AFRL-RI-RS-TR-2021-155

SNOWCAT AND CAVA: VISUALIZATION TOOLS FOR INTERACTING WITH AUTOML AND KNOWLEDGEBASES

TUFTS UNIVERSITY

SEPTEMBER 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-2021-155 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
PETER A. JEDRYSIK
Work Unit Manager

/ S /
JULIE BRICHACEK
Chief, Information 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)*
SEPTEMBER 2021	FINAL TECHNICAL REPORT	MAR 2017 – MAR 2021

4. TITLE AND SUBTITLE

SNOWCAT AND CAVA: VISUALIZATION TOOLS FOR INTERACTING WITH AUTOML AND KNOWLEDGEBASES

5a. CONTRACT NUMBER
FA8750-17-2-0107

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62702E

6. AUTHOR(S)

Remco Chang

5d. PROJECT NUMBER
D3MP

5e. TASK NUMBER
S0

5f. WORK UNIT NUMBER
12

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Tufts University
75 Kneeland Street, Suite 950
Boston MA 02111-1906

8. PERFORMING ORGANIZATION REPORT NUMBER

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

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

DARP/I2O
675 N. Randolph St.
Arlington VA 2203-2114

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

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2021-155

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

The goal of our project is to create visualization systems that enable subject matter experts (SMEs) to construct, curate, evaluate, and assess data-centric machine learning models. Our systems allow a SMEs to explore data visually, construct objective functions via intuitive interfaces, and submit them to an AutoML system to generate machine learning models. By integrating visual exploration of input data, model outputs, and visualization results, our systems supports model development, tuning, formalization, validation in an intuitive manner decoupled from the underlying modeling techniques.

15. SUBJECT TERMS

Automated machine learning; visualization; model learning; model comparison; data augmentation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**PETER A. JEDRYSIK**
U	U	U	UU	32	19b. TELEPHONE NUMBER (*Include area code*) **(315) 330-2150**

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

1.0 SUMMARY

Our project is a part of the DARPA Data-Driven Discovery of Models (D3M) program. The goal of our project is to create visualization systems that enable subject matter experts (SMEs) to construct, curate, evaluate, and assess data-centric machine learning models. SMEs have domain expertise in specific areas, often gained through years of experience. However, they are often not experts in computing and data science, and thus unable to capitalize on the power of modern machine learning techniques. We will construct systems that will allow the SMEs to explore data visually, construct objective functions via intuitive interfaces and submit them to an AutoML system, inspect and compare the models returned by AutoML and choose the optimal model for the analysis goal. By integrating visual exploration of input data, model outputs, and validation results, our systems will allow model development, tuning, formalization, validation and documentation in a manner decoupled from the underlying modeling techniques.

Consider an intelligence analyst tasked with using our visualization system to analyze the political climate of a country based on recent news articles. Since she has direct access to a suite of machine learning models through the visualization, she is able to open a list of predefined high-level analysis tasks in the visualization and select one that suits her goal (e.g., “find something unusual”) without having to directly select the right machine learning algorithms or manually choose their parameters. With AutoML, multiple different models and models with different parameters are constructed, and the outputs are visualized such that they can be assessed, compared, and connected back to the original data. This helps the intelligence analyst focus on their task and data without spending their effort in understanding the nuances of machine learning modeling and parameter tuning.

Further, we consider techniques to augment the SME’s capability in analyzing data by augmenting their data with information in knowledgebases. Knowledgebases are capable of storing vast amount of information and data. For example, WikiData is a knowledgebase that encodes relational information of Wikipedia. Unlike Wikipedia that is meant for humans to read, WikiData stores the information in a structured format such that data can be retrieved through formal query languages such as SPARQL. Integrating rich knowledgebases into an SME’s analysis process can help the SME in exploring new hypotheses that would otherwise not be possible.

Technical Approach

There are four components to our proposed visualization system that will enable a subject matter expert (SME) to utilize complex data analysis algorithms through the use of AutoML (see Figure 1). The four components of our system are: (1) Data Augmentation, (2) Data and Problem Exploration, (3) Model Generation, and (4) Model Exploration and Selection.

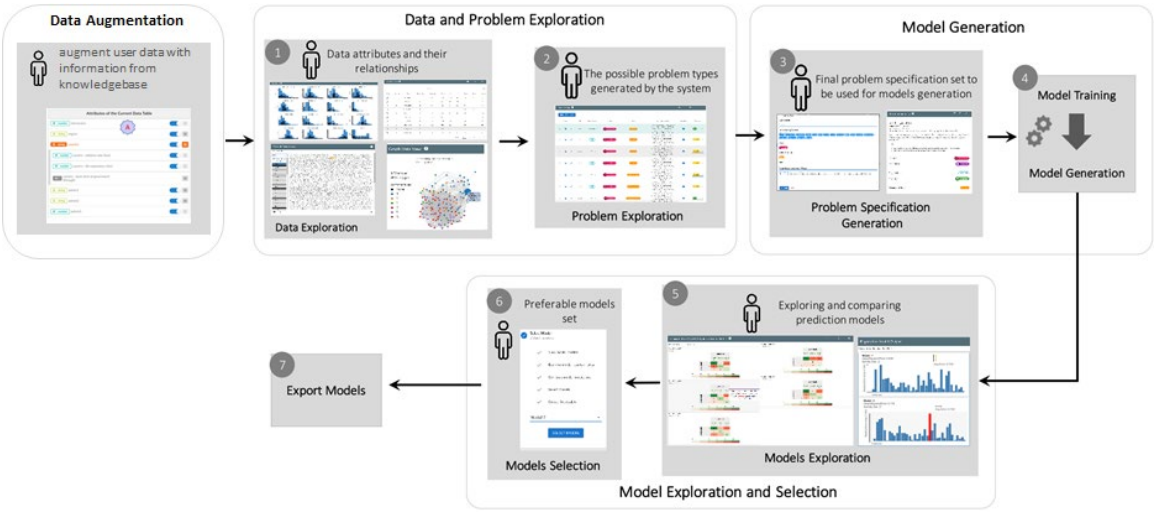

Figure 1: Workflow of the CAVA and Snowcat systems.

(1) Data Augmentation: Machine learning models are only as accurate and useful as the data used to create them. To that end, the first step of a modeling process is to help the SME search, identify, and curate the necessary data for their modeling task. An interactive tool is needed to help an SME augment their initial dataset with additional features derived from a knowledgebase.

(2) Data and Problem Exploration: SMEs have domain expertise but lack data science skills. How can systems support them in creating queries when they do not understand which algorithm to use? Our solution consists of two steps: (a) provide an exploratory visualization interface that allows the SME to inspect the input data, and (b) automatically generate a number of plausible machine learning problems given the data. The SME can refine and update the relevant problem(s) before issuing the tasks to the AutoML system.

(3) Model Generation: We integrate the visualization interface with an AutoML system. Given a problem definition (generated by the SME in the previous step through the use of a visualization interface), our system requests an AutoML system to execute the problem and generate a number of plausible machine learning models.

(4) Model Exploration and Selection: The models generated by the AutoML system will have similar quantitative metrics (such as accuracy, F1-scores, etc.). However, they often differ in qualitative metrics that are relevant to a domain problem. For example, is it plausible that the increase in disease spread follow a piecewise linear fashion or is it more likely for the increase to be quadratic? Towards helping an SME make this determination, our key insight is that the rich data provided in the standard evaluations of model performance provide a rich data set whose exploration will allow an SME to gain insights on the model ensembles in order to perform tasks such as performance prediction, model selection, and trust

characterization. The models generated by AutoML systems will be (in most scenarios) "black boxes" to the SMEs. We, therefore, focus on developing tools that allow the SMEs to perform their tasks using only the input/output pairs that these black boxes generate.

Comparison with Current Technology

Current systems for model construction and curation are designed for data scientists who understand how to interact with the models directly. A data specialist must formalize the question to identify an appropriate approach. They select algorithms and processing pipelines, and manually tune the parameters. Results are then evaluated, potentially leading to further tuning. This approach is inappropriate for SMEs:

- it requires expertise with modeling tools (e.g., R and Python) which involve programming and direct control over algorithmic parameters;
- it requires understanding of the specific machine learning modeling techniques, which is difficult to maintain as the collection of available primitives grows;
- it requires skill with the modeling process, including how to state questions in appropriate forms, how to map questions to modeling pipelines, how to compare many possible models, how to assess results through the design of validation experiments, and how to appropriately document the provenance of the model.

While these interfaces may be usable for data scientists, we need a new user experience for SMEs.

Conclusions

Over the course of the project, we focused on two areas of investigation: (1) design and develop an interactive visualization system that helps a subject matter expert (SME) utilize an AutoML system to generate complex machine learning models without requiring expertise in programming or statistics, and (2) develop a data augmentation tool to help an SME enrich their data with additional information stored in knowledgebases such as WikiData.

We were successful on both fronts. We developed Snowcat as a complete integrated visual analytics system with AutoML. In an evaluation conducted by NIST, Snowcat was found to help SMEs create more accurate machine learning models than the ones generated by AutoML alone and the ones curated (manually) by the SMEs. For data augmentation, we developed CAVA[1] to help SMEs enrich their data. The tool is deployed online and has been found by domain scientists at MITRE to have the capability to help SMEs identify and integrate new data from knowledgebases to answer new analysis questions.

[1] CAVA was originally named Auger. The name was changed to disambiguate from another opensource library in Github.

2.0 INTRODUCTION

Subject matter experts (SMEs) would benefit from using the sophisticated data analysis pipelines that are the purview of practicing data scientists. The overarching goal of this project is to provide data-driven automated assistance to SMEs so that they may produce models as performant as those produced by data scientists. Our role in this research is to provide an interface between the human user (a SME) and machine learning algorithms that generate data analysis pipelines. Within this task, we aim to help the SMEs define their data analysis problem without the need to describe the data analysis primitives being used. We design interactive visualizations to communicate the difference between models and to help the user choose the best model for their task. Lastly, we provide methods of interaction that allow the SME to use their domain knowledge to update and improve the models produced through data augmentation and model refinement.

Existing commercial visual analysis systems such as Tableau and Spotfire allow subject matter experts (SMEs) to visualize their data. However, these systems do not support the SMEs in leveraging the latest machine learning and analytics techniques to perform data analysis. While the visualization of raw data is helpful, without semi-automated analysis, the SME is resorted to manually inspecting their data, which can be tedious and error-prone as the dataset becomes larger and more complex. Conversely, programming languages such as R, Python, and Julia provide a programmer the ability to express data analysis queries. However, the learning curves of these languages are high and may require years of experience before a programmer becomes proficient with them. As such, they are difficult for SMEs who are not familiar with programming or data science to adopt.

The goal of our project is to bridge this gap between the SMEs' analysis needs and the tools that are available to them today. In particular, this project addresses the task of automating the data analysis pipeline by aiding users in selecting complex model pipelines through data-driven automated processes. With our proposed system, a subject matter expert (SME) with little or no background in data science will still be able to perform complex data analysis without understanding the mathematics behind the analysis.

Our tool will provide the interface for the SMEs to express their analysis intent, augment their data, explore the solution models (provided by AutoML systems), and tune and refine the models interactively, all without requiring the SMEs to be have programming skills or deep knowledge about data analysis. In conjunction with advances in AutoML, the resulting system will enable SMEs to utilize cutting-edge machine learning techniques and develop models that can accurately analyze large and complex data.

We envision model development as a continual, iterative process of data exploration and augmentation, user interaction, model selection and refinement, model validation and back to data exploration. In this process, we do not explicitly distinguish raw data from model outputs, but instead treat them as complementary elements that the SME examines, explores, and makes sense of.

With this conceptual formulation, the SME is engaged with the outcomes of the model in a fluid and intuitive manner -- as the SME sees how the model affects the raw data, they can provide direct feedback to the system via the visualization to guide the next step of the computation. Since the intended SME is not presumed to be an expert in data science and machine learning, we do not expect the SME to know, for example, what a "must-link" constraint is, or how to choose the best kernel, or even that clustering is the right kind of model for their needs. Instead, our key contribution is to provide the intuitive abstractions to these complex computational models using interactive visualizations through which the user can effectively utilize a wide range of models without having to directly interact with them.

In the rest of this report, we describe the design and development of two systems toward this goal: Snowcat and CAVA. Snowcat is an end-to-end visual analytics system that helps an SME develop machine learning models without requiring the SME to directly interact with the opaque parameters and programming of AutoML systems (see Figure 1). CAVA is an auxiliary data augmentation tool that can produce rich datasets for Snowcat. CAVA allows an SME to augment their data with knowledge and information stored in knowledgebases such as WikiData – a repository of information in Wikipedia that is queryable using formal languages such as SPARQL. SMEs use CAVA to enrich their data with additional domain knowledge. The resulting augmented data allows for new hypotheses and more accurate predictions in the machine learning models.

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

The goal of this project is to develop visualization interfaces that allows a subject matter expert (SME) to generate machine learning models across different data types and tasks. Under the hood, the visualization connects with an AutoML system for the model generation. Through the use of the visualization, the SME can avoid the need to directly write code to utilize the AutoML or to painstakingly tune opaque parameters to optimize the AutoML system.

In this section, we describe the assumptions we make in developing the two visualization systems toward helping SMEs generate machine learning models: Snowcat and CAVA.

3.1 Assumptions about AutoML Systems and API

AutoML is a broad term used to describe a number of machine learning techniques that can automatically generate or tune machine learning models given some user-specified goal. For example, for clustering images, the traditional approach in machine learning is for a user to: (1) select a clustering algorithm, (2) choose some parameter values, (3) run the clustering algorithm on a training dataset, (4) examine the resulting metrics applied on a testing dataset (accuracy, precision, etc.), and (5) repeat the process until the resulting clusters are satisfactory. In the case of AutoML, a user would instead start by giving example images of the different desired classes (i.e., the training data) and ask AutoML to search for the appropriate algorithm(s) and their corresponding parameter(s) to produce outcome model(s).

Although AutoML systems share the same common goal, their implementations may differ. In addition to using different search algorithms, some AutoML system may return a single most optimal model while others return a number of models that are diverse in algorithms and parameters. In the DARPA D3M program, there is a further requirement that the AutoML system needs to return the data pipeline (i.e., the sequence of operations – referred to as *primitives* – that the resulting model consists of).

To support an SME in exploring data, models, parameters, our visualization system must have close communication to the AutoML system and access to its internal operations. These AutoML systems should generate multiple and diverse models that have similar quantitative metrics for the SME to consider. Further, the AutoML system should be able to provide both quantitative and qualitative information about the models generated. That is, in addition to numeric metrics about the performance of the model, the AutoML system should be able to describe what the models are (e.g., what algorithm the model is using) and the parameters used to construct the model.

The API for communicating with AutoML system is specified by the D3M program. It is relevant to note that this API is only implemented by the AutoML system developed by teams of the D3M program. To our knowledge it is not compatible with other opensource AutoML systems (such as auto-sklearn, TPOT, hyperopt, etc.).

3.2 Methods and Procedures: Snowcat

Snowcat is designed in a modular manner, separated into tasks and subtasks. The specific detail of each module is described in Section 4. At a high level, Snowcat consists of five work components: (1) **task analysis**: understanding the needs of the SMEs, (2) **problem discovery and construction**: given a dataset, generate plausible machine learning tasks, (3) **visualization and interface design**: design and implement a web-based visualization interface, (4) **model assessment, validation, and comparison**: design and implement tools to help an SME inspect and assess models to finally choose the optimal one, (5) **system architecture and scalability**: design the architecture and develop the server-side system with scalability in mind to handle large and complex datasets.

3.3 Methods and Procedures: CAVA

CAVA is a complimentary system to Snowcat that helps the SME augment their data with additional information extracted from a knowledgebase. With CAVA, our goal is to develop an interactive visualization tool for the D3M ecosystem that enables SMEs to build better models through data refinement and augmentation. Our premise is to leverage the SME's domain knowledge by helping them curate new *data features* (e.g., columns in a tabular data) from a collection of datasets to solve a modeling or analysis task.

CAVA can be used independently as a standalone tool for data augmentation, or as an integrated component to Snowcat during the initial data exploration phase (see Figure 1). The initial prototype of CAVA uses WikiData as its knowledgebase, but WikiData can be replaced with other knowledgebases that meet the necessary assumption and criteria (see Section 3.4 below).

The design of CAVA consists of four components, the details of which will be described in Section 4.2. The four components are: (1) **knowledgebase integration**: mechanisms for sending queries to the knowledgebase and parsing the results, (2) **feature engineering**: converting data retrieved from a knowledgebase into a tabular form and appending the result to the SME's original data, (3) **visualization and interface design**: designing a visualization that allows the SME to generate SPARQL queries without programming, and (4) **integration with Snowcat**: system-level integration with the Snowcat system and the rest of the DARPA D3M infrastructure and ecosystem.

3.4 Assumptions about CAVA and Knowledgebases

The use of CAVA assumes an existing knowledgebase that contains relevant information for a given task. For example, a task of predicting food shortage in Ethiopia should have a knowledgebase that might include geographic, weather, farming, population, etc. about Ethiopia. In the case where domain-specific information isn't available, we have found that WikiData can serve as a reasonable proxy for a number of tasks. Although WikiData might not have detailed domain-specific information for all tasks, it contains robust information about a number of topics. The scope of the information contained in WikiData is similar to that of Wikipedia. However, unlike Wikipedia, WikiData is stored in a knowledgebase that is queryable.

One particular feature of WikiData that is utilized in CAVA is its "entity matching" algorithm. This feature of WikiData can return the most similar nodes in the knowledgebase given a string. For example, if the input to the entity matching algorithm is the string "Godfather", WikiData would return a number of plausible nodes, including the movie Godfather, or the secular term referring to a child's guardian. The entity matching feature is not available in all knowledgebases, but its availability is assumed by the CAVA system.

Another assumption CAVA makes about the knowledgebase is the availability of metadata. For example, when querying the size of the state of Massachusetts, the query result would be 27,336. The unit of the number (square kilometers) is typically only available as metadata. CAVA makes use of such metadata to help the user better understand the data presented in the visualization.

4.0 RESULTS AND DISCUSSION

In this section we describe our work on the two systems, Snowcat and CAVA. In Snowcat, we developed a data exploration and analysis visualization system for non-expert subject matter experts (SMEs). Snowcat is integrated with an AutoML system, which together empower SMEs with limited data science skills to make use of machine learning techniques to generate models. As a result, SMEs are be able to make more informed decisions, analyze larger amounts of data, and ultimately enable human-in-the-loop data analysis to scale at the pace of data sensing and storage. In an evaluation independently conducted by NIST, SMEs using Snowcat was found to outperform both AutoML and human experts in undisclosed prediction tasks.

In CAVA we developed an interactive visual analytics tool for the D3M ecosystem that enables SMEs to build better models by augmenting their data. Our premise is that instead of merging datasets for data augmentation, a different approach that can better leverage the SME's domain knowledge is to help the SME extract and curate a new dataset from a knowledgebase to solve a modeling or analysis task. Our evaluations of CAVA found that participants and SMEs were able to use CAVA to find relevant data from a large knowledgebase (WikiData), and the resulting augmented data produce models with higher accuracy when used in conjunction with Snowcat.

4.1 Snowcat

The Snowcat system consists of 5 work modules (task analysis, problem discovery and construction, visualization and interface design, model assessment, validation, and comparison, and system architecture and scalability). We present the results of each effort. Note that research and development for each of the work modules are not always integrated into the Snowcat system. Often the research is published as a standalone prototype. Only when the research finding is appropriate for the goals of DARPA is the research prototype converted to production code that is then integrated into Snowcat.

4.1.1 Task Analysis

This effort involves a human-subject study to better understand the range of data analysis tasks typically performed by a SME. The goal of this work module is to formally document the needs of the SMEs when using machine learning tools. The outcome is used to guide the design and development of the Snowcat system. To complete this work module, we conducted literature survey, user studies, and interviews to develop a list of operations that correspond to data analysis tasks.

Work performed

To gain a better understanding of the processes data scientists employ and what pain points they encounter when determining what analysis their client needs, we performed a structured literature review and conducted semi-structured interviews with 14 data scientists from a variety of data-intensive fields: market research, biomedical research, policy research, and epidemiological and health research. In each field, we sought out professionals who directly interface with clients and either perform data analytics themselves or manage a team of other data scientists.

Results

After analyzing the interviews with the data scientists, we found three common methods data scientists employ to better understand their client's needs that we call *working-backwards*, *probing*, and *recommending*. Each of these methods corresponds to a different level of clarity in the client's need. For example, working backwards serves a client with a high clarity need who can exactly specify their desired analysis outcome. From a well-specified desired outcome, the data scientist can "work backwards" to the appropriate analysis. On the other hand, recommendation serves a client with a low clarity need, who may not know what they are looking for. It consists of the data scientist running a number of different analyses in order to see which results are of most interest to the client.

The combined results from the literature review and the interview studies have been compiled into a paper that was published at EuroVis in 2019 [1]. This paper serves as the design requirement document for the development of the Snowcat system.

4.1.2 Problem Discovery and Construction

We develop methods for automatic discovery of a well-defined "problems" that are formatted and curated for execution by an AutoML system to generate machine learning models. For example, a "problem" can be to classify a number of data points with the goal of maximizing F1-scores using some user-specified attributes in the dataset.

In Snowcat, the system will initially look into the given data set and discover all the possible well-defined problems. User interactions are provided in two ways: the SME can refine a problem by editing a problem that was automatically discovered by Snowcat. Alternatively, the SME can create their own set of problems after exploring the data set. The problem-creation process can be done through interacting with a visualization interface without requiring programming effort by the SME.

Work performed

We designed and developed two software components to accomplish this work module. First, we designed an algorithm to examine an SME's dataset and enumerate all possible problems that can be applied to the dataset given its data

attributes and data properties. These problems are ranked from most relevant to least using a simple heuristic.

Separately, we developed an interactive interface to allow an SME to: (1) inspect the automatically generated problem, (2) refine and edit the problem, or (3) create a new problem manually from scratch. After a problem is curated, the SME can click a button in the interface to initiate the model learning process that is handled by the AutoML system.

Results

The two software components have been integrated into Snowcat. See Figure 1 for how the components fit within the Snowcat workflow. Panel 2 in Figure 1 is an illustration of the visualization interface.

Separately, we investigated alternative ways for an SME to generate "objective functions" for a given task. For example, while maximizing F1-score is a reasonable objective for a task, it might not reflect other considerations by the SME such as the exclusion of certain data points, the annotation of similarity relationships between data points, etc. The result of our research in generating objective functions resulted in a publication in EuroVis 2020 [2]. The work is not integrated into the Snowcat system because the use of the technique requires a specific type of AutoML system that isn't supported in the D3M program.

4.1.3 Visualization and Interface Design

We followed an iterative design process in designing and developing the visualization of Snowcat. The result is two major re-designs based on the feedback from SMEs. The final visualization interface consists of three components: (1) task workflow, (2) data exploration and visualization, and (3) session management.

Task workflow: The most important design change during the revisions is the inclusion of a guided workflow to help the SME progress in their modeling task. Using a "cards" design, the SME can strictly follow the workflow of "data exploration, task selection, model generation, model assessment, and model comparison" (shown in Figure 1), or they can use the "cards" to branch from the default process. For example, during the "task selection" phase, the SME can open the card for data exploration to re-examine the data using a visualization. The modular nature of the cards balances the SME's need for guidance with the flexibility for open-ended exploration.

		Problem Types									
		Classification	Regression	Clustering	Link Prediction	Vertex Nomination	Community Detection	Graph Clustering	Graph Matching	Time Series Forecasting	Collaborative Filtering
Data Types	Tabular	✔	✔	✔	X	X	X	X	X	X	✔
	Graph	✔	✔	✔	✔	✔	✔	✔	✔	X	✔
	Time Series	✔	✔	✔	X	X	X	X	X	✔	✔
	Texts	✔	✔	✔	X	X	X	X	X	X	✔
	Image	✔	✔	✔	X	X	X	X	X	X	✔
	Video	✔	✔	✔	X	X	X	X	X	X	✔
	Audio	✔	✔	✔	X	X	X	X	X	X	✔
	Speech	✔	✔	✔	X	X	X	X	X	X	✔

Figure 2: The tasks and data types that are supported by Snowcat.

Data exploration and visualization: Based on the list of data types specified by D3M, we designed and developed a range of visualizations to support the SME in exploring a variety of data types. The new visualizations that we have developed include visualizations for (1) text data, (2) tabular data, (3) timeseries data, (4) graph data, (5) images, (6) videos, (7) audio, and (8) speech. For text data, we display a list of documents, searchable by content, including highlighting and filtering. For tabular data, we display a coordinated set of barcharts that enables cross-filtering for data exploration. For timeseries data, we display small multiples of line charts for each attribute that varies with time. For graph data, we display node link diagrams with highlighting of predicted edges and nodes. For tabular data, we use feature histograms. For images we display the images grouped by their attributes. A user can select any of the images to enlarge it for further inspection. For video, audio, and speech data, we display side-by-side panels of the data using both a time-series visualization and a player for watching or listening to the raw data. Figure 2 shows the list of data types supported by Snowcat. Figure 3 shows examples of these visualization designs.

Session Management: We added support for *sessions* in the Snowcat system. Prior to the addition of the session management capability, Snowcat was "memory-less" in the sense that SMEs were not able to "go back" to see models generated by an AutoML system under a different variation of the same data set (e.g., after data augmentation) or under a different problem description (e.g., changing the objective from maximizing accuracy to maximizing F1-score). Currently, Snowcat provides two types of sessions: (1) sessions across different problem descriptions (with the same data), and (2) sessions across different datasets (e.g., as a result of performing data augmentation). We implemented a workflow in the visualization to support both types of sessions that allow the SMEs to compare model generated from

different sessions.

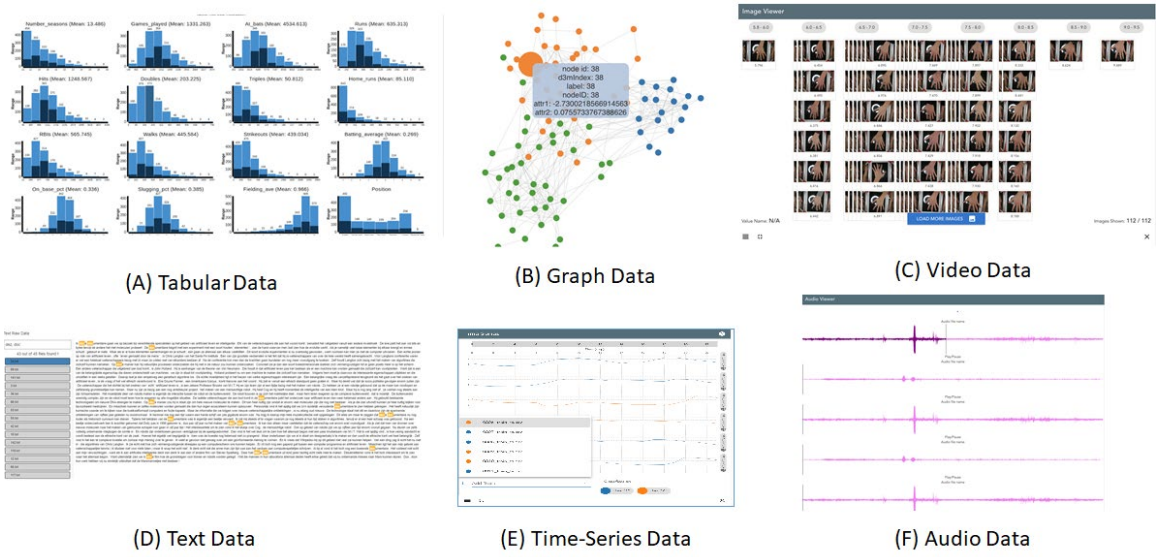

Figure 3: Examples of visualizations supported in Snowcat.

Additional Features: Other features in Snowcat include: (1) model pipeline visualization, (2) "data fact" visualization, and (3) integration with D3M Datamarts and other data augmentation systems. Model pipeline visualization refers to visualizing the procedures, hyperparameters, and parameters of a machine learning model generated by a D3M AutoML system. "Data fact" visualization is a component that visualizes high-level data characteristics along with description of the characteristic in natural language (see Figure 4). This work is based on a prior publication by members of the development team [3]. It is used as an exploratory tool for SMEs who prefer a summarization of an unfamiliar dataset prior to performing data analysis and exploration. Lastly, Snowcat is integrated to support data augmentation. The integration can be with the Datamarts system developed within the D3M ecosystem or the CAVA system that we describe in Section 4.2.

Work performed

The visualization system is developed in Javascript using the VUE.js library such that it can be run in modern browsers. The four components mentioned above are implemented and integrated into a cohesive visualization system that allows an SME to have full view and control over the process of using AutoML to generate machine learning models. Each of the components has been designed with scalability in mind (e.g., through the use of sampling techniques) to ensure fluidity and interactivity during an SME's analysis.

Results

The web-based visualization was completed and integrated with the Snowcat back-end server and D3M's AutoML systems. Aspects of the developed visualizations

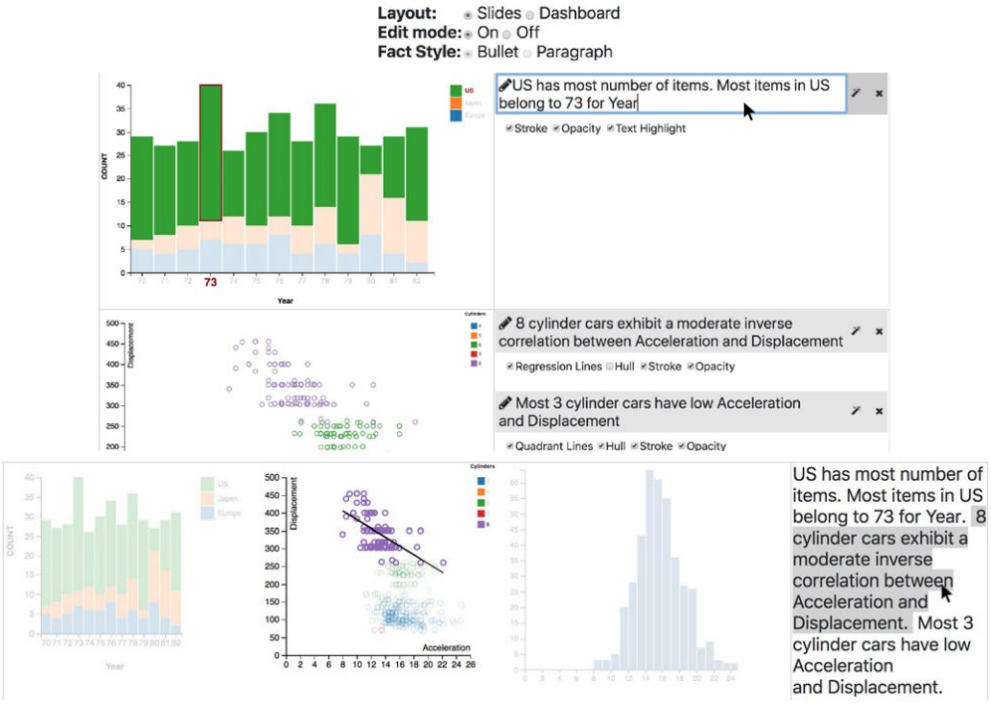

Figure 4: The "data facts" visualization design

were based on ongoing research. Work on the design of multi-class scatterplot visualizations was published in arXiv [4]. Our work on visualizing recurrent neural networks, in particular the vanishing gradients was published in the IEEE CG&A journal in 2018 [5]. Lastly, a technique based on inferential learning that the team developed for evaluating Snowcat was published in an IEEE VIS workshop in 2019 [6].

4.1.4 Model Assessment, Validation, and Comparison

We design the process of model assessment, validation, and comparison using a "guided" approach where the task is broken into multiple stages. The system guides the SME user through the tasks for each stage.

The Snowcat system supports a number of model outputs for machine learning modeling problems, including (1) classification, (2) regression, (3) clustering, (4) link prediction, (5) vertex nomination, (5) community detection, (5) graph clustering, (6) graph matching, (7) time-series forecasting, and (8) collaborative filtering. These different visualizations are integrated into our web visualization framework that take predictions generated by the AutoML system and display them to the user. The visualization system steps an SME through the process of selecting an optimal model for their task. Similar to the visualizations for the different data types, we invested in a modular, card-based interface that helps the SME follow a default workflow while providing the flexibility to allow open exploration. All the visualizations support cross-filtering, for example, between the tabular data and raw data and between the input data and the model output, such that the user can

examine the connection between data and machine learning models. Figure 2 shows the list of machine learning tasks (and the resulting models) supported by Snowcat.

Work performed

The model assessment, validation, and comparison component of Snowcat is implemented in Vue.js using a "cards" metaphor similar to that of the visualization component. The use of the cards allows the two components to be fully integrated such that while an SME is examining the models, they can bring up a visualization card to examine the raw data. Further, because of the cross-filtering mechanics, the SME can select a part of the model (e.g., a number in a confusion matrix) and observe the corresponding data points highlighted in the visualization card.

Results

The model assessment, validation, and comparison component has been completed and fully integrated into Snowcat. Along the way we conducted a number of research studies that led to the final design. First, we completed a research project that uses interactive visual interfaces to help users understand and compare object embeddings, an important data structure that is often used as models of data in intermediary steps of processing pipelines. This model comparison approach is a step towards allowing users to compare entire pipelines that lead up to the final models used for decision making. The result of this work is published in EuroVis 2018 [7] and an extension published in the journal IEEE TVCG in 2020 [8].

Second, we examined neural networks discovered in a neural architecture search. The team worked with external collaborators at Carnegie Mellon and IBM Research in Cambridge to develop visual encodings for convolutional neural network (CNN) architectures to allow hundreds of architectures to all be compared at the same time. We used these new visual encodings to visualize the vast amount of data generated during a neural architecture search. These visualizations were used to compare the discovery process of multiple meta-learning algorithms. The result of this work was published in an ICLR workshop [9] and the journal IEEE TVCG in 2019 [10].

Finally, we developed approaches for analyzing different machine learning models. For analyzing the results of discrete-choice classifiers we developed an approach that enables users to interactively explore experiments run to test different classifiers. The ideas have been built into a prototype system called "Boxer". The result of this work is published in EuroVis 2020 [11]. For regression models, we developed a technique for interactive steering and inspection of multiple regression models. This work was published in the journal IEEE CG&A in 2019 [12].

4.1.5 System Architecture and Scalability

Considerable amount of time and effort were devoted to the design and development of the Snowcat system toward the goals of scalability and modularity. The three unique aspects of the Snowcat architecture are that: (1) it supports a web-based client visualization for different data types and task types, (2) it connects with an AutoML system through a middleware server to request or get the results with

training and testing data, (3) it allows for multiple simultaneous connections from different SME users working on different datasets, and (4) it interoperates data augmentation modules to allow for dynamic, updating datasets.

Work performed

The Snowcat system is developed using a client-server architecture. The client visualization is developed in Javascript to run in modern web browsers. The server is composed of multiple interconnected components. The primary interface with the client is implemented as a Node.js server. User-uploaded data is stored in a Redis database. AutoML is treated as an independent component. Communication between these different components is done using Google's Protobuf message-passing protocol.

To support SME's exploration and analysis of different types of data and types of tasks, we developed an approach where each "card" in the front-end visualization is supported by a corresponding process in the server. For example, the card for visualizing tabular data uses a number of coordinated barcharts. The operation for discretizing the raw data into bins (where each bin corresponds to a bar in a barchart) is executed on the server by a dedicated process. With this architecture design, large amounts of raw data do not need to be send from the server to the client needlessly. The implementation of these card-specific processes is done in a mix of Javascript and Python. Python is specifically chosen because of the abandon libraries for performing machine learning tasks (most notably the scikit-learn library).

Lastly, to support multiple simultaneous client connections, Snowcat uses Node.js threads. Each connection to the Snowcat server is handled by an idle thread taken from a thread pool. The session information for each of connection is stored in the Redis database (along with the user-uploaded data). The information is retrieved by the threads as needed.

Results

The final Snowcat system delivers on all the D3M requirements. It can be integrated with all AutoML systems that implement the specified D3M API. The system has been made into a Docker container such that it can be deployed on most systems. The description of the Snowcat system, including the design process and the evaluation can be found in our paper published at EuroVis 2019 [13].

4.2 CAVA

Our goal for CAVA is to develop an interactive visual analytics tool for the D3M ecosystem that enables a subject matter expert (SME) to build better models by augmenting their data. Figure 5 illustrates our conceptual framework of data augmentation using a knowledgebase. As a result of the data augmentation process, additional columns of data are added to an SME's original dataset (referred to as "seed data" in the figure) with information extracted from a knowledgebase.

To achieve this goal, we leverage the effort of other D3M performers where they have developed a knowledgebase that is composed of relevant datasets to the D3M program. Given such a knowledgebase, CAVA supports an SME to explore, search, and combine information in the knowledgebase and transform it into new data features that can be added to the SME's original data, resulting in a new, augmented dataset. This dataset can then be ingested by Snowcat for generating machine learning models, thus completing a full data workflow.

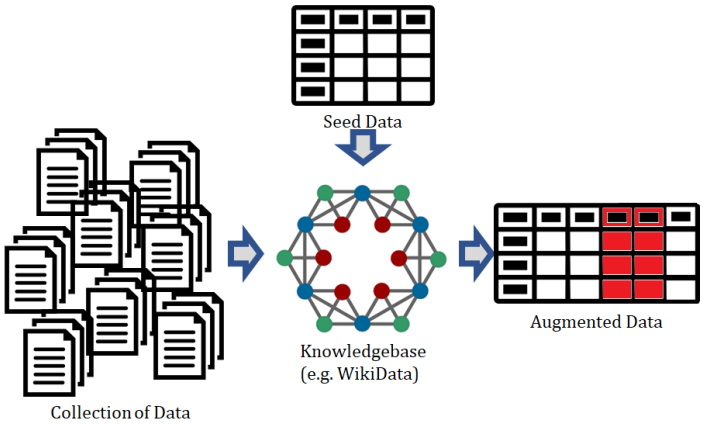

Figure 5: Conceptual framework of data augmentation using a knowledgebase

There are four elements to the CAVA system: (1) knowledgebase integration, (2) feature engineering of knowledgebase into tabular form, (3) design of interactive visual interfaces, and (4) integration with the D3M ecosystem.

4.2.1 Knowledgebase Integration

We consider a knowledgebase to represent a collection of data. The graph structure of a knowledgebase is critical as it captures the data in a structured form, therefore allowing the graph to include data from public sources as well as datasets specific to D3M. The graph in a knowledgebase allows for exploration and connection by following relations. It enables exploration, discovery, and extraction as following relations (i.e., edges) in the graph. The structure enforces semantic and syntactic compatibility of relationships among entities. The challenge is that it requires converting standard data operations, such as query, summarization, and extraction, into graph operations.

We overcome this challenge by defining a set of operational primitives that support the required range of scenarios. For example, we create interfaces that enumerate possible query answers, suggest possible queries that may lead to successful extractions, and assess the data extracted from the queries.

Work performed

We incorporated the WikiData knowledgebase into the CAVA system. CAVA communicates with WikiData via SPARQL queries. As such, CAVA is not bound to using WikiData but can connect with other knowledgebases that support SPARQL

queries. As noted in Section 3.4, CAVA makes some assumptions about the capability of the knowledgebase. In an effort to make CAVA generalizable to a variety of knowledgebases, we reimplemented the "entity resolution" feature that is used extensively by CAVA.

Entity resolution is the process of mapping elements in the user-uploaded data file to objects in the knowledgebase (e.g., matching the string "Massachusetts" from data to the object Q771 in WikiData). In CAVA, we initially made use of WikiData's label-service function (via the WikiData API) to perform this matching operation. As we switched to using other knowledgebases, we implemented our new entity resolution approach that makes use of the knowledge graph's topology. First, we find the "most commonly shared" nodes in the entries in the uploaded data. For example, we find that the strings MA, PA, NM, etc. all share the same common nodes (e.g., "State", "USA", etc.). As such, when an ambiguity arises (such as whether GA refers to Georgia or Gabon), we check which of the two possibilities best match the topology of the other entries.

Results

We have completed the integration of CAVA with WikiData and with knowledgebases created by other D3M teams. Our new entity matching solution has worked well for most test cases but can fail when the uploaded data itself is messy, or if there's limited data in WikiData to establish the "baseline" topology (such as the case of regions in Ethiopia).

4.2.2 Feature Engineering of Knowledgebase into Tabular Form

Once relevant information is found in a knowledgebase, CAVA performs the necessary transformations to convert the information to match the SME's input data in tabular form. In particular, as knowledgebases are commonly represented as semantic graphs, this goal is similar to converting graph information into tabular data.

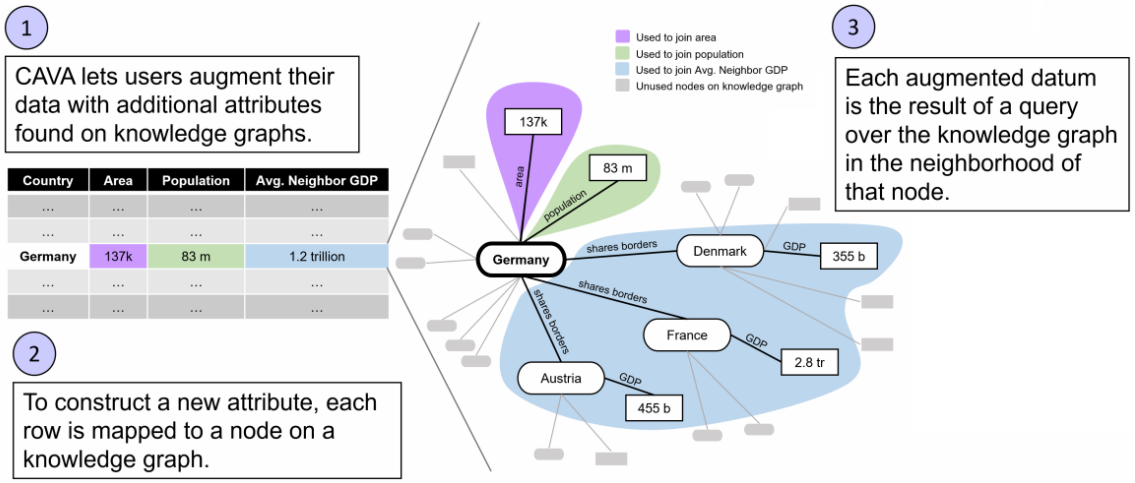

Figure 6: An example of using CAVA for data augmentation

Consider the scenario shown in Figure 6 where an SME wants to augment their country dataset with: Area, Population, and Average neighboring countries' gross domestic product (GDP). First, each row of a dataset (e.g., "Germany") is mapped to an entity in a knowledge graph. The SME can use the knowledge graph to identify the Area and Population information which are connected to the node "Germany". For selecting the Average Neighbor GPD, the SME needs to perform a nested operation to first identify all neighboring countries to Germany ("Denmark", "France", "Austria", etc.), recursively retrieve each of their GDP information, then compute the average of these GDPs.

The above example illustrates the power of using a knowledgebase for data augmentation. The recursive exploration of the graph, combined with the use of algebraic operators, can help an SME to generate complex and nuanced data be added to the SME's original dataset. Without a knowledge structure, these operations would be tedious and difficult for an SME to perform.

Work performed

We are successful in using SPARQL to generate recursive complex queries in CAVA. Through using the visualization (see Section 4.2.3 for more detail), an SME can construct arbitrary complex recursive queries without programming. In addition to the recursive queries, CAVA supports two operations for feature engineering: algebraic operations and operations with temporal data.

First, CAVA supports algebraic operations for a tuple of values. These operations include min, max, count, average, etc. depending on the data types. As shown in the example in Figure 6, these operations can be used in conjunction with the query for data-oriented tasks.

Second, CAVA supports an SME to perform data augmentation with temporal data. The need for curated temporal queries occurs when there are multiple entries of the same data attribute, but recorded at different times. For example, the "population" attribute of the entity "Germany" has many entries, since the population in Germany changes from year to year. To help the SME augment their data with the most pertinent entry, our system supports the SME in choosing from the following options: (1) select an operation (e.g., max, min, average, etc.) over all the entries, (2) select the most recent entry, (3) match the closest entry to a given date (entered by the user), and (4) have the system automatically infer the appropriate entry based on a column in the data. Option (4) is particularly interesting because if the input data contains time-stamp information, our system can retrieve an entry that most closely matches the given time-stamp.

Results

We successfully implemented the query engine, including the generation of SPARQL queries that incorporate both algebraic operations and temporal functions and parsing the results from the knowledgebase. In our testing, we find the query generation robust to a variety of complex, nested, and recursive queries.

4.2.3 Visualization and Interactive Interface Design

We designed the interface of CAVA as a lightweight interactive web-based visualization. To goal of the visualization is to help SMEs express and articulate complex queries outlined in the section above, but without requiring the SME to explicitly write SPARQL queries. In addition, the visualization needs to help the SME build confidence in their data augmentation process, for example by showing the quality of the data to be augmented and providing a preview of the resulting augmented data.

Work performed

We developed an interactive visual interface for SMEs to explore all the data attributes in the original dataset and its relational datasets. Figure 7 shows an overview of the CAVA interface. The view starts with listing all the attributes of the original dataset in one column. For each data attribute, this view provides four functions: “related attribute”, “distribution”, “add”, and “delete”. An SME can click on the “related attribute” button to expand the dataset of the next hierarchy level and list all the related attributes in a new column. By clicking on the “distribution” button, the SME would be able to see the distribution chart of the selected data attribute. If the SME is satisfied with the selected attribute, they can click on the “add” button to add that attribute to the output dataset which will be passed to the next problem stage. On the other hand, if the SME does not want a previously selected attribute, they can use the “delete” button to remove it from the output dataset.

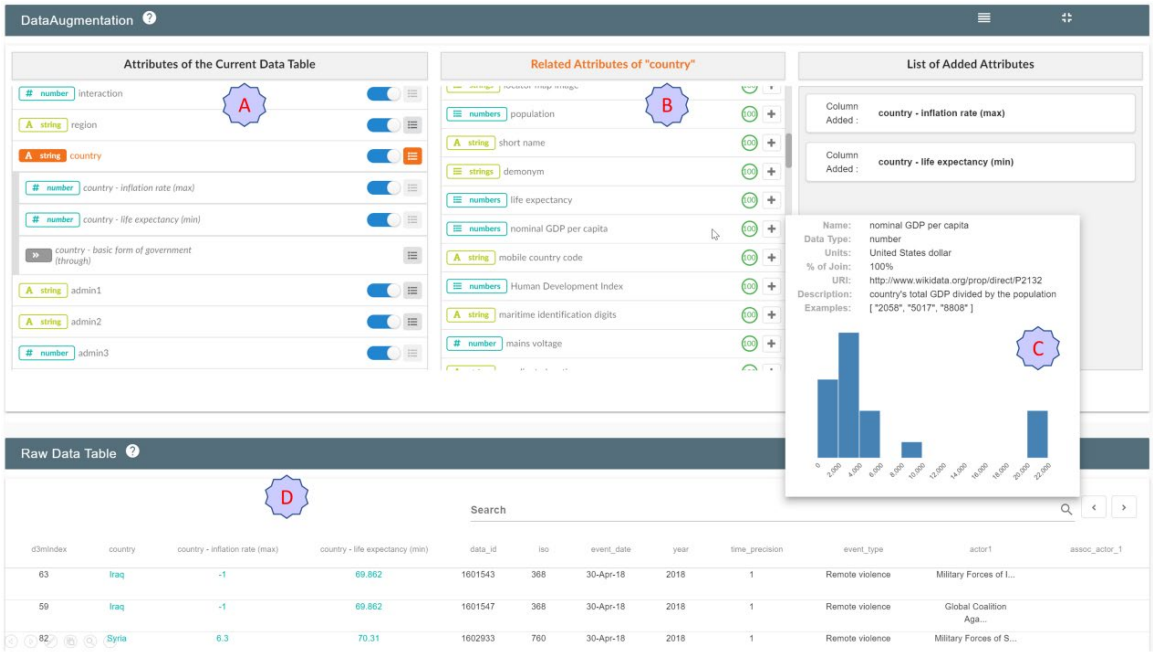

Figure 7: Interface of CAVA

Furthermore, we have developed three visualization interfaces to help the SME with the data augmentation process:

i) Sampling-based data preview: To address the challenge of improving scalability to handle large datasets while minimizing the wait time for the user, we developed a sampling-based approach to provide the user with an approximate preview of the user's data augmentation operations (see Panel D in Figure 7). This "sampling-based preview" extends to a number of places in the visualization, including but not limited to: (1) an estimation of the quality of the data attribute to be included in the augmentation, (2) a visualization of the expected distribution of the data attribute, (3) examples of the resulting rows in the data after the augmentation. Without the sampling-based approach, a full "join" of the complete dataset (that can have millions of rows) could take minutes if not hours to complete. The sampling-based approach reduces that wait time to seconds while providing the user with the intuition needed to make decisions about the augmentation operations and process.

ii) A "join-through" visualization explanation: Since data augmentation can be complex, especially when the data to be joined are 3 or more "hops" away in the knowledge graph, we have observed that these complex augmentation procedures can overwhelm users who might not be familiar with knowledge graphs and data augmentation processes. To help alleviate the difficulties in using our data augmentation tool, we have developed a visualization for explaining these complex augmentation processes. With the use of this visualization, the user gets a preview of the augmentation operations and can make adjustments before committing to the operation.

iii) Support for real-time iterative modeling: Since the purpose of data augmentation in D3M is to improve the quality of the resulting machine learning model, we have added support for direct integration of the modeling tools in our system. Specifically, in our system, the user can quickly check to see if the newly added data feature improves the model (and if not, the feature can be removed). The system keeps track of the iterations of the models (and the data features used to generate the models), allowing the user to see the progression and jump back to a previous model to try a different augmentation idea.

Results

We successfully implemented the web-based visualization interface. In an evaluation with participants recruited at Georgia Tech, participants found the visualization to be intuitive to use, especially in finding relevant data for augmentation. Further, the resulting augmented data are shown to improve model accuracy generated by an AutoML system (see next section, Section 4.2.4 below) when compared to using the original (not augmented) data.

4.2.4 Integration with D3M Ecosystem

CAVA is designed to integrate with the existing D3M ecosystem, including Snowcat. The use of CAVA represents a left most panel in the data workflow shown in Figure 1. With this additional step in the workflow, an SME would first augment the data, specify the problem, and build the models using an AutoML system. If the SME is not

satisfied with the results, the SME would iterate over the process by returning to data augmentation.

In order to support the use of CAVA in the D3M ecosystem, we design CAVA such that it can function as a standalone service that can support multiple simultaneous connections, or used in a dedicated fashion by fully integrating with a system such as Snowcat.

Work performed

CAVA was originally developed with the assumption that it will be integrated in a modeling system similar to Snowcat. However, over the course of the D3M project, we came to realize that such integration is difficult and therefore unlikely due to the reason that other D3M modeling systems make different assumptions about the programming language, the data workflow, etc. As such, the final implementation of CAVA is a standalone system in that CAVA outputs a dataset that other D3M modeling systems would ingest. In particular, towards the integration of CAVA with the rest of the D3M ecosystem, we developed the following features

i) Upload and Download of CSV Files: A key feature of CAVA is the ability for an SME to upload (and download) their own data as a CSV file. Although this feature might seem trivial, there are a number of important innovations that we have implemented to support this goal. First, as the user uploads their own CSV file, our system provides the user with the opportunity to "correct" the data. For example, if a column in the data that represents zipcode is mis-classified as a number, the user can manually choose for the zipcode to be treated like a unique identifier (or a string). As a string, the system can then look into WikiData to find the entry. In contrast, if left uncorrected, a zipcode like 00155 will be considered as the number 155, which would not be meaningful to perform augmentation with.

   Second, to support the downloading the data after augmentation, CAVA includes a number of system optimizations to improve speed and performance. As noted, CAVA uses a "sampling" based approach to ensure responsiveness from the server (especially when the size of the data can be large with millions of records). For the downloading process, CAVA only materializes the full dataset by executing the augmentation queries against all rows in the data upon the SME's request to download the full dataset.

ii) Deployment on D3M Servers: CAVA is currently deployed on the D3M server. Without specifying a domain-specific knowledgebase, CAVA connects to the public WikiData.org website to perform data augmentation. One benefit of using the public WikiData.org knowledgebase is that WikiData.org continues to grow in size (and is constantly maintained and updated to correct for errors and mistakes). As such, an SME can take advantage of "update to date" information.

iii) Support of Multiple Simultaneous Users: CAVA is designed to support multiple SME users simultaneously using CAVA to augment their own datasets. To allow multiple simultaneous users, the CAVA server is built with Node.js for efficiency and uses a multi-threaded architecture. In addition, in order to support SMEs in uploading their own CSV files that could be gigabytes in size, CAVA makes use of Redis as a data management engine.

Results

The complete CAVA system is deployed both on D3M's cluster as well as on Tufts University's server. The system has been evaluated both with participants recruited from a university as well as with SMEs at MITRE. In both cases, CAVA is found to be useful, especially when a problem-specific knowledgebase is available for a task. WikiData can fulfill some of the SMEs' needs. However, for domain-specific datasets and tasks, WikiData might not always have the necessary data for the SME to perform the desired augmentation.

The design, implementation, and evaluation of CAVA has been published in the journal IEEE TVCG in 2020 [14].

5.0 CONCLUSIONS

We successfully developed two systems as part of the D3M program. First, we developed the Snowcat system that allows a subject matter expert (SME) to make use of an automated machine learning (AutoML) system to generate machine learning models. Snowcat is designed to be easy to use without requiring the SME to have knowledge or skill in machine learning, statistics, or programming. In an independent evaluation conducted by NIST, SMEs using Snowcat were able to create machine learning models that were more accurate than models generated automatically by AutoML (without SME intervention) and models that were manually curated by the domain experts.

Second, we developed the CAVA system to help the SMEs augment their input data. Since the machine learning models can only be as accurate as the data that were used to train the models, appropriate data augmentation can significantly improve the outcome of the machine learning models. CAVA uses knowledgebases as the source of augmentation. Through interacting with a web-based visual interface, SMEs can perform complex data augmentation operations without programming or writing explicit database queries. In evaluations with participants recruited at a university and with SMEs from MITRE, CAVA was found be useful in helping the SME identify relevant data for augmentation. Further, machine models generated from augmented datasets using CAVA were more accurate than models generated from the original (not augmented) data.

Source code to both systems are available on Github as open source projects. The two systems are also available as Docker containers for easy deployment.

6.0 REFERENCES

[1] Mosca, A., Robinson, S., Clarke, M., Redelmeier, R., Coates, S., Cashman, D. and Chang, R., "Defining an Analysis: A Study of Client-Facing Data Scientists". *EuroVis (Short Papers).* pp. 73-77, 2019.

[2] Das, S., Xu, S., Gleicher, M., Chang, R. and Endert, A., "QUESTO: Interactive construction of objective functions for classification tasks". *Computer Graphics Forum*, Vol. 39, No. 3, pp. 153-165, 2020.

[3] Srinivasan, A., Drucker, S.M., Endert, A. and Stasko, J., "Augmenting visualizations with interactive data facts to facilitate interpretation and communication". *IEEE transactions on visualization and computer graphics*, Vol. 25, No. 1, pp.672-681. 2019.

[4] Heimerl, F., Chang, C.C., Sarikaya, A. and Gleicher, M., "Visual designs for binned aggregation of multi-class scatterplots". *arXiv preprint arXiv:1810.02445*. 2018.

[5] Cashman, D., Patterson, G., Mosca, A., Watts, N., Robinson, S. and Chang, R., "RNNbow: Visualizing learning via backpropagation gradients in RNNs". *IEEE Computer Graphics and Applications*, Vol. 38, No. 6, pp.39-50. 2018.

[6] Cashman, D., Wu, Y., Chang, R. and Ottley, A., "Inferential tasks as a data-rich evaluation method for visualization". *EVIVA-ML: IEEE VIS Workshop on EValuation of Interactive VisuAl Machine Learning systems* (Vol. 7). 2019.

[7] Heimerl, F. and Gleicher, M., "Interactive analysis of word vector embeddings". *Computer Graphics Forum*, Vol. 37, No. 3, pp. 253-265. 2018.

[8] Heimerl, F., Kralj, C., Moller, T. and Gleicher, M., "embComp: Visual Interactive Comparison of Vector Embeddings". *IEEE Transactions on Visualization and Computer Graphics*, doi: 10.1109/TVCG.2020.3045918.

[9] Cashman, D., Perer, A. and Strobelt, H., "Mast: A tool for visualizing CNN model architecture searches". *ICLR 2019 Debugging Machine Learning Models Workshop*. 2019.

[10] Cashman, D., Perer, A., Chang, R. and Strobelt, H., "Ablate, variate, and contemplate: Visual analytics for discovering neural architectures". *IEEE transactions on visualization and computer graphics*, Vol. 26, No. 1, pp. 863-873. 2019.

[11] Gleicher, M., Barve, A., Yu, X. and Heimerl, F., "Boxer: Interactive comparison of classifier results". *Computer Graphics Forum*, Vol. 39, No. 3, pp. 181-193. 2020.

[12] Das, S., Cashman, D., Chang, R. and Endert, A., "Beames: Interactive multimodel steering, selection, and inspection for regression tasks". *IEEE computer graphics and applications*, Vol. 39, No. 5, pp.20-32. 2019.

[13] Cashman, D., Humayoun, S.R., Heimerl, F., Park, K., Das, S., Thompson, J., Saket, B., Mosca, A., Stasko, J., Endert, A., Gleicher, M., and Chang, R., "A User-based Visual Analytics Workflow for Exploratory Model Analysis". *Computer Graphics Forum*, Vol. 38, No. 3, pp. 185-199. 2019.

[14] Cashman, D., Xu, S., Das, S., Heimerl, F., Liu, C., Humayoun, S.R., Gleicher, M., Endert, A. and Chang, R., "CAVA: A Visual Analytics System for Exploratory Columnar Data Augmentation Using Knowledge Graphs". *IEEE Transactions on Visualization and Computer Graphics*, Vol. 27, No. 2, pp.1731-1741. 2020.

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

API	Application Programming Interface
AutoML	Automated Machine Learning
CG&A	Computer Graphics and Applications
CNN	Convolutional Neural Network
CSV	Comma-Separated Values
D3M	Data-Driven Discovery of Models
DARPA	Defense Advanced Research Projects Agency
GDP	Gross Domestic Product
ICLR	International Conference on Learning Representations
IEEE	Institute of Electrical and Electronics Engineers
ML	Machine Learning
NIST	National Institute of Standards and Technology
SME	Subject Matter Expert
SPARQL	SPARQL Protocol and RDF Query Language
TVCG	Transactions on Visualization and Computer Graphics