AFRL-RI-RS-TR-2021-184

THE HUMAN DATA INTERACTION PROJECT

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

OCTOBER 2021

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2021-184 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
MICHAEL J. MANNO
Work Unit Manager

/ S /
PATRICK D. SCOTT
Deputy Chief
Intelligence Systems Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
OCTOBER 2021	FINAL TECHNICAL REPORT	MAY 2017 – MAY 2020

4. TITLE AND SUBTITLE

THE HUMAN DATA INTERACTION PROJECT

5a. CONTRACT NUMBER
FA8750-17-2-0126

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62702E

6. AUTHOR(S)

Massachusetts Institute of Technology

5d. PROJECT NUMBER
D3ME

5e. TASK NUMBER
00

5f. WORK UNIT NUMBER
12

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Massachusetts Institute of Technology
77 Massachusetts Ave
Cambridge MA 02139

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIEA
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2021-184

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This project removes costly bottlenecks from an end-to-end data science endeavor, while efficiently redistributing the endeavor's most vital resource – human effort. To accomplish this, we first apply structure and systematically to the otherwise ad hoc processes of transforming (1) temporal, relational, and transactional data into features and labels, (2) features and labels into trained models, and (3) models into insights. We establish that structure in a generalizable, intuitive way so as to encourage participation from a broad user base. And, in so doing, our system creates opportunities to automate, optimize, and learn across disparate data science endeavors.

15. SUBJECT TERMS

Deep Feature Synthesis (DFS), ATM systems, BTB systems, Hyperparameters

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**MICHAEL J. MANNO**
U	U	U	UU	18	19b. TELEPHONE NUMBER (*Include area code*) **N/A**

TABLE OF CONTENTS

1.0 SUMMARY

The past decade has seen both significant widening of big data processing's computational bottlenecks and meaningful improvements to the accessibility of machine learning algorithms. However, manual processes - such as interpreting and preparing data, performing feature engineering, and analyzing models - still take up 90% of a data scientist's overall time. Under this regime, most data scientists spend months getting data ready and only weeks actually using machine learning algorithms.

This project removes costly bottlenecks from an end-to-end data science endeavor, while efficiently redistributing the endeavor's most vital resource – human effort. To accomplish this, we first apply structure and systematically to the otherwise ad hoc processes of transforming (1) temporal, relational, and transactional data into features and labels, (2) features and labels into trained models, and (3) models into insights. We establish that structure in a generalizable, intuitive way so as to encourage participation from a broad user base. And, in so doing, our system creates opportunities to automate, optimize, and learn across disparate data science endeavors.

2.0 INTRODUCTION

Our core objectives are to develop and open source Featuretools - a feature engineering library for temporal, relational, and transactional data - and ATM/BTB/MLBLOCKS, a system for autotuning models. Featuretools implements the Deep Feature Synthesis (DFS) algorithm to automate feature engineering with additional functionality. This serves to (a) improve the data science workflow from development to deployment,(b) enable scaling DFS to big data, and (c) provide a framework to transfer building blocks generated using human intuition from one domain to another.

ATM and BTB are systems which (a) tune models and hyperparameters and (b) take advantage of parallel computing resources. Each of these sub goals, as reflected in our tasks, requires new algorithms to be designed. By focusing on structure, ease of use, API design, and demonstrations on industrial scale datasets, we can deliver a product which brings research from Feature Labs and MIT into widespread use.

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

There are two simultaneous aspects of our work. They are (1) Tool Building: make an open source library with bleeding-edge feature engineering tools which is compliant with the Technical Area APIs and (2) Adoption: organize and run workshops, demonstrations and work with other teams to integrate our software.

(1) Tool Building: An average python user is capable of implementing a machine learning algorithm if the question is sufficiently defined and data is sufficiently prepared. How can we build tools that would allow those users to more easily formulate a question and answer it quickly? Our approach is to provide users with a natural set of feature engineering primitives for commonly repeated tasks in problem formulation and feature engineering.
(2) Adoption: What can we do to make our system easy to use? We provided demonstrations, workshops (pilot workshop already hosted at CMU) and industry-driven examples (2 published), and got feedback from open source - feedback we needed to build an excellent API. We integrated our tools with the D3M's TA1 to make it widely available. .

Throughout our participation in the program we focused on the following items, each of which we will focus in the sections below:

1. **Featuretools – Technical Area 1**: This is our automated feature engineering library that we developed. It implements our algorithm called Deep Feature Synthesis published in 2015. We fully developed Featuretools in the open source, built a community around it, integrated with the TA1 system in DARPA D3M program and added any other primitives as required.

Core development of the Featuretools algorithm base: The Featuretools core consists of Deep Feature Synthesis (DFS) and a granular way of handling time varying and relational data. We made several technological improvements to DFS by improving the output in particular ways, namely, giving the user access to better heuristics and a way to search the space of features generated by the algorithm. During the course of the project we (a) made improvements to the Featuretools API; (b) improved DFS heuritics; (c) intelligent searching in the space of features by project end.

Featuretools primitives: Featuretools ships with a set of feature functions that Deep Feature Synthesis can use to produce feature matrices from raw datasets. The modularity of those primitives gives a low barrier to entry for subject matter experts (SME) to create their own primitives and use Featuretools on their problem. Cultivating and curating a set of those primitives is essential to the success of the project as a whole.

Featuretools demonstrations: We executed several projects which effectively demonstrate the efficacy and generalizability of Featuretools. These add value not only by expanding the user base, but also by helping us to refine Featuretools and add functionality and workflow where it will be most beneficial. In the table below we list the demonstrations we did on a variety of problems. A total of 11 demonstrations have been made. The software for all these demonstrations

are open source and are already in wide use. In most cases, users find the demo that is most pertinent to their need, retrofit the data to present the data in the format that the demo requires, and create and assess a machine learning model for their problem.

Featuretools TA1: We integrated Featuretools with the TA1 package and supported its use throughout different evaluations. In addition to collaborating with a working group to develop the TA1 API, we focused making Featuretools as usable as possible for the other research teams involved in this project. Pragmatically, that means we work one-on-one with other teams to see how our tools can best match their needs. For instance, teams working on dimensionality reduction (e.g generalized low rank models from Cornell team), supervised learning (SVMs from vencorelabs team), or human computer interaction (query builder from Tufts team), will require Featuretools in order to integrate with relational or transactional datasets. This also has the aforementioned benefits of giving us feedback on functionality and workflow. We maintained the Featuretools TA1 primitive throughout our participation in the program and made several submissions at the integration events.

Table 1

Library	Releases	Issues	PR's	Stars	Forks
Featuretools	53	465	773	~5200	678
Predict Next Purchase	Demo	11	9	370	125
Predict Malicious Cyber Connections	Demo	2	4	26	9
Predict Correct Answer	Demo	2	4	25	9
Predict Customer Churn	Demo	8	16	270	155
Predict Loan Repayment	Demo	1	4	42	24
Predict Appointment Noshow	Demo	1	3	20	19
Predict Olympic Medals	Demo	2	4	19	12
Predict Remaining Useful Life	Demo	8	5	135	74
Predict Household Poverty	Demo	0	3	19	12
Automated vs Manual Feature Engineering Comparison	Demo	8	3	274	140
Predict Taxi Trip Duration	Demo	3	5	53	21

Current Featuretools usage and community development: As of date (November, 2020), Featuretools package has been downloaded 1.25Million times and has a download rate of 70,000 per month. Its users have written numerous blogs, presented at many conferences and has become a tool used by many in industry. In our own work, we have used it to solve an industrial scale fraud prediction problem (BBVA bank), predicting delays in project deliveries (Accenture) and many others.

2. **MIT-Technical Area 2 system** - We developed the MIT TA2 system. This system evolved over time. It started with us using a simple system and our tuning library (BTB). As we expanded it to support multiple data modalities (time series, images, audio and others) we built an equivalent of TA1 (as TA1 of the D3M core was still under development) and ultimately integrated it with D3M TA1. Several subsystems resulted from this and are documented in Section below.

We developed several modules to support the MIT TA2 system. When we started the program the MIT TA2 system used inbuilt functions to support different data modalities. At this time, D3M TA1 did not exist. The program's TA1 was under development. For the second and third evaluation to scale and support our TA2, we developed a mini TA1 while the program's TA1 was under development. We call this MLPrimitives. Ultimately we fully integrated and used programs of D3M TA1. Below we list the variety of components we developed, their current status and a list of open source libraries that spun out of our TA2.

MLPrimitives: This repository contains primitive annotations to be used by the MLBlocks library, as well as the necessary Python code to make some of them fully compatible with the MLBlocks API requirements. There is also a collection of custom primitives contributed directly to this library, which either combine third party tools or implement new functionalities from scratch.

MLBlocks: MLBlocks is a simple framework for composing end-to-end tunable Machine Learning Pipelines by seamlessly combining tools from any python library with a simple, common and uniform interface. Features include:

- Build Machine Learning Pipelines combining any Machine Learning Library in Python.
- Access a repository with hundreds of primitives and pipelines ready to be used with little to no python code to write, carefully curated by Machine Learning and Domain experts.
- Extract machine-readable information about which hyperparameters can be tuned and within which ranges, allowing automated integration with Hyperparameter Optimization tools like BTB.
- Complex multi-branch pipelines and DAG configurations, with unlimited number of inputs and outputs per primitive.
- Easy save and load Pipelines using JSON Annotations.

MIT-D3M-TA2: This repository contains the TA2 submission for the Data Driven Discovery of Models (D3M) DARPA program developed by the DAI-Lab and Featuretools teams.

Bayesian tuning and bandits (BTB): BTB ("Bayesian Tuning and Bandits") is a simple, extensible backend for developing auto-tuning systems such as AutoML systems. It provides an easy-to-use interface for *tuning* models and *selecting* between models. It is currently being used in several AutoML systems:

- ATM, a distributed, multi-tenant AutoML system for classifier tuning
- MIT's system for the DARPA Data-driven discovery of models (D3M) program
- AutoBazaar, a flexible, general-purpose AutoML system

Library	Releases	Issues	PR's	Stars	Forks
MLPrimitives	17	135	111	42	30
MLBlocks	20	67	59	77	30
MIT-D3M-TA2	6	14	32	6	4
Autobazaar	3	11	16	18	10
PIEX	3	6	5	11	4
BTB	22	112	100	154	36

Throughout our engagement in the program, we participated in all events, and most dry runs.

TA2-TA3 Integration
We integrated with the several TA3s and at one point were the most used TA2 system in the program.

We integrated with the following TA3 systems for evaluation:

- Purdue
- Tufts
- Harvard
- Brown
- CMU

TA2 submissions
Throughout our participation there were two integration events every year - summer and spring. We took part in both events, submitted our system and got evaluated against multiple systems. As we progressed we improved coverage of our TA2 system to support different data modalities and different task types.

TA1 submission
Our core TA1 submission was Featuretools. We submitted this and updated it several times to conform to the updates in TA1 API as the community evolved. We took part in several discussions around the API as well to inform the unique needs of feature engineering for multi-entity, temporal, relational data.

3. **Other systems**: To support our mission of enhancing data scientists productivity we focused on developing systems like Compose (that enables data labeling), Trane (that allows formulation of prediction problems automatically).

TRANE:
A necessary component of predictive modeling is defining the problem to be solved and extracting historical training examples according to that definition. TRANE is a formal language to define prediction problems over raw, transactional data when the label you want to predict doesn't already exist as a column in the dataset. This is needed to automate a data science process end to end. We developed this language and wrote an interpreter that converted it into executable code that generates labels (and cutoff times). Once data scientists are able to successfully define predictive problems in TRANE and solve them using Featuretools, our next goal will be to automatically synthesize interesting or meaningful prediction problems, solve them, and present them to SMEs.

To date we have finished:
(a) completed an API specification of TRANE
(b) an open source implementation
(c) automatically synthesize meaningful prediction problems
(e) Open source the software framework and documentation.

Trane could be accessed at: https://github.com/HDI-Project/Trane

Compose
We introduced "prediction engineering" as a formal step in the predictive modeling process. We define a generalizable framework that allows data scientists to label data by simply defining a labeling function. This methodological development was to address the growing demand for predictive models. The framework provides abstractions for data scientists to customize the process to unique prediction problems.

4. **Open source**: We have built a prolific ecosystem of open source libraries which we summarize in multiple sections below. Our open source is available for anyone to use and indeed is used by several thousands of users. We also created community oriented infrastructure - slack, stackoverflow, demos tutorials. We applied our open source libraries to numerous industrial scale problems with promising results.

4.0 RESULTS AND DISCUSSION

Over the course of the D3M Program we collaborated with both Technical Area 1 and Technical Area 2 performers to make Featuretools are usable as possible for the other research teams involved with the program. We have worked with teams one-on-one to see how our tools can match their needs early on in the program, while providing feedback on functionality and workflow. The Feature tools API was submitted to the primitive repository, and integrated with other teams.

Our goal was to bring advances in feature engineering and data science workflows to a general audience. Current packages like Scikit-Learn (scikit-learn.org) accomplished this past by standardizing the usage of desperate machine learning implementations for both novice and advanced python users, while other packages such as Pandas (pandas.pydata.org) made data manipulation more intuitive by implementing the innovative DataFrame API that was more accessible than SQL. Featuretools is compatible with these technologies and adopts similar strategies to further reduce the barriers to getting involved in the data science process.

5.0 CONCLUSIONS

While there are existing tools for automated feature extraction for text and image data (eg deep learning methods), there aren't similar tools available for addressing the relational and time varying data that is typical in many predictive modeling use cases.

We will bring this type of automation to everyone with the open source release and continued development of Featuretools. If we are successful in building a tool that is ready for public consumption, there will be an increase in problems being solved with data science by SMEs and substantial savings in time and money saved per-project.

For example, an initial experiment compared the Deep Feature Synthesis algorithm in Featuretools against nearly 1000 data scientists in 3 competitions to model raw relational and time varying datasets. The fully automated result enabled by DFS achieved, on average, 92% of the best human submissions and outperformed 615 teams across the three competitions.

We have worked with 16 industry datasets in the past few years, and were able to beat (or generate) a baseline predictive analytics solution with highly interpretable features in approximately a week.

6.0 REFERENCES

We created an API for users to contribute custom feature functions. Please see:
https://featuretools.alteryx.com/en/stable/getting_started/primitives.html

We released the Compose framework as a github library, now available at:
https://github.com/alteryx/compose

A publication prior to the start of the program describes this methodology:
https://dai.lids.mit.edu/wpcontent/uploads/2017/10/Pred_eng1.pdf

A preprint publication that explains TRANE: MLFriend: Interactive Prediction Task Recommendation for Event-DrivenTime-Series Data.
https://arxiv.org/pdf/1906.12348.pdf

APPENDIX A – PUBLICATIONS AND PRESENTATIONS

List the dates, times, title, event and speakers of any presentations made under this effort and the title author and publication information for any publication made under this effort.

Publication 1

Title Indicate the title of the publication.	The Machine Learning Bazaar: Harnessing the ML Ecosystem for Effective System Development
Author(s) Indicate the authors of the publication.	Micah J. Smith, Carles Sala, James Max Kanter, Kalyan Veeramachaneni
Publication Date Indicate the date of publication.	June 2020
Publication Venue Indicate the journal, Proc. 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD '20) conference, or magazine name.	Proc. 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD '20)
Keywords Enter at least three keywords for the publication.	Machine Learning, ML primitives, AutoML
URL Enter the URL associated with this publication (if any.)	https://arxiv.org/abs/1905.08942
Comments Enter any relevant comments about this publication.	

Publication 2

Title Indicate the title of the publication.	Solving the False Positives Problem in Fraud Prediction Using Automated Feature Engineering
Author(s) Indicate the authors of the publication.	Roy Wedge, James Max Kanter, Kalyan Veeramachaneni, Santiago Moral Rubio, Sergio Iglesias Perez
Publication Date Indicate the date of publication.	2018
Publication Venue Indicate the journal, Proc. 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD '20) conference, or magazine name.	European Conference, ECML PKDD 2018: Machine Learning and Knowledge Discovery in Databases (pp. 372-388)
Keywords Enter at least three keywords for the publication.	automated feature engineering, fraud prediction, machine learning
URL Enter the URL associated with this publication (if any.)	http://www.ecmlpkdd2018.org/wpcontent/uploads/2018/09/567.pdf
Comments Enter any relevant comments about this publication.	

Publication 3

Title Indicate the title of the publication.	Augmenting Software Project Managers with Predictions from Machine Learning
Author(s) Indicate the authors of the publication.	Benjamin Schreck, Nitin John James, Shankar Mallapur, Rajendra Prasad, Sanjeev Vohra, Kalyan Veeramachaneni
Publication Date Indicate the date of publication.	December 2018
Publication Venue Indicate the journal, Proc. 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD '20) conference, or magazine name.	Proceedings of IEEE, International Conference on Big Data
Keywords Enter at least three keywords for the publication.	Predictive models, measurement, software, machine learning, data models, training, tools
URL Enter the URL associated with this publication (if any.)	https://ieeexplore.ieee.org/document/8622586
Comments Enter any relevant comments about this publication.	

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

ATM	Auto Tune Models
BTB	Bayesian Tuning and Bandits
MLBLOCKS	Machine Learning Blocks
DFS	Deep Feature Synthesis
SME	Subject Matter Experts
CMU	Carnegie Mellon University
MIT	Massachusetts Institute of Technology
TA1	Technical Area 1
TA2	Technical Area 2
D3M	Data-Driven Discovery of Models