PRELIMINARY AND INCOMPLETE

Technical Report: Identifying PhDs and Postdocs in the LEHD

James C. Davis, Holden A. Diethorn, Gerald R. Marschke, Andrew J. Wang

Abstract

This paper details our construction of a linked employer-employee longitudinal dataset of the doctoral workforce that enables researchers to analyze the labor market outcomes of STEM PhD graduates and postdocs. The dataset is primarily derived from annual ACS and LEHD files held within the secure environment of a Federal Statistical Research Data Center (FSRDC). However, since neither the ACS nor LEHD data contain a label for whether an individual is or has been employed as a postdoc, we must utilize a recently developed data source, UMETRICS, to predict postdoc status for our ACS-LEHD sample. By merging the Employee Transactions File of UMETRICS to our ACS-LEHD doctorate dataset, we obtain a (UMETRICS) subset of the ACS-LEHD data that *does* contain such a label. By utilizing machine learning algorithms such as random forests trained on the UMETRICS subset of the ACS-LEHD doctorate data, we can predict postdoc status for the rest of the ACS-LEHD sample, which will then enable us to examine the economic effects of postdoc employment. In this paper, we discuss the formation of our ACS-LEHD panel of doctorates, the random forest method used for predicting postdoc status, and the diagnostic tests used to evaluate predictive performance. The machine learning methods used to augment the ACS-LEHD doctorate sample is general enough to be applied by other researchers merging datasets of disparate size.

James C. Davis
Boston Census Research Data Center
National Bureau of Economic Research
1050 Massachusetts Avenue
Cambridge, MA 02138
james.c.davis@census.gov

Holden A. Diethorn
State University of New York at Albany
Economics Department
1400 Washington Avenue
Albany, NY 12222
hdiethorn@albany.edu

Gerald R. Marschke
State University of New York at Albany
Economics Department
1400 Washington Avenue
Albany, NY 12222
and National Bureau of Economic Research
gerald.marschke@gmail.com

Andrew J. Wang
National Bureau of Economic Research
1050 Massachusetts Avenue
Cambridge, MA 02138
awang@nber.org

This technical report is based upon work supported by the National Science Foundation under Grant No. DGE-1661278. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

1. Introduction

If in developed economies economic growth and sustained increases in living standards primarily arise from scientific and technological advances, then the state of the STEM workforce is of first-order importance. Are our educational institutions producing enough of the right kinds of STEM workers? Are these institutions passing over some households due for reasons of ethnicity, gender, geography, or income, thus leaving valuable human capital unharnessed? Do federal and state policies designed to grow the STEM workforce `work', for example, do they produce better jobs, increase innovation rates, increase the tax base, and stimulate local, regional, and national economies? What is the value to a student and to society of a STEM graduate degree or postgraduate training? Are there enough students in the STEM educational pipeline to meet current and future STEM demand?

To answer these questions, detailed, comprehensive data are needed that follow persons through the educational system into the workforce and beyond. Ideally, such data would contain information at the person level about programs of study, the identities of mentors and types and amounts of hands-on research, degrees or postgraduate training received, demographic characteristics, undergraduate (and earlier) preparation. The data should include employment information before and after university training, including salaries, occupations, industry, and detailed employer data that includes measures of firm inputs, outputs, and R&D activity.

In this paper, we document our first steps towards creating a new database that will enable researchers to measure the labor market outcomes of STEM PhD graduates and postdocs: what occupations and industries they work in, how much they earn, and how their careers develop over time. To prepare STEM students for the world of work, policy-makers, education officials, and the students themselves need to know where STEM graduates go, what they do, their career paths, how the labor market valuates their skills, and where in the economy their skills are most valued. Policy-makers also need appropriate data and models of STEM workforce demand to wisely allocate scarce educational resources and forecasts far enough in advance (the STEM PhD and postdoc pipeline is decades long) to formulate and implement policies to head off STEM shortages or gluts. Our envisioned data set ticks most of the boxes above to track the job history of STEM graduates and to investigate their contribution to production that is superior to or complements extant data sets for these purposes. Once this new dataset is complete, we plan to measure flows of STEM graduates into different sectors of the economy, estimate the returns to education for STEM PhDs and postdocs, and analyze the determinants of STEM labor demand in industry. Additionally, we will examine how the returns to education for STEM PhDs and postdocs vary in different industries and types of firms (e.g. research-intensive industries, start-up firms, established firms) and for different groups of workers (e.g. women, minorities). We are also interested in investigating the role of STEM workers in creating knowledge spillovers from universities to private business and in assessing the complementarity between STEM workers and innovation and technological change in industry. We will formulate and estimate new models of labor demand based on state-of-the-art econometric methods and innovative identification strategies that are made possible by the new, longitudinal data on new STEM workers created in this project.

The literature on the career paths of PhD scientists is largely based on two longstanding NSF surveys. The NSF's Survey of Earned Doctorates (SED) is an annual census of students earning a doctorate from a U.S. institution. Stephan (2006), for example, uses these data to show that only 37 percent of PhDs stay in their state of training, less for PhDs trained in Midwestern universities. The NSF's Survey of Doctorate Recipients (SDR) is a biennial survey of U.S. PhDs whose sampling frame is the individuals in the SED. The SDR follows respondents until they are 76 years of age and is rich in demographic information and information about employment activities. This is the most commonly used data set for research that requires documenting or analyzing careers of U.S. PhDs as it is the only data set containing a large, representative sample of doctorate recipients with long panels (e.g., Fox and Stephan, 2001; Ginther and Kahn, 2009).

Many critical issues related to the doctoral workforce cannot be answered with the SDR, however. To develop useful models and forecasts of PhD workforce demand, for example, firm-based data are necessary. There is a long-standing literature on the complementarity between technology and skills (e.g., Acemoglu, 1998; Goldin and Katz, 1998; Bresnahan et al., 2002). Firm panel data with detailed information on firm inputs and outputs is also necessary to evaluate the extent to which STEM workers stimulate the use of new technology. In contrast to the SDR, a linked employer-employee dataset of the doctoral workforce would enable researchers to investigate how technological change interacts with the utilization of STEM PhDs in private business and the determinants of PhD workforce demand.

A factor determining the demand for STEM PhDs and postdocs is private business's access to the fruits of university research. More broadly, understanding how knowledge spillovers across institutions within economies work is of interest because of the role spillovers likely play in both local economic development and national economic growth. Studies in both the economics and sociology of innovation literatures argue that new scientific knowledge is frequently "tacit" and difficult to transmit to the uninitiated via spoken or written communication (Polanyi, 1958, 1966). The most efficient means of transmission across organizational boundaries for tacit knowledge may be via person-to-person contact involving a transfer or exchange of personnel. Kaiser (2005) argues that the use of Feynman diagrams diffused relatively slowly and only through face-to-face interactions between physicists. The literature on science and innovation regularly find geographical limitations to the diffusion of ideas (e.g., Jaffe, 1989; Jaffe, Henderson, and Trajtenberg,1993; Audretsch and Feldman, 1996; Zucker, Darby, and Brewer, 1998; Mowery and Ziedonis, 2001) and these studies are often interpreted as evidence of the tacitness of knowledge (e.g., Feldman, 1994). Cohen, Nelson, and Walsh (2000) surveyed R&D managers on the means by which they gather and assimilate new technologies and find that firms access externally-located technology partly through the hiring of and collaboration with researchers from the outside. Moreover, they find that hiring/collaboration with outside researchers is complementary to other means of accessing externally produced knowledge, such as through informal communications with outsiders and more formal (such as consulting) relationships with outsiders.

Much of the literature that examines the mobility of scientists and innovators as a source of knowledge transmission focuses on the movement of academic scientists from academe to industry. Certainly, universities and academic ideas are important to the high-technology sector.

A number of studies offer strong evidence for geographically localized spillovers occurring in areas around major universities (Jaffe, 1986, 1989; Audretsch and Feldman, 1996; Henderson et al., 1998), suggesting both that academe is an important source of commercially-important ideas and that such ideas are not easily transmitted from the university labs in which they originate to the firms where they can be turned into commercial products. Work by Jensen and Thursby (2001), Agrawal and Henderson (2002), and Thursby and Thursby (2002) find that the best predictor that an academic idea leads to successful product roll-out is the participation of the inventing scientist. Thus, the hands-on involvement of academic scientists may in fact be necessary for an academic idea to take root in industry. In the biotechnology sector, Darby and Zucker and co-authors have examined the importance of working relationships between firms' bench scientists and top academic, or "star", scientists. They find that firms in the U.S. and Japan are more likely to enter the biotechnology industry in regions where star academics publish (Zucker, Darby, and Armstrong, 1998, 2002; Zucker, Darby, and Brewer, 1998; Zucker and Darby, 2001). They also find that university influence on nearby firm R&D productivity exists almost exclusively in firms whose bench scientists have working relationships with star academic scientists.

Some of the short-comings of the NSF surveys vis-a-vis investigating demand, technological complementarity, and university-trained workforce mediated spillovers are addressed by a new research platform developed by the U.S. Census in conjunction with prominent U.S. research universities called UMETRICS, which draw on university administrative databases to capture data on individuals—students, postdocs, faculty, staff—who work on federal and other sponsored research grants (UMETRICS is described in Lane et al., 2015). The advantage of UMETRICS for answering some of the questions that are out of the SDR's reach is that it can now be linked to Census data on employee jobs, individual demographic and socioeconomic characteristics (Buffington et al., 2016), and firms and business establishments (Zolas et al., 2015).

In this paper, we utilize UMETRICS data linked to both the ACS and LEHD to construct a new panel data set of PhD-holders and postdocs that contains detailed demographic information, employment information, and employer information, and that will allow researchers to track PhDs and postdocs forward and backward relative to their university training. The current paper focuses on the development of a machine learning strategy to predicting the postdoc status of university employees with PhDs. Since machine learning methods may be unfamiliar to many social scientists, we discuss the machine learning model used in this paper, random forests, in some detail, as well as the standard diagnostics used to assess the performance of these methods.[1] In future work, we will utilize this dataset to describe the career trajectories we observe in our new data set, formulate and estimate models of STEM PhD and postdoc demand, including evaluating the interplay between firm innovativeness and employment of STEM PhDs and postdocs, and measure the earnings gains from PhD and postdoc training.

[1] Future drafts will also test the quality of our imputation method by comparing the characteristics of predicted postdocs with those present in SED and SDR data.

The rest of the paper is organized as follows: In the next section, we discuss recent applications of machine learning in economics research and introduce the general idea behind our machine learning method used to impute postdoc status. We then discuss how we link multiple data sources available in the Census Bureau's Federal Statistical Research Data Centers to create a linked employer-employee dataset of the PhD workforce. Next, we describe the machine learning model, random forests, that we use to form postdoc predictions in our main analysis, as well as describe the standard diagnostics used to assess the performance of competing predictive models. These methods have been successfully implemented using our linked employer-employee dataset of the PhD workforce from within the Census Bureau's Federal Statistical Research Data Centers, but the results of this implementation are pending Census disclosure. Therefore, in order to clarify the methods we use to select and assess the machine learning model we use for prediction, we show our method applied to a publicly-available dataset of spam and non-spam emails available through the UC Irvine Machine Learning Repository. Since no single machine learning algorithm dominates all others across all applications (James, Witten, Hastie, and Tibshirani, 2013), we then develop a method that enables us to compare the performance of our random forest model with other models, including a rival machine learning model known as boosted trees. Lastly, we conclude with the main takaways from our analysis and plans for future work.

2. Machine Learning: An Application to Merging Datasets of Disparate Size

Improvements in data storage capacity and computing power have led to the increasing proliferation of "big data" and an ever-widespread use of machine-learning techniques that detect complex interactions among variables within these data with the aim of optimizing out-of-sample predictive performance for key variables of interest. Applications abound: machine learning is used in 1) information retrieval tasks such as predicting useful responses to search engine queries or the preferences of consumers, 2) speech-recognition software powering virtual assistants such as Siri and Alexa, and 3) image-detection technology used in self-driving cars and for face-recognition. These methods have not only gained favor due to their out-of-sample predictive performance, but also because many of these methods can be applied in cases where conventional methods fail, such as when the number of observations in a dataset is greatly exceeded by the number of potential predictors ($n << p$) or when a great number of interaction effects may exist among the predictors but is not known to the researcher *ex ante*. For these reasons, machine learning has gained a foothold in genomics by enabling researchers to predict cancer subclasses, treatment outcomes, and drug responses for new patients utilizing microarray gene-expression data that contains information on thousands of genes for a relatively small sample of individuals previously diagnosed with cancer (Tibshirani, Hastie, Narasimhan, and Chu, 2002; Berrar, Sturgeon, Bradbury, Dubitzky, 2003). The ways in which machine learning is transforming and will transform the products and services we use are seemingly innumerable. The ways in which these techniques can broaden and improve research in economics and other social sciences, however, are just beginning to be discovered.

A recent emergence of machine learning methods in applied economic research has taken place. Chalfin, Danieli, Hillis, Jelveh, Luca, Ludwig, and Mullainathan (2016) show that machine learning algortihms can aid in the hiring decisions of police departments to reduce the prevalence of police misconduct and in the teacher retention decisions of school districts to increase student test score gains. Kleinberg, Lakkaraju, Leskovec, Ludwig, and Mullainathan (2017) find that using a machine learning model to predict the crime risk of defendants could improve the bail decisions of judges in New York City such that a crime reduction of up to 24.7% could be achieved with no change in the jailing rate, or a 41.9% decrease in the jailing rate could be achieved without a change in the crime rate. These predicition policy problems (Kleinberg, Ludwig, Mullainathan, and Obermeyer, 2015) are not the only area where machine learning can play a role in economic research: Athey and Imbens (2017) survey ways in which machine learning methods can aid in the estimation of both average and heterogenous treatment effects.

Machine learning techniques are also useful in record linkage, where observations from separate datasets that lack a common unique identifier are matched using probabilistic methods. Feigenbaum (2016) outlines a machine learning approach that automates the linkage of fathers from the 1915 Iowa State Census to their sons in the 1940 Federal Census, creating a dataset capable of measuring intergenerational income mobility over the sample period. Chang, Emad, Lane, Tokle, and Weinberg (2016) link the Survey of Earned Doctorates (SED) to UMETRICS source data by utilizing a machine-learning based two-stage approach that leverages the availability of a greater number of link keys for a subset of UMETRICS observations.

With the data revolution upon us, administrative data collected by businesses, government agencies, and academic institutions play an increasing role in economic research (Einav and Levin, 2014; Varian, 2014). Existing survey-based or administrative-based big data sources such as the American Community Survey (ACS) and the linked employee-employer Longitudinal Employer-Household Dynamics (LEHD) database are rich sources of demographic and economic information, but, as with any dataset, are lacking in a wide-range of details of interest to researchers.[2] To obtain these details of interest not present in the big data source, it is common practice to identify a new dataset that does contain these variables, and then link across these sources using the "conventional method" illustrated in Figure 1: a new Dataset A is merged to an existing big Dataset B in order to create a new Dataset C comprised of all matched observations between the two datasets; the potential majority of the observations in big Dataset B that go unmatched are deleted as they lack variables that are key to the anticipated analysis.[3] Thus, part of the value of big datasets, namely a part of what makes them "big" in the first place

[2] For example, researchers wanting to use the ACS or LEHD to examine the career trajectories of Ph.D. recipients who have completed a postdoc face signinifcant difficulty as there is no postdoc occupation category in the ACS and no occupation categories at all in the LEHD. Researchers could try to infer such information using *ad-hoc* methods based on an individuals age and earnings, but assessing the accuracy of these methods can be problematic.

[3] In Figure 1 we assume, for the sake of simplicity, that Dataset A shares a single unique identifier with Dataset B, that each observation in Dataset A matches to an observation in Dataset B, that $N > n$, and that the j variables in Dataset B are distinct from the k variables in Dataset A (the "+1" variable in each dataset is the common unique identifier).

(i.e. containing many observations), is lost in the conventional process due to a tradeoff between more variables and more observations.

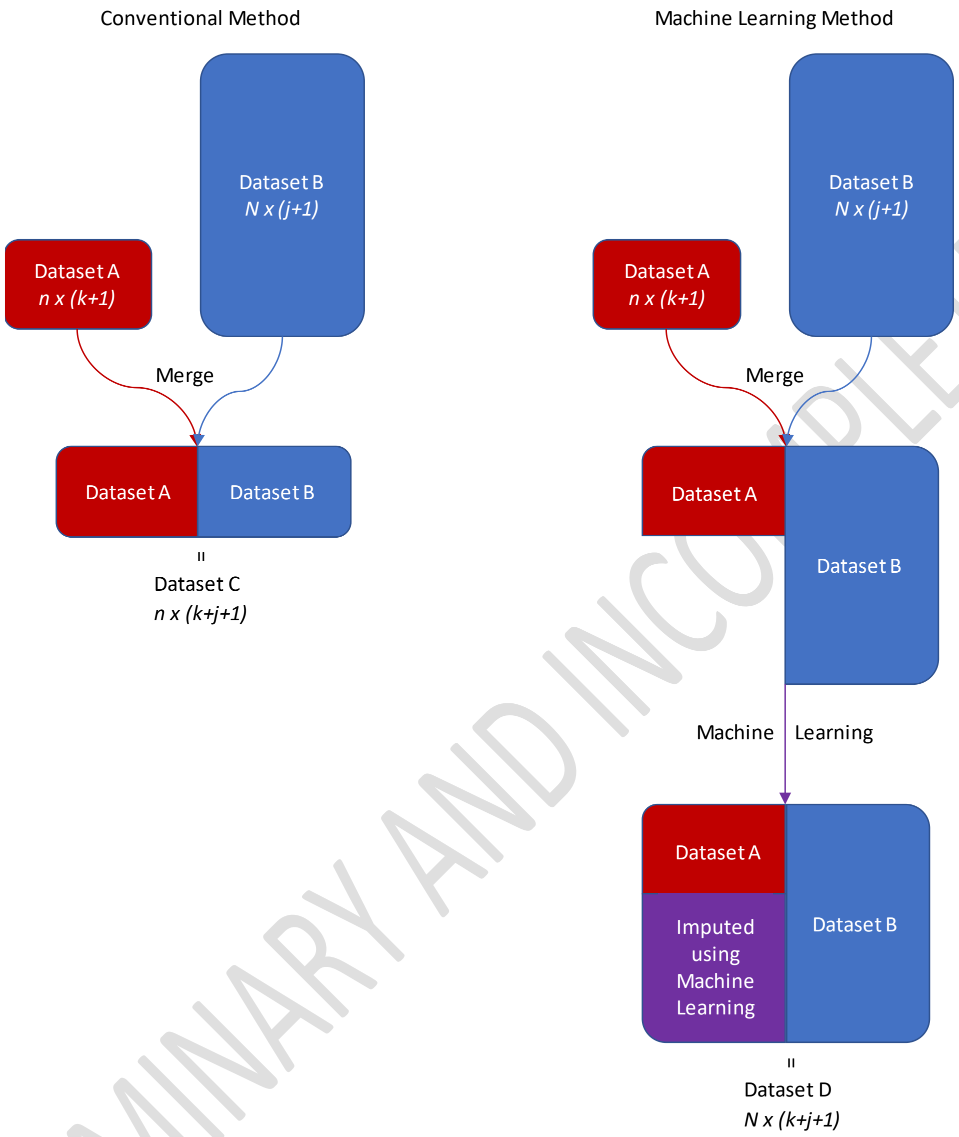
Depending on the nature of the two data sources, one may not need to eat away at big data to obtain additional variables of interest.[4] Machine learning methods can allow researchers to extract new variables from smaller datasets and impute their values for unmatched observations in big datasets. This is illustrated in the "machine learning method" in Figure 1: First, the new Dataset A is merged with big Dataset B, but, in contrast to the conventional method, we retain the unmatched observations from big Dataset B. Then, for each variable originally unique to Dataset A, the researcher trains a machine learning algorithm on the matched observations, using only those variables found in Dataset B as predictors. Lastly, the trained algorithm is used to predict the values of the key variables for the unmatched observations. Of course, the efficacy of this method depends on the extent to which the variables in Dataset B are predictive of the key variables in Dataset A, but this can be transparently assessed using methods discussed in this paper. Another key assumption is that the observations in Dataset A are representative of those contained in Dataset B (i.e. that we do not have a problem of *selective labels* as discussed in Kleinberg, Lakkaraju, Leskovec, Ludwig, and Mullainathan, 2017 and Mullainathan and Obermeyer, 2017).[5]

A major focus of this paper is to describe and demonstrate the efficacy of the machine learning approach to merging datasets of disparate size. Our application of this method utlizes a new university-based administrative data source, UMETRICS, in conjunction with matched ACS-LEHD data on U.S. doctorates, to create a new longitudinal database that enables researchers to measure the labor market outcomes of STEM Ph.D. graduates and postdocs over their career. As we anticipate that machine learning methods will increasingly be used by economic researchers for both analysis and data construction, we describe these methods in some detail, and refer the reader interested in the technical details to expert sources (e.g. Hastie, Tibshirani, and Friedman, 2009). The strategies implemented in this paper are sufficiently general to be applied to any context where researchers merge datasets of disparate size, and so should be of interest to a broad community of researchers.

[4] Of course, a researcher can always eat their data and have it too by contrasting the results of an analysis based on using the conventional method of merging datasets of disparate size with those obtained utilizing the machine learning approach outlined in this paper.

[5] This problem can be difficult to avoid. In our application, we only know whether an individual from a UMETRICS university is a postdoc or not if the person is grant-funded (this includes both federal and non-federal grants). Thus, our labels are selected based on grant-funded status, and so using our algorithm to predict on a sample that also contains individuals who are not grant-funded could be problematic if there are significant differences between individuals who are and are not grant-funded. As most postdocs are paid from grants (NSF Survey of Graduate Students and Postdoctorates in Science and Engineering, 2015), we believe this problem may not be a large issue. To check our priors, we will in the future compare the characteristics of our predicted postdocs to those found in the NSF's Survey of Earned Graduates (SED), Survey of Doctorate Recipients (SDR), and Survey of Graduate Students and Postdoctorates in Science and Engineering (GSS).

Figure 1: Conventional and machine learning methods of merging datasets of disparate size



Our research follows work by Goldschlag, Jarmin, Lane, and Zolas (2017) who use random forests to scale UMETRICS data with LEHD data in order to impute the research-trained status of workers found in the LEHD. Our machine learning based approach is similar to their approach, but differs in three fundamental ways. First, we compare the predictive power of different machine learning models based solely on their out-of-sample properties –rather than a balancing of in-sample and out-of-sample properties—since in-sample predictions are not needed in a subsequent analysis where the true postdoc status is known for in-sample observations. Second, we strictly separate the data used to select our preferred model from that used to estimate its out-of-sample error. A strict separation between the data used for model selection and that used for model assessment is necessary to protect against overly optimistic estimates of prediction error. Third, we implement a strategy for dealing with class imbalance by altering the probability cutoffs for positive prediction of our variable of interest (postdoc status), which leads to better predictive performance. Before discussing our machine learning strategy in detail, we describe how we link across multiple Census data sources to obtain a linked employer-employee longitudinal dataset of the doctoral workforce.

3. Description of Data Sources and Data Linkage Process

We utilize three major data sources in the construction of our analytical sample: 1) the American Community Survey (ACS), 2) Longitudinal Employer-Household Dynamics (LEHD) data, and 3) UMETRICS. These data are accessible to researchers with Special Sworn Status on approved projects via the Federal Statistical Research Data Centers (FSRDC) maintained by the U.S. Census Bureau.

The ACS is an annual survey administered by the U.S. Census Bureau and contains information on the occupations, educational attainment, and background characteristics (e.g. age, sex, place of birth, etc.) of survey respondents. After the 2000 Census, the ACS replaced the "long-form" of the decennial census which had previously been used to collect this information. Each annual ACS contains a nationally-representative snapshot of the American population covering approximately 3.5 million addresses annually (U.S. Department of Commerce, 2013). We utilize the annual ACS person files for years 2002-2015 in forming our analytical sample.

For each year of the ACS, we limit our sample to persons who indicate that they hold a doctorate degree, where persons are uniquely identified by nine-digit Protected Identification Keys (PIKs).[6] We then append each yearly ACS doctorate dataset to form an "ACS Doctorate Panel" that spans the years 2002-2015. Individuals may appear more than once in the ACS doctorate panel only if they are randomly surveyed in multiple ACS years. Thus, the unit of observation for the ACS doctorate panel is person-year (or PIK-year).[7]

LEHD data is maintained by the U.S. Census Bureau and is primarily based on administrative data collected by U.S. States such as Unemployment Insurance (UI) earnings data, as well as the Quarterly Census of Employment and Wages (QCEW). We utilize two LEHD data sets when creating our analytical sample: 1) The Employment History Files (EHF) and 2) The Employer Characteristics Files (ECF). For both datasets, we utilize all observations between 2002-2014. The EHF contains information on where individuals work each year and the earnings generated from their job(s) in each quarter. As in the ACS, individuals are uniquely identified by their PIK. Firms are identified by state employer identification numbers (SEINs) and establishments within each firm are identified by the SEIN reporting unit (SEINUNIT) so that an establishment is uniquely identified by SEIN-SEINUNIT (Vilhuber and McKinney, 2014). The raw EHF dataset is structured as a yearly job-level dataset where the unit of observation is an employee-employer combination within the given year (PIK-SEIN-SEINUNIT-year). For each

[6] PIKs are internal Census identifiers randomly generated for each individual in order to protect the privacy of each individual person while also facilitating linkage across Census data platforms (Mulrow, Mushtaq, Pramanik, and Fontes, 2011).

[7] We keep all ACS observations for which the person ever reports earning a doctorate – that is, if a person is surveyed in 2005 and reports not having earned a doctorate, but then is surveyed again in 2011 and reports having earned a doctorate, we keep both observations.

employer-employee combination, we have the quarterly earnings, and so we reshape the EHF into a quarterly job-level dataset so that the unit of observation becomes PIK-SEIN-SEINUNIT-year-quarter.[8]

The ECF contains establishment level information on US employers, including the establishment's federal Employer Identification Number (EIN), industry (six-digit NAICS code), and measures of the size and age of the firm associated with the establishment. The ECF is an annual establishment-level dataset and thus unique at the SEIN-SEINUNIT-year level.

To create our "LEHD panel", we link the EHF and ECF datasets by merging on establishment-year (SEIN-SEINUNIT-year). This effectively gives us the job profile for all individuals in LEHD states during 2002-2014 who have positive earnings reported in state UI data.[9] Since we are only interested in the job profile of individuals who have earned doctorates, we keep only those observations that are associated with PIKs found in our previously created ACS doctorate panel.[10] This LEHD doctorate panel is then linked to the ACS doctorate panel by person-year (PIK-year), creating our ACS-LEHD doctorate panel, where the unit of observation is person-establishment-year-quarter (PIK-SEIN-SEINUNIT-year-quarter). Figure 2 gives a diagrammatic summary of the process described above that is used to create our ACS-LEHD doctorate panel.

Our ACS-LEHD doctorate panel has one shortcoming that prevents us from carrying out a comparative analysis of PhD-holders who have and have not completed a postdoc: there is no way for us to tell which observations correspond to a quarter in which a person is employed as a postdoc, nor are we able to tell if a person has ever been employed as a postdoc. Therefore, we must introduce a third data source, UMETRICS, to obtain labels for postdoc status for a subset of our ACS-LEHD doctorate observations; once we have postdoc labels for a subset of our observations, we can use a machine learning approach to predict the postdoc status of the unlabeled subset of our ACS-LEHD doctorate observations. This then will allow us, in future work, to examine how employment as a postdoc affects career trajectories and how the postdoc-trained workforce impacts the firms where they work.

UMETRICS (Universities: MEasuring The impacts of Research on Innovation, Competitiveness, and Science) is a database maintained by the Institute for Research on Innovation & Science (IRIS) at the University of Michigan and is accessible to Special Sworn Status researchers on approved projects in the Census FSRDCs[11]. UMETRICS is based on

[8] Individuals employed at an establishment but who do not have strictly positive earnings at their employing SEIN in a given quarter will not have earnings reported within that SEIN for that quarter (Vilhuber and McKinney, 2014). Therefore, we code earnings as zero for any quarter where earnings is missing.

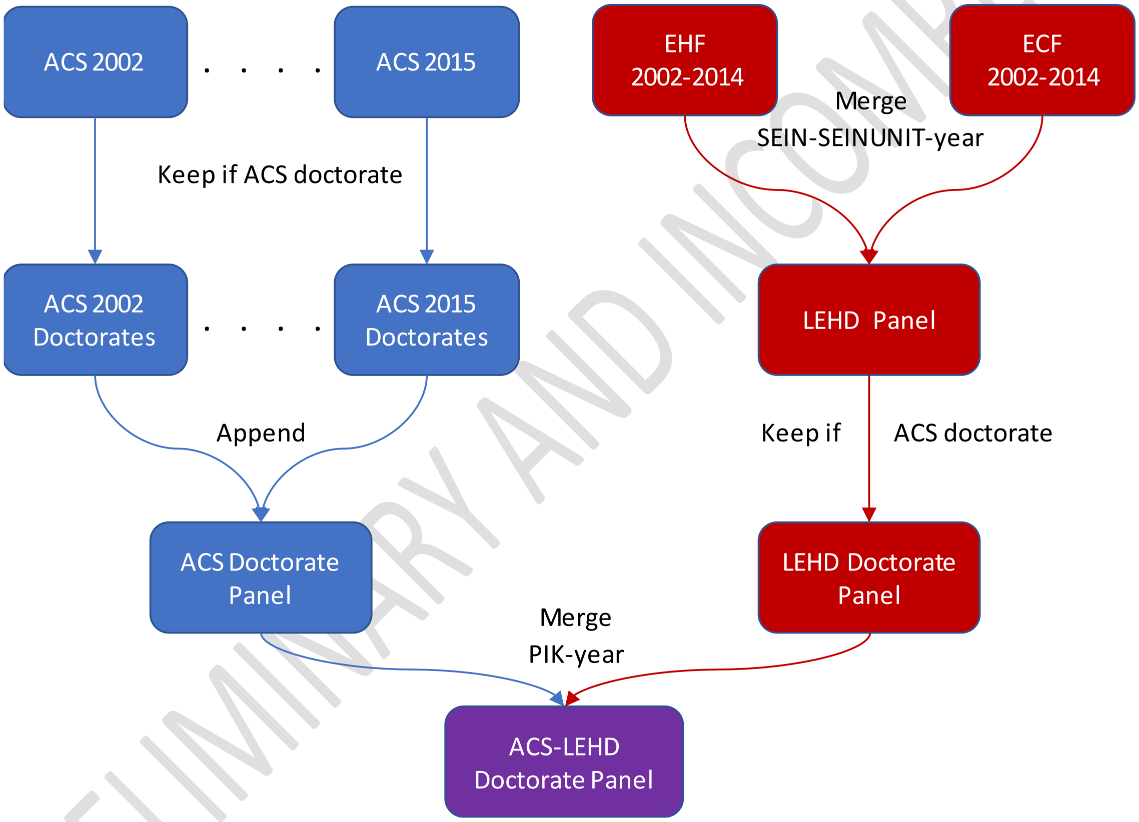
[9] U.S. states can voluntarily opt into or out of the LEHD program. LEHD states refer to those states that participated in the LEHD program in a given year.

[10] These observations are dropped either due to the PIK not being found in the ACS for years 2002-2015 or due to the PIK not having earned a doctorate in any years surveyed as part of the ACS.

[11] UMETRICS data can also be accessed via the IRIS virtual data enclave.

administrative data obtained from 19 IRIS member universities that contains information on grants or awards received by each university including which university employees are paid from each award and what vendors receive funds from the awards in exchange for goods and services (The Insititute for Research on Innovation & Science, 2017).[12] UMETRICS data spans the years 2001-2015, with more universities being added to the sample over time.

Figure 2: Creation of ACS-LEHD Doctorate Panel



We utilize the Employee Transaction File (ETF) of the UMETRICS 2016Q3a release. The ETF contains university payroll transactions for employees paid on any (1) research-related federal or non-federal grants or (2) non-research-related activities such as work-study programs. Each IRIS-member university is assigned a unique “institutionid” for de-identification purposes, and each university employee paid on a grant or award is assigned an institution-specific “employeeid” so that individuals within UMETRICS can be uniquely identified by institutionid-

[12] The 19 IRIS member universities are as follows: Boston University, Michigan State University, New York University, Northwestern University, Ohio State University, Pennsylvania State University, Princeton University, Purdue University, Rutgers University, Stony Brook University, University of Arizona, University of Hawaii, University of Illinois at Urbana-Champaign, University of Iowa, University of Kansas, University of Michigan, University of Missouri, University of Pittsburgh, and University of Wisconsin.

employeeid.[13] Each observation contains a "unique award number" that identifies an award and its funding source, the employeeid and institutionid of the person being paid off of the award, the period start date and end date that represents the beginning and end of the monthly pay period, and the occupational class of the employee. The occupational class encompasses 5 major groups of workers: Faculty, Staff, Post Graduate Researcher (i.e. Postdoc), Graduate Student, and Undergraduate.[14] Persons in UMETRICS have been matched to Census persons and assigned a PIK, and so we are able to match individuals in UMETRICS to our ACS-LEHD doctorate panel. Once we link UMETRICS ETF to our ACS-LEHD doctorate panel, we obtain the occupational classification from the ETF for matching observations in the ACS-LEHD doctorate panel, resulting in a subset of observations in the ACS-LEHD doctorate panel where true postdoc status is known.

Figure 3 shows the steps used to merge our UMETRICS data into the previously created ACS-LEHD Doctorate Panel—we detail these steps in the Data Appendix. A quick summary of these steps is the following: First, we identify the postdoc status of a subset of our ACS-LEHD doctorate panel by merging with UMETRICS ETF data. Next, we keep only those observations in our overall sample where the employee is working in either "Colleges, Universities, and Professional schools" (NAICS=611310) or "General Medical and Surgical Hospitals" (NAICS=622110) since the vast majority of UMETRICS university employees are classified as working in these industries. This should improve the representativeness of our UMETRICS subsample, which is important since the UMETRICS subsample will be used to train our machine learning model used to predict the postdoc status of the non-UMETRICS subsample. This leads us to our final prediction sample referred to as the "ACS-LEHD Academic Doctorate Panel with UMETRICS" in Figure 3, which is unique on person-year-quarter (PIK-year-quarter).[15]

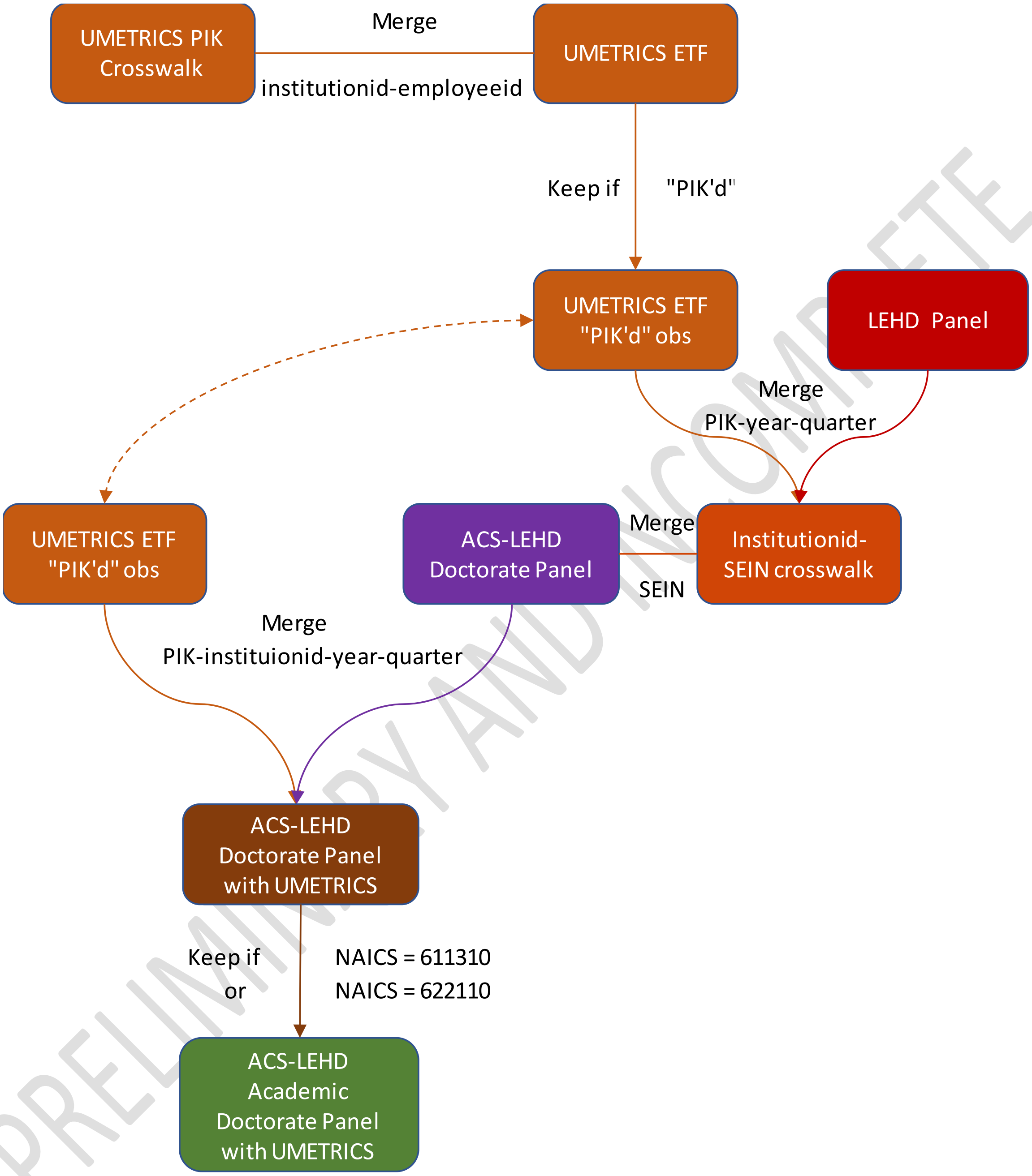
Table A.1 displays the variable names and definitions for this dataset. Our goal is to predict, for each individual in the prediction sample, which quarters between 2002-2014 (if any) represent a period of employment as a postdoc. Our method, as discussed in the next section, is to utilize the UMETRICS subset of our data to train a machine learning algorithm where our postdoc indicator variable is the target, or the variable to be predicted, and the rest of the variables/features listed in Table A.1 are the predictors.

[13] While a single individual will only have one employeeid within a single institutionid, if that individual moves to a different IRIS member university, he will be identified by a new institutionid-employeeid. However, UMETRICS data has been matched to Census PIKs, which should be able to identify the same individual as they move from one university or job to another.

[14] Employees classified as Staff are then classified into one of 10 subcategories so that there is a total of 14 occupational categories in the data altogether.

[15] We are interested in predicting which quarters (if any) of an individual's career are spent as a postdoc, and so a dataset unique on person-year-quarter is sufficient for this purpose. Our LEHD-based variables (see Table A.1), such as the job count variables and total earnings variables, are created to incorporate useful information from our more "general" job-level and non-NAICS restricted intermediate datasets used to form our final NAICS-restricted quarterly person-level prediction sample.

Figure 3: Creation of ACS-LEHD Academic Doctoral Panel with UMETRICS



4. Random Forests: What They Are and How to Tune Them

We utilize the random forest algorithm originally developed by Breiman (2001) and implemented in the R package randomForest (Liaw and Wiener, 2002) to predict the postdoc status of our ACS-LEHD doctorate sample. Random forests are one of the most popular out-of-the box machine learning methods, being utilized in a variety of tasks such as image classification (Bosch, Zisserman, and Munoz, 2007), gene selection (Díaz-Uriarte and Alvarez de Andrés, 2006), and land cover classification (Gislason, Benediktsson, and Sveinsson, 2006). Random forests work by “growing” an ensemble of decision trees, obtaining predictions from each of these trees, and then averaging the predictions across these trees to generate a final prediction.[16] In this subsection, we give a summary of classification trees and the random forest algorithm used for classification.

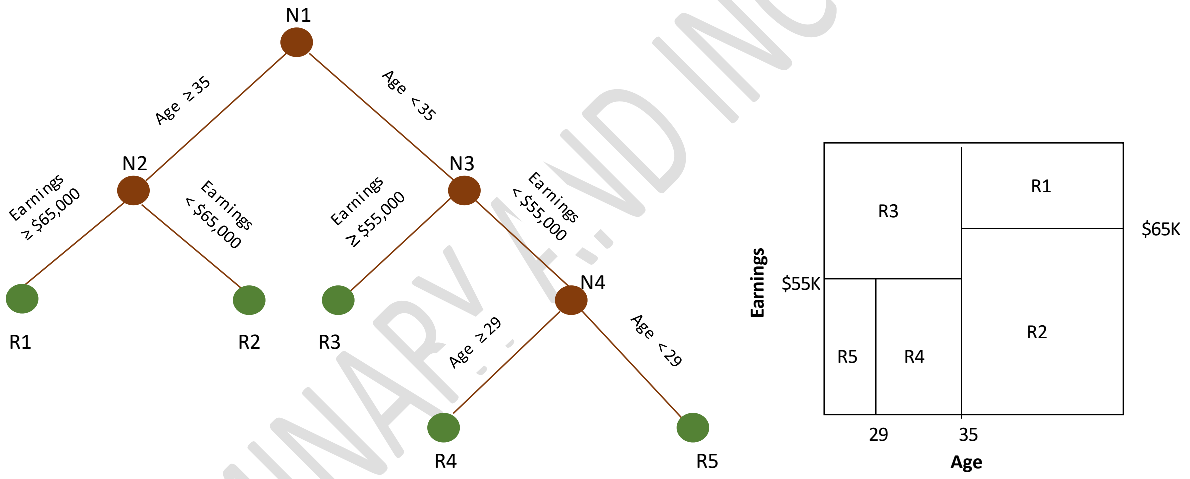
Classification trees are grown by iteratively partitioning a sample of data to group together observations with the same class label (e.g. “postdoc” or “not postdoc”) in a process known as recursive binary splitting. Figure 4 shows a fictional classification tree based on two predictors, along with its equivalent predictor-space representation. Classification trees partition the data at each step by selecting a predictor-cutpoint combination as the basis for the split; for example, in Figure 4 at internal node *N1*, observations are split based on the predictor age and the cutpoint of 35 years, resulting in the two daughter nodes *N2* and *N3*. Generally, to determine how to split the observations, an optimal cutpoint for each predictor is calculated, and the predictor-cutpoint combination that gives the greatest gain in node purity is chosen to divide the data into two daughter nodes.[17] This process is continued until a stopping criterion, such as the minimum number of observations allowed in a node, is satisfied. Observations in each terminal node (or leaf) of the decision tree are then predicted as belonging to the class held by the majority of observations in that terminal node. In Figure 4, the terminal nodes are labeled *R1-R5* since the observations grouped into these nodes are the same that would appear in the identically-named regions in the predictor-space representation of the classification tree.

[16] See Hastie, Tibshirani, and Friedman (2009) for an extensive and technical treatment of machine learning, and see James, Witten, Hastie, and Tibshirani (2013) for an introductory treatment with applications using R statistical software. Breiman, Friedman, Olshen, and Stone (1984) is the classic reference for classification and regression trees. As a note of terminology, a decision tree is referred to as a classification tree when the variable to be predicted is a categorical variable, and is referred to as a regression tree in cases where the variable is non-categorical (e.g. continuous variables and count variables).

[17] Let p_{mk} be the proportion of observations at internal node *m* that are of class *k*. Then the Gini index at that node is calculated as $\sum_{\{k=1\}}^{K} p_{mk}(1-p_{mk})$ where a smaller value of the Gini index represents a node of greater purity. The predictor-cutpoint combination used to split at an internal node is chosen so that the resulting two daughter nodes give the largest decrease in the Gini index, where the decrease in the Gini index is calculated as follows: first, the Gini index for each daughter node is calculated and weighted by the proportion of parent-node observations falling into that node, and then these measures are subtracted from the value the of the Gini index of the parent node. Recursive binary splitting is referred to as a top-down, greedy approach because at each stage, the data is partitioned to maximize the gain in node purity at that step without considering how a given partition will affect future partitioning of the data and thus ultimate node purity at the terminal nodes – this is done for computational feasibility.

A strength of classification trees is that they automatically capture interaction effects among predictors without the user needing to specify a set of interaction terms *ex ante*.[18] A major weakness of classification trees is that they suffer from high variance: the structure of a given decision tree is highly dependent on the data used to train the model such that a small change to the data may result in a non-negligible change in the tree structure, which can cause a noticeable change in the predictive performance of the model as measured by the model's out-of-sample (or test) error. In order to mitigate the weaknesses of unstable (high variance) learners such as classification trees, Breiman (1996) introduced an ensemble method known as bagging (Bootstrap AGGregatING).[19] This method works by taking B bootstrap samples from the available training data, fitting a classification tree to each of the bootstrap samples, generating a prediction from each tree for each observation, and then classifying each observation based on a majority vote – that is, the final prediction for each observation is the most commonly predicted class among the B predictions.[20] Figure 5 gives a schematic representation of a bagged tree model.

Figure 4: Example of a classification tree and its equivalent predictor-space representation



Random forests improve upon bagged trees by introducing a source on randomness into the tree growing process: at each internal node in each tree, a random subset of the available predictors is first chosen, and then the best split among these randomly chosen predictors is used to split at the node; this contrasts with bagged trees, where the best split among all available predictors is chosen. It may seem odd that random forests typically perform better than bagged

[18] Mullainathan and Spiess (2017) highlight this aspect of decision trees in a regression context.

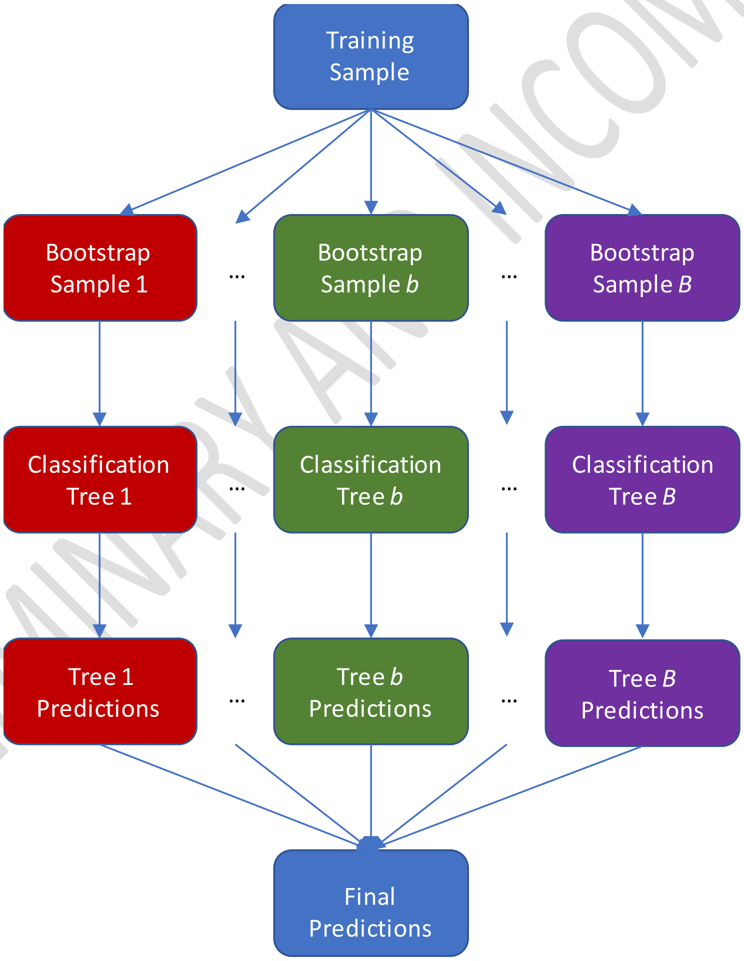
[19] Ensemble methods are methods that generate predictions by combining the predictions of a set of "base-learners" such as decision trees. Popular ensemble methods include bagging, boosting, averaging, and stacking.

[20] This is assuming the default threshold of 50%.

trees given that the only difference between these two methods is that random forests restrict the available information considered at each node of each tree. However, the intuition for the performance improvement of random forests over bagged trees stems from the fact that the variance of an average of identically distributed random variables is decreasing in the pairwise correlation of these variables. By introducing a source of randomness into the tree-growing process, random forests decorrelate the trees, thus leading to a smaller variance in prediction relative to bagged trees.[21]

Each tree in a random forest is grown on a bootstrapped sample of the original training data which, due to sampling with replacement, contains approximately two-thirds of the original

Figure 5: Schematic representation of Bootstrap Aggregated ("Bagged") Classification Trees



[21] See Hastie, Tibshirani, and Friedman (2009) Ch. 15 for technical details.

training observations. The approximately one-third of the original training observations that are not used to train a given tree are referred to as the out-of-bag (OOB) observations of that decision tree. It follows that each observation of the original training data will be in the OOB sample of approximately one-third of the decision trees grown in a random forest. The OOB error rate of a random forest is obtained by generating predictions for each original training observation from only those trees for which it is part of the OOB sample, measuring the average error in classification for each observation based on its OOB predictions, and then averaging these error rates across all observations.[22] The OOB error rate is a measure of the predictive performance of a random forest and is used to select the number of decision trees grown in a random forest model: The number of trees is selected to be large enough that the OOB error rate becomes relatively stable – with no risk to overfitting by growing too many trees.

Random forests contain one hyperparameter that the user tunes to obtain the best random forest model: the number of randomly selected variables considered for node splitting at each node in each decision tree, which we refer to as the number of "splitting variables."[23] One way to tune a random forest model is to compare the OOB error rates that are obtained by changing the number of splitting variables and then selecting the hyperparameter value that yields the lowest OOB error rate. However, since the OOB error rate is a measure of the overall classification error rate of the model, it is sensitive to the probability cutoff used for positive prediction. By default, the cutoff is set to 0.5 meaning that, in our application, all observations with a predicted probability of being a postdoc greater than 0.5 would be classified as postdocs.[24] While a seemingly reasonable default, the 0.5 probability threshold may not be optimal as there is no guarantee that this threshold minimizes classification error, and even if it does achieve the minimum classification error, such a property may not be desirable in the presence of class imbalance since prediction will tend to favor the most commonly occurring class, leading, in our case, to a greater prevalence of false negative predictions compared to false positive predictions. A threshold that balances the two types of errors may be more desirable, and so tuning a random forest model using a metric that is sensitive to the choice of the cutoff should generally be avoided in the case of class imbalance.[25]

[22] The OOB error rate and predictions are calculated automatically in the implementation of the random forest algorithm in the R randomForest package.

[23] For classification problems, the recommended default value for the number of splitting variables (m) is equal to the square root of the total number of predictors (p) (Hastie, Tibshirani, and Friedman, 2009). Another hyperparameter that could be adjusted for a random forest is the size or depth of the individual trees making up the random forest ("minimum observations in node"); however, Hastie, Tibshirani, and Friedman (2009) suggest that tuning this parameter does not typically lead to large changes in predictive performance, especially in the case of classification (596). We leave the minimum observations in node hyperparameter set to one, which is the default value for classification trees (Hastie, Tibshirani, and Friedman, 2009).

[24] The probability of being a postdoc is calculated as the proportion of decision trees in a random forest that predict an observation as belonging to the postdoc classification.

[25] If costs differ between false positive and false negative errors, a threshold minimizing the cost could be selected.

Table 1: Confusion matrix

	Actual	
Predicted	**Not Postdoc**	**Postdoc**
Not Postdoc	True Negative (TN)	False Negative (FN)
Postdoc	False Positive (FP)	True Positive (TP)

A preferred alternative is to rely on a method that explicitly considers the tradeoff between false positive and false negative errors as the cutoff is altered. One such method is to use the OOB predictions to graph a Receiver's Operating Characteristic (ROC) curve for each value of the hyperparameter and choose the number of splitting variables that maximizes the area under the ROC curve.[26] To understand the reasoning behind this method, it is helpful to first introduce what is referred to as a classification method's confusion matrix, as shown in Table 1. A confusion matrix counts the number of true positive, true negative, false positive, and false negative predictions made by a classifier. For a random forest model, we can obtain a confusion matrix based on how well the model predicts the classes of OOB observations. From there, we can calculate the various error and accuracy measures shown in Table 2.

Table 2: Accuracy and Error Measures

Name of Measure	**Definition**
Accuracy	(TP + TN) / (TP + TN + FP + FN)
Misclassification/Error Rate	(FP + FN) / (TP + TN + FP + FN) ≡ 1 - Accuracy
True Positive Rate (TPR)	TP / (TP + FN)
False Positive Rate (FPR)	FP / (TN + FP)
True Negative rate (TNR)	TN / (TN + FP) ≡ 1-FPR
False Negative Rate (FNR)	FN / (TP + FN) ≡ 1-TPR
Positive Predictive Value (PPV)	TP / (TP + FP)
Negative Predictive Value (NPV)	TN / (TN + FN)

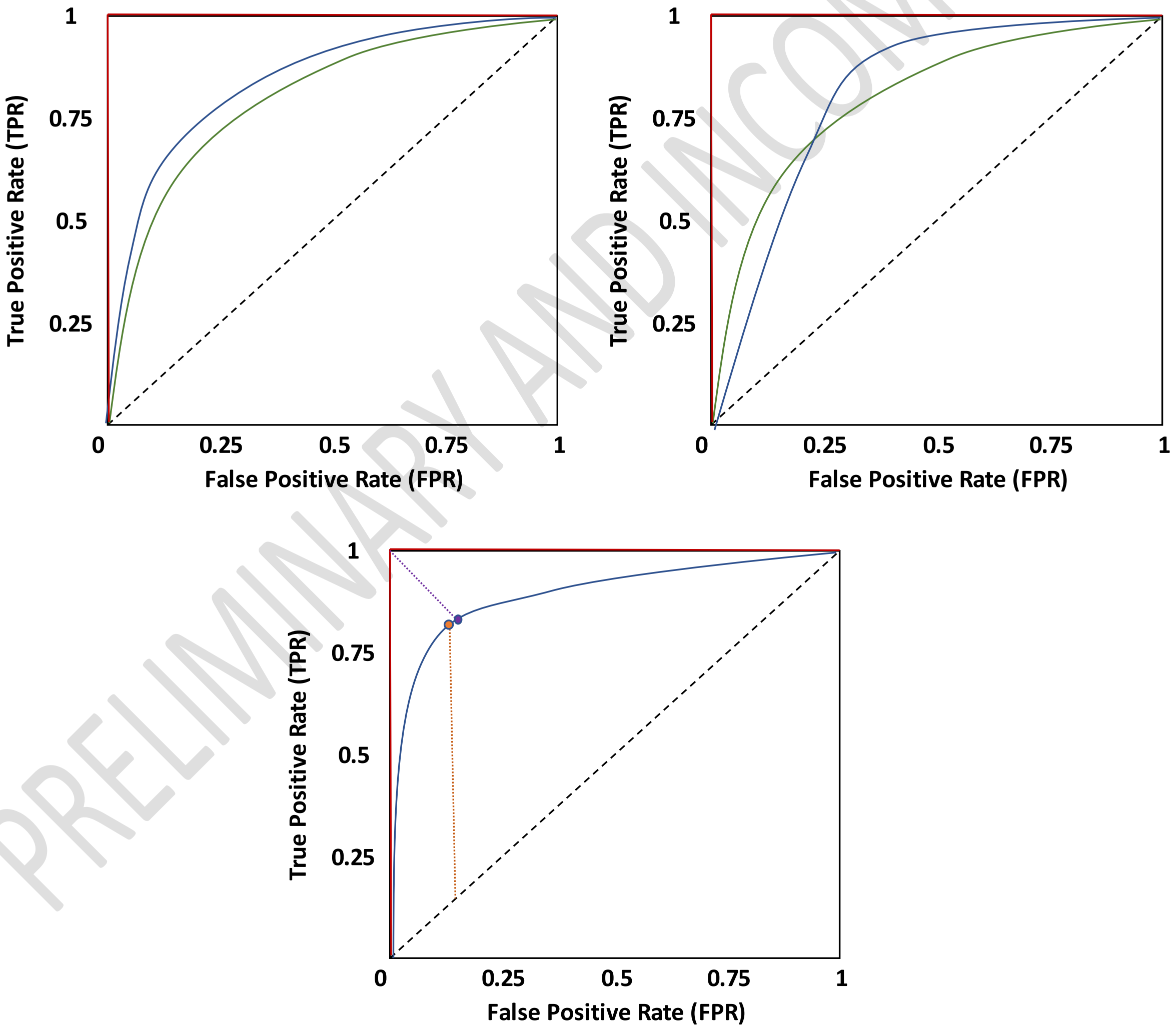
An ROC curve is simply a plot of the true positive rate vs. the false positive rate achieved by a given predictive model across all alternative probability cutoffs. The top panel of Figure 6 shows two examples of ROC plots. The dotted diagonal line in each plot represents the performance expected using random guessing for prediction. The connected red lines touching the border represents the performance of a perfect predictive model since it intersects with point

[26] See Lahiri and Yang (2013) and Kuhn and Johnson (2013) for an overview of ROC curve analysis.

(0,1) in ROC space, which is associated with a 100% TPR and 0% FPR. The blue and green curves lying above the diagonal are two ROC curves, each associated with a separate hypothetical predictive model such as two random forests with different values for the number of splitting variables. Each point on the ROC curve gives the (FPR, TPR) combination achieved by a particular probability threshold; points farther to the right along a given ROC curve correspond to lower probability cutoffs.[27]

In the upper left-hand panel of Figure 6, we see that the model corresponding to the blue ROC curve strictly dominates the model corresponding to the green ROC curve since the "blue

Figure 6: Example Receiving Operator Characteristic (ROC) plots



[27] Keeping in mind that TPR ≡ 1 – FNR, an ROC curve explicitly shows that lowering the probability cutoff results in a lower incidence of false negative errors at the cost of an increase in false positive errors.

model" achieves a higher true positive rate for any given false positive rate, and thus outperforms the "green model" across all probability thresholds. However, deciding between models based on visual inspection of ROC curves is not always so straightforward. For example, in the right-hand panel of Figure 6, we have ROC curves that intersect and overlap, meaning that what model is "better" depends on the probability threshold under consideration. Without a particular probability threshold in mind *a priori*, a judicious approach is to select the model that exhibits the greatest "global" skill over all possible probability thresholds, rather than a model that exhibits the greatest "local" skill at a particular probability threshold such as the 0.5 default. This can be done by calculating the area under each ROC curve and then selecting the model that gives the maximum area under the curve (AUC). [28] Since the AUC of a model takes account of a model's performance across all probability thresholds, it is a more appropriate metric to use when tuning a random forest compared to the OOB error rate of the model which is necessarily dependent upon the choice of a probability cutoff.

After tuning a random forest model by selecting the number of splitting variables that maximizes AUC, one still needs to determine the appropriate probability cutoff to use for prediction. This is particularly important in cases where class imbalance is an issue, as the default cutoff is likely to overpredict the most commonly occurring class. This can be done by selecting the threshold that maximizes some measure of local skill. Two popular cutoff choices are those that either minimize the sum of squared false positive and false negative rates ($FPR^2 + FNR^2$) or maximize the sum of true positive and true negative rates (TPR + TNR). We refer to the cutoff that minimizes the sum of squared false positive and false negative rates as the "top-left" cutoff, as this cutoff identifies the point on the ROC curve closest to point (0,1) in ROC space. This cutoff is represented in the bottom panel of Figure 6 as the purple point on the ROC curve. We refer to the cutoff that maximizes the sum of true positive and true negative rates as the "Youden" cutoff since this cutoff maximizes the Youden Index (Youden, 1950): TPR + TNR – 1. The Youden cutoff identifies the point on the ROC curve where the model is most skilled relative to random guessing, that is, where the vertical distance between the ROC curve and the no-skill diagonal is greatest. This cutoff is represented in the bottom panel of Figure 6 as the orange point on the ROC curve. To choose between these cutoffs, one can use the accuracy and error measures in Table 2 and choose the cutoff that is most desirable, in terms of the metrics viewed as most important, for the application at hand.

5. Model Selection and Assessment of Random Forests: An Application for Predicting Postdoc Status

In this section, we describe our machine learning based strategy for using the UMETRICS subsample of the ACS-LEHD doctorate panel to predict postdoc status for the rest of the observations in the ACS-LEHD doctorate panel. Unfortunately, the exact results of our method are pending Census disclosure. As we await disclosure review, and because we think it is instructive to clarify how our methods work with an example, we show our method applied to a publicly-available dataset of spam and non-spam emails available through the UC Irvine

[28] AUC lies in the range [0,1], with a perfect predictive model having AUC=1 and random guess having AUC=0.5.

Machine Learning Repository.[29] This dataset contains information from 4601 emails including how many times an exclamation mark appears in the email and the longest string of ALL CAPS in the email, as well as whether the email was ultimately classified as spam or not spam. This enables the dataset be used to predict whether an email is spam based on 57 characteristics of the text. All graphics pertaining to the performance of different models are based on this spam data; all tables describing model performance are also based on spam data where not left blank. Notwitstanding that these results are based on the spam dataset, we describe our method in terms of predicting postdocs so that the methods can be understood in terms of the context of our aims, which is to obtain accurate predictions of postdoc status as an essential step towards producing a linked employer-employee longitudinal dataset of the doctoral workforce that enables researchers to analyze the labor market outcomes of STEM PhD graduates and postdocs.

A quick summary of our method is as follows: First, we split the UMETRICS subset of our ACS-LEHD doctorate data into a training set (50%) and a test set (50%). The training set is used to train competing random forest models, the area under the ROC curves generated from the OOB predictions from each model are used to compare our competing random forest models (i.e. tune the splitting variables hyperparameter), and then the OOB predictions are used to identify alternative probability cutoffs for positive prediction to mitigate for class imbalance. To assess the predictive accuracy of our tuned random forest model, we estimate the misclassification error using the test data. Once our model is assessed, we then retrain the model on all the UMETRICS data and use this trained random forest model to predict the postdoc status of the non-UMETRICS observations in our ACS-LEHD data. Table 3 outlines this strategy for model selection, assessment, and prediction. We give the rationale for our method in what follows.

Table 3: Random Forest Model Selection (Steps 1-3), Assessment (Step 4), and Prediction (Step 5)

1. Partition data into a training set and test set (50%-50% split).
2. For random forest models with different number of splitting variables:
    i. Train model on the training set.
    ii. Output model that performs the best in terms of AUC using OOB predictions.
3. Identify alternative cutoffs/thresholds based on OOB predictions.
4. Estimate generalization error using the test set.
5. Retrain selected model on all labeled /UMETRICS data and use to predict postdoc status for all non-labeled/non-UMETRICS observations.

The training (or apparent) error rate of a predictive model is an overly optimistic measure of prediction error; this is because when any model is trained or estimated on a given dataset, it is likely not only to discover signals in that data that are useful for out-of-sample prediction, but

[29] It is known as the “Spambase Data Set” on the UCI Machine Learning Repository, and can be found at https://archive.ics.uci.edu/ml/datasets/spambase, or easily accessed via the R package “ElemStatLearn.”

also to fit sample-specific noise. To properly assess the predictive power of a model, a portion of data should be withheld during the training process so that performance of the model on out-of-sample data can be accurately assessed. Therefore, we partition the data into a training set used to train and tune our random forest model (model selection), and save the other 50% of our UMETRICS data to be used as our test set to estimate the out-of-sample performance of our tuned random forest model (model assessment). It is important that no data used in model assessment is used in model selection (i.e. feature selection, hyperparameter tuning, model comparison), and vice versa.[30] Therefore, we utilize OOB predictions, rather than test set predictions, as the basis of our performance measures used for model tuning. Our test set predictions are only used when estimating the generalization error of our tuned random forest model.

After splitting our UMETRICS subsample into a training set and test set, we train four random forest models, each with a unique value for the number of splitting variables considered at each tree node.[31] Figure 7 shows the OOB error rate for each of the four different random forest models, where we use the spam data from the UCI Machine Learning Repository as the basis for these figures as our real results are pending Census disclosure. As we can see, the OOB error rate for each model becomes relatively stable such that the model with the number of splitting variables (m) equal to the square root of the number of available predictors (p) tends to perform best on this metric. Based on Figure 7, we use 1000 trees in our random forest, which is well past the point where the error rates for each model stabilize. As noted in the previous section, the OOB error rate measures the classification error rate of the model, and is thus sensitive to the probability cutoff used for positive prediction. Rather than selecting the random forest model with the lowest OOB error, we would like to select a model based on the global skill of that model over all possible probability thresholds. Therefore, we tune the number of splitting variables by first calculating the area under the ROC curve (AUC) for each random forest model and then selecting the model with the number of splitting variables that maximizes AUC. Figure 8 shows that the random forest model with the number of splitting variables equal

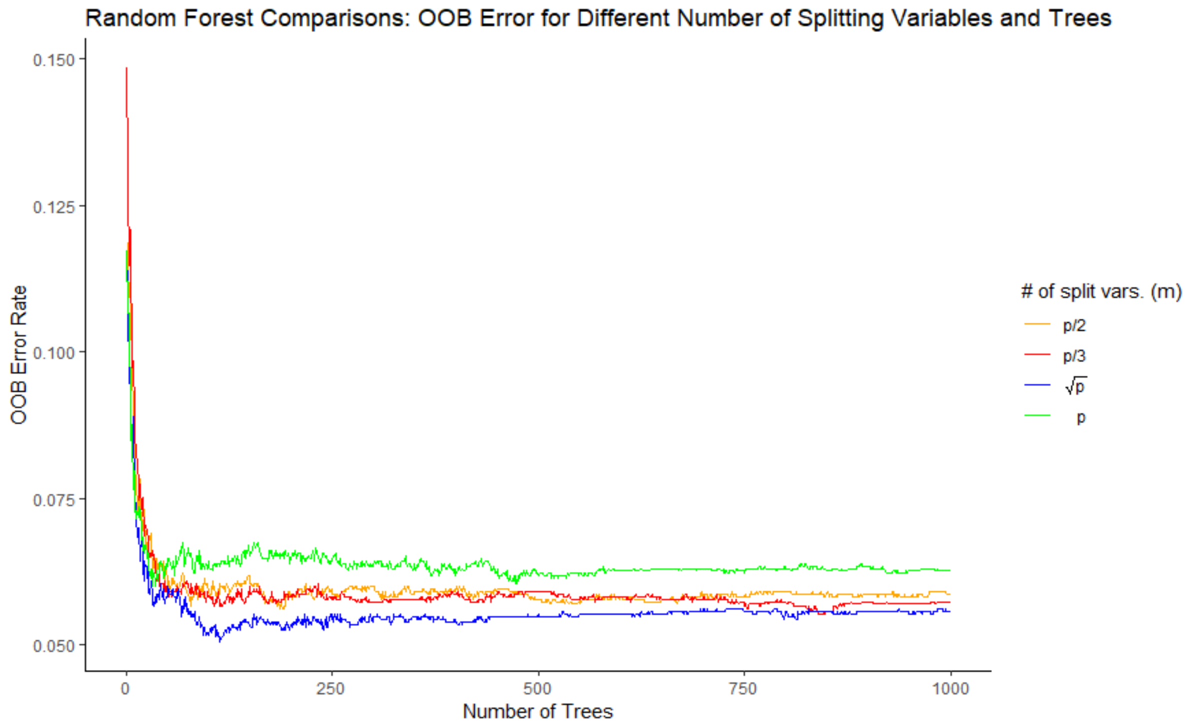
[30] Ambroise and McLachlan (2002) show that using a full dataset for feature selection prior to partitioning the data into a training set and test set leads to an optimistic bias in cross-validation (CV) error estimates. Varma and Simon (2006) show that the CV error rate used to tune a model underestimates the generalization error of the model, although Tibshirani and Tibshirani (2009) provide evidence that this mostly occurs in cases where the number of features greatly exceeds the number of observations ($n << p$). Cawley and Talbot (2010) show that tuning a model's hyperparameters using the full set of data prior to partitioning the data and calculating the test set error will lead to an optimistically biased estimate of the generalization error. Hastie, Tibshirani, and Friedman (2009) warn that tuning hyperparameters or selecting a model based on minimizing the test set error will cause the test set error to underestimate the generalization error.

[31] For computational feasibility, Kuhn and Johnson (2013) suggest only tuning over a limited number of values for the number of splitting variables. Our first three values are chosen following the exposition in James, Witten, Hastie, and Tibshirani (2013) whom compare the default value of $m = \sqrt{p}$ with $m = p/2$ and $m = p$ (bagged trees). We also consider $m = p/3$, which corresponds to the suggested default value for random forests in a regression context (Hastie, Tibshirani, and Friedman, 2009) and puts the number of splitting variables roughly halfway between $m = \sqrt{p}$ and $m = p/2$ in our application. In all four cases, we round the number of splitting variables to the interger value closest to these targeted values.

to the square root of the number of available predictors achieves the greatest AUC, and thus represents our tuned random forest model. Again, Figure 8 is based on the spam data from the UCI Machine Learning Repository as our postdoc prediction results are pending Census disclosure.

Having selected the random forest model with the greatest global skill, we now identify alternative probability thresholds to use for positive prediction of postdoc status. Kuhn and Johnson (2013) suggest considering alternative probability cutoffs for positive prediction to account for class imbalance since class imbalance leads to error rate imbalance – a predictive model seeking to minimize classification error will tend to favor predicting the most commonly occurring class. In our case, since postdoc is the rarer class, we would expect the false negative rate to exceed the false positive rate (or equivalently, the true negative rate to be greater than the true positive rate). Therefore, in addition to considering the default 50% cutoff and the cutoff that minimizes the total classification error, we also examine two thresholds that are less sensitive to class imbalance. The first of these alternatives is the "top-left" cutoff which corresponds to the point on the ROC curve that minimizes the Euclidean distance between the ROC curve and point (0,1) in ROC space. The second alternative is the "Youden" cutoff which corresponds to the point on the ROC curve that is the greatest vertical distance away from the no-skill (random) forecast represented by the diagonal line in Figure 8. All cutoffs are derived using the OOB predictions of our random forest model.[32]

Figure 7:



[32] Refer to our discussion of ROC curves in the previous section for details on how to compute these cutoffs.

Figure 8:

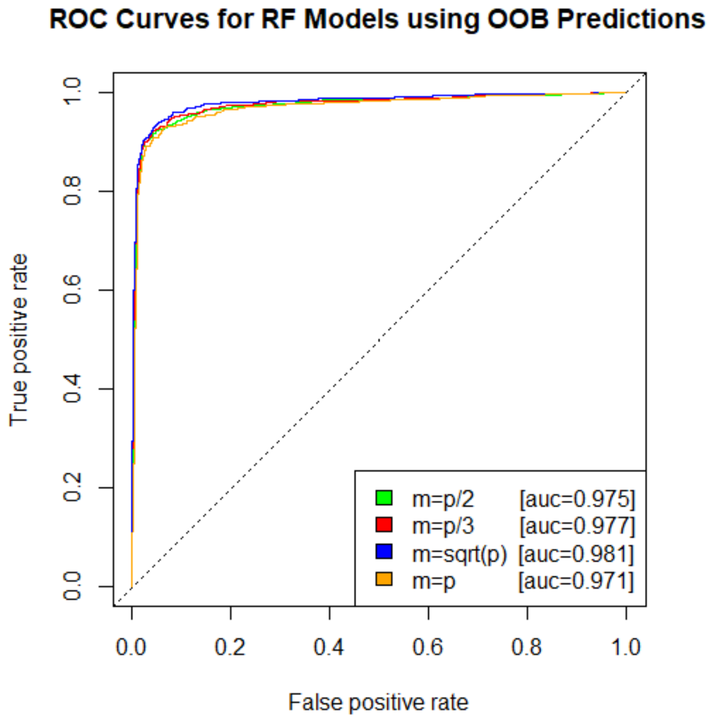


Table 4 gives the different probability cutoffs that we consider for our model, with the corresponding OOB error rates used to derive the cutoff values. These results are based on the spam data, but we discuss these results as if they were the result of our postdoc prediction to keep the focus on the decision-making context relevant to our actual model selection task. As we can see, the cutoff that minimizes the OOB error is close to the default 0.5 threshold, and the top-left and Youden cutoff are close in value. As is typically the case, moving from the default cutoff to the Youden or top-left cutoff results in a drop in total accuracy, but a more even distribution of errors in terms of false negatives and false positives. Altering the probability cutoff to the Youden or top-left cutoff helps counter the problem of class imbalance that typically results in prediction models that favor predicting the most commonly occurring class. If the ultimate objective is to accurately predict as many true postdocs as possible, then the top-left cutoff appears to be the best choice as it achieves the highest TPR of any cutoff considered in Table 4. A TPR of 94.09% means that 94.09% of all true postdocs will be predicted as being postdocs by our model. However, the increase in TPR comes at the expense of lowering the purity of our predicted postdoc sample; while we increase the percentage of true postdocs that we predict as postdocs, we also increase the percentage of nonpostdocs that we incorrectly predict as being postdocs. This shows up in a decrease in the PPV of the random forest model when moving from the default 0.5 cutoff to the top-left cutoff. While achieving the best TPR of all the cutoffs, we can see that the top-left cutoff also achieves the worst PPV of any of these cutoffs. A PPV of 91.97% means that, of all those observations that we predict as being postdocs, only 91.97% are truly postdocs.[33] Without information on the relative costs of false positive and false negative errors, it is somewhat a matter of researcher preference as to which cutoff is best. Ultimately, we favor using the Youden cutoff. This is because we are interested in both accurately predicting true postdocs (high TPR) and ensuring that our predicted postdocs are indeed postdocs (high PPV); the Youden cutoff strikes the most equitable balance between these two measures. Additionally, the Youden cutoff represents the cutoff at which our random forest is most skilled relative to random guessing, as discussed in the previous section, and this adds some intuitive appeal. Thus, we select the random forest model with $\sqrt{p}$ predictors and a probability cutoff of 0.43 as our final prediction model.

Table 4: Random Forest Accuracy by Cutoff using Training Set OOB Predictions

Cutoff				Accuracy				
Default	**Min Error**	**Top-left**	**Youden**	**Total**	**TPR**	**TNR**	**PPV**	**NPV**
0.50	…	…	…	94.43%	91.02%	96.68%	94.75%	94.28%
…	0.56	…	…	94.87%	90.36%	97.84%	96.49%	93.91%
…	…	0.39	…	94.39%	94.09%	94.59%	91.97%	96.05%
…	…	…	0.43	94.70%	92.99%	95.82%	93.61%	95.41%

It is important to note that, while useful for deciding on which probability threshold to select, the accuracy measures given in Table 4 are an optimistically biased estimate of the generalization accuracy of each model since selection of the random forest model with $\sqrt{p}$ splitting variables was chosen based on its performance on the OOB observations. Therefore, having finished our model selection, we consider how well our selected model predicts on the test set observations that were not used during any of the model selection steps to get unbiased measures of the generalization accuracy of our selected model. Table 5 displays our calculated accuracy measures for these test set predictions when our method is applied to the spam dataset. Reassuringly, our model performs strongly on data it had never "seen" at any point in the model selection process.

Table 5: Random Forest Accuracy using Test Set Predictions

Accuracy				
Total	**TPR**	**TNR**	**PPV**	**NPV**
94.61%	92.89%	95.72%	93.30%	95.44%

[33] While these measures are close in value for this application to spam data, one could hypothetically obtain a predictive model with a high PPV and low TPR—for example, imagine a sample with 100 postdocs and 900 nonpostdocs. If only one observation were predicted to be a postdoc, and if this prediction was correct, the model would have a 100% PPV and a 1% TPR. Likewise, one can (easily) obtain a high TPR and a low PPV by simply classifying all observations as postdocs—in our example, this would lead to a 100% TPR and 10% PPV. Therefore, it is important to consider both measures when choosing among different cutoffs, rather than one in isolation.

The Table 5 measures of accuracy can likely be viewed as conservative estimates of the generalization accuracy since they are based on a model trained on only 50% of the available UMETRICS-ACS-LEHD data. Since it is likely that training on more data will add to the predictive power of the model, these estimates likely exhibit a pessimistic bias. When making our predictions for the non-UMETRICS subset of our ACS-LEHD doctorates, we train our tuned random forest model on all the UMETRICS data, so we would like an estimate of the generalization error of this model that uses 100% of the available data. Unfortunately, we are not aware of a method to measure this error in an unbiased and computationally-feasible way. Therefore, we measure this error using the OOB error of the random forest model trained on all the UMETRICS data, noting that this measure of error may be optimistically biased; our results as applied to the spam dataset are in Table 6. In an informal sense, we can view the accuracy measures in Table 5 and Table 6 as a lower bound estimate and upper bound estimate of the generalization error, respectively.

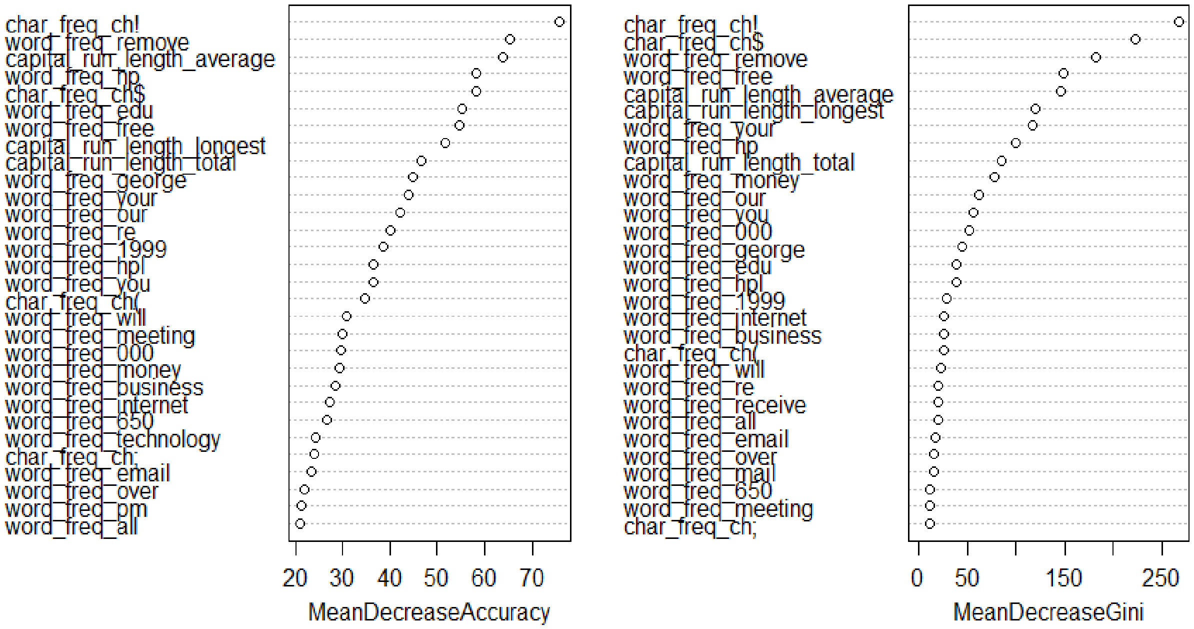
Table 6: Random Forest Accuracy using OOB Predictions from Random Forest Trained on Full Data

Accuracy				
Total	**TPR**	**TNR**	**PPV**	**NPV**
95.41%	94.54%	95.98%	93.87%	96.43%

The random forest algorithm has two built-in methods for evaluating the importance of different predictors. The importance measures used by the random forest algorithm are descriptive in nature. The first importance measure is referred to in the randomForest package as "mean decrease accuracy" and is described in Breiman (2001). The method works as such: first, the OOB accuracy for each tree is recorded.[34] Then, the OOB accuracy for each tree is calculated after randomly permuting the value of each predictor, one predictor at a time; by randomly shuffling a predictor's values in this way, any link between the predictor and postdoc status is effectively broken, and so the OOB accuracy should decrease in proportion to the importance of the variable in prediction. For each predictor, the decrease in OOB accuracy is averaged over all trees and normalized by the standard deviation of these differences. The second measure of predictor importance, called "mean decrease Gini", reports the decrease in the Gini index, a measure of node impurity, from splitting on each predictor, averaged over all trees in the random forest. Figure 9 gives our results using these measures of importance on the spam data from the UCI Machine Learning Repository, as our true results are pending Census disclosure.

[34] OOB accuracy = 100% - OOB Error Rate

Figure 9: Random forest importance measures of predictors



6. Comparison of random forest model with other predictive models

Since no single machine learning algorithm dominates all others across all applications (James, Witten, Hastie, and Tibshirani, 2013), it is useful to compare the performance of our random forest model with other models. If a different model does better than random forests, then we may as well adopt that model for prediction. One popular alternative to random forests is known as boosted trees. Boosting, like bagging, is an ensemble method based on averaging predictions across many simple learners such as classification trees.[35] However, these two approaches differ in several aspects. First, with bagged trees, each tree is grown to be large, while in boosting, typically trees with only a few splits each are grown. A second and more significant difference is that with bagging, each tree is grown independent of the other trees in the ensemble, whereas with boosting trees are grown sequentially, with each tree's structure depending on the structure of the trees before it. Specifically, each successive tree in a boosted tree model places more weight on correctly predicting observations for which previous trees in the ensemble performed poorly. The predictions of the model are updated as each tree is grown, with more weight being applied to trees that achieve greater accuracy. The rate at which this updating occurs is controlled by a shrinkage parameter. Altogether, boosted trees contain three hyperparameters that the user must tune: the number of trees, the size of each tree (interaction depth), and the rate of learning (shrinkage) across trees. Typically, the choice of a smaller shrinkage parameter will necessitate growing a larger number of trees, and in practice, Hastie, Tibshirani, and Friedman (2009) suggest choosing the size of the trees to be such that the number

[35] Our description of boosting is based on the Adaboost algorithm developed in Freund and Schapire (1997).

of terminal nodes is around 6, finding that variation in the size of the trees seldom provides significant improvement.[36] Gradient boosted machines, which are a generalization of boosted trees introduced in Friedman (2001), are implemented in the R package gbm (Ridgeway, 2007).

Table 7: Machine Learning Model Selection (Steps 1-4), Assessment (Step 5), and Prediction (Step 6)

1. Partition data into a training set, validation set, and test set (50%-25%-25% split).[37]
2. For each machine learning algorithm:
   i. Train model on the training set using different values of hyperparameters.
   ii. Use repeated CV to estimate AUC for different hyperparameter combinations.
   iii. Output model that performs the best in terms of AUC as measured by repeated CV.
3. Pick machine learning algorithm whose tuned model gives the best AUC on validation set.
4. Identify alternative cutoffs/thresholds based on validation set.
5. Estimate generalization error using the test set.
6. Retrain selected model on all labeled /UMETRICS data and use to predict postdoc status for all non-labeled/non-UMETRICS observations.

To compare different types of machine learning models, we adopt the method of model selection and assessment outlined in Table 7. While similar to our strategy in Table 3, there are two main differences. First, we partition the UMETRICS subsample into three sets (a training set, validation set, and test set) rather than two sets (a training set and test set). The training set is used to train and tune each individual machine learning model, the validation set is used to compare between different machine learning models and identify alternative cutoffs, and the test set is used to estimate the generalization error of our selected model as before. The second difference is that we use repeated *K*-fold cross-validation (CV) to tune our different machine learning models.[38] *K*-fold CV works as follows: 1) the training set is partitioned into *K* folds, 2) For each fold *k:* the model is trained on all folds except fold *k*, and then predictions are made for fold *k* and AUC is calculated, 3) the *K* AUC calculations are averaged to obtain a single CV

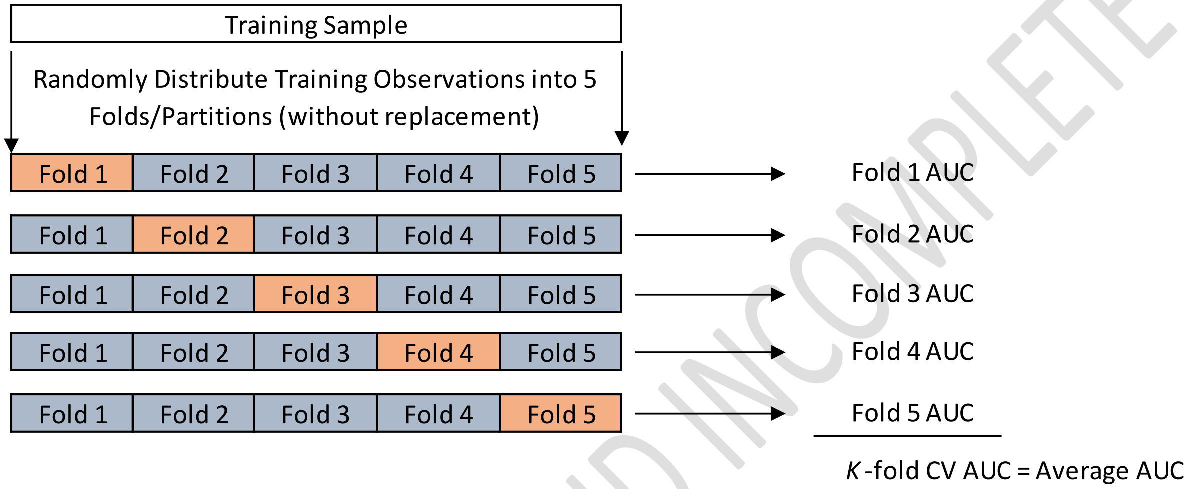
[36] A tree with 6 terminal nodes allows for fifth-order interactions between the predictors.

[37] Hastie, Tibshirani, and Friedman (2009) state that it is too difficult to give a general rule for how to partition the data, but that a typical split might be 50%-25%-25%. The validity of our method does not hinge on the choice of data partitions.

[38] While we could again use AUC calculated from ROC curves based on OOB observations instead of *K*-fold CV to tune our random forest model, we choose the latter so as to follow the general strategy put forth in Table 7 which can be applied for any machine learning model, including those not considered here, e.g. support vector machines, neural networks, etc.

estimate of AUC. The calculated *K*-fold AUC depends on the partitioning of the original data, and so one way to reduce this source of variance is to repeat *K*-fold CV multiple times.[39] We give a schematic representation of *K*-fold CV in Figure 10.

Figure 10: *K*-fold cross-validation (CV) method for calculating AUC with *K* = 5


Training Sample
Randomly Distribute Training Observations into 5 Folds/Partitions (without replacement)
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Fold 1 AUC
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Fold 2 AUC
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Fold 3 AUC
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Fold 4 AUC
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Fold 5 AUC
K-fold CV AUC = Average AUC

Notes: "Fold *k* AUC" is the AUC calculated by first training a machine learning model on all observations not in the *k* th fold (i.e. the blue folds), and then using this model to predict the classes of fold *k* observations (the beige fold); such predictions can be used to graph an ROC curve and calculate the area under the curve.

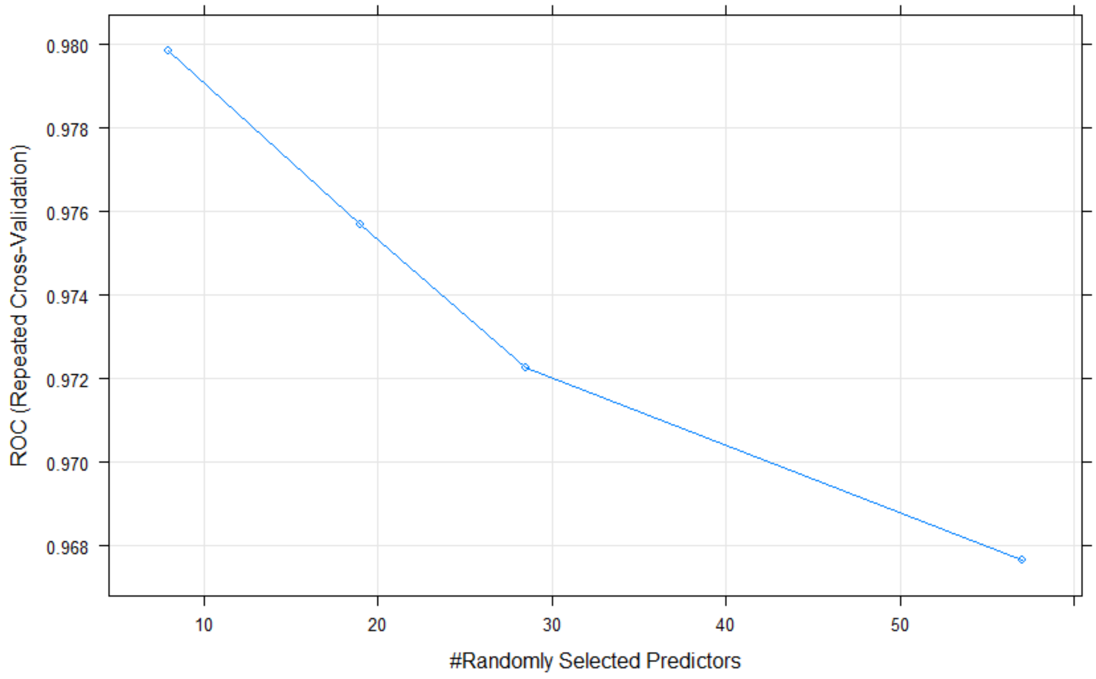
We compare two machine learning algorithms, random forests and boosted trees, using the method described in Table 7. We also compare the performance of these two algorithms with a linear probability model and logit model where each predictor enters the model additively with no interaction terms. As we will see, both random forests and boosted trees achieve greater accuracy in prediction compared to a linear probability model and logit model. Additionally, we find that random forests and boosted trees are comparable in performance. Due to the ease and speed of tuning random forests relative to boosted trees, we prefer a random forests approach to prediction.

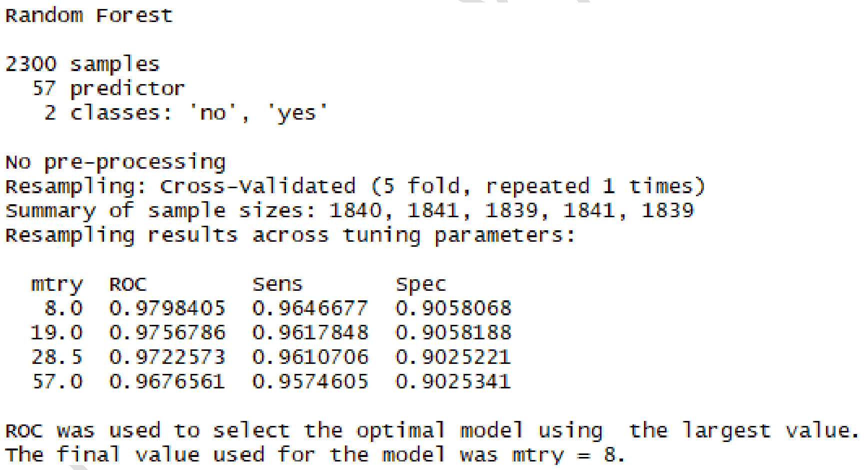
We tune our random forest model over the same values of the splitting variables that we considered in the previous section. For boosted trees, we tune over combinations of the following parameter values: Number of trees = {5000, 8000, 110000}, shrinkage rate = {0.001, 0.01, 0.1}, and interaction depth = {1, 2, 3}. Our tuning results for random forests and boosted trees are found in Figures 11 and 12, respectively, which are based on the spam data. As we can see, the random forest model with $\sqrt{p}$ splitting variables performs best amongst random forests models, and the boosted trees model with {number of trees, shrinkage rate, interaction depth} = {5000,

[39] Kim (2009) compares repeated 10-fold CV to other methods of comparative computational requirements and recommends repeated CV for general use. We use the R package caret (Kuhn, 2008) to perform repeated CV. We only use 5-fold CV for spam data for computational convenience, but use 10-fold CV repeated 5 times when tuning models used for postdoc prediction.

0.01, 3} performs best amongst the boosted trees models considered, and so these two models represent our tuned random forest model and boosted trees model, respectively.

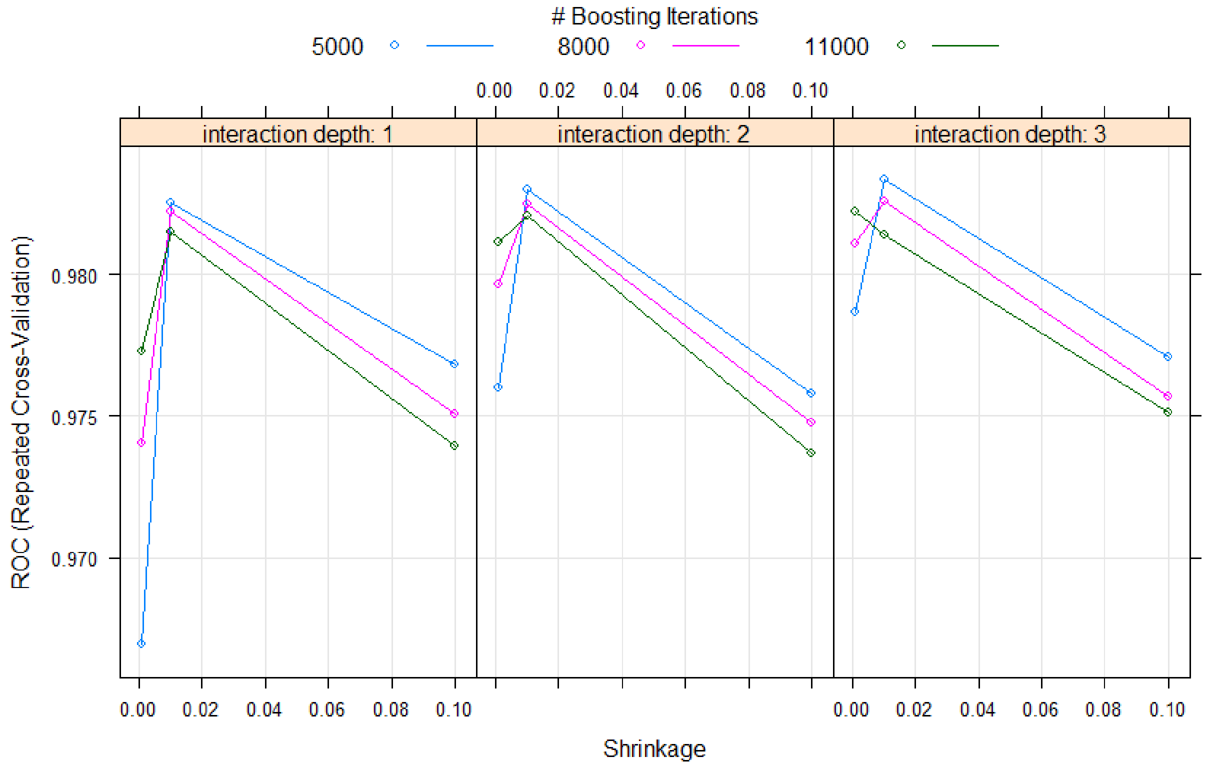
Figure 11: 5-fold CV Estimate of AUC for Random Forest Models

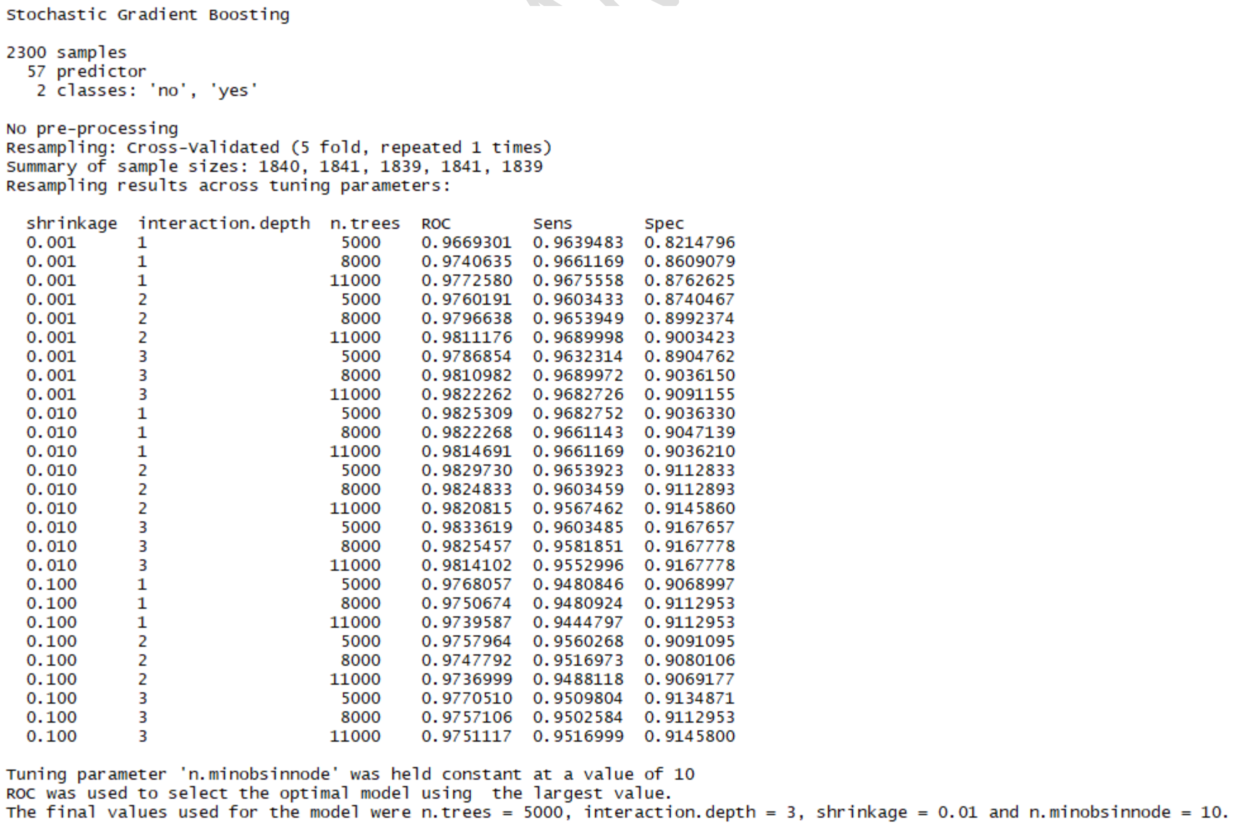




```
Random Forest

2300 samples
  57 predictor
   2 classes: 'no', 'yes'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 1 times)
Summary of sample sizes: 1840, 1841, 1839, 1841, 1839
Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec
   8.0  0.9798405  0.9646677  0.9058068
  19.0  0.9756786  0.9617848  0.9058188
  28.5  0.9722573  0.9610706  0.9025221
  57.0  0.9676561  0.9574605  0.9025341

ROC was used to select the optimal model using  the largest value.
The final value used for the model was mtry = 8.
```

Figure 12: 5-fold CV Estimate of AUC for Boosted Trees Models

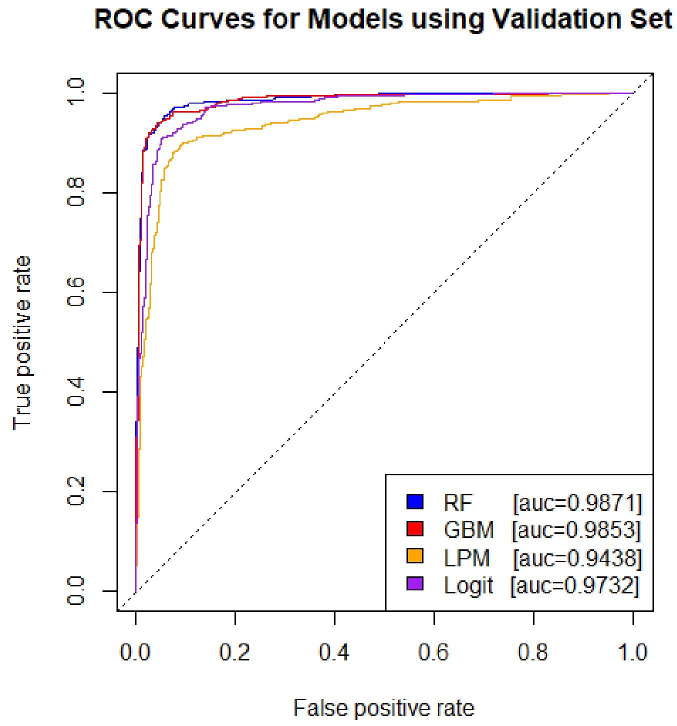




```
Stochastic Gradient Boosting

2300 samples
  57 predictor
   2 classes: 'no', 'yes'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 1 times)
Summary of sample sizes: 1840, 1841, 1839, 1841, 1839
Resampling results across tuning parameters:

  shrinkage  interaction.depth  n.trees  ROC        Sens       Spec
  0.001      1                   5000    0.9669301  0.9639483  0.8214796
  0.001      1                   8000    0.9740635  0.9661169  0.8609079
  0.001      1                  11000    0.9772580  0.9675558  0.8762625
  0.001      2                   5000    0.9760191  0.9603433  0.8740467
  0.001      2                   8000    0.9796638  0.9653949  0.8992374
  0.001      2                  11000    0.9811176  0.9689998  0.9003423
  0.001      3                   5000    0.9786854  0.9632314  0.8904762
  0.001      3                   8000    0.9810982  0.9689972  0.9036150
  0.001      3                  11000    0.9822262  0.9682726  0.9091155
  0.010      1                   5000    0.9825309  0.9682752  0.9036330
  0.010      1                   8000    0.9822268  0.9661143  0.9047139
  0.010      1                  11000    0.9814691  0.9661169  0.9036210
  0.010      2                   5000    0.9829730  0.9653923  0.9112833
  0.010      2                   8000    0.9824833  0.9603459  0.9112893
  0.010      2                  11000    0.9820815  0.9567462  0.9145860
  0.010      3                   5000    0.9833619  0.9603485  0.9167657
  0.010      3                   8000    0.9825457  0.9581851  0.9167778
  0.010      3                  11000    0.9814102  0.9552996  0.9167778
  0.100      1                   5000    0.9768057  0.9480846  0.9068997
  0.100      1                   8000    0.9750674  0.9480924  0.9112953
  0.100      1                  11000    0.9739587  0.9444797  0.9112953
  0.100      2                   5000    0.9757964  0.9560268  0.9091095
  0.100      2                   8000    0.9747792  0.9516973  0.9080106
  0.100      2                  11000    0.9736999  0.9488118  0.9069177
  0.100      3                   5000    0.9770510  0.9509804  0.9134871
  0.100      3                   8000    0.9757106  0.9502584  0.9112953
  0.100      3                  11000    0.9751117  0.9516999  0.9145800

Tuning parameter 'n.minobsinnode' was held constant at a value of 10
ROC was used to select the optimal model using  the largest value.
The final values used for the model were n.trees = 5000, interaction.depth = 3, shrinkage = 0.01 and n.minobsinnode = 10.
```

Next, we compare our random forest model and boosted tree model directly by calculating the AUC of each method when applied to the validation set. As we can see in Figure 13, these two methods achieve comparable performance on the spam data, and each method performs better than a linear probability model and logit model, where our LPM and logit specifications contain all predictors but no interaction terms.

Figure 13:



On this basis, we select the random forest model since it achieves the highest AUC based on the validation set. From here, we can use the validation set to identify alternative probability thresholds to mitigate for class imbalance. While we believe picking the predictive model that achieves the highest AUC is a prudent approach, especially when considering many types of models, we show the error rates for alternative cutoffs for each of the four types of models considered in this section in Table 8. As we see in Table 8, boosted trees and random forests outperform both our LPM and logit model, likely due to the fact that these models can automatically capture complex interaction effects in the data without these interactions needing to be specified by the researcher *ex ante*. We also see that random forests and boosted trees perform quite similarly. Since boosted trees are more difficult and computationally intensive to tune, we favor using the random forest model over boosted trees when results between the two methods are similar. Again, we choose the random forest model with Youden cutoff as our final prediction model.

Table 8: Error Rates by Cutoff using Validation Set Data

	Cutoff				Accuracy				
Model	**Default**	**Min Error**	**Top left**	**Youden**	**Total**	**TPR**	**TNR**	**PPV**	**NPV**
LPM	0.50	…	…	…	88.43%	78.04%	95.36%	91.82%	86.69%
LPM	…	0.42	…	…	90.87%	88.04%	92.75%	89.01%	92.09%
LPM	…	…	0.38	…	90.43%	90.00%	90.73%	86.61%	93.15%
LPM	…	…	…	0.42	90.87%	88.04%	92.76%	89.01%	92.09%
Logit	0.50	…	…	…	92.87%	88.70%	95.65%	93.15%	92.70%
Logit	…	0.44	…	…	93.30%	90.87%	94.93%	92.27%	93.97%
Logit	…	…	0.43	…	93.22%	91.09%	94.64%	91.89%	94.09%
Logit	…	…	…	0.44	93.30%	90.87%	94.93%	92.27%	93.97%
GBM	0.50	…	…	…	95.22%	92.83%	96.81%	95.10%	95.29%
GBM	…	0.52	…	…	95.30%	92.61%	97.10%	95.52%	95.17%
GBM	…	…	0.38	…	95.04%	94.13%	95.65%	93.52%	96.07%
GBM	…	…	…	0.40	95.13%	93.91%	95.94%	93.91%	95.94%
RF	0.50	…	…	…	94.96%	91.96%	96.96%	95.27%	94.76%
RF	…	0.55	…	…	95.30%	91.74%	97.68%	96.35%	94.66%
RF	…	…	0.34	…	94.87%	95.43%	94.49%	92.03%	96.88%
RF	…	…	…	0.34	94.87%	95.43%	94.49%	92.03%	96.88%

Similar to before, the accuracy measures given in Table 8 are an optimistically biased estimate of the generalization accuracy of each model since selection of the random forest model was chosen based on its validation set performance. To get a measure of the generalization accuracy, we use our random forest model with a cutoff of 0.34 to predict the spam status of our test set observations, and then calculate our accuracy measures for these test set predictions. Table 9 displays our results.

Table 9: Random Forest Accuracy using Test Set Predictions

Accuracy				
Total	**TPR**	**TNR**	**PPV**	**NPV**
93.74%	94.55%	93.25%	89.66%	96.51%

As before, we view Table 9 measures of accuracy as conservative estimates of the generalization accuracy since they are based on a model trained on 50% of the available data. Therefore, we measure this error using the OOB error of the random forest model trained on all the UMETRICS data, noting that this measure of error is likely optimistically biased. We informally view our accuracy measures in Table 9 and Table 10 as a lower bound estimate and upper bound estimate of the generalization error, respectively.

Table 10: Random Forest Accuracy using OOB Predictions from Random Forest Trained on Full Data

Accuracy				
Total	**TPR**	**TNR**	**PPV**	**NPV**
95.02%	96.47%	94.08%	91.38%	97.62%

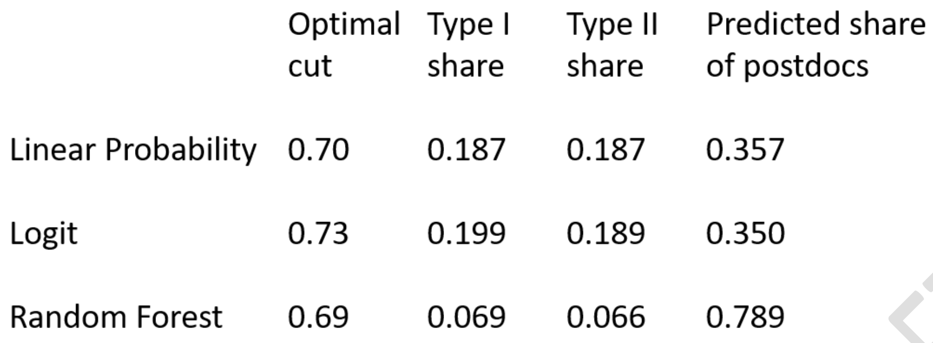
The results in Tables 9 and 10 are similar to those in Tables 5 and 6, respectively, but there are some differences emanating from the fact that the Youden cutoff obtained when using the OOB observations (0.43) is higher than the Youden cutoff obtained when using the validation set observations (0.34), leading now to a higher TPR and lower PPV relative to the final model obtained in the last section. Another difference is that the test dataset used to produce Table 9 results was one-half the size of the test set used for Table 5 because, unlike before, we had to reserve some data to form a validation set.

Overall, random forests compare favorably to the other prediction models considered in this section. Random forests performed better than the LPM and logit model and about the same as boosted trees. The advantage of random forests over boosted trees is that they are considerably easier and less computationally intensive to tune. We may be able to obtain a boosted tree model that would marginally outperform random forests if we tune over more tuning parameter values, but given the high performance of random forests already, the expected return to doing so is small.

While we are unable to report our exact postdoc prediction results as they are pending Census disclosure, we discuss our general findings here: first, as demonstrated using the spam data, random forests and boosted trees perform similarly, and both these models significantly improve over the LPM and logit model. Overall, the random forest algorithm predicts postdoc status extraordinarily well in terms of accuracy, true positive rate, true negative rate, and area under the ROC curve, with age and earnings being two of the most important predictors of postdoc status.

In the past, we utilized a simpler approach to predicting postdoc status using random forests, and had these results disclosed. Prediction is based on a reduced set of variables including age, sex, quarterly earnings, and whether the individual is employed in a university. The results from this random forest model are reported in Table 11. Despite predictions being based on a small number of predictors, random forests performed reasonably well in terms of FPR (“Type I”) and FNR (“Type II”). Our improved method and set of predictors builds on these results.

Table 11: Postdoc Prediction Using a Reduced Set of Predictors



	Optimal cut	Type I share	Type II share	Predicted share of postdocs
Linear Probability	0.70	0.187	0.187	0.357
Logit	0.73	0.199	0.189	0.350
Random Forest	0.69	0.069	0.066	0.789

7. Conclusion

In this paper, we detailed our construction of a linked employer-employee longitudinal dataset of the doctoral workforce that enables researchers to analyze the labor market outcomes of STEM PhD graduates and postdocs. This dataset contains demographic information such as age, race, and sex for each individual from the annual ACS files, as well as key quarterly economic information from the LEHD about where each individual works, how much they earn, and how their careers develop over time. By matching a new university-based administrative dataset, UMETRICS, to our ACS-LEHD Doctorate Panel, we were able to develop a machine learning procedure to predict, at a high degree of accuracy, the postdoc status of individuals for whom true postdoc status is unknown. Since the actual results of our postdoc prediction exercise are pending Census disclosure, we utilized a publicly available dataset of spam emails to demonstrate our methods. In doing so, we rigorously compared the prediction performance of our preferred model, random forests, to other predictive models including a linear probability model (LPM), logit, and boosted trees, another popular machine learning model, and found that random forests outperformed the standard approaches and achieved performance comparable to boosted trees, with the advantage of being more computationally efficient. Our methods are sufficiently general to be applied in other research contexts, and we view our method as a way to reliably augment the research capabilities of existing big datasets cheaply.

The next step we would like to accomplish is to validate our postdoc prediction algorithm by comparing the characteristics (e.g. age, sex, race, foreign-born status, salary, etc.) of our predicted postdocs to the characteristics of postdocs found in the Survey of Earned Doctorates (SED), Survey of Doctorate Recipients (SDR), and the Survey of Graduate Students and Postdoctorates in Science and Engineering (GSS). The SED contains a census of individuals obtaining PhDs through a US institution and contains information on whether a new doctorate is pursuing a postdoc. The GSS is a census that covers all US academic institutions granting doctorates in STEM fields, from which we can obtain the number of postdocs in each university by different demographic characteristics. An important distinction is that individuals who obtained a PhD outside the US and then took a postdoc position at a US academic institution will be counted in the GSS, but will not be in the SED. Postdocs in UMETRICS data may have obtained their PhD in the U.S. or abroad, so it is important to compare their characteristics with postdocs found both in SED and GSS. The SDR is longitudinal and contains individual-level information on STEM PhDs including salary for job in year of survey and, starting in 2010, salary for the first job after obtaining a PhD. Thus, we can compare the earning trajectory of our predicted postdocs with those in the SDR. If characteristics of our predicted postdocs match those in the SED, SDR, and GSS reasonably well, we can be even more confident in our methods used for postdoc prediction.

We would also like to apply our method to predicting spells in doctoral programs for individuals in the ACS-LEHD Doctorate Panel. A future goal is to have Proquest linked to the LEHD, which would enable us to identify individuals in the LEHD whom have doctorates and when they graduate and from what institution. This would also be useful for our postdoc prediction exercise as date of graduation and university attended could be a useful predictor.

Lastly, once all data construction is complete, we will begin using the dataset to study a wide-range of pertinent topics including: (1) the returns to education for STEM PhDs and postdocs, and how these differ by demographics, (2) the determinants of STEM labor demand, including an assessment of the complementarity between STEM workers and firm R&D activity, and (3) how the labor mobility of STEM doctorates impacts R&D spillovers, and how the earnings of STEM doctorates depend on measures of their past R&D exposure.

References

Acemoglu, D. (1998). Why Do New technologies Complement Skills? Directed Technical Change and Wage Inequality. *The Quarterly Journal of Economics*, *113*(4), 1055-1089.

Agrawal, A., & Henderson, R. (2002). Putting Patents in Context: Exploring Knowledge Transfer from MIT. *Management Science*, *48*(1), 44-60.

Ambroise, C., & McLachlan, G. J. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences*, 6562-6566.

Athey, S., & Imbens, G. (2017). The State of Applied Econometrics: Causality and Policy Evaluation. *Journal of Economic Perspectives*, 3-32.

Audretsch, D. B., & Feldman, M. P. (1996). R&D Spillovers and the Geography of Innovation and Production. *The American Economic Review*, *86*(3), 630-640.

Berrar, D. P., Sturgeon, B., Bradbury, I., & Dubitzky, W. (2003). Microarray Data Integration and Machine Learning Techniques for Lung Cancer Survival Prediction. *Proceedings of the CAMDA-2003*.

Bosch, A., Zisserman, A., & Muñoz, X. (2007). Image Classifcation using random Forests and Ferns. *Computer Vision*.

Breiman, L. (1996). Bagging Predictors. *Machine Learning*, 123-140.

Breiman, L. (2001). Random Forests. *Machine Learning*, 5-32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees.* Chapman & Hall.

Bresnahan, T. F., Brynjolfsson, E., & Hitt, L. M. (2002). Information Technology, Workplace Organization, and the Demand for Skilled Labor: Firm-Level Evidence. *The Quarterly Journal of Economics, 117*(1), 339-376.

Buffington, C., Cerf, B., Jones, C., & Weinberg, B. A. (2016). STEM Training and early career Outcomes of Female and Male Graduate Students: Evidence from UMETRICS Data Linked to the 2010 Census. *American Economic Review: Papers & Proceedings*, 333-338.

Cawley, G. C., & Talbot, N. L. (2010). On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *Journal of Machine Learning Research*, 2079-2107.

Chalfin, A., Danieli, O., Hillis, A., Jelveh, Z., Luca, M., Ludwig, J., & Mullainathan, S. (2016). Productivity and Selection of Human Capital with Machine Learning. *American Economic Review: papers and Proceedings*, 124-127.

Chang, W.-Y., Emad, A., Lane, J., Tokle, J., & Weinberg, B. (2016). Linking Survey and Transaction Data: New Techniques. (unpublished paper).

Cohen, W. M., Nelson, R. R., & Walsh, J. P. (n.d.). *Protecting Their Intellectual Assets: Appropriability Conditions and Why U.S. Manufacturing Firms Patent (or Not).* NBER Working Paper No. 7552.

Díaz-Uriarte, R., & Alvarez de Andrés, S. (2006). Gene selection and classifcation of microarray data using random forest. *BMC Bioinformatics*.

Einav, L., & Levin, J. (2014). The Data Revolution and Economic Analysis. In J. Lerner, & S. Stern (Eds.), *Innovation Policy and the Economy, Volume 14* (pp. 1-24). University of Chicago Press.

Feigenbaum, J. J. (2016). A Machine Learning Approach to Census Record Linking. Retrieved from https://jamesfeigenbaum.github.io/research/pdf/census-link-ml.pdf

Feldman, M. P. (1994). Knowledge Complementarity and Innovation. *Small Business Economics, 6*(5), 363-372.

Fox, M. F., & Stephan, P. E. (2001). Career of Young Scientists: Preferences, prospects, and Realities by Gender and Field. *Social Studies of Science, 31*(1), 109-122.

Freund , Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and Systsem Sciences*, 119-139.

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, 1189-1232.

Ginther, D. K., & Kahn, S. (2009). Does Science Promote Women? Evidence from Academia 1973-2001. In R. B. Freeman, & D. L. Goroff (Eds.), *Science and Engineering Careers in the United States: An Analysis of Markets and Unemployment* (pp. 163-194). Chicago, IL: University of Chicago press.

Gislason, P. O., Benediktsson, J. A., & Sveinsson, J. R. (2006). Random Forests for land cover classification. *Pattern Recognition Letters*, 294-300.

Goldin, C., & Katz, L. F. (1998). The Origins of Technology-Skill Complementarity. *The Quarterly Journal of Economics, 113*(3), 693-732.

Goldschlag, N., Jarmin, R., Lane, J., & Zolas, N. (2017). "The Link between University R&D, Human Capital and Business Startups". In *Measuring and Accounting for Innovation in the 21st Century. (forthcoming).*

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition.* New York: Springer.

Henderson, R., Jaffe, A. B., & Trajtenberg, M. (1998). Universites as a Source of Commercial Technology: A Detailed Analysis of University Patenting, 1965-1988. *The Review of Economics and Statistics, 80*(1), 119-127.

Jaffe, A. B. (1986). Technological Opportunities and Spillovers of R&D: Evidence from Firms' Patents, Profits, and Market Value. *The American Economic Review*, 984-1001.

Jaffe, A. B. (1989). Real Effects of Academic Research. *The American Economic Review*, 957-970.

Jaffe, A. B., Trajtenberg, M., & Hendersen, R. (1993). Geographic Localization of Knowledge Spillovers as Evidenced by Patent Citations. *The Quarterly Journal of Economics, 108*(3), 577-598.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R.* New York: Springer.

Jensen, R., & Thursby, M. (2001). Proofs and Prototypes for Sale: The Licensing of University Inventions. *The American Economic Review, 91*(1), 240-259.

Kaiser, D. (2005). *Drawing Theories Apart: The Dispersion of Feynman Diagrams in Postwar Physics.* Chicago, IL: University of Chicago Press.

Kim, J.-H. (2009). Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics and Data Analysis*, 3735-3745.

Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J., & Mullainathan, S. (2017). Human Decisions and Machine Predictions. *The Quarterly Journal of Economics*.

Kleinberg, J., Ludwig, J., Mullainathan, S., & Obermeyer, Z. (2015). Prediction Policy Problems. *American Economic Review: Papers & proceedings*, 491-495.

Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. *Journal of Statistical Software, 28*(5).

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling.* New York: Springer.

Lahiri, K., & Yang, L. (2013). Forecasting Binary Outcomes. In G. Elliott, & A. Timmerman (Eds.), *Handbook of Economic Forecasting, Volume 2B* (pp. 1025-1106). Elsevier.

Lane, J. I., Owen-Smith, J., Rosen, R. F., & Weinberg, B. A. (2015). New linked data on research investments: Scientific workforce, productivity, and public value. *Research Policy*, 1659-1671.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News, 2*(3), 18-22. Retrieved from http://CRAN.R-project.org/doc/Rnews/.

Mowery, D. C., & Ziedonis, A. A. (2001). *The Geographic Reach of Market and Non-Market Channels of Technology Transfer: Comparing Citations and Licenses of University Patents.* NBER Working Paper No. 8568.

Mullainathan, S., & Obermeyer, Z. (2017). Does Machine Learning Automate Moral Hazard and Error? *American Economic Review: Papers & Proceedings*, 476-480.

Mullainathan, S., & Spiess, J. (2017). Machine Learning: An Applied Econometric Approach. *Journal of Economic Perspectives*, 87-106.

Mulrow, E., Mushtaq, A., Pramanik, S., & Fontes, A. (2011). *Assessment of the U.S. Census Bureau's Person identification Validation System.* Chicago: NORC at the University of Chicago.

Polanyi, M. (1958). *Personal Knowledge: Towards a Post-Critical Philosophy*. Chicago, IL: University of Chicago Press.

Polanyi, M. (1966). *The Tacit Dimension.* Garden City, NY: Doubleday.

Ridgeway, G. (2007). Generalized Boosted Models: A Guide to the gbm Package. Retrieved from http://cran.r-project.org/web/packages/gbm/vignettes/gbm.pdf

Stekhoven, D. J., & Bühlmann, P. (2012). MissForest – non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 112-118.

Stephan, P. (2006). Wrapping It Up in a Person: The mobility Patterns of New PhDs. *Innovation Policy and the Economy*, *7*, 71-98.

The Insititute for Research on Innovation & Science. (2017). *Summary Documentation for UMETRICS 2016Q3a Dataset.*

Thursby, J. G., & Thursby, M. C. (2002). Who is Selling the Ivory Tower? Sources of Growth in University Licensing. *Management Science, 48*(1), 90-104.

Tibshirani, R. J., & Tibshirani, R. (2009). A Bias Correction for the Minimum Error Rate in Cross-Validation. *The Annals of Applied Statistics*, 822-829.

Tibshirani, R., Hastie, T., Narasimhan, B., & Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 6567-6572.

U.S. Department of Commerce. (2013). *American Community Survey: Information Guide.*

Varian, H. (2014). Big Data: New Tricks for Econometrics. *The Journal of Economic Perspectives*, 3-27.

Varma, S., & Simon, R. (2006). Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*.

Vilhuber, L., & McKinney, K. (2014). *LEHD Infrastructure Files in the Census RDC - Overview*.

Youden, W. J. (1950). Index for Rating Diagnostic Tests. *Cancer*, 32-35.

Zolas, N., Goldschlag, N., Jarmin, R., Stephan, P., Owen-Smith, J., Rosen, R. F., . . . Lane, J. I. (2015). Wrapping it up in a person: Examining employment and earnings outcomes for Ph.D. recipients. *Science*, 1367-1371.

Zucker, L. G., & Darby, M. R. (2001). Capturing technological Opportunity via Japan's Star Scientists: Evidence from Japanese Firms' Biotech Patents and Products. *Journal of Technology Transfer, 26*(1-2), 37-58.

Zucker, L. G., Darby, M. R., & Armstrong, J. (1998). Geographically Localized Knowledge: Spillovers or Markets? *Economic Inquiry, 36*(1), 65-86.

Zucker, L. G., Darby, M. R., & Armstrong, J. S. (2002). Commercializing Knowledge: University Science, Knowledge Capture, and Firm Performance. *Management Science, 48*(1), 138-153.

Zucker, L. G., Darby, M. R., & Brewer, M. B. (1998). Intellectual Human Capital and the Birth of U.S. Biotechnology Enterprises. *The American Economic Review, 88*(1), 290-306.

Data Appendix: Merging UMETRICS to the ACS-LEHD Doctorate Panel

Here, we outline the steps shown in Figure 3 used to merge the UMETRICS Employee Transactions File (ETF) to our ACSD-LEHD Doctorate panel.

The first step in preparing UMETRICS ETF data for merging into our ACS-LEHD data is to obtain the PIKs for each ETF observation via a merge with the UMETRICS employeeid-PIK Crosswalk available in the Census FSRDC. After the merge, we keep only observations for which there is a PIK available (i.e. “PIK’d” observations). We subsequently reformat the resulting ETF file into a quarterly dataset where the unit of observation is PIK-institutionid-year quarter since an individual may be employed at multiple IRIS member universities within the same quarter. We also code a postdoc indicator variable for each observation in this dataset which equals 1 if a person is ever employed as a “Post Graduate Researcher” in the given quarter.

In order to merge the UMETRICS ETF PIK’d observations with our ACS-LEHD Doctorate Panel by PIK-institutionid-year-quarter, we must first bring the institutionid variable into the ACS-LEHD Doctorate Panel. Therefore, we construct an Institutionid-SEIN Crosswalk by merging together our LEHD Panel with a version of the UMETRICS ETF “PIK’d” observations that is unique on PIK-year-quarter.[40] After the merge, we keep observations associated with either “Colleges, Universities, and Professional schools” (NAICS=611310) or “General Medical and Surgical Hospitals” (NAICS=622110). For each institutionid, we keep any SEIN that is associated with at least 100 observations and make our institutionid-SEIN Crosswalk unique on SEIN. We then merge the institutionid-SEIN Crosswalk with our ACS-LEHD Doctorate Panel to bring the institutionid variable into our ACS-LEHD Doctorate Panel. At this stage, we then merge the ACS-LEHD Panel with the UMETRICS ETF “PIK’d” observations by PIK-institutionid-year-quarter to create an ACS-LEHD Doctorate panel with the UMETRICS subset of this data being identified and containing an indicator variable for postdoc status. UMETRICS mostly contains information on individuals employed in NAICS = 611310 or NAICS = 62210 industries. Therefore, we remove observations from our ACS-LEHD Doctorate Panel with UMETRICS that have different NAICS codes to improve the representativeness of our UMETRICS subsample which will be used to train our random forest model used to predict the postdoc status of the non-UMETRICS subsample. This leads us to our prediction sample “ACS-LEHD Academic Doctorate Panel with UMETRICS” which we make unique on PIK-year-quarter.

[40] This drops observations where a single person is employed at multiple IRIS member universities within the same quarter, but this should have a negligible effect in construction of the institutionid-SEIN crosswalk. The version of UMETRICS ETF “PIK’d” observations that we use later when directly merging with our ACS-LEHD Panel is unique on PIK-institutionid-year-quarter.

Appendix Tables

Table A.1: Variables included in ACS-LEHD Academic Doctoral Panel with UMETRICS

Variable	Definition
postdoc[1]	If occupational class = "Post Graduate Researcher", then postdoc==1; otherwise, postdoc==0.
Year	Year between 2002-2014
Quarter	Quarter between 1-4
age[2]	Year - birth year
male[2]	If male, then male==1; otherwise, male==0
white[2]	If white, then white==1; otherwise, white==0
black[2]	If black, then black==1; otherwise, black==0
native[2]	If Native American, then native==1; otherwise, native==0
asian[2]	If Asian, then asian==1; otherwise, asian==0
hispanic[2]	If Hispanic, then Hispanic==1; otherwise, Hispanic==0
other[2]	If other race, then other==1; otherwise, other==0
race_miss[2]	If race missing, then race_miss==1; otherwise, race_miss==0
stateborn[2]	Born in US State
terrborn[2]	Born in US Territory
foreign[2]	Foreign born
homelang[2]	Speaks another language at home
eng[2]	English speaking ability: 1= Very Well, 2 = Well, 3 = Not well, 4 = Not at all. If surveyed in multiple ACS years, uses worst reported language ability
military[2]	Ever serve in military?
disable[2]	Report a disability?
ind_1[2]	industry code = [awaiting disclosure]
ind_2[2]	industry code = [awaiting disclosure]
ind_3[2]	industry code = [awaiting disclosure]
ind_4[2]	industry code = [awaiting disclosure]
ind_5[2]	industry code = [awaiting disclosure]

ind_miss^{2}	industry code missing
occ_1^{2}	occupation code = [awaiting disclosure]
occ_2^{2}	occupation code = [awaiting disclosure]
occ_3^{2}	occupation code = [awaiting disclosure]
occ_4^{2}	occupation code = [awaiting disclosure]
occ_5^{2}	occupation code = [awaiting disclosure]
occ_miss^{2}	occupation code missing
univ^{3}	Employed in a NAICS = 611310 job (Colleges, University, and professional Schools) during quarter
hosp^{3}	Employed in a NAICS = 622110 job (General Medical and Surgical Hospitals) during quarter
univ_earn^{3}	Quarterly earnings across all NAICS = 611310 jobs
hosp_earn^{3}	Quarterly earnings across all NAICS = 622110 jobs
totearn^{3}	Quarterly earnings across all jobs
jobs^{3}	Total number of jobs during quarter as counted by number of SEINs
totseinunits^{3}	Total number of jobs during quarter as counted by number of SEINUNITs
univ_jobs^{3}	Total number of NAICS = 611310 jobs during quarter (SEIN)
hosp_jobs^{3}	Total number of NAICS = 622110 jobs during quarter (SEIN)
univ_qtrs_max^{3}	maximum # of quarters spent in a single SEIN where NAICS = 611310
univ_qtrs_min^{3}	minimum # of quarters spent in a single SEIN where NAICS = 611310
hosp_qtrs_max^{3}	maximum # of quarters spent in a single SEIN where NAICS = 622110
hosp_qtrs_min^{3}	minimum # of quarters spent in a single SEIN where NAICS = 622110
job_qtrs_max^{3}	maximum # of quarters spent in a single SEIN
job_qtrs_min^{3}	minimum # of quarters spent in a single SEIN

Notes: Superscripts indicate data sources used to create variable: 1 = UMETRICS, 2 = ACS, 3 = LEHD. The variable “postdoc” is only available for the UMETRICS subsets of our ACS-LEHD prediction sample