Machine Intelligence vs. Human Judgement in New Venture Finance*

Christian Catalini
MIT Sloan

Chris Foster
Harvard Business School

Ramana Nanda
Harvard Business School

August 2018

Abstract

We use data from a leading startup accelerator in the US to study how supervised machine learning models trained to mimic human evaluators performed relative to models trained purely to maximize financial success (regardless of whether they were selected by the accelerator). We find that (1) models trained to mimic the picks of humans performed well out-of-sample, implying that humans had a systematic pattern of early stage investing that could be identified and replicated; (2) Models trained to maximize success strongly outperformed 'mimic human models' when picking from a common out-of-sample applicant pool, implying that heuristics used by these evaluators were systematically overlooking certain high-potential applications that were identifiable ex ante; (3) comparing the focus of the two models suggests that the differences arose in part due to human heuristics systematically under-emphasizing more 'cognitively demanding' elements of the applications. Our findings have important implications for the selection and financing of high potential ideas, and more broadly for how Artificial Intelligence can help humans screen and evaluate information in an era of increasing 'information overload'.

*We are grateful to Ajay Agrawal, Rem Koning, Josh Lerner and participants at the NBER Productivity lunch for very helpful discussions. Catalini recognizes support of the Kauffman Foundation Junior Faculty Fellowship and MIT Sloan. Foster and Nanda thank the Division of Research and Faculty Development at HBS for financial support. All errors are our own.

Figure 1: Comparing Rank Correlation and Correlation of Weights across the two models

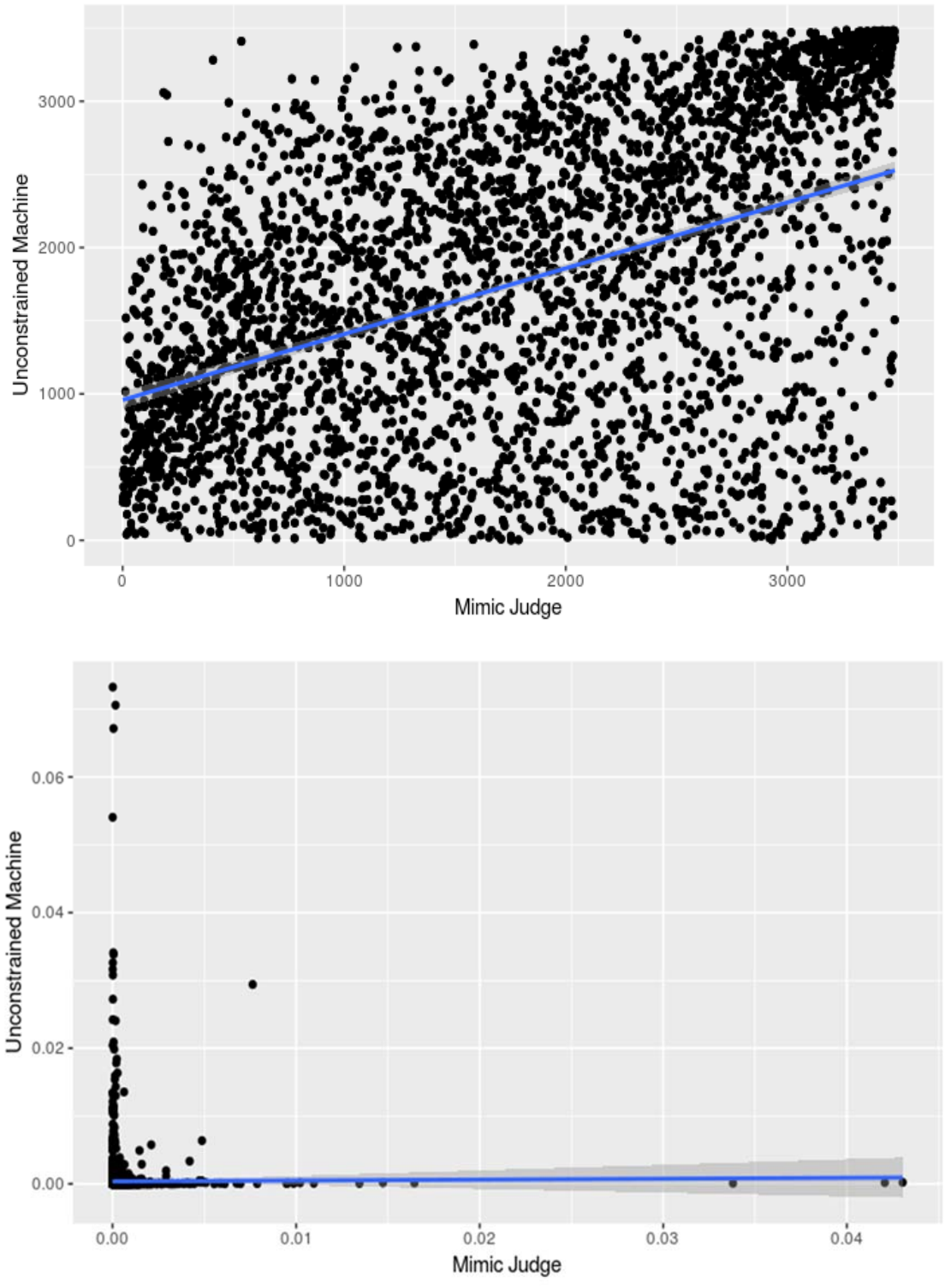

Figure 2: Comparing Rank Correlation and Correlation of Weights across the two models, by category

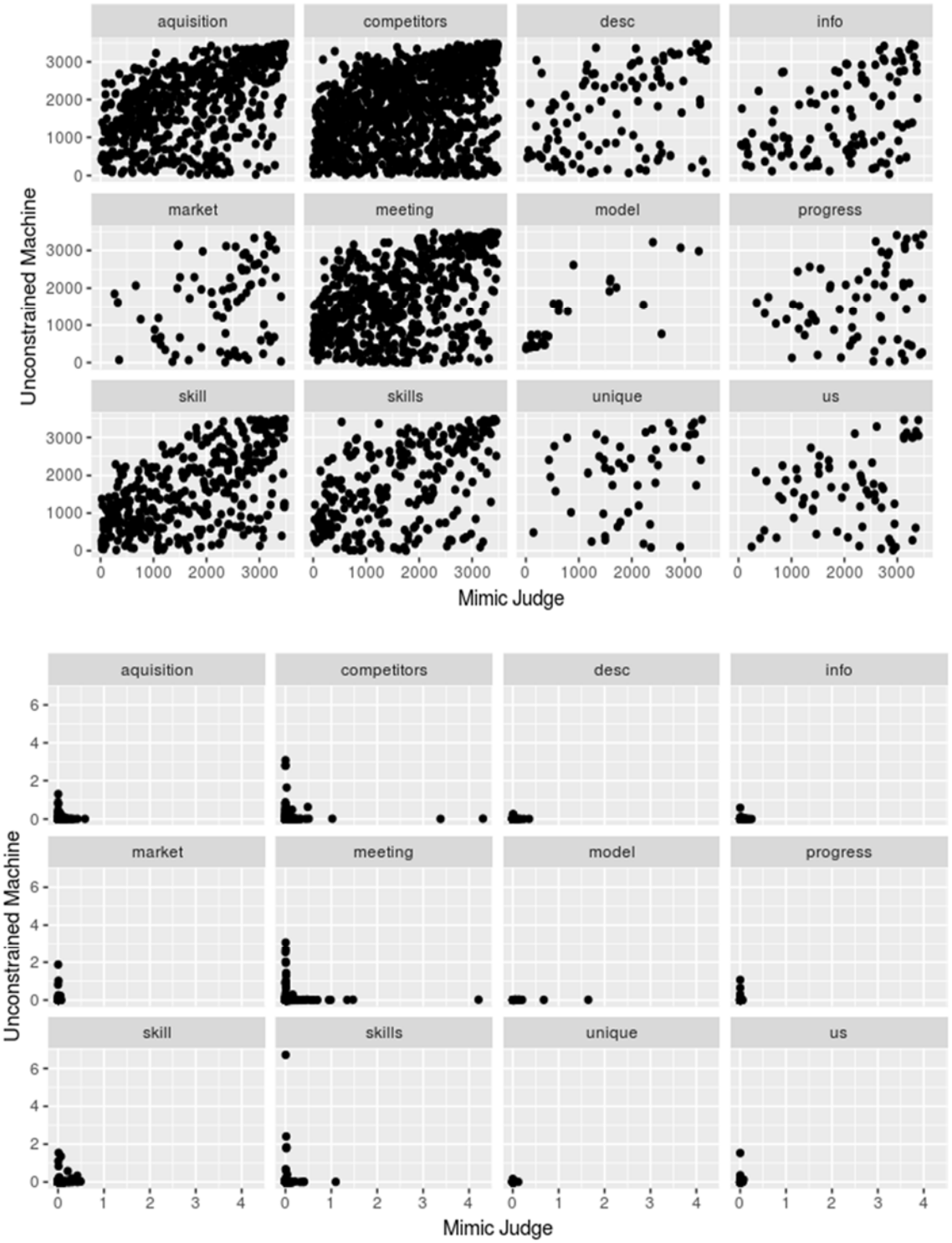

Table 1A: Descriptive Statistics

This table reports descriptive statistics on the sample of 14,423 complete applications we use in this study. These applications were received by the accelerator across 17 cohorts between 2013 and 2016. Of these 14,423 applications, we identified 979 ventures (including name changes, and regardless of whether they were accepted by the accelerator) that went on to raise sufficient funding from angel and institutional investors so as to be recorded in the following three databases of venture capital financing: AngelList, VentureXpert and Preqin. In instances where we did not have information on subsequent financing, we conservatively assumed the amount of money raised by the venture subsequent to application was zero. Given this strong assumption imposed on applications without outcome information, we separately report descriptive statistics and subsequent results for the full sample (including imputed outcomes values) and the sub-sample of potentially higher quality applicants where we located outcome information. Panel A reports details on the number of applicants, the scores they received and their outcomes; Panels B and C report descriptive statistics on the structured and unstructured data in the applications that we use to train the machine learning algorithms.

PANEL A: NUMBER OF APPLICATIONS, JUDGES AND DOLLARS RAISED

	All Applications	Applications with outcome Information
Number of observations	14,423	979
Average application year	2014.6	2014.4
Average score received from evaluators (Maximum=4)	1.86	3.28
Average number of evaluators per application	1.9	2.4
Average correlation in score between the evaluators	0.49	0.6
Average $ raised / startup since application if selected	319,625	3,411,398
Average $ raised / startup since application if not selected	649,813	2,819,046

Table 1B (continued): Descriptive Statistics

Notes: Please see Table 1A.

PANEL B: STRUCTURED AND SEMI-STRUCTURED DATA: MEANS FOR KEY VARIABLES

	All Applications	Applications with outcome Information
Total number of members of founding team	2.4	3.0
Number of developers/engineers	1.9	2.1
Share Incorporated at time of application	0.32	0.40
Share that claim to have identified a customer base	0.29	0.26
Total Capital Raised at the time of application	285,756	288,099
Share of equity held by largest shareholder	56%	49%
Runway Left (Months)	11.30	7.33
Do you have a fundraising goal?	0.41	0.50
If so, how much are you looking to raise?	1,305,317	766,706

PANEL C: UNSTRUCTURED DATA: AVERAGE NUMBER OF WORDS IN TEXT-BASED FIELDS

	All Applications	Applications with outcome Information
Describe yourself (applicant)	10.9	10.3
Applicant's Skills	13.9	15.0
Team's Skills	10.0	10.7
Why did you choose the accelerator?	52.1	53.5
How did the founders meet?	61.1	83.6
Will everyone attend the program?	5.2	5.2
Short info about the company	10.1	11.2
What makes the company unique?	18.9	18.8
Describe Problem You are Solving	105.7	105.7
How will you acquire Customers	67.0	67.3
Who are your competitors	152.8	166.1
What is your revenue model	2.6	3.0
Will you work on this project full time?	7.6	9.9
How have you validated the market	75.5	80.3
What progress have you made	69.5	70.6

Table 2: Predicting evaluator scores

This table reports the results from three common machine learning algorithms (OLS, SVR and Random Forest) as well as an Ensemble algorithm that takes the best fit across all three. The goal of all algorithms was to use structured and unstructured data described in Table 1 to predict the average score an application received across all evaluators that scored it (Min=1, Max=4). Panel A reports results on applications with outcome information and Panel B reports results on all applications. For each dataset , we randomly divided the applications into a training sample (80% of the observations, which we use to train the algorithms) and a hold-out sample (20% of the observations which we use to make out-of-sample prediction). We ran 100 iterations of this exercise (that is, 100 different randomly drawn training and hold-out datasets). All models were run on the same 100 training and hold-out samples and the reported performance refers to the average across all 100 runs. We separately report the results of our prediction performance for the training and hold-out samples, both with and without cross-validation. The numbers in brackets in the hold-out sample column are 95 percent bootstrap confidence intervals for hold-out prediction performance.

PANEL A: APPLICATIONS WITH OUTCOME INFORMATION

	Prediction Performance R^2			
	Without cross-validation		*With cross-validation*	
Model used	Training Sample	Hold-out Sample	Training Sample	Hold-out Sample
OLS	50%	42% [30% , 53%]		
SVR	35%	29% [15% , 41%]	41%	36% [24% , 48%]
Random Forest	93%	44% [31%, 54%]	91%	45% [53%, 54%]
Ensemble	75%	43% [30%, 53%]	81%	44% [32%, 54%]

PANEL B: ALL APPLICATIONS

	Prediction Performance R^2			
	Without cross-validation		*With cross-validation*	
Model used	Training Sample	Hold-out Sample	Training Sample	Hold-out Sample
OLS	55%	52% [47% , 57%]		
SVR	54%	46% [49% , 55%]	68%	54% [51% , 57%]
Random Forest	95%	59% [56%, 61%]	97%	60% [58%, 62%]
Ensemble	80%	59% [56%, 61%]	82%	61% [59%, 63%]

Table 3: Comparing outcome of top ranked ventures as chosen by actual evaluators with top ranked ventures as chosen by model trained to mimic evaluators

This table compares the amount raised by ventures ranked highest by evaluators with ventures ranked highest by the model trained to mimic evaluators. We use ventures ranked highest by evaluators as the baseline instead of those actually accepted to the accelerator so that we don't conflate judging with potential adverse selection if the very best ventures choose not to join the accelerator. We use the same 100 randomly drawn test and hold-out samples documented in Table 2 to run the following exercise: for each hold-out sample, we calculate the number of ventures that were actually accepted to the program. For this number X in each of the 100 runs, we take the top X ranked ventures in the hold-out sample based on the actual scores received by evaluators and the top X ranked ventures in the hold-out sample based on the prediction of the ensemble model trained to mimic the scores assigned by evaluators. We calculate the average amount raised across each of these top X ventures, across all 100 runs and report these values below. Amount raised is winsorized at the 99th percentile (~$20 million) for the few ventures who have raised substantially more, to prevent outliers from skewing results. We also report the results of a test for equality between the amount raised and in both panels, cannot reject the hypothesis that the difference is zero.

PANEL A: APPLICATIONS WITH OUTCOME INFORMATION	*Amount Raised*
Ventures ranked the highest by evaluators	2,208,437
Equivelent # of ventures ranked by model trained to mimic evaluators	2,037,459
P-value for the difference	0.18
PANEL B: ALL APPLICATIONS	***Amount Raised Imputed***
Ventures ranked the highest by evaluators	366,354
Equivelent # of ventures ranked by model trained to mimic evaluators	451,099
P-value for the difference	0.29

Table 4: Outcomes for top ranked ventures chosen by model trained to maximize success

This table compares the amount raised by ventures ranked highest by model trained to maximize success, with the ventures ranked highest by the evaluators and with ventures ranked highest by the model trained to mimic evaluators. We use the same 100 randomly drawn test and hold-out samples documented in Table 2 to run the following exercise: for each hold-out sample, we calculate the number of ventures that were actually accepted to the program. For this number X in each of the 100 runs, we take the top X ranked ventures in the hold-out sample based on the actual scores received by evaluators and the top X ranked ventures in the hold-out sample based on the prediction of the ensemble model trained either to mimic the scores assigned by evaluators or to maximize success. We calculate the average amount raised across each of these top X ventures, across all 100 runs and report these values below. Amount raised is winsorized at 99th percentile to prevent outliers from skewing results. We report results for three different versions of the model trained to maximize success. In column 1, we report results where model trained to maximize success is given continuous values of amount raised in the training dataset. In columns (2) and (3), we train the maximize success model to identify either the top 5% of applicants by amount raised (col 2) or the top 5% of applicants by cohort (col 3). For these models, the dependent variable in the training dataset is an indicator and these colums show that the success of this model is not driven by any skewness in the outcome measure. We also report the results of a test for equality between the amount raised and show that there is a statistically significant difference in the performance between the maximize success model compared to both the evaluator picks and the picks based on the model mimicking evaluators.

PANEL A: APPLICATIONS WITH OUTCOME INFORMATION

	Amount Raised		
	(1)	*(2)*	*(3)*
Model trained to maximize success	4,813,809	4,795,369	4,503,133
Model trained to mimic evaluators	2,208,437	2,208,437	2,208,437
Ventures ranked the highest by evaluators	2,037,459	2,037,459	2,037,459
P-value for the difference between:			
(1) Maximize success model and evaluators	<.001	<.001	<.001
(2) 'Maximize success model' and 'mimic evaluators model'	<.001	<.001	<.001

PANEL B: ALL APPLICATIONS

	Amount Raised Imputed		
	(1)	*(2)*	*(3)*
Model trained to maximize success	994,456	697,720	5,019,499
Model trained to mimic evaluators	451,099	451,099	451,099
Ventures ranked the highest by evaluators	366,354	366,354	366,354
P-value for the difference between:			
(1) Maximize success model and evaluators	0.002	0.01	<.001
(2) 'Maximize success model' and 'mimic evaluators model'	0.01	0.01	<.001

Table 5: Overlap in applications picked by model trained to mimic evaluators and model trained to maximize success

This table documents overlap in the picks and outcomes of our two models, both of which were trained on the same 100 training samples and predicted from the same 100 hold-out samples noted in Tables 2, 3, and 4. Panel A reports results for applications with outcome information and Panel B reports results for all applications. Looking across both panels shows that among applications selected by at least one model, the bottom left box has the highest outcomes, followed by the top left box followed by the top right box. This suggests that the model trained to mimic humans is systematically missing some extremely promising applications that the machine maximizing success can identify ex ante. Applications picked by both models are ones where outcomes are towards the lower end of the picks for the machine maximizing success but towards the higher end of the picks for the machine mimicking evaluators.

PANEL A: APPLICATIONS WITH OUTCOME INFORMATION

	Maximize success model picked	Maximize success model did not pick
Mimic evaluators model picked	$3,839,238 [0.4%]	$2,094,849 [6.2%]
Mimic evaluators model did not pick	$4,881,690 [6.2%]	$3,249,010 [87.2%]

PANEL B: ALL APPLICATIONS

	Maximize success model picked	Maximize success model did not pick
Mimic evaluators model picked	$627,397 [0.03%]	$362,619 [1.8%]
Mimic evaluators model did not pick	$999,708 [1.8%]	$199,648 [96.4%]

Table 6: Correlation in Ranks and Correlation in Weights assigned to different attributes across the two models

This table reports the correlation in the ranks (Spearman-Brown Correlation) and correlation in the weights (Pearson Correlation) assigned to different attributes across the two models. For each of the 100 runs of our bootstrap estimations on the training and hold-out sets, we compare the emphasis the two models place on different attributes used to predict the 'top applications' (either in terms of the predicted score assigned by evaluators for the mimic evaluator model or the predicted amount raised for the maximize success model). Column 1 uses an ordinal ranking of the features to create a Spearman-Brown Correlation coefficient. Column 2 uses a cardinal ranking of the features to create a Pearson correlation. Comparing the columns shows that the ranks tend to be quite highly correlated while the weights are quite uncorrelated, particularly for unstructured data. This suggests that one reason the model maximizing success performs better is that humans might be systematically over- or under-weighting certain attributes -- particularly in text-based fields.

PANEL A: APPLICATIONS WITH OUTCOME INFORMATION

	Spearman-Brown Correlation (rank correlation)	Pearson Correlation (correlation of weights)
All Variables	0.31	0.006
Structured and Semi-Structured Data	0.68	0.48
Unstructured Data	0.28	-0.002

PANEL B: ALL APPLICATIONS

	Spearman-Brown Correlation (rank correlation)	Pearson Correlation (correlation of weights)
All Variables	0.44	0.002
Structured and Semi-Structured Data	0.84	0.10
Unstructured Data	0.48	0.004

Table 7: Systematic differences in importance placed on 'more cognitively demanding' variables across the two models

To examine systematic differences across the two models as a way to gain insight into human decision making, we tag sets of variables as having a certain attribute related to them being more cognitively demanding. In column 1, we tag variables as being more cognitively demanding if they were in text fields where the average response was more than 10 words. In column 2, we tag variables as being more cognitively demanding if they appeared in passages of the application that had gunning fog index greater than 7. In column 3, we tag variables as being more cognitively demanding if they appeared as a main feature in less than 15 of the 100 runs, suggesting they were infrequently seen in applications. We take advantage of the fact that total weight assigned to all variables in the random forest model sum to 1. We run an OLS regression where the dataset includes two observations for every variable used in the machine learning models - one with the average weight across 100 runs assigned to the variable in the maximize success model and the second with the average weight across 100 runs assigned to the variable in the mimic evaluator model. We report coefficients for the relevant tag in each column (i.e. whether that variable was coded as cognitively complex using a given measure), an indicator for whether the observation corresponds to the mimic human model and an interaction between the two. Robust standard errors are reported in parentheses, clustered by variable. *, ** and *** refer to significance at the 10%, 5% and 1% level respectively.

PANEL A: APPLICATIONS WITH OUTCOME INFORMATION

	Information Load	Reading Difficulty Index	Uniqueness of Attribute
	(1)	*(2)*	*(3)*
Indicator for High Cognitive Load	0.405	0.000	
	(0.619)	(0.039)	
MHM [Mimic Human model indicator]	0.760	0.134	
	(0.498)	(0.579)	
MHM x Indicator for High Cognitive Load	-2.382***	-0.173***	
	(0.876)	(0.056)	
Number of Observations			

PANEL B: ALL APPLICATIONS

	Information Load	Reading Difficulty Index	Uniqueness of Attribute
Indicator for High Cognitive Load	-0.044	-0.008	
	(0.481)	(0.047)	
MHM [Mimic Human model indicator]	-0.704	1.126*	
	(0.429)	(0.668)	
MHM x Indicator for High Cognitive Load	-1.496**	-0.142**	
	(0.681)	(0.066)	
Number of Observations			