data generation has grown exponentially during the last decade. New and increasingly important data sources for econometric analysis include the Internet, networks of interconnected devices and retail scanner data for example (Einav & Levin, 2014). Advances have been also made in computation to answer to the increased demand for data analysis. This practically means that new algorithms or models optimized for both wider and longer data sets have emerged along with parallelized computing environments and advanced storage systems (Varian, 2014). Modern technologies are out the scope of this thesis, but 'tech savvy' readers are advised to take a look at distributed cloud platforms and relevant computing frameworks like Apache Spark ('http://spark.apache.org/') and Hadoop ('https://hadoop.apache.org/').

These recent developments are changing the landscape for econometric analysis. Data scarcity simply isn't there for some problems where it used to be a persistent issue. Other problems may have ended up on the opposite end of the spectrum: The sheer amount of available data makes it hard to choose the most representative subset for analysis. New problems can be addressed and models can be built with greater precision. Several data sources like US open data 'www.data.gov' contain thousands of unique data sets concerning topics like business, agriculture, schooling, education and climate for example. Most of these sets can also be joined with each other based on unique identifiers like state codes or household IDs. This type of data aggregation allows the simultaneous analysis of hundreds or thousands of independent variables at a time. Einav & Levin (2014) propose that in addition to novel research designs and better measurements, this new data might even change the way economists approach empirical research. Even if the scope in machine learning is different, many researchers are starting to see value in statistical algorithms and a more 'data- driven' approach to empirical economics (Varian, 2014, Sengupta, 2015).

Naturally, the new data landscape brings new challenges for econometrics. Einav & Levin (2014) summarize these challenges broadly as gaps in knowledge and domain expertise, data accessibility, computation and 'asking the right questions'. Additionally, big data sets contain features like high dimensionality, nonlinear effects and abnormal structures that complicate

analysis. These features lead to specific issues like spurious correlations, incidental endogeneity and noise accumulation (Fan, 2014). In several occasions, classic econometric methods like linear regression fall short as they haven't been designed to deal with the scale and structure of modern data sources (Burger & Repiský, 2012). This chapter identifies the specific features and requirements that big data brings to analysis and evaluates these requirements with examples of linear regression applications.

2.2 Challenges

Computational requirements and domain knowledge. Tapping into big data requires some expertise in computer science. Efficiently processing large amounts of data practically calls for a parallel environment with multiple CPUs. At the very least, the practitioner needs to be familiar with relational databases and their usage with SQL, or 'structured query language'. As processing complexity and data quantity increase, modern computation frameworks and programming paradigms like 'Map- Reduce' become crucial (Jacobs, 2009).

Data accessibility. While large- scale data sets on topics like labour economics, consumption, productivity and macroeconomic indicators are freely available through government resources ('www.census.gov'; 'www.data.gov') and the World Bank ('data.worldbank.org'), several other sources are restricted or simply hard to access. First of all, some of the most rigorous micro- level data sets are collected by retailers, telecommunication companies and banks, i.e. the private sector. Gaining access to this data is therefore highly regulated and often requires a confidentiality contract limiting research questions and publications (Einav & Levin, 2014). Additionally, several internet data sources like ecommerce stores, social media and geo-locations require advanced collection methods like web crawlers and aggregators (Vargiu & Urru, 2013). Developing these methods is hardly a trivial task. Luckily however, upcoming privacy laws and open data schemes are expected to bring a change to this setting, at least in the case of privately collected big data (Shadt, 2012).

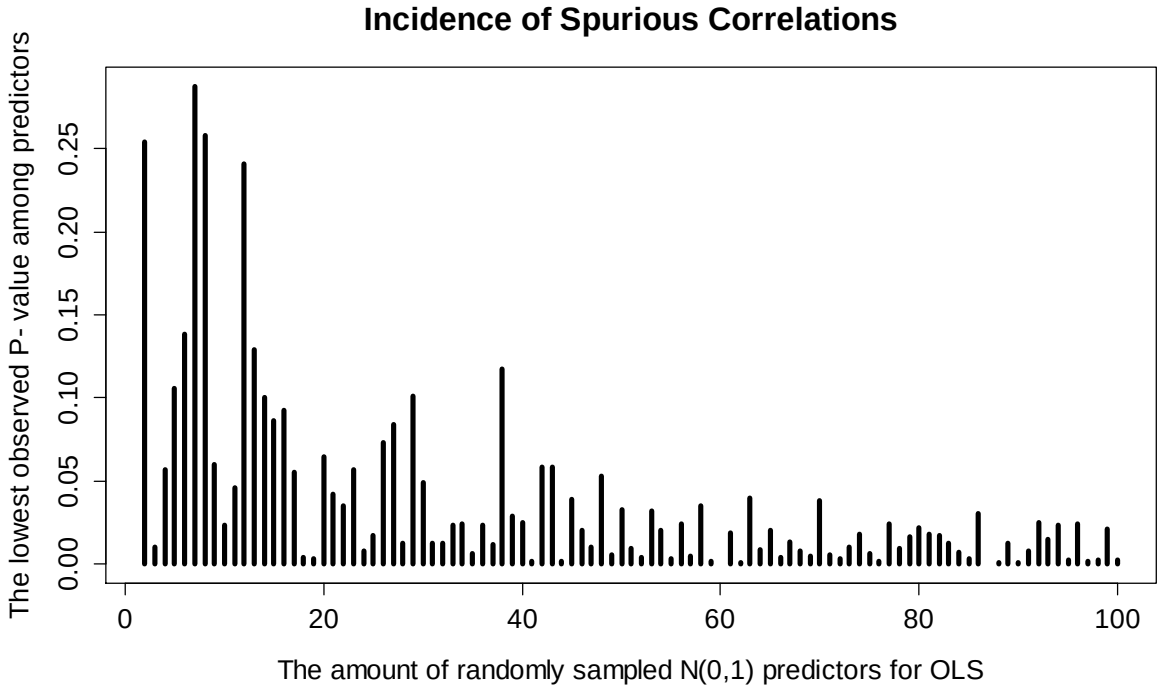
Asking the right questions and applying the correct methods. The analysis or large data sets is not trivial. Simple questions like 'What problems does this data support' become more complex as tasks like running visualisations and variable summaries become more computationally intensive. The local computer is often not the place for these exploratory analyses (Einav & Levin, 2014). As many statistical packages (like R, 'cran.r-project.org')

load data into memory, simple read operations might reserve the majority of a normal laptop's resources with big data sets, leaving little capability for actual analysis. Modelling is increased in complexity as well: The specific features often present in big data lead to situations where linear estimators either don't converge or offer biased results (Burger & Repiský, 2012). Additonally, spurious correlations due to high- dimensions reduce the credibility of statistical tests and confidence intervals for variable importance. Nonparametric estimators and cross- validation become crucial as data size and complexity increases.

2.3 Features of Big Data

There are specific features in big data sets that do not emerge in in low-dimensional data or small samples. Specifically, high dimensionality is known to promote spurious correlations, incidental endogeneity and noise accumulation (Fan, 2014).

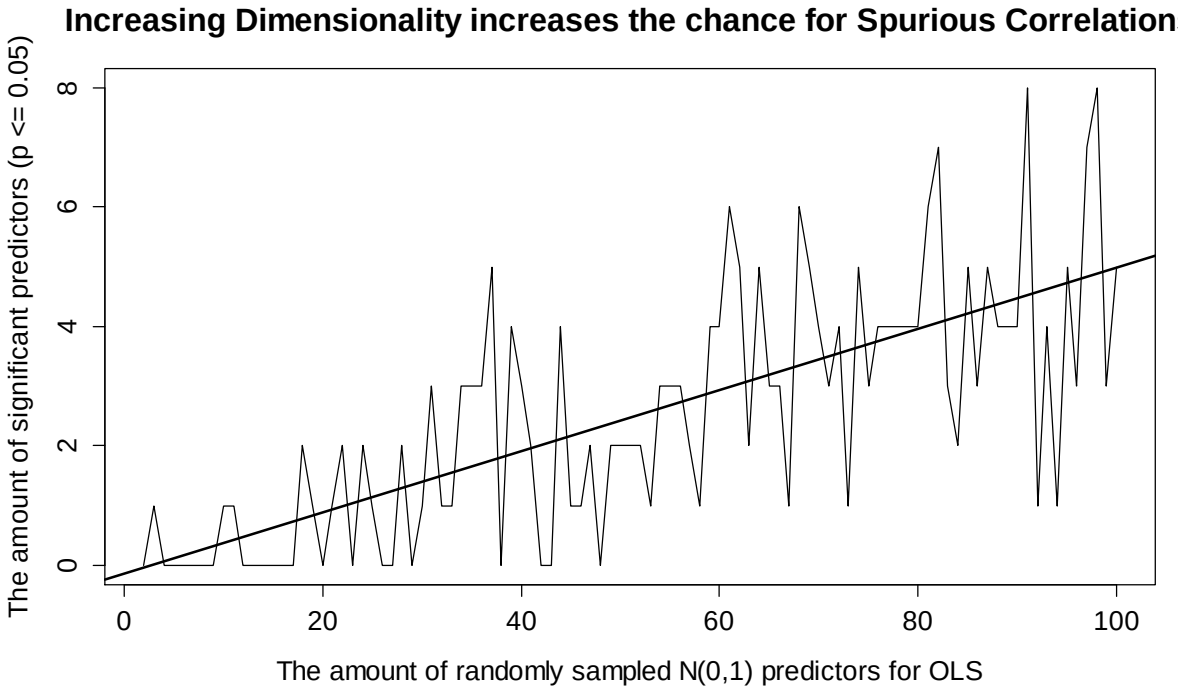
Figure 1: Incidences of random correlations



Spurious correlations. These are correlations that aren't theoretically sound, but have high support in some sample data. Fan (2014) provides an example: Consider a random sample of size n with p independent standard random variables. The population correlation between any

two random variables will be zero in this setting. Taken that the amount of individual variables (dimensionality) is small compared to sample size, the sample correlation between any random variables should also be close to zero. Increasing dimensionality (p) will however lead to increased correlations between different independent variables p. These correlations are spurious.

Figure 2: Correlation between dimensionality and random significance



Selecting variables based on random correlations lead to biased models and skewed results. This issue can be avoided in modelling by applying feature reduction techniques and validating built models on separate hold-out data sets. If the correlation captured is spurious, the corresponding model should lead to poor results in a carefully selected validation set.

Figures 1 and 2 capture the effect that increased dimensionality has on statistical tests. The simulations were conducted in R by randomly sampling independent variables from the normal distribution and iteratively running OLS regression on the data. The first sampled variable was held dependent. As one can easily observe in figure 2, increased dimensionality is highly correlated with falsely significant independent variables.

Incidental endogeneity. Adding dimensionality can lead to an unwanted situation where some covariates are actually correlated with the residual noise instead of the dependent variable. This leads to inconsistency in model- and variable selection. Validating the assumption of exogeneous variables, $E(\varepsilon X) = 0$, can be tricky and is often ignored in practice. Fan (2014) proposes the 'Generalized Method of Moments' to deal with this phenomenon.

The Random Forests approach applied in this study offers an alternative where model stability is achieved through bootstrapping and variable selection is based on 'average information gain' over the whole system.

Noise accumulation. If a method is dependent on estimating multiple parameters, estimation errors can accumulate. Noise accumulation is generally more severe for high- dimensional problems and can sometimes even dominate underlying signals in data. In some applications, models can capture noise instead of actual relationships in data, leading to negligible predictive power (Fan, 2014).

2.4 Common pitfalls of linear regression

How do the features of big data materialize in the case of a well- known method in econometrics, linear regression? Ordinary least squares regression is one of the most used statistical methods in science. In addition to vast applications in econometrics it's heavily used in fields like psychology, medicine and finance for example. OLS, or linear regression, makes the assumption that the dependent variable is at least to some extent a linear function of the independent variables. In other words, it estimates a dependent variable Y with a linear formula (Burger & Repiský, 2012).

$$Y = a_0 + a_1 X_1 + a_2 X_2 + a_3 X_3 + ... + a_n X_n + \varepsilon$$

The least squares algorithm works by choosing the constants $a_{1,}\ldots,a_n$ that minimize the squared error between observed dependent variable values and model predictions. In a bivariate case with a single independent variable, this problem equals minimizing Q in the following formula (Burger & Repiský, 2012):

$$Q = \sum_{i=1}^{n} [y_i - f(\hat{\alpha}_0 + \hat{\alpha}_1 x_i]^2$$

Linear regression with a single independent variable can be represented as a line. A regression with two independent variables is a three- dimensional plane and further added predictors result in different hyperplanes, or generalized lines in the corresponding feature spaces. Figure 4 illustrates the three- dimensional case of linear regression.

In addition to high- dimensions, big data can contain strong nonlinear relationships that render linear estimators unusable (Varian, 2014). Other common OLS pitfalls include outliers, dependence between independent variables (or multicollinearity) and heteroskedasticity. Most seasoned econometrics practitioners are no doubt aware of these issues and may have successfully avoided them in related work. However, when the size of data scales up, the chance that some or all of these effects are present increases as well (Fan, 2014). Discarding observations based on the presence of these effects alone might lead to significant information loss and suboptimal results. One should therefore also consider alternative approaches in the domain of statistical learning.

Outliers. The least squares algorithm is highly sensitive to even single outliers. Abnormal observations of the dependent variable skew resulting coefficients due to the sum of squares estimation. In addition to bias, outliers can also lead to excessively large regression constants. Burger & Repiský (2012) propose replacing least squares with the median of squares and algorithmic outlier detection for adjusting for outliers in linear models. Regularization methods like 'Support Vector Machines' and the 'Lasso'- and 'Ridge- regression that penalize for dependence between predictors are also commonly applied (Varian, 2014; Einav & Levin, 2014).

In practice, extreme outliers are relatively easy to detect with methods like 'winsorization' (clamping extreme values at some specific thresholds; setting limits at 99% of the global maximum and minimum of a variable for example) and domain knowledge. The more subtle outliers can become a problem however as their removal can result in poor model performance and lost information. The process of thorough data understanding and preparation increases in importance as the size of data (and complexity) scales up.

Figure 3: An extreme outlier

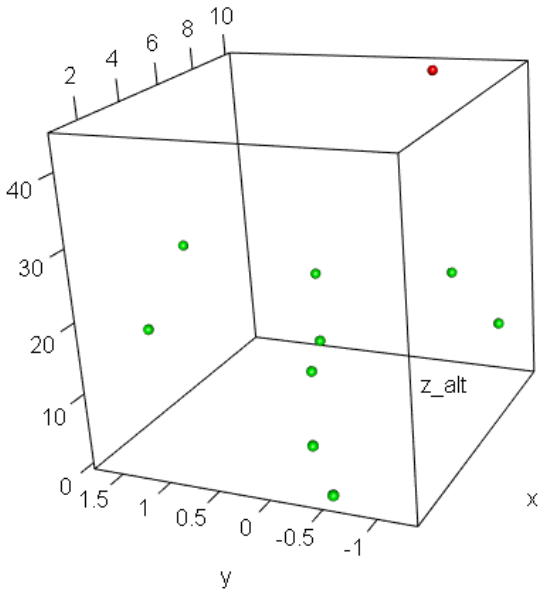
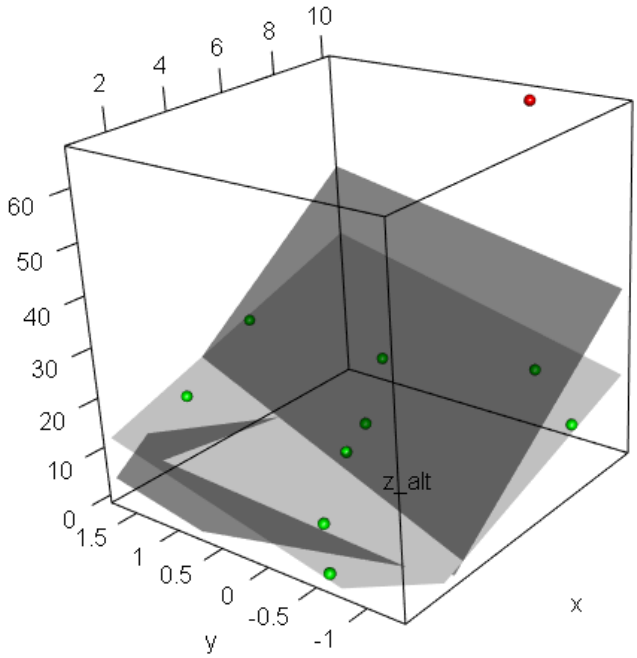


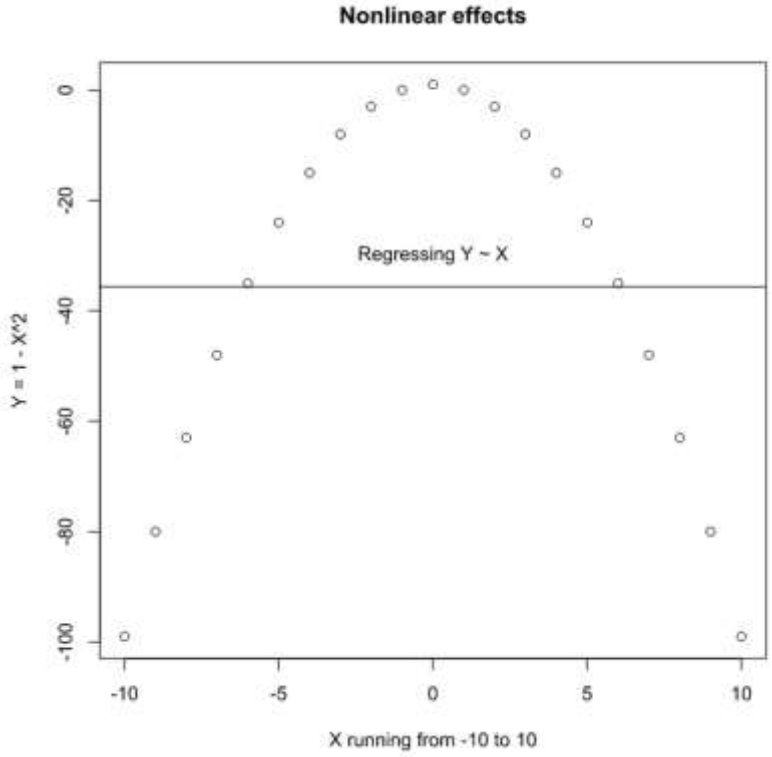
Figure 4: Two regression planes, with and without outlier inference



Nonlinearities. Most systems aren't truly linear. Some relationships just resemble linearity more than others. If the system under study is highly nonlinear, linear models will perform

very poorly. Consider fitting a linear regression to data generated by the equation $Y = 1 - X^2$ for reference. As you see in figure 5, the model completely fails to capture the phenomenon (Burger & Repiský, 2012). A straightforward way to effectively model nonlinear relationships is to apply models that do not make assumptions about the underlying distributions in data. Semi- and nonparametric methods like generalized linear models, decision trees and even neural networks tend to capture complex nonlinear effects relatively well (Caruana & Niculescu-Mizil, 2006).

Figure 5: A nonlinear relationship



Instead of assuming distributions and data structures, these semi- and nonparametric estimators contain different parameters that are then optimized for the data set in question. This is essentially a process of trial and error where holdout sets of the data are used to assess improvements in the system ('cross- validation'). The optimal parameters are 'learned' for each unique problem. The Random Forests algorithm applied in this paper expands upon standard nonparametric decision tree predictors in both stability and predictive power. The need for parameter tuning and exhaustive search to arrive at good results is also reduced. Due

to these properties, Random Forests are considered to be among the best- performing statistical learning algorithms currently available (Caruana & Niculescu-Mizil, 2006).

High- dimensional data. Linear regressions are especially prone to this problem. When the amount of variables (columns) exceeds the amount of training points (rows) in a data set, the least squares algorithm will not be able to provide a unique solution and the model will fail (Burger & Repiský, 2012). Several papers propose different dimensionality reduction techniques and regularized regressions for these situations. Methods like Principal Component Analysis and 'Lasso'- or 'Ridge'- regression are commonly applied (Einav & Levin, 2014). Decision tree approaches and Random Forests are also known to be generally efficient with high- dimensional data (Breiman, 2001; Siroky, 2009).

Dependence among independent variables. Big data brings forth spurious correlations, what is the exact effect of this on linear regression? With many correlations between independent variables, the least squares algorithm can find many solutions that it considers 'equally good' and can't decide upon. This can create very volatile models as in reality some of these alternatives are far from optimal. Let's look at an example by Burger & Repiský (2012) to examine the issue in more detail.

Dependence between independent variables (Burger & Repiský, 2012)

Consider two independent variables X_1 and X_2 that are perfectly correlated. In this case the Ordinary Least Squares algorithm may find multiple 'optimal' solutions that it cannot decide upon. Some of these solutions can lead to extremely biased results.

Assume regression equations Y_1 and Y_2 for this example as follows

$$Y_1 = 0.5X_1 + 0.5X_2$$
$$Y_2 = 1000X_1 - 999X2$$
$$when\ X_1 = X_2$$
$$Y_1 = 0.5(X_1) + 0.5(X_1) = X1$$
$$Y_2 = 1000(X_1) - 999(X_1) = X_1$$
$$Y_1 = Y_2$$
$$\frac{\partial Y_1}{\partial X_1} = \frac{\partial Y_2}{\partial X_1}$$

When X_1 is perfectly correlated with X_2, $Y_1 = Y_2$ and the slope for Y in X_1 is the same for both equations. However just slight alterations in the relationship between X_1 and X_2 results in heavily biased predictions for Y_2.

$$if\ X_1 = 0.99X_2$$
$$Y_1 = 0.5(X_1) + 0.5(0.99X_1) = 0.995X_1$$
$$Y_2 = 1000(X_1) - 999(0.99X_1) = 10.99X_1$$
$$Y_1 \sim Y_2/11$$

The first model (Y_1) has a relatively robust reaction to the alteration but the alternative (Y_2) greatly exaggerates the output. High dependence among predictor variables can therefore lead to extreme coefficient variations in linear regression and very small variations in sample data can cause large swings in predictions.

Heteroskedasticity. Heteroskedasticity is present when some data points in a data set are more likely than others to be affected by noise. More formally, data points in a given data set have unequal variances in their values along the feature axis (Burger & Repiský, 2012). Consider the relationship of earnings and age for example. Annual earnings for adults are likely to vary a lot more than those for kids. For age and height, the relationship is inverted. In both cases the level of noise in data is unequally split between different regions of the feature space and least squares estimators have trouble in arriving at an optimal result.

Burger & Repiský (2012) propose weighting observations for the least squares algorithm based on the magnitude of heteroskedasticity, this approach is known as the weighted least squares. Weighted least squares constrains parameter estimation to reduce the bias originating from heteroskedastic samples. Bagged decision trees and Random Forests can also effectively deal with heteroskedasticity due to the majority voting- or averaging process. In these approaches biased trees fit on a heteroskedastic subset of data are outweighed by the results of other trees, leading to an unbiased end result (Breiman, 2001; Payne, 2014).

Limiting data resources and potential variables based on theory alone might lead to systemic error and suboptimal results when working with large data sets. Then again, more complex data sets complicate analysis and specific situations arise where classic econometric methods give biased results. Regularization methods provide efficient tools in controlling for outliers, high- dimensionality and dependence between independent variables. Nonparametric estimators offer solutions for problems with nonlinearity and heteroskedasticity. To excel in econometric analysis with modern data sets, one should integrate these methods in his workflow.

A great deal of effort has been put in developing modern algorithms for data analysis in statistics and machine learning. Applications in econometric studies are still scarce however (Taylor et al., 2014). When correctly applied, the methods discussed here offer solutions to new and perhaps even some old problems where standard econometric models have been ineffective or biased. The end goal in this paper is to shed light in these tools, present a theoretical framework for incorporating decision trees and Random Forests into the econometrics workflow and possibly raise an interest towards a 'new approach' to empirical economics.

2.5 A rising interest in machine learning

Up until the last few years, machine learning hasn't been very popular among applied economists (Varian, 2014; Taylor et al., 2014). This is mostly due to a different approach to modelling: Instead of causal inference and bivariate relation analysis, machine learning focuses purely on prediction. Maximizing predictive power is effectively a shift from including a strictly supervised set of variables (econometrics) to simply including every variable that significantly reduces some error term on a hold-out sample of the data. Also, using nonparametric estimators and building model ensembles can lead to complex systems that are hard to decompose. For econometric purposes, an additional layer of complexity is added in inference. As we are about to realize however, this increase in complexity does not make causal inference or treatment- control analysis impossible. When correctly applied, machine learning can actually improve upon linear estimators for causal analysis by enabling it on bigger and more complex data sets (Athey & Imbens, 2015).

Several econometrics papers and talks during the last few years signal a sparked interest in machine learning (Taylor et al., 2014; Varian, 2014; Einav & Levin, 2014). The earliest adopters and frontrunners in this field include Stanford Professors Susan Athey & Guido Imbens and Google's chief economist Hal Varian. Even if the field is relatively new and a lot of papers are still 'work-in-progress', there's some highly interesting literature available. Current research is mostly focused around three high-level topics: (1) The opportunities and challenges that big data brings to econometrics, (2) actual machine learning algorithms that could potentially be applied to solve econometric problems and (3) the changes that must be made to adjust machine learning towards the end goal of econometrics, causal inference.

The need for an update in econometric methods for big data analysis is generally agreed upon. Many researchers see new opportunities in modern data sources and expect empirical economics to evolve in the coming years (Varian, 2014; Einav & Levin, 2014). Some have strong beliefs that machine learning will be key, while others stress the need to further develop and adjust the current methods available (Taylor et al., 2014).

Current literature identifies a specific subset of machine learning methods for econometric applications. These are regularized regressions, decision trees and model- testing procedures like cross- validation. Causal inference with machine learning is also a key research area and the contributions on it by Athey & Imbens (2015) establish a framework to incorporating learning algorithms in econometrics. The methodological focus in this paper is restricted to decision tree approaches as they're well represented in existing literature. The literature will be naturally extended in this part by introducing Random Forests, an ensemble decision tree algorithm that offers several interesting properties for econometrics.

The following chapter provides an overview of the core methods selected for this paper. Machine learning is first defined along with its main objectives, different approaches and the types of problems it's commonly used for. A brief history of classification- and regression trees follows along with their specific strengths and weaknesses. The cross- validation procedure for model tuning and testing is also reviewed. Finally, ensemble methods and Random Forests are introduced to expand upon the strengths of standard decision trees in statistical learning.

The goal of the overview chapter is not only to compare machine learning tools, but to understand them and realize both their potential value and weaknesses in econometric analysis for big data, especially in the case of Random Forests. The final literature review chapter adds a finishing touch to the conceptual framework used in the empirical part of this paper by summarizing recent literature in causal decision tree applications.

After the theoretical framework is set up, a Random Forest classifier is built on the American Community Survey Public Use Microdata Sample to find factors that drive earnings for the top 1% of population in the United States. The 5- year aggregate ACS data serves as a good test for the methods reviewed. It features high dimensionality, various data types and possible multicollinearity.

3 MACHINE LEARNING IN ECONOMETRIC APPLICATIONS

3.1 Machine learning explained

Machine learning is a broad field that consists of topics like artificial intelligence, computer vision and statistical learning. Statistical learning, the subset of machine learning with lucrative applications for econometrics, is an umbrella term for different predictive and prescriptive methods for data analysis. These methods are the building blocks for most intelligent services and applications in the modern economy. Fields like finance, advertising and telecom feature some of the earliest applications in statistical learning to support day-to-day business activities (Hastie et al., 2009)

Statistical learning divides into two distinct families of algorithms depending on whether the goal is to model a known phenomenon or to describe previously unknown relationships and structures in data. Supervised learning refers to cases where a phenomenon of interest is identified and labelled as a data set consisting of a target (or dependent) variable and predictors (or independent variables). Supervised methods can be further divided into classification models where the output variable is categorical (or binary) and regression models where the output is continuous. In econometrics, classification models like the logistic regression are often referred to as binary response models (Horowitz & Savin, 2001). Most applications in econometrics fall in the supervised learning category. Some other examples include:

1. Predicting movements in stock prices and finding other interesting correlations (https://cloudplatform.googleblog.com/2016/03/TensorFlow-machine-learning-with-financial-data-on-Google-Cloud-Platform.html).
2. Assessing the probability for a patient having a certain eye disease based on images of the retina; 'Diabetic Retinopathy Detection', a past competition in data mining at www.kaggle.com funded by the California Healthcare Foundation.
3. Traditional credit scoring in banks (Khandani et al., 2010).
4. Recommendations at Spotify & Netflix (http://techblog.netflix.com/; https://labs.spotify.com/).

It's worth to note that even if this paper is focused around decision tree applications, supervised learning is a broad field that includes a wide- array of different algorithms for a variety of tasks. The topics covered here include decision trees, Random Forests and bootstrap resampling, or 'bagging'. Other important methods in the field include Support Vector Machines, boosted decision trees (via Adaboost or Gradient Boosting) and Neural Networks to name a few. See the book 'The Elements of Statistical Learning' by Hastie, Tibshirani and Friedman (2009) for reference.

Figure 6: An overview of machine learning techniques
(http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/)

Unsupervised learning refers to methods that are used to find previously unknown relationships and structures in data. Common applications include tasks like customer segmentation for marketing and market basket analysis. The K-Means clustering algorithm (and its variations) which finds similar observations and groups them by minimizing the within cluster sum of squares is among the most applied methods in this area. Other unsupervised learning methods include association and sequence mining techniques (like the 'Apriori' algorithm) for market basket analysis and a more general 'Self Organizing Maps' algorithm based on neural networks that produces a low- dimensional discrete representation

of the input space for a given data set. Unsupervised learning can offer interesting tools for economists in the initial data analysis phase, but in this paper they are left out of consideration (Hastie et al., 2009).

The focus on this section is to review the classification- and regression tree family of supervised learning methods along with model- testing and fitting procedures like cross-validation and bootstrap resampling. The concept of model combinations, or ensembles, is then introduced and the Random Forest algorithm is proposed for econometric analysis. The goal is to provide the reader with enough knowledge to be able to compare nonparametric tree estimators with standard econometric methods and understand their corresponding strengths and weaknesses for economic analysis. The basis for causal inference in this study is established by reviewing the study on causal decision trees by Athey & Imbens (2015).

3.2 Why decision trees and Random Forests?

The motivation behind choosing Random Forest specifically relates to its generalizing ability and strong performance in multiple studies (see Figure 8). The algorithm provides several options for variable selection, offers a way to analyse bivariate relationships between dependent and independent variables in a ceteris paribus (all else held constant) setting and avoids over-fitting through bootstrap resampling. In other words, it's a natural extension to machine learning methods for econometrics as it builds upon single Classification- and Regression Trees already proposed in the literature.

Table 1: Description of the data sets for algorithm testing
(Caruana & Niculescu-Mizil, 2006)

PROBLEM	#ATTR	TRAIN SIZE	TEST SIZE	%POZ
ADULT	14/104	5000	35222	25%
BACT	11/170	5000	34262	69%
COD	15/60	5000	14000	50%
CALHOUS	9	5000	14640	52%
COV_TYPE	54	5000	25000	36%
HS	200	5000	4366	24%
LETTER.P1	16	5000	14000	3%
LETTER.P2	16	5000	14000	53%
MEDIS	63	5000	8199	11%
MG	124	5000	12807	17%
SLAC	59	5000	25000	50%

Caruana and Niculescu- Mizil (2006) have studied the predictive power of multiple supervised learning methods in several different learning problems (Tables 1 & 2). The methods compared include Support Vector Machines (SVM), Naïve Bayes (NB), Artificial Neural Networks (ANN), Logistic Regression (LR), Boosted (BST-DT) and Bagged Decision Trees (BAG-DT) and Random Forests (RF). The values in Fig. 8 are normalized averages of 8 different accuracy metrics for each algorithm on a specific problem. The columns from 'COVT' to 'BACT' refer to individual problems and data sets used to compare these methods, problem specifics are found in Fig. 7. The 'Mean' column refers to an average accuracy per algorithm across all 11 learning problems. The column 'CAL' refers to the model calibration method (Platt- scaling, Isotonic Regression or no calibration). The problems used for testing were all binary classification problems, but the data contents and features varied considerably.

Table 2: A comparison of supervised learning methods
(Caruana & Niculescu-Mizil, 2006)

MODEL	CAL	COVT	ADULT	LTR.P1	LTR.P2	MEDIS	SLAC	HS	MG	CALHOUS	COD	BACT	MEAN
BST-DT	PLT	**.938**	.857	**.959**	**.976**	.700	.869	**.933**	.855	**.974**	**.915**	.878*	**.896***
RF	PLT	.876	.930	.897	.941	**.810**	.907*	.884	.883	.937	.903*	.847	.892
BAG-DT	–	.878	.944*	.883	.911	.762	.898*	.856	**.898**	.948	.856	**.926**	.887*
BST-DT	ISO	.922*	.865	.901*	.969	.692*	.878	.927	.845	.965	.912*	.861	.885*
RF	–	.876	.946*	.883	.922	.785	.912*	.871	.891*	.941	.874	.824	.884
BAG-DT	PLT	.873	.931	.877	.920	.752	.885	.863	.884	.944	.865	.912*	.882
RF	ISO	.865	.934	.851	.935	.767*	**.920**	.877	.876	.933	.897*	.821	.880
BAG-DT	ISO	.867	.933	.840	.915	.749	.897	.856	.884	.940	.859	.907*	.877
SVM	PLT	.765	.886	.936	.962	.733	.866	.913*	.816	.897	.900*	.807	.862
ANN	–	.764	.884	.913	.901	.791*	.881	.932*	.859	.923	.667	.882	.854
SVM	ISO	.758	.882	.899	.954	.693*	.878	.907	.827	.897	.900*	.778	.852
ANN	PLT	.766	.872	.898	.894	.775	.871	.929*	.846	.919	.665	.871	.846
ANN	ISO	.767	.882	.821	.891	.785*	.895	.926*	.841	.915	.672	.862	.842
BST-DT	–	.874	.842	.875	.913	.523	.807	.860	.785	.933	.835	.858	.828
KNN	PLT	.819	.785	.920	.937	.626	.777	.803	.844	.827	.774	.855	.815
KNN	–	.807	.780	.912	.936	.598	.800	.801	.853	.827	.748	.852	.810
KNN	ISO	.814	.784	.879	.935	.633	.791	.794	.832	.824	.777	.833	.809
BST-STMP	PLT	.644	**.949**	.767	.688	.723	.806	.800	.862	.923	.622	.915*	.791
SVM	–	.696	.819	.731	.860	.600	.859	.788	.776	.833	.864	.763	.781
BST-STMP	ISO	.639	.941	.700	.681	.711	.807	.793	.862	.912	.632	.902*	.780
BST-STMP	–	.605	.865	.540	.615	.624	.779	.683	.799	.817	.581	.906*	.710
DT	ISO	.671	.869	.729	.760	.424	.777	.622	.815	.832	.415	.884	.709
DT	–	.652	.872	.723	.763	.449	.769	.609	.829	.831	.389	.899*	.708
DT	PLT	.661	.863	.734	.756	.416	.779	.607	.822	.826	.407	.890*	.706
LR	–	.625	.886	.195	.448	.777*	.852	.675	.849	.838	.647	.905*	.700
LR	ISO	.616	.881	.229	.440	.763*	.834	.659	.827	.833	.636	.889*	.692
LR	PLT	.610	.870	.185	.446	.738	.835	.667	.823	.832	.633	.895	.685
NB	ISO	.574	.904	.674	.557	.709	.724	.205	.687	.758	.633	.770	.654
NB	PLT	.572	.892	.648	.561	.694	.732	.213	.690	.755	.632	.756	.650
NB	–	.552	.843	.534	.556	.011	.714	-.654	.655	.759	.636	.688	.481

The general result is obvious, different ensemble tree methods topped the chart across problems. In this study, Boosted Trees (BST-DT) take first place by edging out Random

Forests (RF) with a 0.04 percentage point difference in average accuracy. A simpler version of Random Forests, Bagged Decision Trees (BAG-DT), comes third. The key observation here is that when compared to other methods, Random Forests (and other tree ensembles) offer consistent predictive power over all the problems under study. Logistic Regressions for example appear to offer top results for the 'BACT' data set, but for the unbalanced problem 'LETTER.P1' (details in Fig. 8) they are at the bottom of the list (Caruana & Niculescu-Mizil, 2006).

The stable and accurate results of Random Forests persist in a later study by Caruana et al. (2008), where supervised learning methods are compared for high- dimensional problems. In this study, Random Forests offer the most accurate and stable results, followed by Artificial Neural Networks, Boosted Trees and Support Vector Machines. The robustness of decision tree methods and Random Forests over different data sets and problems support their applicability in the big data environment for econometrics.

The following chapters provide a detailed overview of decision tree algorithms, ensemble models and Random Forests.

3.3 Classification- and Regression Trees

CART, or Classification- and Regression Trees, is a classic decision tree learning method initially introduced by Breiman et al. in 1984. Adjusted CART- trees are used as the 'base-units' (or 'learners') in many modern applications of statistical learning. In fact, model ensembles like Random Forests and Boosted Decision Trees can contain hundreds or thousands of singular trees that are built with different regulations and procedures. The ensemble models' predictions are then averages of the base learners' predictions for regression problems or majority votes for classification problems. With specific restrictions in the base learner fitting process, these types of ensembles can offer far greater predictive power and generalizing ability when compared with a single decision tree (Caruana & Niculescu-Mizil, 2006). Understanding CART serves as a starting point for working with Random Forests.

The basic idea behind CART is fairly straightforward. A set of observed predictors is used to recursively partition the data until the values of the response variable become homogenous

within each sub- partition. The measure of contribution towards this homogeneity can be used as an indicator of variable importance. These 'impurity' measures will be discussed in further detail later. CART uses binary splits (i.e. splitting on one variable at a time) to partition the tree. The best partitioning variable at each split is determined by minimizing the sum of squared errors in regression or alternatively finding the predictor that best splits the response variable into separate classes in classification (Siroky, 2009). When the best split is found, the partitioning continues until some stopping rule is realized. 'Node size', i.e. the amount of observations inside each partition could be used as a stopping rule for example: Using a node size of 5 stops the partitioning process when the less than five observations are available at a specific node. The end result for a CART tree for both classification and regression can be summarized as a series of different logical conditions (Fig. 9) and the model response is a constant based on observations on the final node (mean for regression, vote for classification). Hastie et al. (2009) provide a formal definition:

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

The model response is a constant c_m in each region R_m (Hastie et al., 2009).

Figure 7: The CART partitioning process (Hastie et al., 2009)

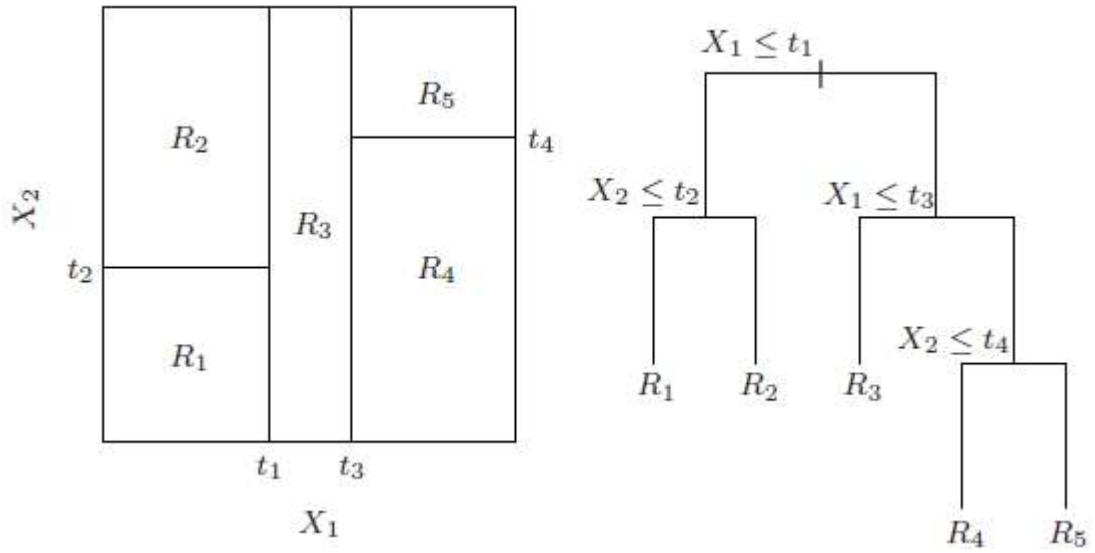


Figure 9 provides two representations of the same CART partitioning in a case with two predictors (X_1 and X_2). As visualized on the leftmost picture, CART can effectively deal with

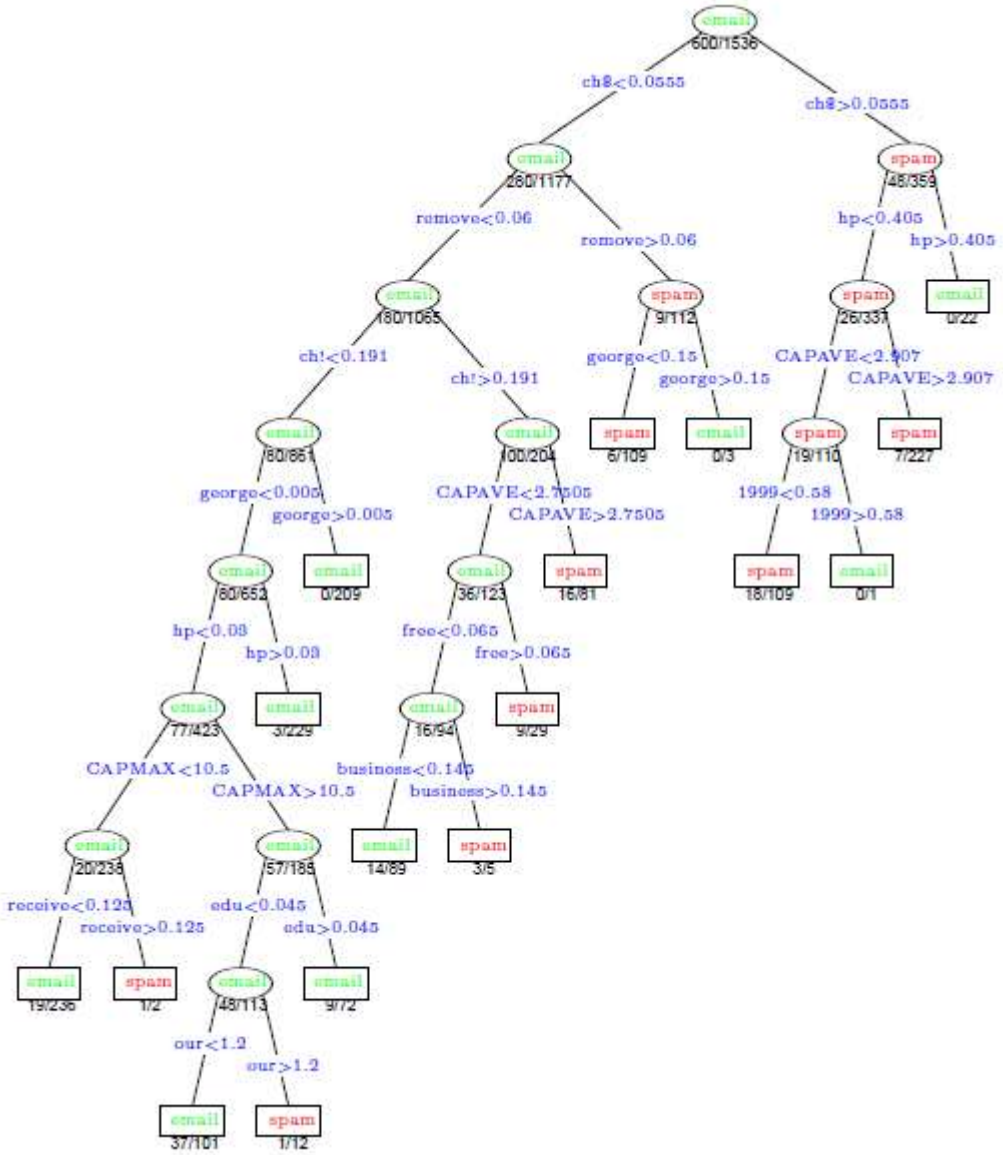
nonlinear relations by partitioning observations in homogenous sub- groups. The decision tree diagram on the right is an insightful visualization of the splitting process used to arrive at partitions $R_1,\ldots,R_5$. One can easily observe the most important variables at each split in determining a set outcome, starting from the top with the full set of data (Hastie et al., 2009).

The optimal size of a tree remains a problem; big trees tend to over-fit and fail to generalize the structure while small trees may be unable to capture essential details in data. In CART, this process is often handled by first fitting a maximum size tree based on some stopping rule and then 'pruning' it. Pruning uses a function to find the weakest node (or split) in a tree in terms of complexity versus information added and then collapses it. Optimized pruning is a delicate process that varies between data sets. A method called 'cross- validation' is used to arrive at the optimal tree structure (Siroky, 2009). For more information on pruning, see Hastie et al. (2009).

Cross- validation is the main method to optimize the generalizing ability of a model and is therefore crucial for all machine learning applications. In k- fold cross- validation, a data set is split into k partitions of (approximately) equal size. In order to optimize some parameter x in a model, the model is fit k times on k-1 partitions of the data and the error rate is evaluated on the hold-out partition. The parameter value that minimizes the hold-out error rate is then optimal in terms of generalizing ability in the model. In the case of CART, this is the correct way to arrive at an optimal tree- depth through the pruning- process (Siroky, 2009).

CART's main strengths are its informative variable importance measures (impurity) and the ability to handle practically all variable types and highly nonlinear relations between them. In addition, CART's results are easily interpretable through visualizations and binary splits. Its main weaknesses are in the need for careful cross- validation to arrive at optimally pruned trees, poor results in low dimensions, instabilities and discontinuous boundaries (Siroky, 2009). At the cost of visually interpretable results, the Random Forests algorithm proposed in this paper extends the capabilities of CART and effectively eliminates some its biggest problems.

Figure 8: A CART classifier detecting spam messages in email (Hastie et al., 2009)



3.4 Model ensembles: Boosting, Bagging & Stacking

Ensemble models represent the high-end of statistical learning methods in terms of predictive power. Like observed in the study comparing supervised learning methods by Caruana & Niculescu-Mizil (2006), model ensembles tend to outperform their simpler counterparts by a large margin. The logic behind combination models is in fact relatively simple: As different methods and algorithms tend to learn certain relationships or structures differently, combining the outputs of these models can lead to a system with more information and therefore better predictive power. The same goes for combinations of single algorithm fits on

different subsets of a data set. There are several approaches to building model ensembles like Stacking, Boosting and Bagging and the choice of method depends on preferences and the type of question at hand (Hastie et al., 2009).

Stacking refers to a 'meta-learning' ensemble method where different models are combined by a second- level model trained on new data (same features and first- level model responses). This second level model takes the predictions of first-level models as input and predicts a final output. Stacking is used to enhance predictive power through learning specific situations where certain first- level models perform well or the opposite. The 'stacker' is often a linear model (Hastie et al., 2009). *Boosting* on the other hand is based on the idea of combining several 'weak' classifiers to form a 'strong' classifier. In 'AdaBoost' (a commonly applied boosting algorithm), this is done by fitting a model to data, reweighting the data based on poorly classified observations, fitting a second model, repeating and so on. The end result is a series of models where each model captures some specific relationship in the data very well (due to weighting), but may perform relatively poorly on the whole data set alone. The end result of AdaBoost is a majority vote by all classifiers, which tends to be far more accurate than any single model in the series (Freund & Schapire, 1999). 'Gradient Boosting' is a newer boosting algorithm based on differentiable loss functions as model combiners (Hastie et al., 2009). Gradient Boosted Decision Trees (GBDTs or GBMs) are highly common in modern data mining applications, see 'www.kaggle.com' for reference.

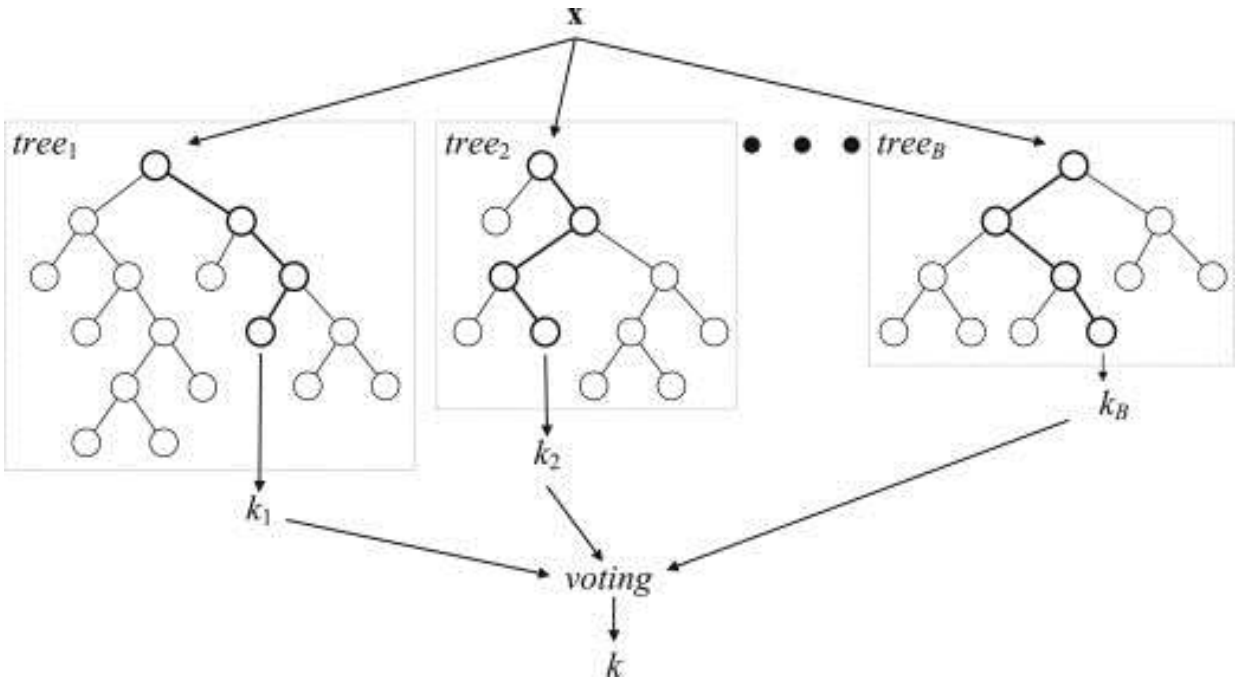
Stacking and Boosting are both widely applied ensemble methods, but can also be unstable and lead to over- fitting when incorrectly applied. *Bagging* (bootstrap resampling) brings an alternative approach to ensembles by building models on subsets of the data through resampling with replacement (Stephenson et al., 2010). Bagged regressions average the prediction while classifiers do a majority vote. By averaging or voting results from models fit on different subsets of the data, predictions become more resistant to noise and models are less likely to overfit. Bagging was among the first methods for decision tree combinations and serves as the starting point for Random Forests (Siroky, 2009).

3.5 Random Forests

Originally introduced by Leo Breiman in 2001, Random Forests (and more generally decision forests) is a combination of tree predictors in which each tree is built on a unique subset of

the data with random variable selection. The development of Random Forests was motivated by the promising results in earlier tree- bagging (Breiman, 1996) and –boosting (Freund & Schapire, 1999) applications. Random Forests are known to be highly resistant to over- fitting and to effectively handle noise and heteroskedasticity (Payne, 2014). They also expand upon the strengths of standard decision tree predictors in detecting nonlinear relationships and working with high- dimensional data sets (Siroky, 2009; Caruana et al., 2008). Unlike most modern algorithms in statistical learning, Random Forests maintain stability across data sets without a need for an excessive parameter tuning process (Caruana & Niculescu-Mizil, 2006). In econometrics, Random Forests have been applied in GDP forecasting (Biau & D'Elia, 2011).

Figure 9: A conceptual visualization of the Random Forests algorithm (Nguyen et al., 2013)



Random Forests build multitudes of individual trees on unique data subsets through bootstrap resampling with replacement. Unlike CART, the tree predictors in Random Forests are built in full- length without pruning. In addition to sampling unique data for each tree predictor, an additional layer of randomness is added at each node by randomly selecting a subset of variables to split on. The resulting 'forest' is then a combination of individual trees built on randomly selected observations and variables. This process greatly reduces the dependence between individual trees and thus promotes the generalizing power of the model. The effects

of heteroskedasticity, outliers and other data- anomalies are also reduced due to the large amount of individual tree learners and majority voting (Breiman, 2001).

Random Forests provide unbiased estimates of model generalization error through testing on 'out-of-bag' observations, removing the need for a separate testing set. As the bootstrap resample for tree k_i in a forest consists of approximately two thirds of all observations, the last third of the cases remains unused for this tree. This set of unused data is unique for each tree $k_1,\ldots,k_N$ in the forest. To arrive at the unbiased error estimate, Random Forests puts these out-of-bag observations through all the trees where they weren't part of the initial training sample. Approximately one third of the forest is then used to vote for each out-of-bag sample. The proportion of times this classification doesn't equal the actual observed value becomes the out-of-bag error rate (Breiman, 2001).

The pseudocode for Random Forests can be defined as follows (Siroky, 2009):

---

1. For $B_i, i = 1, \ldots, B$
   a. Draw a bootstrap sample S of size N from the training data.
   b. Grow an unpruned random-forest tree, T_b using the bootstrapped data, until the minimum terminal node size, n_{min}, is obtained by recursively following the sub-algortihm.
      Randomly select p variables from the total set of P variables.
      Select the optimal variable/split-point among the p variables
      Split node into two daughter nodes.
2. Output ensemble of trees $\{T_b\}_1^B$
3. Predict new observations, or out-of-bag observations
   For regression: $\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$
   For classification: Let $\hat{C}_b(x)$ be the class prediction of the b^{th} random forest tree $\rightarrow \hat{C}_{rf}^B(x) =$ majority vote $\{\hat{C}_b(x)\}_1^B$

The number of variables to try on each node of the forest, or 'mtry', is the main tuning parameter in the model. The base values for these parameters in most Random Forest implementations are sqrt(p) for regression and p/3 for classification (where p is the total amount of predictors in the model), as Breiman (2001) suggests. The out-of-bag error can often be at least somewhat enhanced by optimizing mtry through cross- validation. One can also experiment with the node size stopping rule as it can help control overfitting in noisy

data. One should remember however that highly randomized variable selection enhances generalization by reducing competition between highly correlated variables and variables that only fit a subset of the data well. Reducing the amount of randomization in the splitting process can therefore negatively affect model performance when these effects are present. Altering the size of trees in the forest is also available through the 'ntree' parameter. The Law of Large Numbers ensures that big Random Forests will not over- fit (Breiman, 2001). When adding trees has a significant impact on the out-of-bag error rate, it's justified to grow the forest further. Generally, keeping the forest 'large enough' will always guarantee unbiased estimates of the generalization error while reducing trees effectively reduces the degree of randomization and can lead to increased bias (Siroky, 2009).

Random Forests have several interesting properties in terms of modern econometric applications. In addition to the generalizing ability and high stability across data sets of different types and structures, the 'impurity' measures of variable importance and partial dependence plots for bivariate relation analysis offer effective tools for dissecting complex data sets.

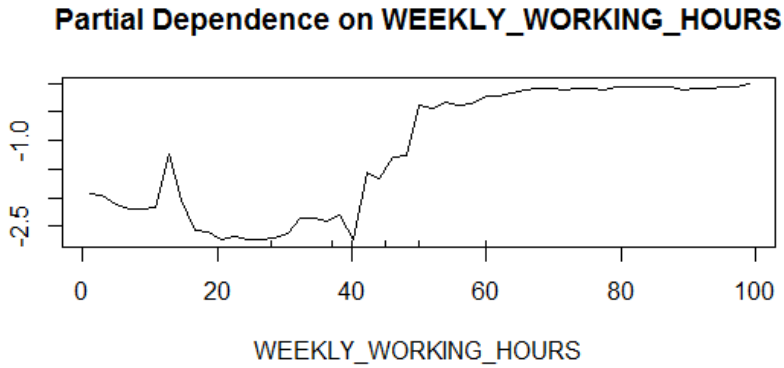
Impurity measures (for classification) in decision trees rate variables based on their ability to divide observations into homogenous sub- groups during the tree- splitting process. As belonging to the 99th percentile in earnings is a binary- response problem, this paper bases variable selection on the 'Gini impurity' importance measure. Siroky (2009) defines Gini impurity as "The sum of all reductions in the forest due to the ith variable, divided by the total amount of trees in the forest". In more general terms, Gini impurity measures how often a randomly chosen observation would be incorrectly labelled if the labelling was a random selection based on distributions in data, instead of a selection based on reductions in heterogeneity within tree partitions.

$$I_{x_i} = \frac{1}{k} \sum_{z} [d(i,z) I(i,z)]$$

The above equation is the formal definition of impurity measures in decision trees. Here, z is a node in each tree that relies on a heterogeneity index and d(i,z) is a decrease in heterogeneity at node z induced by x_i. I(i,z) is an indicator function that equals 1 when the ith

variable is selected for a split at node z. x_i is chosen for the split from the randomly selected set of variables X_w at node z, if d(i,z) > d(w,z). It's worth to note that as standard statistical tests tend to fail in variable selection when data is high- dimensional (Grömping, 2009), impurity measures in Random Forests can overcome this issue by averaging the overall information gain per variable across the forest.

Figure 10: A partial plot of high earnings and weekly working hours



Random Forests allow bivariate relationship analysis between the outcome- and predictor variables through partial dependence plots. The method is based on a 'ceteris paribus'-setting, where all other factors are held constant to compute the marginal effects of individual variables. Figure 12 contains an example of partial dependence plotting in the 'randomForest' R- package. The data in Fig. 12 is derived from the model built on American Community Survey data in chapter 5. The dependent variable is the average weekly hours worked in the past 12 months. The y- axis signals the estimated effect in changed likelihood for being among the top 1% of population in earnings based on changes hours worked, holding all other factors constant. Hastie et al., (2009) define the partial dependence function for regression as:

$$\tilde{f}_S(X_S) = \frac{1}{N}\sum_{i=1}^{N} f(X_S, x_{iC}),$$

Where Xs is the predictor (or dependent variable) on which partial dependence is estimated. $\{X_{iC}\}$ are the X_c values from the Xs in training data. The partial dependence plot shows the

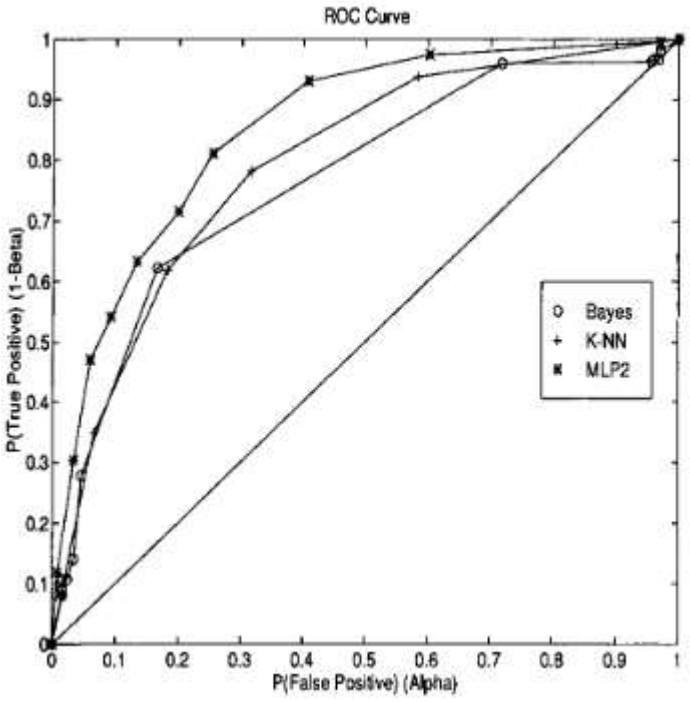
effect of X_s on f_x after adjusting for the effect of X_c on f_x. In other words, partial dependence functions show how changing a specific independent variable affects the corresponding dependent variable after adjusting for the average effects of other independent variables (Hastie et al., 2009). As such, partial dependence plots provide a useful method in interpreting the effects captured in Random Forest models and work as an effective baseline evaluation for dissecting the models built in chapter 5.

3.6 Model selection based on 'Area Under the Curve'

Model evaluation based on the AUC measure, or 'Area Under the ROC Curve' is the final machine learning concept to review before moving on to empirical analysis. As we're modelling the top 1% in income, we're effectively dealing with what's called a 'very rare phenomenon' in statistics (King & Zeng, 2001). This is when standard metrics like classification accuracy might not tell much about how well the model built actually distinguishes between classes. As a simple example: If one stated that no individual belongs to the highest earning 1% of population, the accuracy of this statement would be identically 99%. The Area Under the ROC Curve is a better measure of how well the model separates between different classes, and is used here as a basis for model selection and parameter optimization.

In layman's terms, the AUC measure estimates how likely it is for a true positive to receive a higher prospensity score than a true negative when estimated with a classifier. Therefore an AUC of 0.85 would indicate that there's a high probability that the corresponding classifier distinguishes between true positives and true negatives. AUC is scaled between 0.5 and 1 with a random guess getting 0.5 and perfect information yielding an AUC of 1. Practically an AUC of over 0.6 would indicate that there is a detectable phenomenon in the data, 0.7 would indicate an 'acceptable' model, 0.8 a 'good' model and so on. AUC is generally recommended as the main metric for analysing binary classification models (Bradley, 1997). The detailed mathematical concepts of estimating AUC are not reviewed here, but Bradley's (1997) article "The use of the area under the ROC curve in the evaluation of machine learning algorithms" is a good resource for further reading.

Figure 11: ROC curves (Bradley, 1997)



Looking at figure X from Bradley (1997), AUC refers to the area of the square covered by specific lines (models). The straight line refers to an AUC of 0.5. The separate lines in this figure represent different machine learning classifiers trained on the same data, these are naïve bayes, k- nearest neighbors and a multi- layer perceptron neural network. In this case it's easy to see that the neural network (MLP2) performs best in all regions and therefore has the highest AUC.

In this paper, AUC is used to arrive at optimal randomization for the Random Forest model. Specifically the mtry- and node- size parameters are adjusted based on AUC gains.

4 ESTIMATING TREATMENT EFFECTS WITH DECISION TREES

4.1 Causal inference in machine learning

A family of machine learning methods has been reviewed for applications in econometrics. Decision tree algorithms and especially Random Forests offer unbiased estimates of model performance through methods like bootstrapping and out-of-bag evaluation. If a hold- out data set represents an accurate sample of the phenomenon under study, building a machine learning classifier or regression model through cross- validation ensures optimal generalization and predictive power. Many of the aforementioned features in big data that spell trouble for linear estimators can be dealt with by choosing the correct ML algorithm and optimizing the model- fitting process. However machine learning systems aren't exactly geared towards causation (but correlation) and don't therefore directly provide support for the end goals of many econometric studies: treatment effects and causal relations. This chapter concludes the literature review by looking at recent approaches to 'causal decision trees' and synthetic treatment effect estimation.

As we've observed, the performance of machine learning algorithms and other statistical methods can be optimized by tuning the model based on some hold- out sample of the data. The big problem in causal inference is the fact that both 'treatment'- and 'no-treatment' outcomes can never be observed for the same unit at the same time. Designing controlled experiments can also be tricky in several econometric problems. Consider the case of college graduation and government investment per student for reference: One person can only graduate or not graduate once at a given point in time. There is therefore no 'ground truth' to test if a treatment effect has been correctly predicted and methods like cross- validation can't directly optimize the accuracy of these treatment predictions (Athey & Imbens, 2015).

In their 2015 paper 'Machine Learning Methods for Estimating Heterogenous Causal Effects', Susan Athey & Guido Imbens introduce different approaches for incorporating treatment effect estimation into decision trees. The 'Two Trees' approach fits the scope of this study best, and is now explained in detail.

4.2 The 'Two Trees' Algorithm

We want to get a sense of the approximate effect of a certain treatment on an individual. Optimally we'd like to rank the treatment effects of different independent variables to provide an efficient base for speculation and conclusions. This requires both a predictive model and the dissection of the most important predictors in terms of treatment effects.

Suppose we build a model that predicts whether one belongs in the top 1% of the U.S. population in terms of earnings. The model is accurate, we find that both a PhD and weekly working hours are statistically highly significant predictors (or informative in terms of the decision tree impurity criterion). So we're now able to distinguish between high- earners and others based on a few details about the individual, that's good. However, which effect has the biggest impact on the outcome? If we hold working hours constant, what's the difference between those with a doctoral degree and those without? In the actual data we never observe these things at the same time for the same individuals. We need to somehow approximate the treatment effects of the most important predictors. This is where the 'Two Trees' algorithm and synthetic treatment effect estimation can be useful.

Athey & Imbens (2015) provide the formal description:

$$Y_i = Y_i(W_i) = \begin{cases} Y_i(1) & if\ W_i = 1, \\ Y_i(0) & otherwise. \end{cases}$$

From this we want to estimate the conditional average treatment effect (CATE):

$$\mu(w, x) = \mathbb{E}[Y_i | W_i = w, X_i = x]$$
$$\tau(x) = \mu(1, x) - \mu(0, x)$$

Here Y_i is the estimated outcome taken the treatment, W_i. We want to estimate the treatment effect t(x) per individual and finally get the average treatment effect by aggregating it over all observations.

The Two Trees algorithm is based on analysing the treatment and control group separately: A treatment model $\mu(1,x)$ is built by fitting a tree to the subset of data where $W_i = 1$, similarly the control classifier $\mu(0,x)$ is built on $W_i = 0$. The synthetic treatment effect is then estimated by $\mu(1,x) - \mu(0,x)$ for each individual. The equation $t(x) = \mu(1,x) - \mu(0,x)$ is called the 'conditional average treatment effect', or 'CATE' (Athey & Imbens, 2015).

In practice, this would mean splitting the training data set into two separate partitions based on a specific treatment. Let's assume we want to analyse the effects of getting a PhD. We split a training data set into partitions A and B, where A only contains people with PhDs and B people with a level of education below PhD. All other predictors in the training data set remain untouched. We now fit a treatment classifier on partition A and a control classifier on partition B. When both the treatment and control models have been estimated, we use both to predict on a third hold- out data set. We can now estimate the average treatment effect from the hold- out predictions with CATE.

The motivation behind working with the Two Trees approach in this paper lies in its effectiveness in situations where treatment effects vary with covariates (Athey & Imbens, 2015). This is exactly the case when looking at high income predictors and working with the American Community Survey data. Let's go back to the example with PhD and weekly working hours: When estimating the treatment effect that getting a PhD has on annual earnings, the effect's magnitude will almost certainly differ between people who work 30-hour weeks and those that work more than 50 hours for example.

Additionally, the Two Trees approach does not require the reverse- engineering of classic decision tree algorithms, and can easily be implemented with forests of trees as well (Foster et al., 2011). In fact, estimating treatment effects with two full decision forests could theoretically enhance causal estimation due to the stability of bootstrapped ensemble models (Breiman, 2001). The specific implementation in R can be found in appendix A, but basically for each treatment and control in question two Random Forest models are built for the corresponding data subsets.

5 PERSONAL INCOME DRIVERS: A RANDOM FORESTS APPROACH

5.1 The American Community Survey

The American Community Survey (ACS) is an ongoing mandatory survey by the United States Census Bureau. The survey is sent to a statistically representative subset of the American population annually and contains highly detailed information for both individuals and households. The information contains but is not limited to: income, education, occupation, race, ancestry, family, disability, region, house type, military history and so on. The motivation to use this data as a base set for income analysis rose from the fact that it allows the simultaneous analysis of hundreds of potential variables in modelling. Additionally, regional files (called 'PUMAs') can be joined to the ACS data to include aggregated economic indicators for individuals or households for example.

The most granular level of the ACS data available is called 'PUMS', or the 'Public Use Microsample Data'. The records in this data refer to individual people and households. The information has been anonymized and at parts transformed so no person can be identified based on this set alone, extremely high incomes have been 'top coded' for example. The PUMS is a sample of actual answers to the survey and contains variables for nearly every question (this adds up to approximately 200 variables). The PUMS files are available in 1-, 3- and 5- year sets, where the biggest 5- year set contains information from approximately 5 percent of the United States population. In this paper, the biggest 5- year PUMS set (concerning years 2009- 2013) is used for modelling and model generalization error is evaluated by testing on a validation set that spans the whole 5- year timeframe.

The ACS data sets are freely available through the US Census portal 'www.census.gov'. The data can be accessed in readily tabulated format for specific use cases, downloaded as excel, ,stata- or comma- separated files from the site or through the ftp service 'ftp2.census.gov'. For the ftp option, username 'anonymous' has to be used and no password is required. For the bigger PUMS data sets, ftp options are greatly recommended as compressed files can add up to 5 gigabytes or more.

5.2 Data collection, aggregation and the research question

The PUMS data collection process was very straightforward. The 5- year files for years 2009-2013 were first downloaded through the Census Bureau ftp service and then uploaded into Google BigQuery, a column- oriented database in the Google Cloud Platform. The data could now be accessed and manipulated via an SQL- like language in BigQuery. As the latest 5-year PUMS data takes approximately 20 gigabytes of disk space when uncompressed, some database approach is recommended for analysis. A distributed environment is definitely not a requirement and a local database- engine like 'sqlite' could work as a simple alternative ('www.sqlite.org').

The PUMS data consists of separate records for housing and population units that can be joined by a unique row identifier, 'SERIALNO'. The data comes with a 205- page 'data dictionary' that describes each variable in the population- and housing sets. The housing- set was not included in this analysis as the focus here is on individual factors instead of household averages. For studies on poverty however, the housing record offers multiple potential variables concerning topics like family and apartment size, rent, gas prices etc.

The optimal environment for validating a machine learning method for econometrics would ideally include a big data set with a large number of potentially important variables. A theoretical relationship to the dependent variable should be available for most variables, but specific assumptions about the nature of these relationships (linear, nonlinear etc.) need not be made. Additionally, some potentially redundant variables could be included to find new interesting correlations and at the same time see how the model deals with noise and increased dimensions. For the scope of this study, the ACS PUMS was a perfect match. It's also worth to note that the ACS data supports state- level joins to other third- party data sources, and could work as basis for further expanded research in income or poverty for example.

The goal is to find drivers that influence extremely high incomes for individuals (based on a large set of personal factors) and then examine these relationships in more detail. This is reached by building a Random Forest classifier and minimizing its prediction error on out-of-bag samples and a hold-out testing set. A low generalization error on a hold- out set consisting of entirely new people and spanning multiple years can be considered to be a

strong signal that the effects learned have real support. The most influential variables can be derived from this classifier based on the Gini- impurity measure, and used as a basis for the Two Trees approach to treatment effect estimation. Considering the way Random Forests work, the importance of redundant variables should diminish as forest size grows (Breiman, 2001). An additional logistic regression is also run to compare between traditional econometric approaches to binary response problems and Random Forests.

It is interesting to see what variables remain for the final set, as in how does the Gini criterion rank working hours, education and occupation (factors that each individual can decide upon) versus factors like race, nativity and sex (traits that cannot be altered). Further analysis can then take place to rank predictors in terms of treatment effect magnitude and draw conclusions.

5.3 Creating a subset of the ACS data for income analysis

To arrive at a valid data set, certain parameters in the PUMS data were controlled: The variable 'ESR', or 'EMPLOYMENT STATUS RECODE', was restricted so that only people who currently possess a job get selected. This equals the set {1,2,4,5} for ESR. Additionally, 'PINCP' (total income last 12 months) was adjusted with the 'ADJINC' variable as advised in the PUMS readme file. It was also later found that observations for working people with very low education are sparse and an additional limit was set at SCHL > 15, this equals filtering out people who haven't acquired at least a high school diploma. This way the SCHL variable can be treated continuous, ranging from 16 (high school diploma) to 24 (PhD). The target variable for the model was a binary construction based on the 99^{th} quantile of the PINCP*ADJINC combination. If an individual's income is at least this much, the target receives value 1 and is otherwise 0. Statistically, being in the 99^{th} quantile of income is a 'rare phenomenon' and equals earnings of more than $350 000 a year in the ACS sample.

The predictors (or independent variables) selected for the initial data set contain both those with theoretical support and those with no support. This is to assess the effects of theoretically sound predictors and also explore new interesting relationships. The data set contains continuous, categorical (or 'multiclass') and binary variables. Using a wide range of predictors from the ACS data set often leads to a high- dimensional sparse matrix, which is the case here as well.

43 initial independent variables were selected and divided into groups as follows (the sequel script for further explanation and reproducible analysis is can reviewed in appendix B):

Personal and family related:

- Age
- Quarter of birth
- Sex
- Has a child
- Married
- Served in the army
- US Native
- Citizenship
- Race

State level

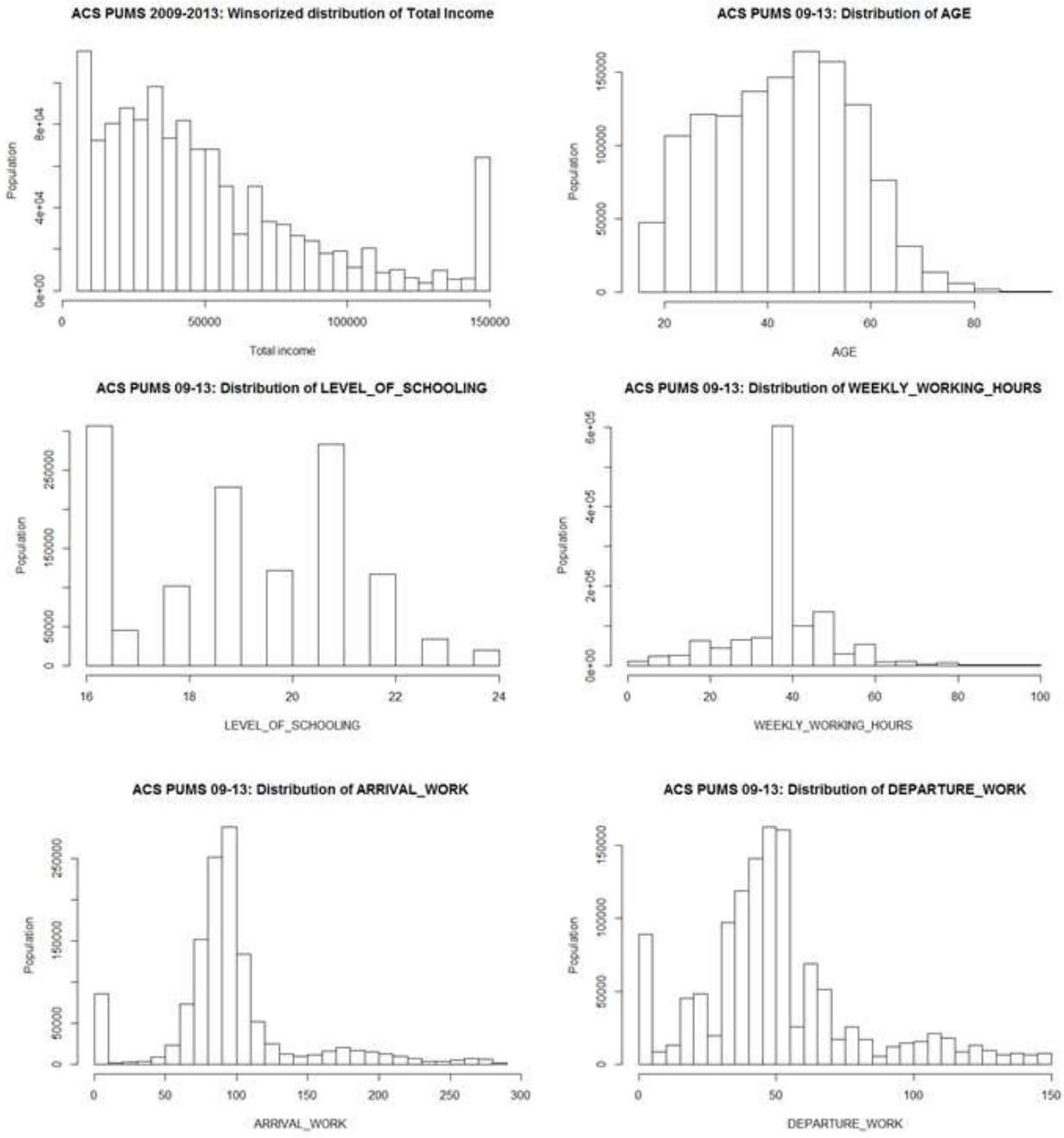
- State code, control to capture regional differences

Educational and work- related:

- Level of schooling, ranging from high school to PhD
- Double degree binary
- Weekly hours worked, 12 month average
- Average arrival- and departure times at work
- Work sector binaries
    - 4 variables: private, nonprofit, state or government and self- employed
- Occupation binaries
    - 24 variables on detailed field of occupation, derived from the 'SOCP00' occupation coding in the PUMS data

Setting the restrictions on employment yields a data set consisting of 1 253 897 observations, or individuals. Of the 43 independent variables five are continuous, two are categorical and the rest binary. Figure 11 contains distributions for several continuous variables in the data subset. The right- hand side peak at the income distribution identifies the point of interest in this study.

Figure 12: Distributions of continuous variables in the ACS PUMS subset



5.4 Building the Random Forest classifier

The following modelling process was adopted:

1. The original dataset was split in three subsets: One for parameter tuning, one for model training and variable importance analysis and one for validation (evaluation of the out-of-bag error). Training and parameter tuning sets were undersampled with a 4 to 1 rate.

2. Using the parameter tuning set, a search function was written to find the optimal values of mtry and nodesize for the Random Forest model. The results can be found in figure 13. A five- fold cross- validation was implemented as a part of this step.

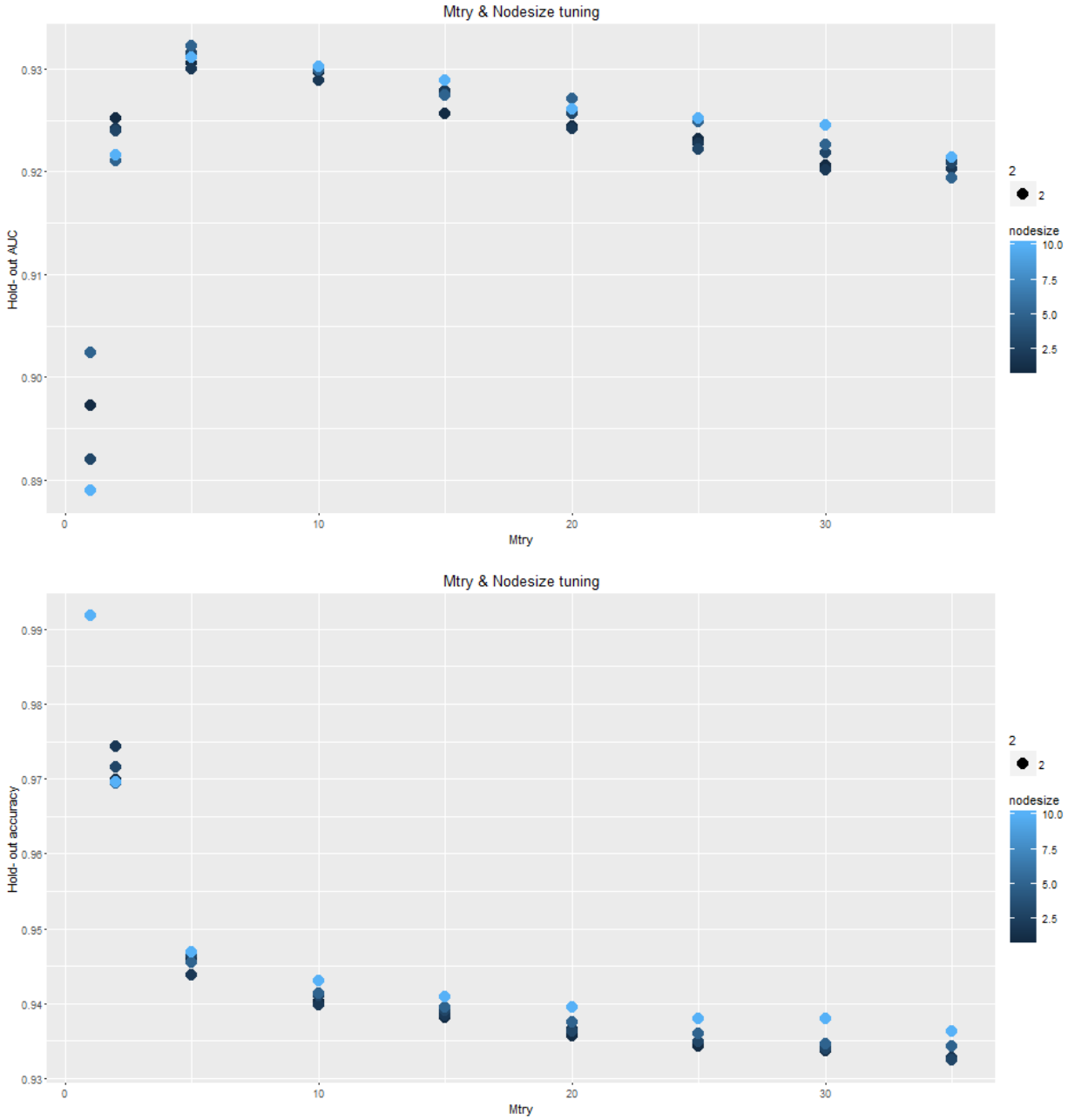
3. A final Random Forest model was built on the training data, based on the optimized parameters from step 2. Variable importance measures were derived from this model, these are found in figure 14. Model interactions were analysed with partial dependence plots, figure 15.

4. A logistic regression was fit on the training set for comparison, figure 16.

5. Finally, the 'Two Trees' algorithm was implemented to estimate average treatment effects for the most important variables, section 5.5.

The validation set was built with 20% of the original data, which equals approximately 250 000 random individuals across the 5 aggregated surveys. The tuning set was of identical size. An optimal amount of randomization for variable splits was found with mtry = 5, combining this with node size = 5 maximized the model's cross- validated AUC in the tuning set. Visualization of the search process can be observed in figure 13.

Figure 13: Random Forest parameter search



In figure 13, mtry is the x- axis while colours refer to node size. The graph above features AUC as the y- axis while the graph below is based on model accuracy. It's interesting to see that the mtry value chosen based on highest AUC is in fact slightly higher (5) than that based on accuracy (3). Accuracy variations based on node size changes are relatively small, generally increasing this tends to have slight positive impact.

After optimizing model parameters, a final Random Forest model was built with the training set. The results are promising: The model achieves an accuracy of over 90% in the validation set with 250 000 new observations (or people) spanning 5 different surveys. The test set error rate is only 5.67% (AUC 0.9358) and divides into a 5.34% error rate in the majority class (99% of people in terms of income) and a 32.78% error rate in the minority class (top 1%). In the R console printout below the OOB esimate of error refers to the sampled training set while the test set error rate refers to the holdout validation set results.

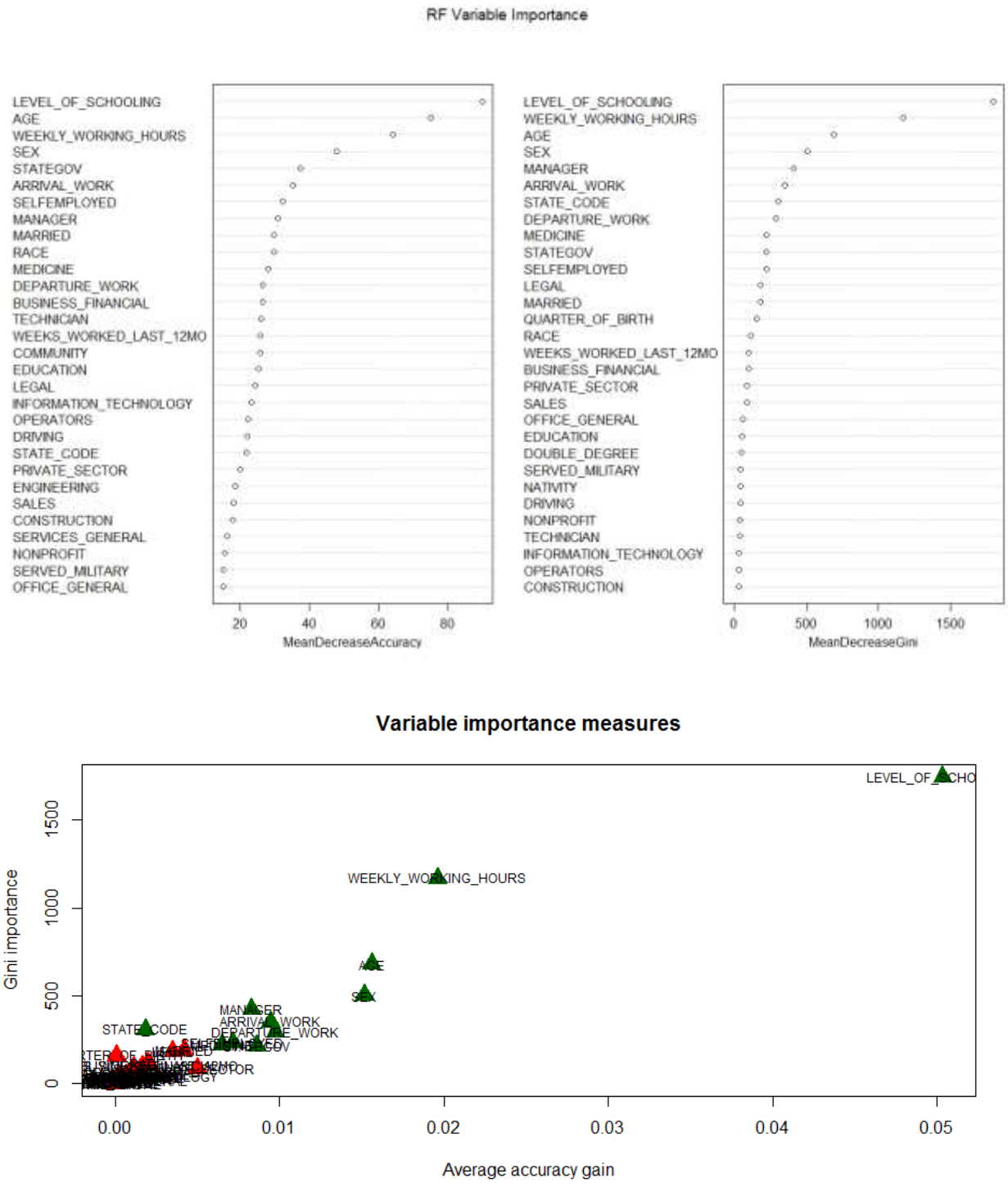
```
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 5

        OOB estimate of  error rate: 10.68%
Confusion matrix:
      0    1 class.error
0 28398 1578  0.05264211
1  2424 5070  0.32345877
               Test set error rate: 5.67%
Confusion matrix:
       0     1 class.error
0 234536 13233  0.05340862
1    987  2024  0.32779807
```

To understand the effects of the most important predictors, variable importance analysis was conducted using two separate importance measures: The average accuracy gain and Gini impurity. Figure 14 (pp. 48) paints a clear picture on the underlying dependencies; while several factors can have an effect, there are four characteristics that greatly differentiate between the highest earning percent and others. These are the highest level of schooling achieved by an individual, age, weekly working hours and sex.

The results appear familiar, and might not be very surprising. It's easy to say that high earnings are often correlated with longer workweeks, experience (as in age) and in some cases high education. The well- documented gender income effect (see Bobbitt- Zeher (2007) for example) is clearly present here as well.

Figure 14: Variable importance measures



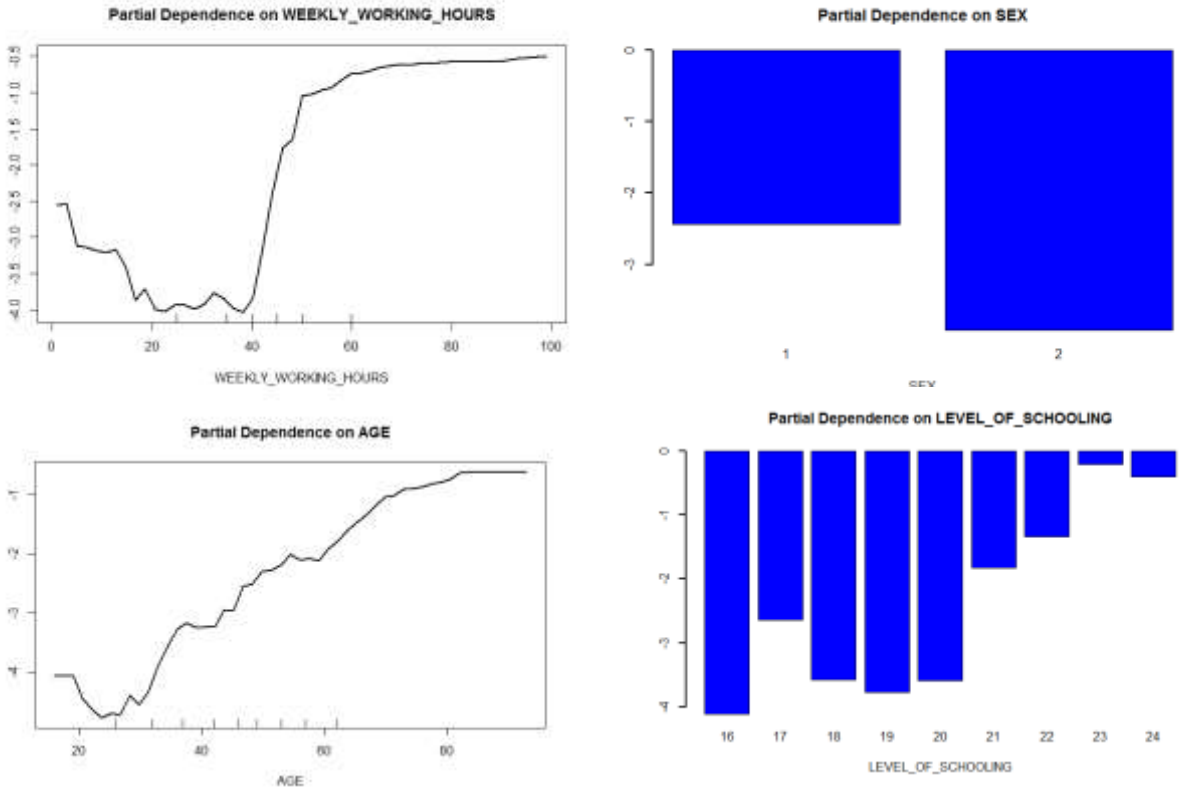
Looking at model performance, a powerful classifier has been established that's really able to distinguish between high earners. Some of the underlying relationships that make the model effective can be nonlinear and possibly complex combinations of multiple features. Partial plots work as an effective tool in better understanding the effects that these top 4 predictors have on the end result.

Following the results of the feature importance analysis, the 4 clear distinguishing predictors are the following:

- Level of schooling
- Weekly working hours
- Age
- Sex

To analyse the effects of these variables, partial plots are leveraged. As discussed above, partial plots offer a way to do ceteris paribus analysis for Random Forests. All other effects are held constant to extract the effect of altering singular predictors. The results can be observed below on figure 15.

Figure 15: Partial dependence plots



Looking at the partial plot on schooling, one can observe that after passing the threshold of academic degrees (20 for associate's degree, 21 for bachelor's), the marginal effects of start to play a bigger role in determining income. The effect is maximized at 23, an MBA. An appropriate treatment group for schooling could be set at >22, which stands for further education (MBA or PhD) after a master's degree. The plot on working hours is fairly simple, chances for bigger earnings really only start to stack up after moving past the average 40-hour workweek What's interesting however is that at least in this data income gains cap out very fast when increasing working hours; extremely long weeks appear to add very little extra when compared with 60- hour weeks for example. In here the treatment could be set at weekly working hours > 40. The partial plot on age describes the effect of experience and a solid track record, but a few points of interest arise as well. Notice the drop at ages 20-25 and peak at around 35: One interpretation could be that the average American is less likely to be a high earner at 24 than at 20 years of age, and that after 35 many reach their maximum potential. A natural explanation for the early twenties drop is likely education. Most people do university at this point. The linear income gains in age can refer to both increased experience and acquired capital. The treatment group for age and experience could be set at age >= 40.

To compare the Random Forest model to a more classic approach to binary response problems, a logistic regression was fit on the training data and its AUC derived from the validation set. The logistic model had a robust performance and was very much in line with Random Forests. The top 4 predictors by Gini impurity were rated very high in terms of statistical significance here as well. The logistic model also achieved a hold- out AUC of 0.9296 which is very good (but still below the decision tree ensemble). Detailed variable importance measures for logistic regression can be observed on page 51.

Table 3: Predictor importance measures in logistic regression

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-7.6300899	1.1483790	-6.644	3.05e-11	***
STATE_CODE	-0.0037950	0.0011666	-3.253	0.00114	**
AGE	0.0463824	0.0017977	25.801	< 2e-16	***
QUARTER_OF_BIRTH2	-0.0409371	0.0522765	-0.783	0.43357	
QUARTER_OF_BIRTH3	-0.1129774	0.0515883	-2.190	0.02853	*
QUARTER_OF_BIRTH4	-0.0988387	0.0520021	-1.901	0.05735	.
SEX2	-1.1711691	0.0465341	-25.168	< 2e-16	***
OWN_CHILD1	-6.8394430	82.1984175	-0.083	0.93369	
MARRIED1	0.4585070	0.0468294	9.791	< 2e-16	***
SERVED_MILITARY1	-0.3566510	0.0591752	-6.027	1.67e-09	***
NATIVITY1	-0.0434394	0.0659910	-0.658	0.51037	
RACEBLACK	-0.1146454	0.1336382	-0.858	0.39096	
RACEMULTIRACE	-0.4997421	0.2494603	-2.003	0.04515	*
RACEOTHER	0.3442827	0.1912628	1.800	0.07185	.
RACEWHITE	0.4769473	0.0911779	5.231	1.69e-07	***
LEVEL_OF_SCHOOLING17	0.4336013	0.1850974	2.343	0.01915	*
LEVEL_OF_SCHOOLING18	0.4753214	0.1165389	4.079	4.53e-05	***
LEVEL_OF_SCHOOLING19	0.6075753	0.0913492	6.651	2.91e-11	***
LEVEL_OF_SCHOOLING20	0.2504596	0.1157829	2.163	0.03053	*
LEVEL_OF_SCHOOLING21	1.7205474	0.0786701	21.870	< 2e-16	***
LEVEL_OF_SCHOOLING22	2.2163839	0.0856990	25.862	< 2e-16	***
LEVEL_OF_SCHOOLING23	3.6797947	0.1092644	33.678	< 2e-16	***
LEVEL_OF_SCHOOLING24	3.1771030	0.1221721	26.005	< 2e-16	***
DOUBLE_DEGREE1	0.0357272	0.0645565	0.553	0.57997	
WEEKLY_WORKING_HOURS	0.0458067	0.0016795	27.275	< 2e-16	***
WEEKS_WORKED_LAST_12MO2	-0.1633715	0.1338707	-1.220	0.22233	
WEEKS_WORKED_LAST_12MO3	-0.1295975	0.1128399	-1.149	0.25076	
WEEKS_WORKED_LAST_12MO4	-0.7632408	0.1568027	-4.868	1.13e-06	***
WEEKS_WORKED_LAST_12MO5	-1.0589510	0.2302975	-4.598	4.26e-06	***
WEEKS_WORKED_LAST_12MO6	-0.7534504	0.2503445	-3.010	0.00262	**
ARRIVAL_WORK	-0.0006273	0.0017084	-0.367	0.71347	
DEPARTURE_WORK	0.0019167	0.0027036	0.709	0.47837	
PRIVATE_SECTOR1	-1.1122127	0.4573936	-2.432	0.01503	*
NONPROFIT1	-1.9739402	0.4632907	-4.261	2.04e-05	***
STATEGOV1	-3.0246919	0.4658602	-6.493	8.43e-11	***
SELFEMPLOYED1	-0.7663328	0.4582874	-1.672	0.09449	.
MANAGER1	2.2912614	1.0385776	2.206	0.02737	*
BUSINESS_GENERAL1	1.6858497	1.0422687	1.617	0.10577	
BUSINESS_FINANCIAL1	2.2647758	1.0407027	2.176	0.02954	*
INFORMATION_TECHNOLOGY1	0.6509439	1.0451366	0.623	0.53340	
ENGINEERING1	0.7963528	1.0444991	0.762	0.44581	
SCIENCE1	0.8912243	1.0485055	0.850	0.39533	
COMMUNITY1	-0.8320440	1.0904028	-0.763	0.44543	
LEGAL1	1.4891824	1.0440907	1.426	0.15378	
EDUCATION1	0.3137127	1.0447240	0.300	0.76396	
ENTREPRENEURS1	1.6496969	1.0439740	1.580	0.11406	
MEDICINE1	1.7970039	1.0410708	1.726	0.08433	.
HEALTHCARE1	1.0725390	1.0748704	0.998	0.31836	
SECURITY1	0.7783095	1.0673589	0.729	0.46588	
FOOD1	-0.0841025	1.0876388	-0.077	0.93836	
CLEANING1	0.0387648	1.0743256	0.036	0.97122	
SERVICES_GENERAL1	0.2869069	1.0728740	0.267	0.78915	
SALES1	1.9101090	1.0394944	1.838	0.06613	.
OFFICE_GENERAL1	0.7715437	1.0430712	0.740	0.45949	
AGRICULTURE1	-0.1015122	1.1589442	-0.088	0.93020	
CONSTRUCTION1	0.7214257	1.0465515	0.689	0.49061	
TECHNICIAN1	-0.6812364	1.0721436	-0.635	0.52517	
OPERATORS1	-0.1437164	1.0537715	-0.136	0.89152	
DRIVING1	0.2575195	1.0483215	0.246	0.80595	
MILITARY1	NA	NA	NA	NA	

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

5.5 Evaluating average treatment effects

What remains is ranking the top predictors in terms of average treatment effects. This section uses the Two Trees approach from Athey & Imbens (2015). The code and specific implementation can be reviewed in appendix A.

When modelling binary response problems and individual prospensity scores, or the 'likelihoods' to belong to the high- earning group in this case, treatment effects can be interpreted as changes in likelihood based on whether an individual receives a given treatment or not. In here the treatment effect could be the increased chance of high earnings by investing (time and money) into further education for example. While the concept of estimating synthetic treatment effects is highly theoretical in nature, it helps in quantifying and ranking the effects of different predictors and treatments and serves as a step towards more 'causally geared' algorithmic modelling.

The Two Trees algorithm is implemented with Random Forests in place of ordinary decision trees. As discussed in chapter 4, we define two Random Forest models as $RF_{treatment}$ and $RF_{control}$. Here $RF_{treatment}$ is fitted on a subset of the training data where the treatment holds true and $RF_{control}$ is fitted on the remaining data. A separate validation data is then scored by both models and the synthetic treatment effect is estimated with $CATE = RF_{treatment}(1,x) - RF_{control}(0,x)$. The average treatment effect is aggregated from individual effects in the validation set. A t- test is also conducted to establish a statistical significance threshold for the effects.

Looking back at the top 4 predictors established by the Gini impurity criterion, we want to find out which can be expected to have the greatest influence in an individual's monetary success:

1. Higher education: Further education above a master's degree (MBA or PhD)
2. Longer hours: Working more than 40 hours a week
3. Seniority: Being over 39 years old
4. Gender: Female

The simulation results can be found below.

Table 4: CATE estimates

```
[1] "TREATMENT: FEMALE"
 Treatment observations    Control observations
                "15948"                 "21522"
       Treatment Effect Significant at p < 0.01
  "-0.0853665962197942"                   "yes"

[1] "TREATMENT: EDUCATION"
 Treatment observations    Control observations
                 "8089"                 "29381"
       Treatment Effect Significant at p < 0.01
    "0.230153022569583"                   "yes"

[1] "TREATMENT: LONG_HOURS"
 Treatment observations    Control observations
                "13632"                 "23838"
       Treatment Effect Significant at p < 0.01
    "0.068276597017306"                   "yes"

[1] "TREATMENT: AGE & EXPERIENCE"
 Treatment observations    Control observations
                "24484"                 "12986"
       Treatment Effect Significant at p < 0.01
   "0.0664906890501635"                   "yes"
```

First of all, all treatment and control groups are of relevant size and the prospensity score differences between groups are statistically significant. The estimated CATEs are as follows:

- Female gender reduces likelihood of high income by 8.5% on average
- Getting either an MBA or a PhD increases likelihood by 23.0%
- Working longer hours increases likelihood by 6.8%
- Seniority and experience increases likelihood by 6.6%

According to the Two Trees simulation, investing in higher education has the biggest impact on increased earnings. Interpreting this can be tricky however, as companies can offer MBA degrees to promising employees for example. This could make MBAs an identity for high earners that are more likely side- products than root- causes. However, taken that all other personal factors like occupation are left unaltered in the simulation, one could conjecture that acquiring higher education certainly can drive income gains. There are several explanations

to this, including simply the educational requirements in certain high- level jobs and the efficiency signalling effect of holding academic degrees.

Another interesting result lies in the relationship between gender gap and hard work and experience. The negative effect of gender is just enough to offset either the positive effect of longer workweeks or the effect of seniority. The positive finding here is however that according to this analysis, one could overcome the negative gender effects with the combination of acquired experience and a strong work- ethic. This is a very interesting finding and could be further researched in a rigorous micro- level study on gender and employment. The ACS data set used here could also complement classic gender- income data sets in econometric studies.

While establishing causality can still be somewhat hard for machine learning and needs to be evaluated case-by-case, these new approaches to decision trees are opening up avenues for empirical econometrics. It's well known that controlled experiments are often very hard to set up in practical economic problems: Consider for example simulating the effects of a tax change in household income. This is a classic problem of choosing the correct policy by maximizing welfare where control groups are virtually impossible to set up without greatly discriminating some parts of the population.

In the modern world data is in abundance, governments and companies have access to highly granular sets concerning personal-, household- and geographical data for example. By tapping into these modern data sets and experimenting with new methods like causal trees and other machine learning applications, economists could potentially implement simulation models and forecasts with great increases in accuracy.

6 CONCLUSION

The goals in this thesis were to review the new opportunities and challenges offered by big data in econometrics, introduce machine learning methods and decision forests and evaluate them by working on a classic econometric study: personal income determinants. The idea was to not set too many limits on these determinants, but let machine learning decide upon the most important factors and focus the analysis on this set of independent variables. The American Community Survey Microsample Data from years 2009 to 2013 was used for this purpose.

The downfalls of classic econometric methods were identified as high dimensionality, non-linear relations and outliers. The concepts of decision trees and ensemble learning were introduced as possible workarounds in analysing big data sets. The effectiveness and predictive accuracy of these methods were reviewed. Additionally, the Random Forest algorithm was proposed for econometrics as a continuation to the modern concept of causal trees in econometrics literature. The Two Trees treatment effect estimator was implemented with Random Forests.

The empirical results were both promising in terms of accuracy but also expected in terms of the most important predictors. The Random Forest classifier achieved an accuracy of over 90% in determining who belongs to the top 1- percentile in annual earnings on a hold out data set. The most important variables derived from this classifier were highly significant in a more traditional econometric model for binary response problems, logistic regression. These variables were related to education, work, experience and gender.

The treatment effect simulation yielded interesting results. The effects of investing in higher education (MBA or PhD) far outweigh all other variables in influencing the likelihood of high earnings. Gender was the only discriminating factor, and it's estimated negative effect was greater than the positive effect of experience (age) or long hours alone. However the simulation also shows that seniority and hard work together tend to outweigh the negative gender bias. This is a very interesting phenomenon and calls for further research.

Establishing causal relations still remains a tough nut to crack for machine learning, but these types of simulations offer interesting alternatives to traditional econometric approaches. Causal decision tree approaches could be leveraged in for example micro- level policy impact estimation models, where controlled experiments are very hard if not impossible to set up.

To conclude, decision trees and regularization methods can be very effective in the econometrics workflow when applied correctly. The causal trees theory and treatment effect simulation is still very new and not many packages support direct implementation, but this field is expected to evolve fast in the coming years. In addition to offering interesting new ways for treatment estimation, machine learning provides effective tools for dissecting multiple data sources for correlations and finding relationships to provide a basis for a more controlled econometric approach. The American Community Survey data is also a very rich source for micro- level studies and can be joined with other sources for an even more complete analysis.

7 BIBLIOGRAPHY

Athey S., and G.W. Imbens (2015): "Machine Learning Methods for Estimating Heterogeneous Causal Effects". CA: Stanford University.

Biau. O., and A. D'Elia (2011): "Euro area GDP forecasting using large survey datasets. A random forest approach". Euroindicators working papers, 2011: 002.

Bobbitt- Zeher, D. (2007): "The Gender Income Gap and the Role of Education". Sociology of Education, vol.80, pp. 1- 22.

Bradley, A.P. (1997): "The use of the area under the ROC curve in the evaluation of machine learning algorithms". Pattern Recognition, vol. 30, pp. 1145- 1159.

Breiman, L. (2001): "Random Forests". Machine Learning, vol. 45, pp. 5 – 32.

Breiman, L., J.H. Friedman, R.A. Ohlsen and C.J. Stone (1984): "Classification and Regression Trees". CA: Wadsworth.

Burger, M., and J. Repiský (2012): "Problems of Linear Least Square Regression And Approaches to Handle Them". Advanced Research in Scientific Areas, December 2012.

Caruana. R., and A. Niculescu- Mizil (2006): "An Empirical Comparison of Supervised Learning Algorithms". NY: Cornell University.

Caruana. R., N. Karampatziakis and A. Yessenalina (2008): "An Empirical Evaluation of Supervised Learning in High Dimensions". NY: Cornell University.

David S. Siroky (2009): "Navigating Random Forests". Statistics Surveys, vol.3, pp.147 – 163.

Einav L., and J. Levin (2014): "The Data Revolution and Economic Analysis". CA: Stanford University.

Fan, J. (2014): "Features of Big Data and the sparsest solution in high confidence set". NJ: Princeton University.

Foster, J.C., J.M. Taylor, and S.J. Ruberg (2011): "Subgroup identification from clinical trial data". Statistics in Medicine, vol. 30, pp. 2867- 2880.

Freund, Y. and R.E. Schapire (1999): "A Short Introduction to Boosting", Journal of Japanese Society for Artificial Intelligence, vol. 14, pp. 771- 780.

Grömping, U. (2009): "Variable Importance Assessment in Regression: Linear Regression versus Random Forest". The American Statistician, vol. 63, pp.308- 319.

Hastie, T., R. Tibshirani, and J. Friedman (2009): "The Elements of Statistical Learning". Second Edition, NY: Springer- Verlang.

Horowitz, Joel L., and N.E. Savin (2001): "Binary Response Models: Logits, Probits & Semiparametrics". Journal of Economic Perspectives, vol. 15, pp.43 – 56.

Jacobs, A. (2009): "The Pathodologies of Big Data". Databases, vol. 7, issue 6.

Khandani, A.E., A.J. Kim, and A. W. Lo (2010): "Consumer Credit Risk Models via Machine-Learning Algorithms". Journal of Banking and Finance, vol. 34, pp. 2767- 2787.

King, G. and L. Zeng (2001): “Logistic Regression in Rare Events Data”. Political Analysis, vol. 9, pp. 137- 163.

Nguyen, C., Y. Wang, and H. N. Nguyen (2013): “Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic”. Journal of Biomedical Science and Engineering, vol. 6, pp. 551-560.

Payne. N. (2014): “Evaluating the Impact of Heteroscedasticity on the Predictive Ability of Modern Regression Techniques”. Master’s thesis: Simon Fraser University.

Reiss. P.C., and F.A. Wolak (2007): “STRUCTURAL ECONOMETRIC MODELING: RATIONALES AND EXAMPLES FROM INDUSTRIAL ORGANIZATION”

Schadt, E. E. (2012): “The changing privacy landscape in the era of big data”. Molecular Systems Biology, 8:612.

Sengupta, Nandata. (2015): “Machine Learning Techniques in Applied Econometrics”. Doctoral Thesis: Carnegie Mellon University.

Stephenson, R.W., A.G. Froelich, and W.M. Duckworth (2010): “Using Resampling to Compare Two Proportions”. Teaching Statistics, vol. 32, pp.66-71.

Taylor, L., R. Schroeder, and E. Meyer (2014): “Emerging practices and perspectives on Big Data analysis in economics: Bigger and better or more of the same?”. Big Data & Society, July- December, 1-10.

US Census Bureau (2015): The American Community Survey Data Dictionary

Vargiu, E. and M. Urru (2013): “Exploiting web scraping in a collaborative filtering- based approach to web advertising”. Artificial Intelligence Research, vol. 2, no. 1.

Varian, H. (2014): “Big Data: New Tricks for Econometrics”. Journal of Economic Perspectives: vol. 28, pp. 3- 28.

Websites:

Google Cloud Platform blog: https://cloudplatform.googleblog.com

Kaggle, a data- mining competition website: www.kaggle.com

Machine learning blog by Jason Brownlee: http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

Netflix technology blog: http://techblog.netflix.com/

Spotify technology blog: https://labs.spotify.com/

The Apache Hadoop project: http://hadoop.apache.org

The Apache Spark project: http://spark.apache.org

The Comprehensive R Archive Network: cran.r-project.org

US Census Bureau website: www.census.gov

US Open Data website: www.data.gov

World Bank open data website: data.worldbank.org

8 APPENDIX A: R CODE

This section contains all relevant R- scripts used in this paper. The scripts are shared (and seeded) for reproducible analysis. R version 3.2.1 (2015-06-18) was used.

Script for Chapter 2.3: 'Features of Big Data': Simulation of 'spurious correlations'

No external packages used.

```
randomCorr <- function(maxdims,obs){

  normalDF <- function(rows,dims) {
    df <- data.frame(matrix(0,rows,dims))
    for (i in 1:dims) {
      df[,i] <- rnorm(rows)
    }
    df
  }


  temp <- data.frame()
  for (j in 2:maxdims){
    df <- normalDF(obs,j)
    colnames(df)[1] <- "Y"
    varnames <- names(df)[2:ncol(df)]
    f <- as.formula(paste("Y~", paste(varnames, collapse = " + ")))
    model <- lm(f,df)

    temp <- rbind(temp,cbind(j,min(summary(model)$coefficients[,4])
                             ,sum(summary(model)$coefficients[,4]<=0.05)
                             ,sum(summary(model)$coefficients[,4]<=0.01)
                             ,sum(summary(model)$coefficients[,4]
                             /length(summary(model)$coefficients[,4]))))
  }

  names(temp) <- c("Predictors","Smallest P- value"
                   ,"Amount of significant predictors (p <= 0.05)"
                   ,"Amount of significant predictors (p <= 0.01)"
                   ,"Average P- value")
  temp
}

set.seed(2015)
corrframe <- randomCorr(100,10000)

plot(corrframe$Predictors,corrframe$`Smallest P- value`,type="hist"
     ,lwd=4,main="Incidence of Spurious Correlations"
     ,xlab="The amount of randomly sampled N(0,1) predictors for OLS"
     ,ylab="The lowest observed P- value among predictors")

plot(corrframe$Predictors,corrframe$`Amount of significant predictors (p <=
0.05)`,type="l",lwd=1
     ,main="Increasing Dimensionality increases the chance for Spurious
Correlations"
     ,xlab="The amount of randomly sampled N(0,1) predictors for OLS"
```

```
     ,ylab="The amount of significant predictors (p <= 0.05)")

abline(lm(corrframe$`Amount of significant predictors (p <=
0.05)`~corrframe$Predictors),lwd=2)
```

Script for Chapter 2.3: 'Features of Big Data': Simulation of outlier effects in three-dimensional case of OLS

Packages used : 'rgl'.

```
library(rgl)
set.seed(123)

x <- seq(from = 1, to =  10, by = 1)
y <- rnorm(length(x))
z <- 2*x + 5*y + rnorm(1)

model <- lm(z ~ x + y)

coefs <- coef(model)
a <- coefs["x"]
b <- coefs["y"]
c <- -1
d <- coefs["(Intercept)"]

z_alt <- z
z_alt[[length(z)]] <- 3*max(range(z))

model_alt <- lm(z_alt ~ x + y)

plot3d(x,y,z_alt,type="s",col=ifelse(z_alt>max(range(z)),"red","green"),siz
e=1)

coefs2 <- coef(model_alt)
q <- coefs2["x"]
w <- coefs2["y"]
e <- -1
f <- coefs2["(Intercept)"]

planes3d(q,w,e,f,alpha=.5)
planes3d(a,b,c,d,alpha=.25)
```

Script for Chapter 5: 'Personal income drivers: A Random Forests approach'

Packages used: 'readr', 'pROC', 'randomForest' & 'ggplot2'

```
set.seed(777)

library(readr)

data <- read_csv("pums_extract.csv")

# auditing data

summary(data)
str(data)

print('Nulls per predictor')
sapply(data,function(x) { sum(is.na(x)) })

print('Unique values per predictor')
sapply(data, function(x) { length(unique(x)) })

data[is.na(data)] <- 0

# factor conversion for binary and categorical predictors (capped at 10
unique values)

for (feature in seq_along(data)) {

  if (is.character(data[, feature]) | length(unique(data[, feature])) <=
10)
  { data[, feature] <- as.factor(data[, feature]) }
}

# setting percentile cutoff values for selected continuous variables

winsorize <- function(x, cutoff = 0.01){

  limits <- quantile(x, c(cutoff, 1 - cutoff))

  x[x < limits[1]] <- limits[1]
  x[x > limits[2]] <- limits[2]

  x
}

# graphing earnings

hist(winsorize(data$TOT_INCOME),
     main = "ACS PUMS 2009-2013: Winsorized distribution of Total Income",
     xlab = "Total income",
     ylab = "Population")

# graphing histograms of other continuous variables

for (feature in seq_along(data)) {

  if (class(data[, feature]) == 'integer' | class(data[, feature]) ==
'numeric') {

    varname <- colnames(data)[feature]
```

```
    hist(data[, feature],
         main = paste("ACS PUMS 09-13: Distribution of ", varname),
         xlab = varname, ylab = "Population")
  }
}


# dropping unnecessary columns and defining the target variable

data$HIGH_INCOME <- as.factor(ifelse(data$TOT_INCOME >=
quantile(data$TOT_INCOME, .99), 1, 0))

data <- data[, 6:ncol(data)]
data <- data[, -which(names(data) %in% "TOT_INCOME")]

# creating data subsets for parameter tuning, training and validation

create.folds <- function(df, k, seed) {

  set.seed(seed)
  # random shuffle
  df <- df[sample(1 * nrow(df)), ]
  partitioning <- cut(seq(1, nrow(df)), breaks = k, labels = FALSE)

  partitioning
}

folds <- create.folds(data, 5, 777)
validation <- data[which(folds == 5), ]
tuning <- data[which(folds == 4), ]
training <- data[which(folds < 4), ]

# creating a balancing function to randomly undersample the majority class
# this is for parameter search and training puposes only

do.undersample <- function(df, target, value, rate, seed) {

  set.seed(seed)

  posObs <- sum(target == value)
  negToAdd <- round(rate*posObs)
  majority_obs <- df[sample(which(target != value), negToAdd), ]
  rare_obs <- df[which(target == value), ]

  balanced <- rbind(majority_obs, rare_obs)
  balanced <- balanced[sample(1*nrow(balanced)), ]

  balanced
}

test <- do.undersample(tuning, tuning$HIGH_INCOME, 1, 4, 777)

# tuning the mtry value for Random Forest, evaluating against OOB ERROR and
AUC

library(randomForest)
library(pROC)

tuning_folds <- create.folds(tuning, 5, 777)
tuning_k <- unique(tuning_folds)
```

```
searchgrid <- expand.grid('mtry_values' = c(1, 2, 5, 10, 15, 20, 25, 30,
35),
                          'nodesize' = c(1, 2, 3, 5, 10))

mtryResults <- apply(searchgrid, 1, function(search) {

  mtry <- search[['mtry_values']]
  nodesize <- search[['nodesize']]

  features <- colnames(tuning)[1:(ncol(tuning) - 1)]
  target <- colnames(tuning)[ncol(tuning)]

  subresults <- lapply(tuning_k, function(cv_iter) {

    set.seed(777)

    holdout <- which(tuning_folds == cv_iter, arr.ind = TRUE)
    train <- tuning[-holdout, ]
    test <- tuning[holdout, ]

    # only the training folds [1, k-1] are undersampled
    train_sampled <- do.undersample(train, train$HIGH_INCOME, 1, 4, 777)

    model <- randomForest(x = train_sampled[, features],
                          y = train_sampled[, target],
                          ntree = 50,
                          mtry = mtry,
                          nodesize = nodesize,
                          do.trace = TRUE,
                          keep.forest = TRUE)

    predictions <- data.frame('actual' = test[, target],
                              'predicted' = predict(model, test[,
features], type = 'class'),
                              'prospensity' = predict(model, test[,
features], type = 'prob')[, 2])

    accuracy <- sum(predictions$actual == predictions$predicted) /
nrow(predictions)
    auc <- auc(roc(predictions$actual, predictions$prospensity))

    return(c(cv_iter, auc, accuracy))
  })

  results <- data.frame(matrix(unlist(subresults),
                               nrow = length(tuning_k),
                               byrow = TRUE),
                        stringsAsFactors=FALSE)

  names(results) <- c('iter', 'auc', 'accuracy')
  auc <- mean(results$auc)
  accuracy <- mean(results$accuracy)

  return(c(mtry, nodesize, auc, accuracy))
})

results <- as.data.frame(t(mtryResults))
names(results) <- c('mtry', 'nodesize', 'auc', 'accuracy')

# plotting results
```

```
library(ggplot2)

qplot(mtry, auc, col = nodesize, cex = 2, data = results,
      main = 'Mtry & Nodesize tuning',
      xlab = 'Mtry', ylab = 'Hold- out AUC')

qplot(mtry, accuracy, col = nodesize, cex = 2, data = results,
      main = 'Mtry & Nodesize tuning',
      xlab = 'Mtry', ylab = 'Hold- out accuracy')

# fitting an optimized Random Forest classifier

training_sampled <- do.undersample(training, training$HIGH_INCOME, 1, 4,
777)

forest <- randomForest(x = training_sampled[, 1:43],
                       y = training_sampled$HIGH_INCOME,
                       xtest = validation[, 1:43],
                       ytest = validation$HIGH_INCOME,
                       do.trace = TRUE, importance =TRUE, ntree = 500,
                       mtry = results$mtry[which.max(results$auc)],
                       nodesize = results$nodesize[which.max(results$auc)],
                       keep.forest=TRUE)

print(forest)

# plotting variable importance measures

importancedata <- data.frame('VariableName' = names(forest$importance[,3]),
                             'MeanDecreaseAcc' = forest$importance[,3],
                             'MeanDecreaseGini' = forest$importance[,4])

plot(importancedata[,2:3],
     col = ifelse(importancedata$MeanDecreaseGini >
                    quantile(importancedata$MeanDecreaseGini,.75),
                  "darkgreen",
                  "red"),
     pch = 17,
     cex = 2,
     ylab = "Gini importance",
     xlab = "Average accuracy gain",
     main = "Variable importance measures")

text(importancedata$MeanDecreaseAcc,
     importancedata$MeanDecreaseGini,
     importancedata$VariableName, cex = 0.75)

gini_ordering <- order(importancedata$MeanDecreaseGini, decreasing = TRUE)
acc_ordering <- order(importancedata$MeanDecreaseAcc, decreasing = TRUE)

print('Gini ranking for predictors:')
head(importancedata$VariableName[gini_ordering], 10)

print('Accuracy ranking for predictors:')
head(importancedata$VariableName[acc_ordering], 10)

varImpPlot(forest, main = 'RF Variable Importance')

# predicting and comparing results with standard logistic regression
```

```
rf_preds <- data.frame('actual' = validation$HIGH_INCOME,
                       'predicted' = predict(forest, validation[, 1:43],
type = 'class'),
                       'prospensity' = predict(forest, validation[, 1:43],
type = 'prob')[, 2])

rf_accuracy <- sum(rf_preds$actual == rf_preds$predicted) / nrow(rf_preds)
plot(roc(rf_preds$actual, rf_preds$prospensity))
rf_auc <- auc(roc(rf_preds$actual, rf_preds$prospensity))

logit <- glm(formula = training_sampled$HIGH_INCOME~.,data =
training_sampled[, 1:43], family = "binomial")
logitpreds <- data.frame('actual' = validation$HIGH_INCOME,
                         'predicted' = predict(logit, validation[, 1:43],
type = 'response'))

plot(roc(logitpreds$actual, logitpreds$predicted))
summary(logit)

# partialPlotting

partialPlot(forest,training_sampled,WEEKLY_WORKING_HOURS,which.class = "1",
lwd = 2)
partialPlot(forest,training_sampled,LEVEL_OF_SCHOOLING,which.class = "1",
lwd = 2)
partialPlot(forest, training_sampled, AGE, which.class = "1", lwd = 2)
partialPlot(forest, training_sampled, SEX, which.class = "1", lwd = 2)


# Creating a 'Two- Trees' treatment-control simulation as proposed in Athey
& Imbens (2015)

two.trees <- function(t, df_train, df_validate, trees, features, target,
seed) {

  set.seed(seed)

  result <- lapply(t, function(estimator) {

    treatment <- unlist(estimator)
    df_t <- df_train[treatment, ]
    df_c <- df_train[!treatment, ]

    treatmentsize <- nrow(df_t)
    controlsize <- nrow(df_c)

    rf_t <- randomForest(x = df_t[, features], y = df_t[, target],
                         ntree = trees,
                         mtry = results$mtry[which.max(results$auc)],
                         nodesize =
results$nodesize[which.max(results$auc)],
                         keep.forest = TRUE)

    rf_c <- randomForest(x = df_c[, features], y = df_c[, target],
                         ntree = trees,
                         mtry = results$mtry[which.max(results$auc)],
                         nodesize =
results$nodesize[which.max(results$auc)],
                         keep.forest = TRUE)

    predictions <- data.frame('target' = df_validate[, target],
```

```
                                'treatment' = predict(rf_t, newdata =
df_validate[, features], type = "prob")[,2],
                                'control' = predict(rf_c, newdata =
df_validate[, features], type = "prob")[,2])


    predictions$difference <- predictions$treatment - predictions$control
    agg <- sum(predictions$difference)/nrow(predictions)
    significance <-
t.test(predictions$treatment,predictions$control)$p.value
    significance_dummy <- ifelse(significance < 0.01,"yes","no")

    return(c('Treatment observations' = treatmentsize,
             'Control observations' = controlsize,
             'Treatment Effect' = agg,
             'Significant at p < 0.01' = significance_dummy))
  })

  result

}

treatments <- list('FEMALE' = training_sampled$SEX == "2",
                   'EDUCATION' = training_sampled$LEVEL_OF_SCHOOLING %in%
c(22, 23, 24),
                   'LONG_HOURS' = training_sampled$WEEKLY_WORKING_HOURS >
40,
                   'AGE & EXPERIENCE' = training_sampled$AGE > 39)

avg_treatment_effects <- two.trees(t = treatments,
                                   df_train = training_sampled,
                                   df_validate = validation,
                                   trees = 200,
                                   features =
colnames(training_sampled)[1:43],
                                   target = colnames(training_sampled)[44],
                                   seed = 777)

for (treatment in names(avg_treatment_effects)) { print(paste('TREATMENT:',
treatment))

print(unlist(avg_treatment_effects[[treatment]])) }
```

9 APPENDIX B: SQL SCRIPT

This section identifies the SQL script used to aggregate the American Community Survey data. This script was run in Google BigQuery, a columnar database in the google cloud platform.

```
SELECT
  *
FROM (SELECT
    -- WEIGHT AND ROW IDENTIFIERS, STATE CODE
    pwgtp WEIGHT,
    RT,
    SERIALNO,
    SPORDER,
    ESR,
    st STATE_CODE,

    -- ADJUSTMENT FOR ANNUAL INCOMES
    pincp*CAST(adjinc AS float)/1000000 TOT_INCOME,

    -- AGE
    agep AGE,
    qtrbir QUARTER_OF_BIRTH,

    -- SEX
    SEX,
    -- CHILDREN, MARRIAGE, MILITARY SERVICE AND NATIVITY

    oc OWN_CHILD,
    CASE WHEN mar = 1 THEN 1 ELSE 0 END MARRIED,
    CASE WHEN mil IN (1,2,3) THEN 1 ELSE 0 END SERVED_MILITARY,
    CASE WHEN nativity = '1' THEN 1 ELSE 0 END NATIVITY,

    -- RACE
    CASE WHEN rac1p = 1 THEN 'WHITE' WHEN rac1p = 2 THEN 'BLACK' WHEN rac1p
= 6 THEN 'ASIAN'
         WHEN rac1p = 8 THEN 'MULTIRACE' ELSE 'OTHER' END RACE,

    -- EDUCATION
    schl LEVEL_OF_SCHOOLING,
    CASE WHEN fod1p > 0
    AND fod2p > 0 THEN 1 ELSE 0 END DOUBLE_DEGREE,

    -- WORKING
    wkhp WEEKLY WORKING HOURS,
    wkw WEEKS_WORKED_LAST_12MO,
    jwap ARRIVAL_WORK,
    jwdp DEPARTURE_WORK,
    CASE WHEN cow = 1 THEN 1 ELSE 0 END PRIVATE_SECTOR,
    CASE WHEN cow = 2 THEN 1 ELSE 0 END NONPROFIT,
    CASE WHEN cow IN (3,
      4,
      5) THEN 1 ELSE 0 END STATEGOV,
    CASE WHEN cow IN (6,
      7) THEN 1 ELSE 0 END SELFEMPLOYED,

    -- OCCUPATION BINARIES
    CASE WHEN socp00 LIKE '11%' THEN 1 ELSE 0 END MANAGER,
    CASE WHEN socp00 LIKE '131%' THEN 1 ELSE 0 END BUSINESS_GENERAL,
```

```
    CASE WHEN socp00 LIKE '132%' THEN 1 ELSE 0 END BUSINESS_FINANCIAL,
    CASE WHEN socp00 LIKE '15%' THEN 1 ELSE 0 END INFORMATION_TECHNOLOGY,
    CASE WHEN socp00 LIKE '17%' THEN 1 ELSE 0 END ENGINEERING,
    CASE WHEN socp00 LIKE '19%' THEN 1 ELSE 0 END SCIENCE,
    CASE WHEN socp00 LIKE '21%' THEN 1 ELSE 0 END COMMUNITY,
    CASE WHEN socp00 LIKE '23%' THEN 1 ELSE 0 END LEGAL,
    CASE WHEN socp00 LIKE '25%' THEN 1 ELSE 0 END EDUCATION,
    CASE WHEN socp00 LIKE '27%' THEN 1 ELSE 0 END ENTREPRENEURS,
    CASE WHEN socp00 LIKE '29%' THEN 1 ELSE 0 END MEDICINE,
    CASE WHEN socp00 LIKE '31%' THEN 1 ELSE 0 END HEALTHCARE,
    CASE WHEN socp00 LIKE '33%' THEN 1 ELSE 0 END SECURITY,
    CASE WHEN socp00 LIKE '35%' THEN 1 ELSE 0 END FOOD,
    CASE WHEN socp00 LIKE '37%' THEN 1 ELSE 0 END CLEANING,
    CASE WHEN socp00 LIKE '39%' THEN 1 ELSE 0 END SERVICES_GENERAL,
    CASE WHEN socp00 LIKE '41%' THEN 1 ELSE 0 END SALES,
    CASE WHEN socp00 LIKE '43%' THEN 1 ELSE 0 END OFFICE_GENERAL,
    CASE WHEN socp00 LIKE '45%' THEN 1 ELSE 0 END AGRICULTURE,
    CASE WHEN socp00 LIKE '47%' THEN 1 ELSE 0 END CONSTRUCTION,
    CASE WHEN socp00 LIKE '49%' THEN 1 ELSE 0 END TECHNICIAN,
    CASE WHEN socp00 LIKE '51%' THEN 1 ELSE 0 END OPERATORS,
    CASE WHEN socp00 LIKE '53%' THEN 1 ELSE 0 END DRIVING,
    CASE WHEN socp00 LIKE '55%' THEN 1 ELSE 0 END MILITARY

  FROM
    TABLE_QUERY(ikonhen,'table_id contains "ss13pus"') )

  /* FILTER THE DATA TO THOSE WHO HAVE JOBS, HAVE ANNUAL INCOME
     AND HAVE ACQUIRED AT LEAST A HIGH SCHOOL DIPLOMA */
WHERE
  ESR IN (1,2,4,5)
  AND TOT_INCOME > 0
  AND LEVEL_OF_SCHOOLING > 15
```