Nailing Prediction

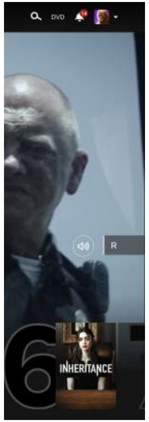
Experimental Evidence on The Impact of Tools in Predictive Model Development

Iavor Bojinov (Harvard Business School, Technology & Operations Unit)
With Daniel Yue and Paul Hamilton

AI is highly visible in key products in our society, *but* is unevenly implemented across businesses.

Industry	Human Resources	Manufacturing	Marketing And Sales	Product and/or Service Development	Risk	Service Operations	Strategy and Corporate Finance	Supply-Chain Management
All Industries	8%	12%	15%	21%	10%	21%	7%	9%
Automotive and Assembly	13%	29%	10%	21%	2%	16%	8%	18%
Business, Legal, and Professional Services	13%	9%	16%	21%	13%	20%	10%	9%
Consumer Goods/Retail	1%	19%	20%	14%	3%	10%	2%	10%
Financial Services	5%	5%	21%	15%	32%	34%	7%	2%
Healthcare/Pharma	3%	12%	16%	15%	4%	11%	2%	6%
High Tech/Telecom	14%	11%	26%	37%	14%	39%	9%	12%

% of Respondents



AI ADOPTION by INDUSTRY & FUNCTION, 2020

Source: McKinsey & Company, 2020 | Chart: 2021 AI Index Report

How can AI be unevenly implemented across businesses if the *technology* is public knowledge?

An accepted explanation is *worker skills*.

Tambe 2014; Tambe and Hitt 2014

While skills are no doubt important, they cannot explain the prevalence of *tools* in AI systems.
Can tools be an (additional) answer to this puzzle?

Amazon SageMaker
aws

+

scikit learn
PyTorch
Hugging Face

+

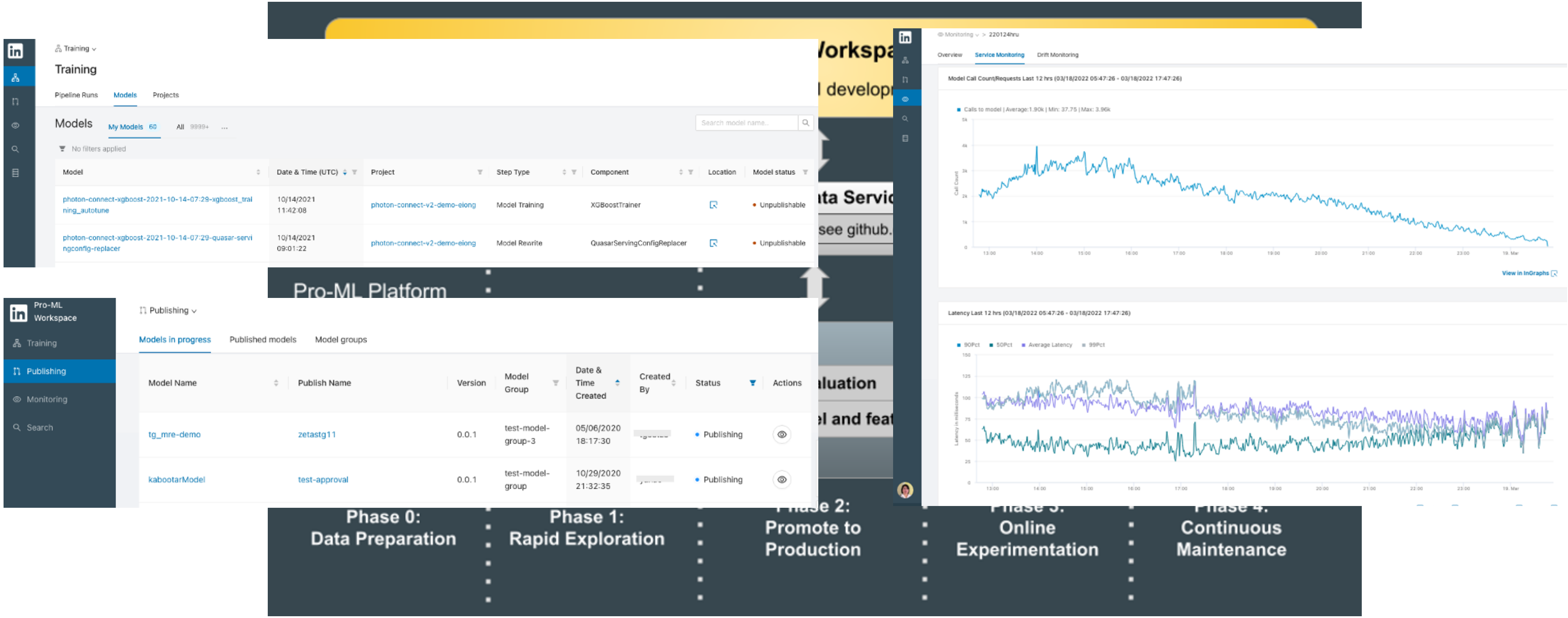
LinkedIn
P&G
Spotify

Cloud Infrastructure

Open Source Machine Learning Software

Internal Tools

LinkedIn's Pro-ML



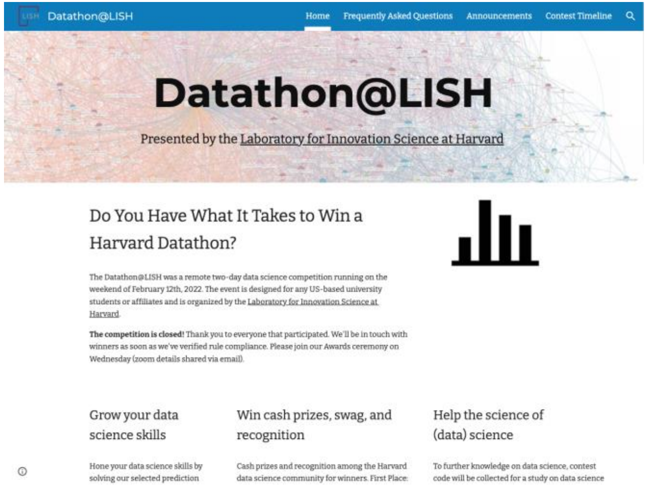
Research Questions

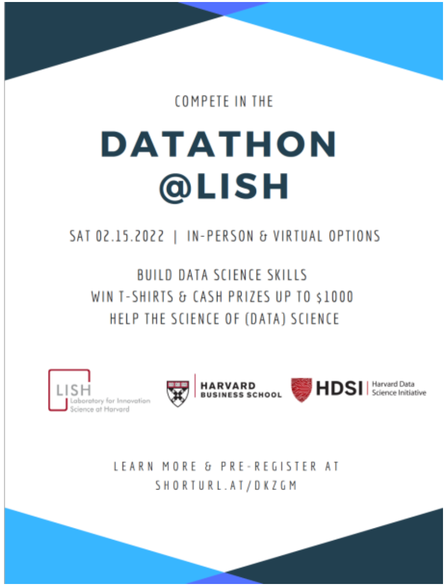
1. What is the effect of *tooling* on *predictive model development*? How does its effect compare to the effect of *methodology* on predictive model development?

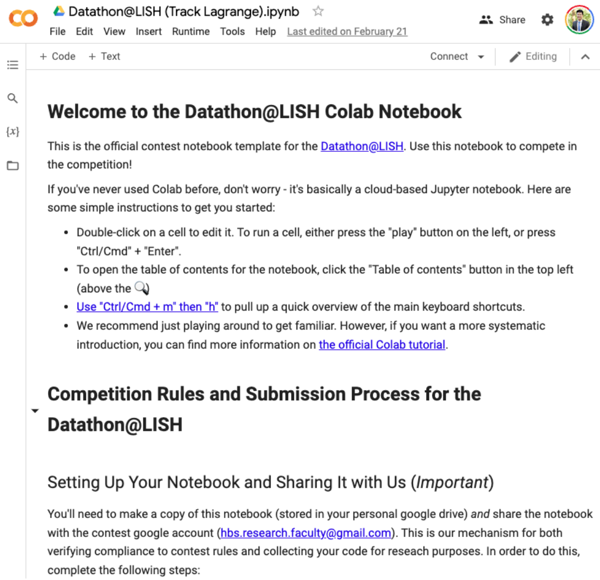
2. How does tooling interact with *skills*? Are tools and skills complements or substitutes?

(our terms will be better defined later…)

A field experiment with data scientists limiting access to key ML *software libraries*.







Outline

1. Conceptual Framework
2. Experiment Design
3. Results

Conceptual Framework

Typical ML Project Steps

SELECTION

Prioritizing & sequencing effectively

DEVELOPMENT

Model selection

EVALUATION

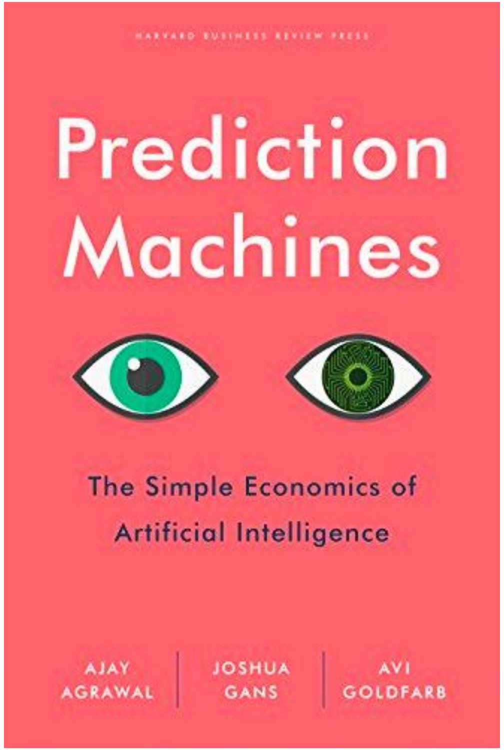
Experimentation (A/B testing)

DEPLOYMENT

Release and drive adoption

MANAGEMENT

Monitor, manage, and improve



Focus on:

Predictive Model Development

What are the *drivers* of predictive model development?

Typical Framework for Analyzing PMD

Data	Compute	Models
How an empirical reality is represented.	*How computational resources are structured and employed.*	*How models are structured so as to capture empirical regularities in the data.*

To better answer our questions, we extend a framework from the IT Productivity Literature.

1. The literature currently makes the distinction between *technology* and *skills* (Tambe 2014; Wu et al 2017).

2. We further separate technology into ***methodology*** and ***tooling***.
   - Methodology is *abstract*, *conceptual knowledge* of how to solve a problem.
   - Tooling is an *implementation* of a methodology, through a combination of hardware and software.

The Drivers of Predictive Model Development

	Data	Compute	Models
	How an empirical reality is represented.	*How computational resources are structured an employed*	*How models are structured so as to capture empirical regularities in the data.*
Methodology	Split Apply Combine One-hot encoding More Observations	Backpropagation / AutoDiff Accelerators (GPUs / TPUs) Infrastructure-as-a-Service	Random Forest BERT (Language Models) AutoML

Methodology is *abstract*, *conceptual knowledge*.

The Drivers of Predictive Model Development

	Data	**Compute**	**Models**
	How an empirical reality is represented.	*How computational resources are structured an employed*	*How models are structured so as to capture empirical regularities in the data.*
Methodology	Split Apply Combine One-hot encoding More Observations	Backpropagation / AutoDiff Accelerators (GPUs / TPUs) Infrastructure-as-a-Service	Random Forest BERT (Language Models) AutoML
Tools	R's `dplyr` Python's `pandas` PostgreSQL	PyTorch NVIDIA's CUDA Google Cloud Platform	*Sci-kit Learn* *Hugging Face Transformers* *AWS SageMaker AutoPilot*

Tooling is a *specific implementation / integration* of known methods.

The Drivers of Predictive Model Development

	Data	**Compute**	**Models**
	How an empirical reality is represented.	*How computational resources are structured an employed*	*How models are structured so as to capture empirical regularities in the data.*
Methodology	Split Apply Combine One-hot encoding More Observations	Backpropagation / AutoDiff Accelerators (GPUs / TPUs) Infrastructure-as-a-Service	Random Forest BERT (Language Models) AutoML
Tools	R's `dplyr` Python's `pandas` PostgreSQL	PyTorch NVIDIA's CUDA Google Cloud Platform	*Sci-kit Learn* *Hugging Face Transformers* *AWS SageMaker AutoPilot*

To better answer our questions, we extend a framework from the IT Productivity Literature.

1. The literature currently makes the distinction between *technology* and *skills* (Tambe 2014; Wu et al 2017).

2. We further separate technology into ***methodology*** and ***tooling***.
    - Methodology is *abstract*, *conceptual knowledge* of how to solve a problem.
    - Tooling is an *implementation* of a methodology, through a combination of hardware and software.

The Drivers of Predictive Model Development

<table>
<tr><th></th><th>Data</th><th>Compute</th><th>Models</th></tr>
<tr><td></td><td>How an empirical reality is represented.</td><td>How computational resources are structured an employed</td><td>How models are structured so as to capture empirical regularities in the data.</td></tr>
<tr><td>Methodology</td><td>Split Apply Combine
One-hot encoding
More Observations</td><td>Backpropagation / AutoDiff
Accelerators (GPUs / TPUs)
Infrastructure-as-a-Service</td><td>Random Forest
BERT (Language Models)
AutoML</td></tr>
<tr><td>Tools</td><td>R's <code>dplyr</code>
Python's <code>pandas</code>
PostgreSQL</td><td>PyTorch
NVIDIA's CUDA
Google Cloud Platform</td><td>Sci-kit Learn
Hugging Face Transformers
AWS SageMaker AutoPilot</td></tr>
<tr><td rowspan="2">Skill</td><td>Domain-Specific Skill</td><td>Computer-Specific Skill</td><td>Modeling-Specific Skill</td></tr>
<tr><td colspan="3" align="center">General Skill</td></tr>
</table>

Skill is further broken into specific skills and general skills.

Research Questions 1

What is the effect of tooling on predictive model development? How does its effect compare to the effect of methodology on predictive model development?

Restricting access to tools reduces log-loss score by ~30% of the gains over baseline.

This corresponds to **reducing the training data set to 10-15%** of its original size (a reduction of 85%!)

Research Question 2

How does tooling interact with skills? Are tools and skills complements or substitutes?

Tooling does not interact with an aggregate measure of skill.

BUT Tools are *complementary* to **specific skills**, but *substitutable* with **general skills**.

Mechanism: *tools encodes general skills*, changing the type of skills needed to develop effective models.

Experiment Design

Experimental Design

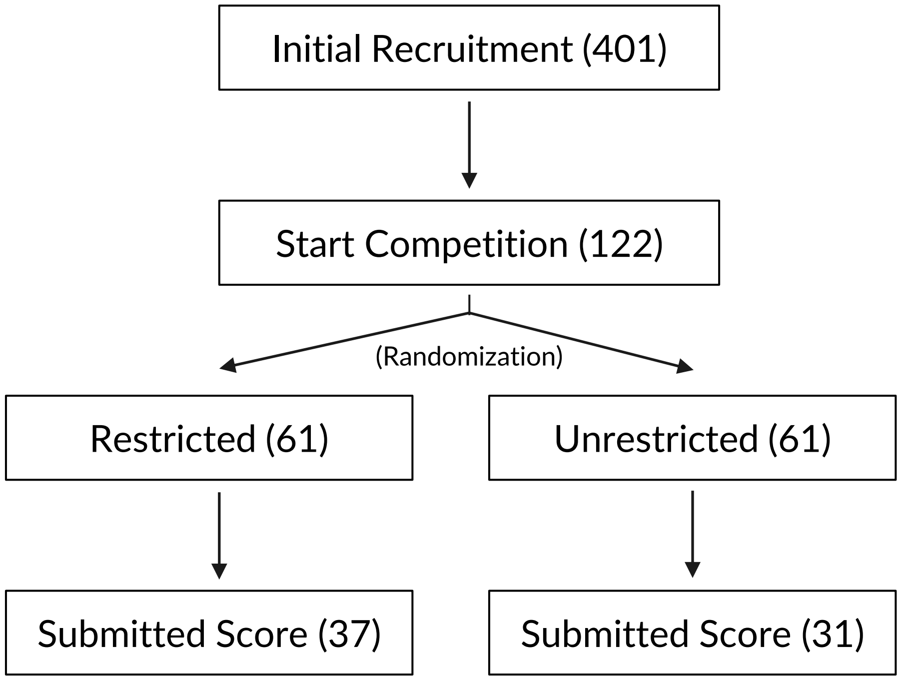
Experimental Setting.
We created a (private) contest on Kaggle, a leading platform for coordinating data science competitions. We recruited teams of either one or two participants from leading US universities for a 48-hour Datathon.

Experimental Task.
We tasked teams with solving a statistical prediction problem from DrivenData. We awarded prizes to participants based on the final loss score of their best submitted model.

Treatment: Teams are restricted from using machine learning modeling functions from software libraries (importantly, *they are free to reimplement* or use standard GLM approaches).

Control: Teams are unrestricted in use of libraries.

Recruitment & Randomization



Teams of 1 or 2 were recruited from a wide set of leading US Universities

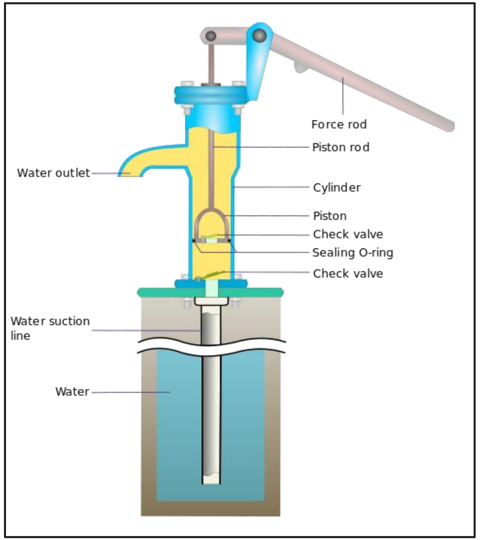
Teams started competition by completing Qualtrics Survey

Teams randomized (Bernoulli) into treatment arm and given separate Colab notebooks

Teams work on problem and submit scores on Kaggle.

The Task: Taarifa's Pump Repairs (Binary Classification)

A smart understanding of which waterpoints will fail can improve maintenance operations and ensure that clean, potable water is available everywhere. In this competition, you will develop a predictive model that solve binary classification task focused on predicting the operational status of water pumps throughout Tanzania, based on some the provided information about their installation context.



Data Given:

- Pump Status
- Type (Funder, Water Source, Installation Details)
- Management (Organization, Payments)
- Location (Region, Lat/Long)
- Geography (Altitude)
- Demographics

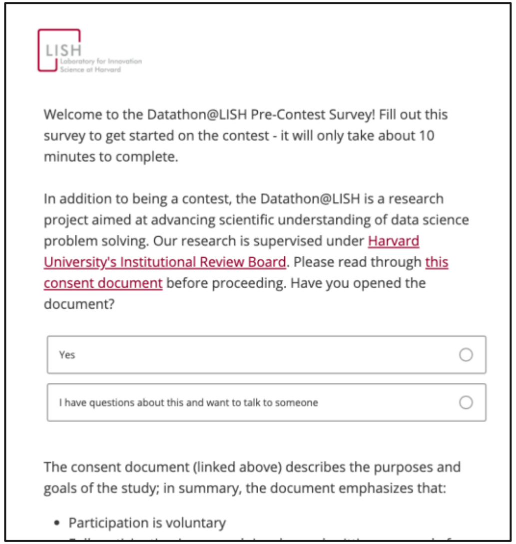
Treatment definition

Restricted use of advanced ML python libraries.

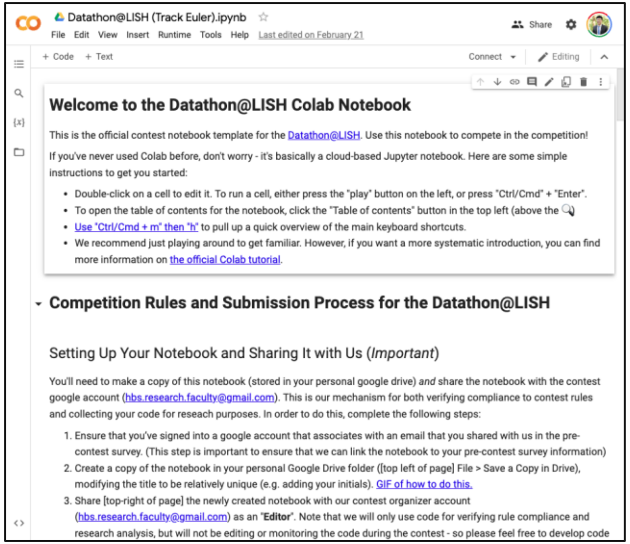
- No libraries that implement anything more advanced than constrained generalized linear models. (So no random forest, neural networks, etc.)
- BUT, can use any functions for feature engineering or other non-modeling related tasks (test-train split etc.).

Automatic checks for compliance—no one broke the rules.

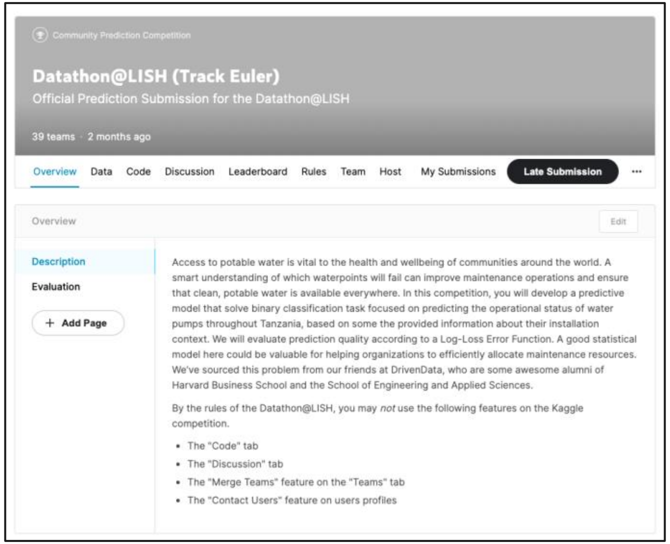
Data Sources



Qualtrics (Background)



Colab (Code)



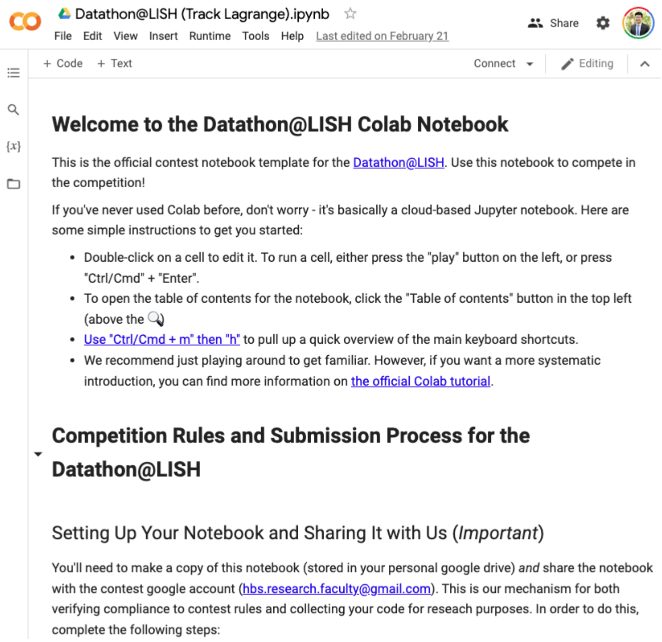
Kaggle (Submissions)

Primary Variable Construction

- Treatment: Restricted use of advanced ML python libraries.
  A dummy variable called “Unrestricted” that takes a value 1 if team can use modeling libraries and 0 otherwise.

- Primary Outcome: A normalized *Score* obtained by applying an affine transformation such that the best score is 1 and the baseline score is 0. [Kaggle]

Additional Data


Datathon@LISH (Track Lagrange).ipynb

File Edit View Insert Runtime Tools Help Last edited on February 21

Share

+ Code + Text

Connect Editing

Welcome to the Datathon@LISH Colab Notebook

This is the official contest notebook template for the Datathon@LISH. Use this notebook to compete in the competition!

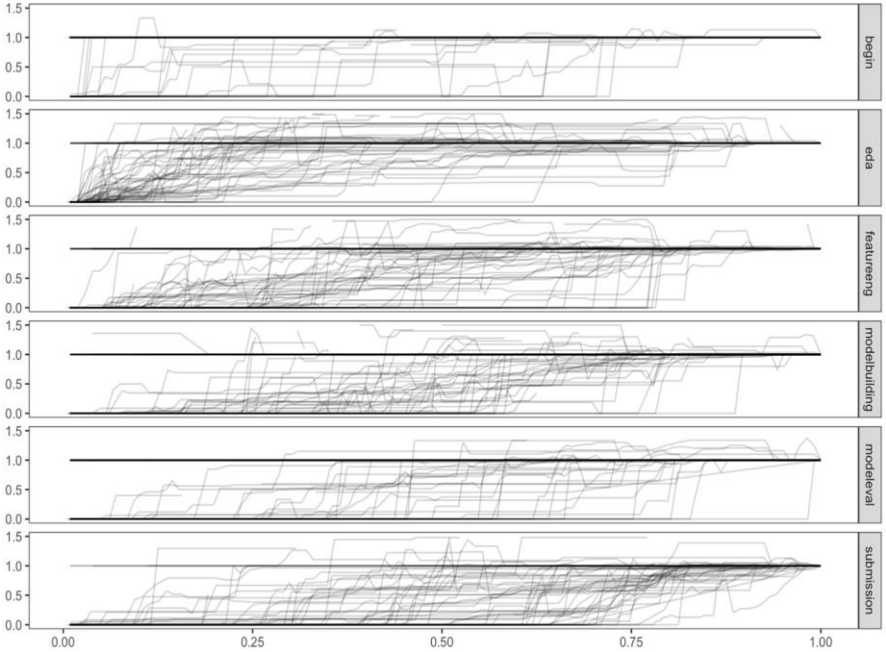
If you've never used Colab before, don't worry - it's basically a cloud-based Jupyter notebook. Here are some simple instructions to get you started:

- Double-click on a cell to edit it. To run a cell, either press the "play" button on the left, or press "Ctrl/Cmd" + "Enter".
- To open the table of contents for the notebook, click the "Table of contents" button in the top left (above the 🔍)
- Use "Ctrl/Cmd + m" then "h" to pull up a quick overview of the main keyboard shortcuts.
- We recommend just playing around to get familiar. However, if you want a more systematic introduction, you can find more information on the official Colab tutorial.

Competition Rules and Submission Process for the Datathon@LISH

Setting Up Your Notebook and Sharing It with Us (*Important*)

You'll need to make a copy of this notebook (stored in your personal google drive) *and* share the notebook with the contest google account (hbs.research.faculty@gmail.com). This is our mechanism for both verifying compliance to contest rules and collecting your code for reseach purposes. In order to do this, complete the following steps:



Results

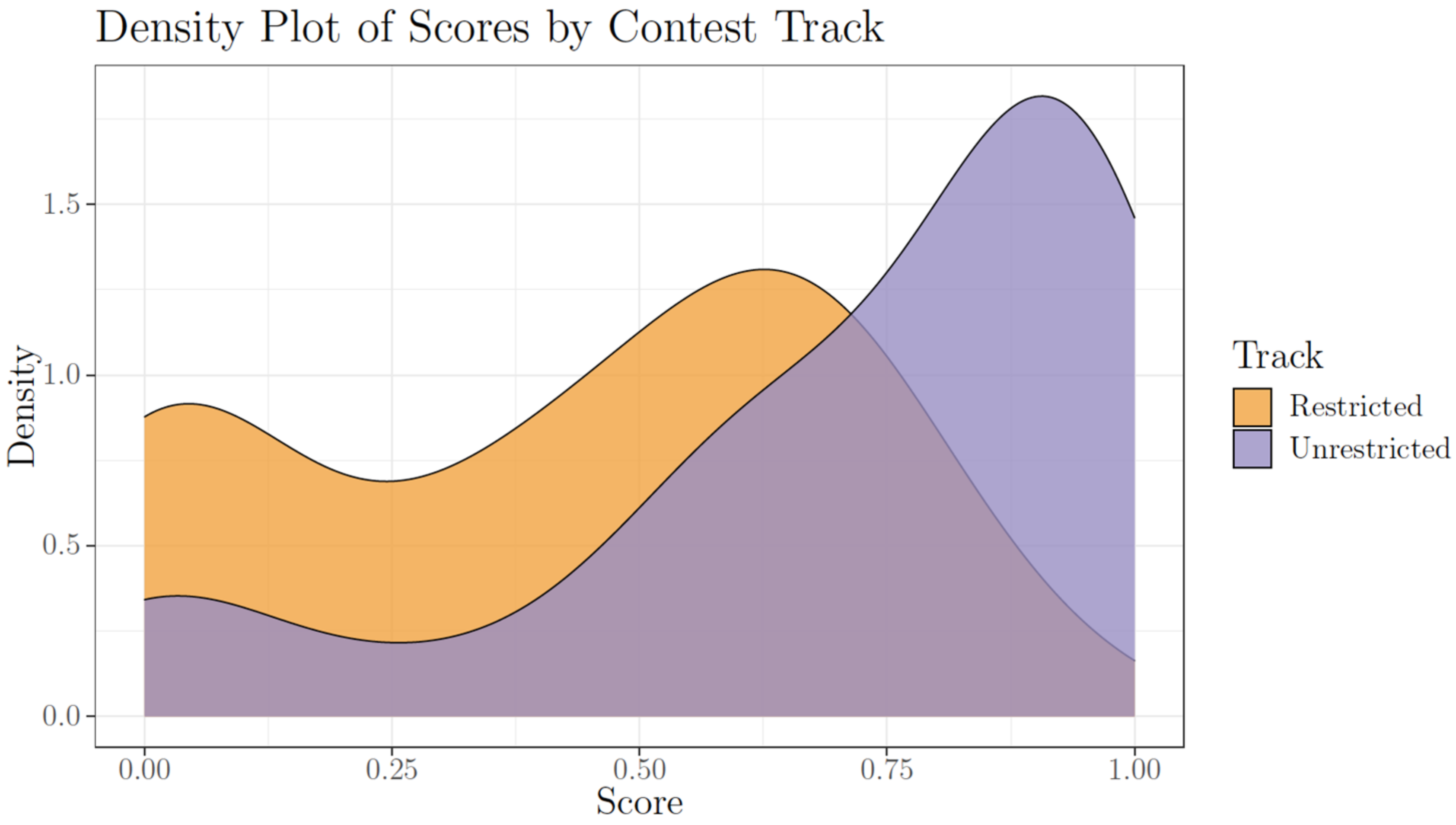
Variable	**N = 100**
Year Finished Undergrad	2021 (2016, 2022)
Current or Most Recent Degree	
Undergraduate	29%
Masters	49%
PhD	22%
Undergraduate Major	
Data Science	27%
Physical Science or Engineering	56%
Social Science	17%
Current Employment Status	
Student (full-time)	83%
Employed (full-time)	11%
Other	6%
Prior Jobs in Software, Research, or Data Science	
None	21%
Internship	43%
Full Time Work	36%
Prior Jobs in (Only) Data Science	
None	53%
Internship	32%
Full Time Work	15%
Gender	
Male	57%
Female	42%
Prefer Not to Say	1%

Question 1 — Analytic Approach

1. *What is the effect of tools [modeling libraries] on predictive model development? How does its effect compare to the effect of methodology [training set size] on predictive model development?*

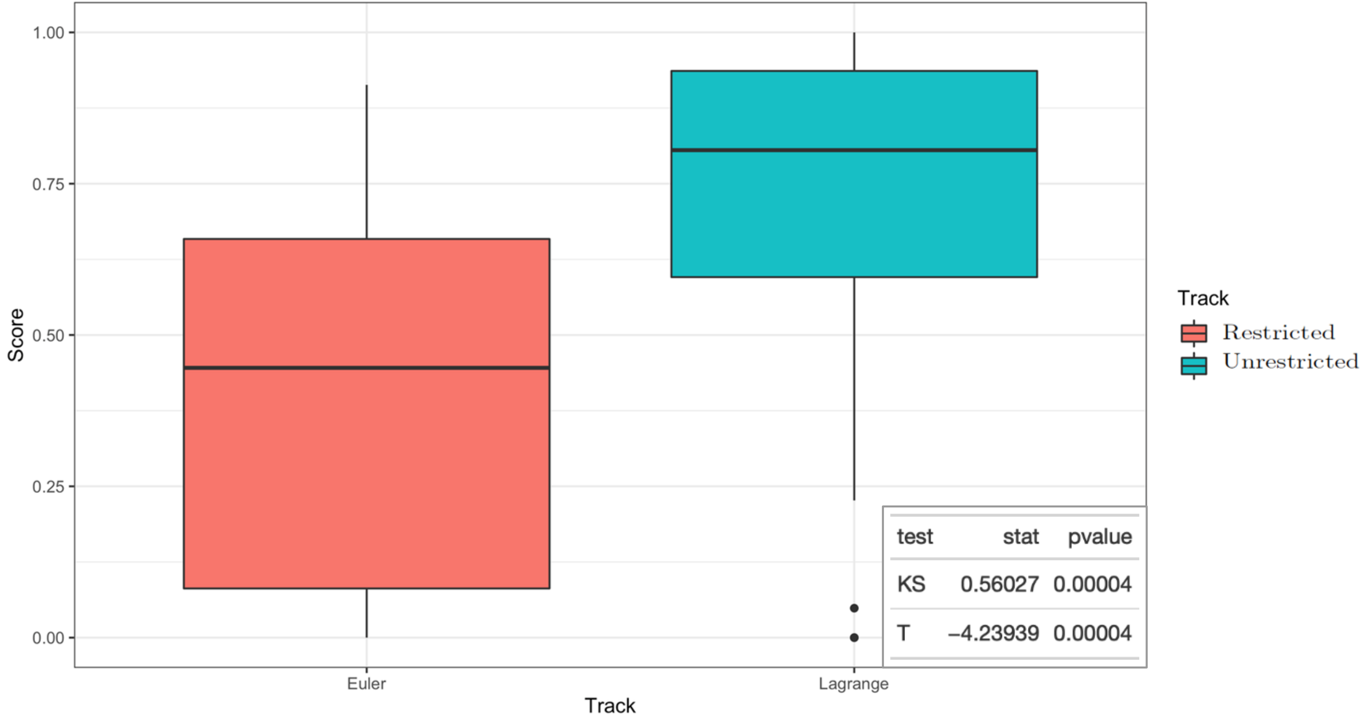
   Nonparametric analysis & regression

   We rerun the models from the winner's of each track on progressively smaller training data sets to compare our estimate to the effect of data set size on predictive model performance.

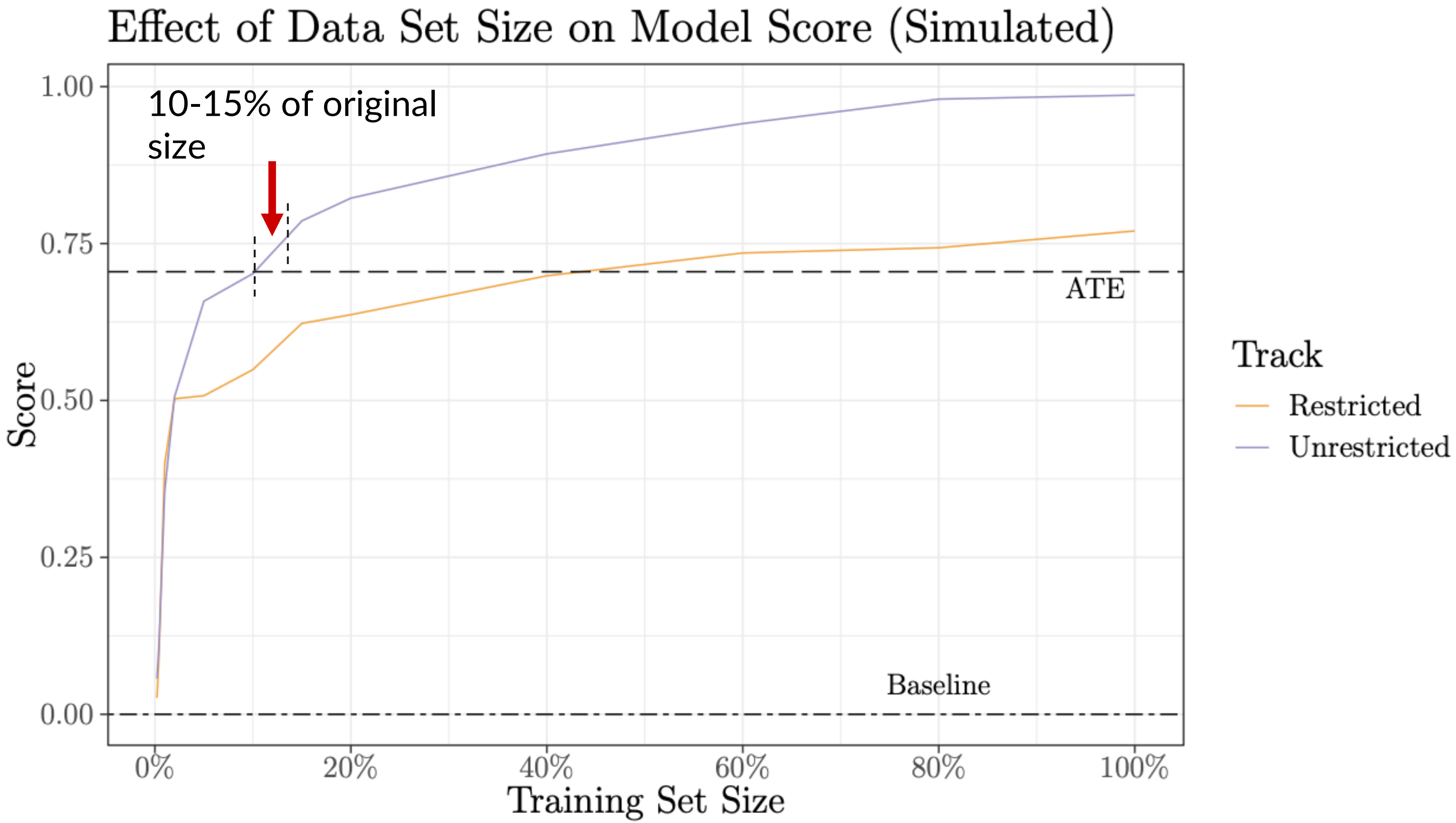

Density Plot of Scores by Contest Track
Density
1.5
1.0
0.5
0.0
0.00
0.25
0.50
0.75
1.00
Score
Track
Restricted
Unrestricted

	Model 1	Model 2
Dependent Var.:	Score	Score
Unrestricted	0.2949*** (0.0719)	0.2879*** (0.0715)
Team is Pair		0.0889 (0.0841)
Comfortable with sklearn		-0.0380 (0.0989)
Comfortable with Feature Engineering		0.1238 (0.0869)
Comfortable with SQL		-0.1503** (0.0710)
Had Prior Data Science Job		-0.0161 (0.0781)
(Intercept)	0.4171*** (0.0479)	0.4118*** (0.0924)
S.E. type	Heteroskedas.-rob.	Heteroskedas.-rob.
Observations	68	68
R2	0.20372	0.25584
Adj. R2	0.19166	0.18265

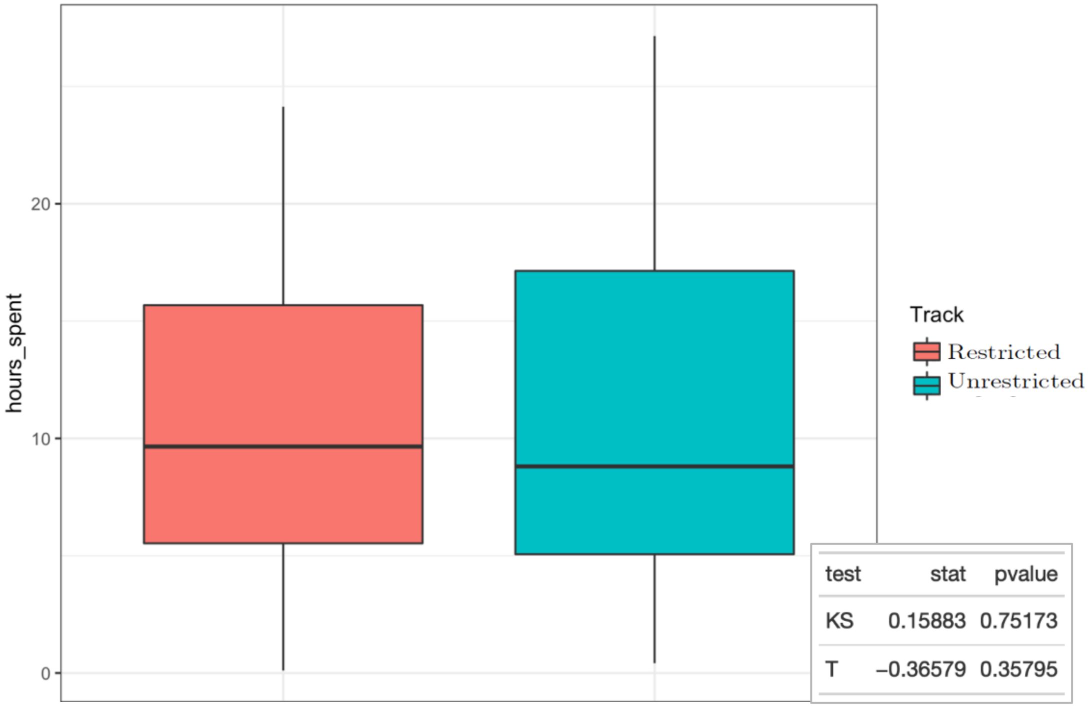
The restrictions reduce the normalized score by almost 30% of the possible gains baseline



test	stat	pvalue
KS	0.56027	0.00004
T	−4.23939	0.00004


Effect of Data Set Size on Model Score (Simulated)
10-15% of original size
1.00
0.75
0.50
0.25
0.00
Score
ATE
Baseline
0%
20%
40%
60%
80%
100%
Training Set Size
Track
Restricted
Unrestricted

The effect was not driven by differences in rate of participation or effort



test	stat	pvalue
KS	0.15883	0.75173
T	−0.36579	0.35795

Research Questions 1

What is the effect of tooling on predictive model development? How does its effect compare to the effect of methodology on predictive model development?

Restricting access to tools reduces log-loss score by ~30% of the gains over baseline.

This corresponds to **reducing the training data set to 10-15%** of its original size (a reduction of 85%!)

The result is not driven by "effort" or time-spent.

Question 2 — Analytic Approach

2. *How does technology interact with skills? Are technology and skills complements or substitutes?*

   We form *skill* indexes aggregating prior experiences of participants. We distinguish further between *general* and *specific* skills. General indicates statistical problem solving abilities (e.g. prior employment as DS). Specific indicates knowledge of modeling and modeling tools (e.g. experience with sklearn).

   We convert the measures into binary indicators and estimate interactions with treatment.

	Model 1
Dependent Var.:	Score
Unrestricted	0.3596** (0.1065)
Unrestricted x High Skill (Total)	-0.1080 (0.1443)
High Skill (Total)	0.0817 (0.1012)
(Intercept)	0.3685*** (0.0829)
S.E. type	Heteroskedas.-rob.
Observations	68
R2	0.21269
Adj. R2	0.17578

	Model 3
Dependent Var.:	Score
Unrestricted	0.2196. (0.1139)
Unrestricted x High Skill (General)	-0.2586* (0.1264)
High Skill (General)	0.0983 (0.1203)
Unrestricted x High Skill (Specific)	0.3422** (0.1262)
High Skill (Specific)	-0.0361 (0.1221)
High Skill (General) x High Skill (Specific)	-0.1426 (0.1294)
(Intercept)	0.4478*** (0.0993)
S.E. type	Heteroskedas.-rob.
Observations	68
R2	0.31640
Adj. R2	0.24916

Research Question 2

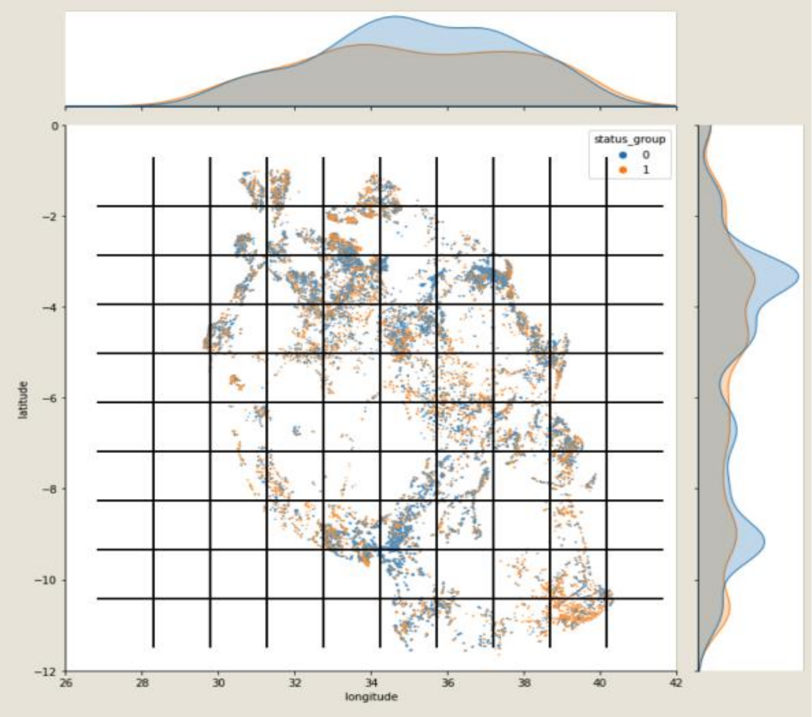
How does tooling interact with skills? Are tools and skills complements or substitutes?

Tooling does not interact with an aggregate measure of skill.

BUT Tools are *complementary* to **specific skills**, but *substitutable* with **general skills**.

Contest Results - Solution Approaches

Different tree-based and Boosting models such as Random Forest, XGBoost, LightGBM, CATBoost were tested and *the best model (CATBoost) was selected based on the validation accuracy*… The model with the best validation log-loss was considered for the final submission on Kaggle.



Unrestricted 1st: Focus on Model Approaches

Restricted 1st: Focus on Feature Engineering

Mechanism – Tools-as-Skills

Modeling libraries act as *substitute* for general skills (like intuitive feature development) – but only when teams had the *complementary* modeling-specific skills needed to use them.

Implication: tools allows for targeted training that can lower the cost of predictive model development for firms.

Implications

1. IT Productivity: we extend the technology-skills framework by distinguishing technology into methodology and tooling. We conceptualize the mechanism by which tools drive predictive model development (“tools as skills”) and present experimental evidence in support of that mechanism.

2. Economics of AI: we are the first to frame predictive model development as a theoretical problem and to contribute a novel conceptual framework and empirical methodology to studying it.

Managerial Implications

Foundational Understanding

Data literacy, use data for judgment and judgment for data

General Skills

Broad knowledge of statistics and computer science

Tool-Specific Skills

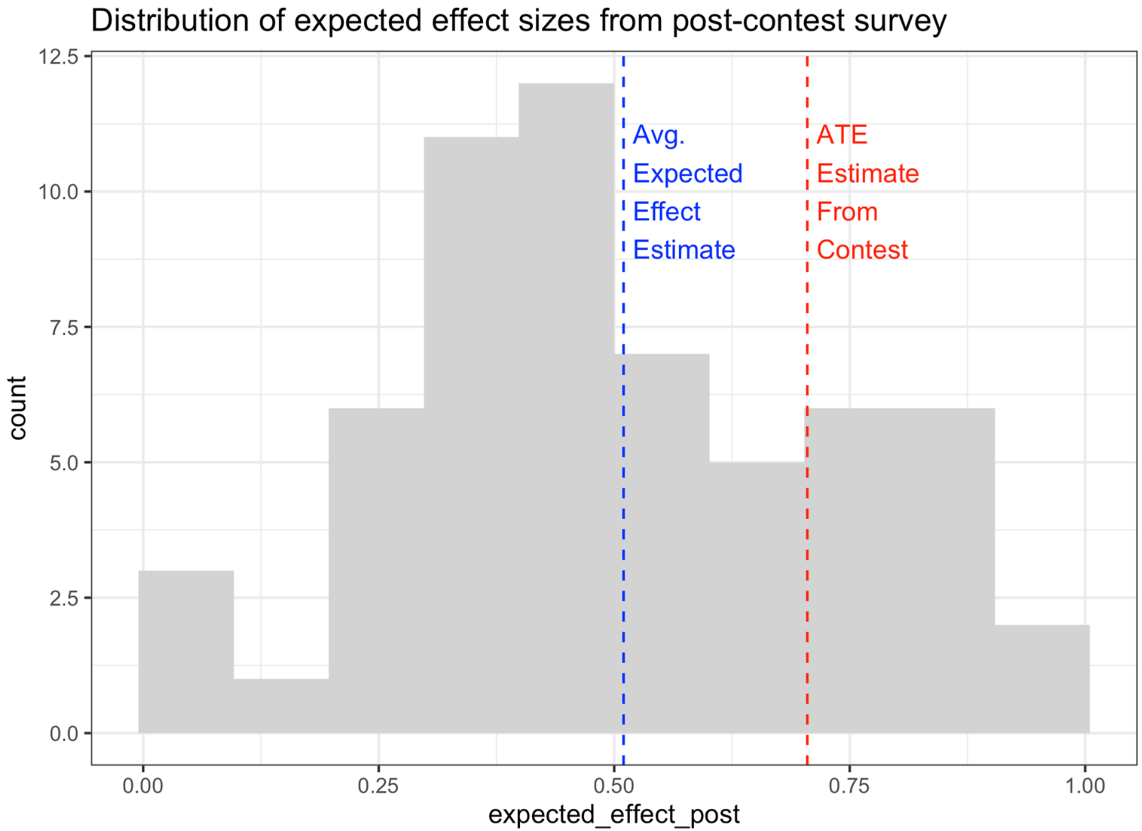
Experience with tools for implementing AI/ML models

Thank you!

Questions or suggestions? Email: **ibojinov@hbs.edu**

Working paper

Expectation result: The participants expected a larger effect than was observed (from post-survey)



	Model 1	Model 2	Model 3
Dependent Var.:	Score	Percent Accuracy	Work Hours
Unrestricted	0.2949*** (0.0719)	0.1237*** (0.0250)	-0.5745 (1.519)
(Intercept)	0.4171*** (0.0479)	0.7691*** (0.0227)	7.237*** (0.9974)
Observations	68	68	63
R2	0.20372	0.24746	0.00236
Adj. R2	0.19166	0.23606	-0.01400

Track	Overall, N = 68[1]	euler, N = 37[1]	lagrange, N = 31[1]	p-value[2]
year_undergrad	2,021.0 (2,016.0, 2,022.1)	2,020.0 (2,015.5, 2,022.0)	2,021.0 (2,016.0, 2,022.5)	0.5
info_educ				0.13
Masters	31 (46%)	21 (57%)	10 (32%)	
PhD	17 (25%)	7 (19%)	10 (32%)	
Undergraduate	20 (29%)	9 (24%)	11 (35%)	
info_employ				0.7
Employed (full-time)	9 (13%)	6 (16%)	3 (9.7%)	
Other	5 (7.4%)	3 (8.1%)	2 (6.5%)	
Student (full-time)	54 (79%)	28 (76%)	26 (84%)	
info_major				0.8
DataSci	19 (28%)	11 (30%)	8 (26%)	
SocSci	10 (15%)	6 (16%)	4 (13%)	
STEM	39 (57%)	20 (54%)	19 (61%)	
info_gender				0.2
Female	35 (51%)	16 (43%)	19 (61%)	
Male	32 (47%)	20 (54%)	12 (39%)	
Prefer Not to Say	1 (1.5%)	1 (2.7%)	0 (0%)	

[1] n (%); Median (IQR)

[2] Pearson's Chi-squared test; Wilcoxon rank sum test; Fisher's exact test

Team-Level Summary Statistics (Demographics)

Most teams graduated after 2015.

Primarily masters students participated. Many had prior work or internship experiences in data science.

The majority of participants are currently students

Most students came from STEM or data science majors.

The gender composition skewed male, reflecting general trends in STEM.

Track	Overall, N = 68[1]	euler, N = 37[1]	lagrange, N = 31[1]	p-value[2]
priorcourses_os				0.3
None	42 (62%)	23 (62%)	19 (61%)	
One Course	17 (25%)	11 (30%)	6 (19%)	
Two Courses or More	9 (13%)	3 (8.1%)	6 (19%)	
prioremploy_datascience				0.6
Full Time Work	14 (21%)	6 (16%)	8 (26%)	
Internship	25 (37%)	14 (38%)	11 (35%)	
None	29 (43%)	17 (46%)	12 (39%)	
None	22 (32%)	16 (43%)	6 (19%)	
prioremploy_softwaredev				0.4
Full Time Work	10 (15%)	4 (11%)	6 (19%)	
Internship	15 (22%)	7 (19%)	8 (26%)	
None	43 (63%)	26 (70%)	17 (55%)	
priorlanguages_sql				0.8
Comfortable	31 (46%)	16 (43%)	15 (48%)	
Heard of it but unfamiliar	15 (22%)	9 (24%)	6 (19%)	
Never heard of it	1 (1.5%)	0 (0%)	1 (3.2%)	
Used / Done Before	21 (31%)	12 (32%)	9 (29%)	

[1] n (%); Median (IQR)
[2] Pearson's Chi-squared test; Wilcoxon rank sum test; Fisher's exact test

Team-Level Summary Statistics (General Exp)

Team had significant prior experiences in employment as both data scientists and software engineers.

Some had key related general skills such as operating systems and SQL knowledge.

Track	**Overall**, N = 68[1]	**euler**, N = 37[1]	**lagrange**, N = 31[1]	**p-value**[2]
priorstatmodels_regul				0.6
Comfortable	45 (66%)	22 (59%)	23 (74%)	
Heard of it but unfamiliar	4 (5.9%)	3 (8.1%)	1 (3.2%)	
Never heard of it	4 (5.9%)	3 (8.1%)	1 (3.2%)	
Used / Done Before	15 (22%)	9 (24%)	6 (19%)	
priormlstages_modelbuild				0.5
Comfortable	41 (60%)	22 (59%)	19 (61%)	
Heard of it but unfamiliar	5 (7.4%)	4 (11%)	1 (3.2%)	
Never heard of it	1 (1.5%)	1 (2.7%)	0 (0%)	
Used / Done Before	21 (31%)	10 (27%)	11 (35%)	
priorlibraries_scipy				0.8
Comfortable	41 (60%)	21 (57%)	20 (65%)	
Heard of it but unfamiliar	6 (8.8%)	4 (11%)	2 (6.5%)	
Used / Done Before	21 (31%)	12 (32%)	9 (29%)	
priorlibraries_sklearn				0.2
Comfortable	51 (75%)	29 (78%)	22 (71%)	
Heard of it but unfamiliar	3 (4.4%)	0 (0%)	3 (9.7%)	
Used / Done Before	14 (21%)	8 (22%)	6 (19%)	

[1] n (%); Median (IQR)
[2] Pearson's Chi-squared test; Wilcoxon rank sum test; Fisher's exact test

Team-Level Summary Statistics (Specific Exp)

Many participants had specific experience in tools and methods most closely associated with data science problem solving.

For example, about 60% of participants were comfortable with modeling building.

60% of participants were comfortable with scientific computing tools like scikit-learn.

	Model 3	Model 4
Dependent Var.:	**`Percent Accuracy`**	**`Percent Accuracy`**
`Unrestricted`	0.1237*** (0.0250)	0.1185*** (0.0258)
`Team is Pair`		0.0287 (0.0266)
`Comfortable with sklearn`		-0.0209 (0.0362)
`Comfortable with Feature Engineering`		0.0311 (0.0359)
`Comfortable with SQL`		-0.0252 (0.0256)
`Had Prior Data Science Job`		0.0150 (0.0287)
(Intercept)	0.7691*** (0.0227)	0.7582*** (0.0360)
S.E. type	Heteroskedas.-rob.	Heteroskedas.-rob.
Observations	68	68
R2	0.24746	0.27237
Adj. R2	0.23606	0.20080

Note: baseline Percent Accuracy is 0.55.

	Model 1	Model 2	Model 3
Unrestricted	0.2936** (0.0732)	0.3134** (0.0670)	0.2295** (0.0622)
High Skill (Total)	-0.0359 (0.0732)		
Unrestricted x High Skill (Total)	-0.0803 (0.1463)		
High Skill (General)		-0.0876 (0.0815)	-0.0784 (0.0622)
Unrestricted x High Skill (General)			-0.3028* (0.1243)
High Skill (Specific)		0.0645 (0.0842)	0.0499 (0.0622)
Unrestricted x High Skill (Specific)			0.3309** (0.1243)
High Skill (General) x High Skill (Specific)			-0.2656* (0.1243)
Unrestricted x High Skill (General) x High Skill (Specific)			0.5071* (0.2487)
(Intercept)	0.5639** (0.0366)	0.5642** (0.0421)	0.5770** (0.0311)
Observations	68	68	68
R2	0.20996	0.26470	0.34072
Adj. R2	0.17293	0.21801	0.26380

Checks for Experiment Validity

- Balance Checks [seems like no difference from drop-outs]