In: 2023 27th International Conference Information Visualisation (IV) 2023 Jul 25 (pp. 316-322). IEEE.

No-Code Platform for Visual Knowledge Discovering in General Line Coordinates: DV 2.0

Lincoln Huber
Dept. of Computer Science
Central Washington University
Ellensburg, WA 98926, USA
HuberLi@cwu.edu

Boris Kovalerchuk
Dept. of Computer Science
Central Washington University
Ellensburg, WA 98926, USA
BorisK@cwu.edu

Abstract—DV 2.0 is a no-code platform contributing to interpretable machine learning via Visual Knowledge Discovery in General Line Coordinates (GLC). GLC-Linear (GLC-L), a type of GLC, provides interactive tools to enhance the model creation process and create interpretable visualizations. DV 2.0 uses these techniques so end-users can directly involve themselves in the model creation pipeline. GLC-L visualizations are easily understood by end-users and allow for complex multidimensional relationships to be visualized. Additionally, using GLC-L allows both classification and regression models to be visualized with high degrees of accuracy. For classification, DV 2.0 also offers additional expansions of GLC-L such as GLC non-Linear (GLC-nL) and GLC Hyperblock Rules Linear (GLC-HBRL) to provide different interpretable non-linear classifiers for more complex data. DV 2.0 was tested on benchmark data from the UCI Machine Learning repository.

Keywords—Interpretable machine learning, general line coordinates, visual knowledge discovery, human-guided, machine learning.

I. Introduction

The goal of **Data-Driven Discovery of Models** (D3M) and automated machine learning (AutoML) tools [1-8] is automated production of machine learning models that allow users with **subject matter knowledge** but **no data science background** to build empirical models of complicated processes on their own without the assistance of data scientists [1].

The philosophy and methodology of D3M and AutoML is presented by the interactive paradigm of **human-guided machine learning** (HGML) [8]. It consists of **intuitive Data Exploration (DE)**, **Model Discovery (MD),** and **Model Interpretation (MI)** and analysis. In this paradigm:

- The domain experts interactively convey (a) research **questions** and (b) **knowledge** to the system on data to guide the automated generation of **AI assistance for data analysis.**

- The intelligent back-end programs **automatically** seek (c) interesting **data relationships** and (d) builds ML predictive **models**.

Multiple D3M and AutoML systems have been created [1-8] with focus on:
(1) supporting interactive **exploratory modeling data analysis,**
(2) automation of production of machine learning models based on existing **analytical** ML algorithms, and
(3) analysis and interpretation of the models.

These works produced several **end-to-end Machine Learning pipelines** [2-8] and an interactive visualization tool to explore and compare the solution space of such ML pipelines [5].

The design of the AutoML systems assumes [1]:

- Basic **building blocks** for complex modeling pipelines.
- Automated **composition** of complex models based on user-specified data and outcomes of interest.
- Human-model interaction for **Subject Matter Experts** (SMEs) to **define** modeling problems and **curate** models.

The D3M ecosystem is presented in [7] with description of how to add datasets, problems, primitives, and pipelines using the D3M standard JSON schemas. This allows to activate the AutoML engines to **auto-solve** new problems by using existing and new primitives. The goal is building and executing ML pipelines automatically from primitives using the task formulation produced with end-user's interfaces.

While D3M and AutoML are intended to help end users who are not data scientists to build models as a **self-service**, in fact they still do not fulfill this goal fully. These visualizations for **exploratory data analysis are limited** by 1-D or 2-D data projections and lossy visualizations of n-D data not allowing the full analyses of n-D data.

One of the goals of the exploratory stage, where the domain expert has a key role, is selecting the right attribute predictors and the right target attribute. The simplest implementation in many systems is a user interface where a domain expert can select appropriate attributes from those available for a dataset. Tanagra [17] is one of such systems.

This assumes that the domain expert already has enough knowledge to go through this simple UI operation. Often this is not the case. Therefore, several systems offer ways to explore

the relation between a particular predictor and a target attribute to help to make decisions. This helps to select some attributes.

However, it is very limited. Many attributes have complex **multidimensional relations** with other attributes and the target attribute, which need to be explored. This is often supported by scatter plots for 2-D and 3-D relations, but lossless scatter plots for 4 and more dimensional relations do not exist.

This analysis shows severe limitations in the common ways to support exploratory data analysis in existing D3M and AutoML systems and requires **new approaches** and **enhancement** of the existing systems in support of exploratory data analysis. Thus, the claims that a system supports exploratory data analysis should not be taken for granted without deep analysis of what kind of support it provides.

The exploratory data analysis task is much more complex when the data are contained in multiple files, in different data structures, and locations. These data need to be organized to a uniform structure required for machine learning. If all this information is already available, then respective tools can help automate this process.

In the TwoRavens system proposed in [8], users can browse openly available event datasets, construct queries to select types of events and sets of actors, view and download resulting data, and export data for AI assisted analysis. The major benefit of domain experts involved in this task is pointing out where the data are **located** and explaining the **structure** and **types** of those complicated data. People without deep domain knowledge often cannot do this.

Next, in AutoML the ML models are built and optimized automatically. The users are not involved in this process and cannot judge the internals of this hidden process. Therefore, the user can get trust of the model only on the next stages of task formulation and analysis and interpretation of the model post factum. Thus, the AutoML pipeline provides a **limited support** for **discovering** more trustable models only by changing input to the automatic model building and optimization process.

This can include changing parameters of the algorithms and adding/removing data and attributes from input data, e.g., in [8] domain experts provide interesting features, which is considered as sufficient for the domain expert to remain central to the task. While it is sufficient to keep the central role of the domain expert in this paradigm, it is not sufficient to meet the challenge of interpreting black box ML models.

The full automation of this stage indirectly assumes that domain experts should trust this automatic process by default. While it can be reasonable for some interpretable models like decision trees, for most other models where interpretability is in question it cannot be assumed.

The **model interpretation** and analysis stage is also quite **limited** because it heavily depends on abilities to **fully visualize** the produced models for the end user who cannot analyze the mathematical properties of the model professionally. Also, the automatic model discovery process limits the abilities of the end users to analyze the resulting model because the process of its construction is hidden for them.

The motivation for the automatic model discovery process was that these users are not data scientists and cannot judge internals of the model design process, so they should be removed from this stage. The problem with this approach is that it reduces the benefits and motivation of the interactive paradigm of **human-guided machine learning** with the end users designing ML models as a self-service. The major benefit of this paradigm is that end users bring deep domain knowledge to this process, whereas AutoML **excludes** this from the actual critical model **discovery** stage.

The **visual knowledge discovery methodology** [9,10] removes this limitation, allowing the end user to conduct all three stages themselves including building the models visually in lossless visualizations on n-D data.

It allows end users to incorporate their implicit and explicit domain knowledge more completely in this **full 2D machine learning** (**Full 2D ML**) paradigm [10,11]. This is possible because the end users work interactively with n-D data without loss of n-D information in 2-D visualization space as was demonstrated in [11-16].

DV 2.0 presented in this paper is one such Full 2D ML system designed specifically to enhance Machine Learning on multidimensional data to create ML models with both high performance and explainability. By using visualization methods, DV 2.0 answers the need for both efficient and interactive tools among domain experts.

II. Design of Data-Driven Model Discovery Systems

Many scientists have declared the fundamental role that images played in their most creative thinking. Albert Einstein sums up this idea with: "the words or the language, as they are written or spoken, do not seem to play any role in my mechanism of thought."

Figs. 1 and 2 show the design of data-driven model discovery systems presented in [1,4] and Fig. 3 shows the design and process of the DV 2.0 system. The major difference of DV 2.0 from these systems is in the reliance on full 2D machine learning in lossless General Line Coordinates visualization space and in producing interpretable rules based on hyperblocks (n-dimensional rectangles).

A hyperblock (HB) is a set of n-D points $\{\boldsymbol{x} = (x_1, x_2, \ldots, x_n)\}$ with a center $\boldsymbol{c} = (c_1, c_2, \ldots, c_n)$ and side lengths $\boldsymbol{L} = (L_1, L_2, \ldots, L_n)$ such that,

$$\forall i \in N, |x_i - c_i| \leq \frac{L_i}{2}.$$

Examples of HBs can be seen in Figs. 12 and 13.

The DV 2.0 system allows observations of multidimensional relations within each n-D point and among n-D points. It visualizes these data and relations losslessly in GLC-L and

DSC2 coordinates. It also allows to simplify visual representations of n-D points for observing multidimensional relations between data points. All these allows discovering interpretable machine learning predictive models interactively in this visualization space.

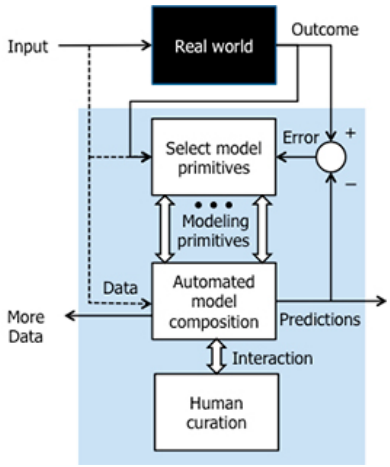

Fig. 1. Data-Driven Discovery of Models (D3M) process from [1].

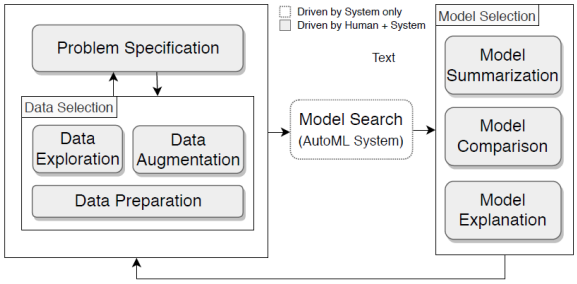

Fig. 2. D3M process from [4].

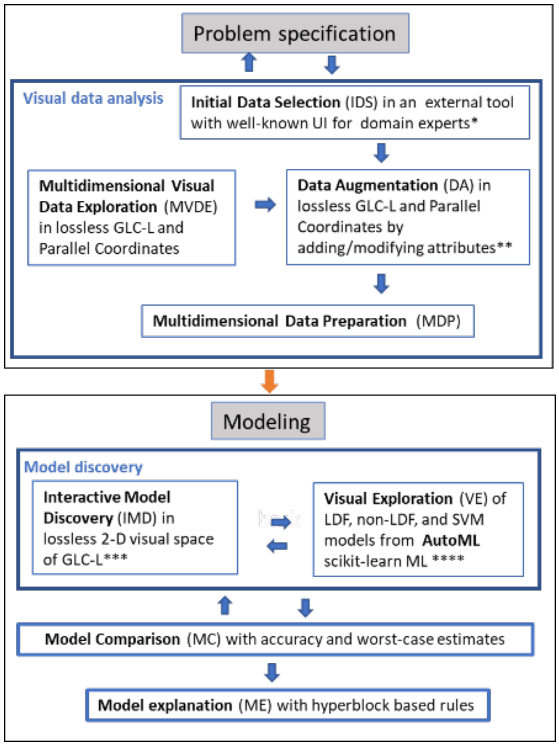

Fig. 3. General design and process of the DV2 system.

* DV 2.0 relies on Excel. Several ML systems like Tanagra are built as Excel Adds-in for this reason.

** A user can add vertical separation attribute for pairs of hyperblocks, reverse attributes, create a new attribute that capture a common visual direction or location of several attributes to simplify visual patterns (see examples in [9]).

*** IMD is conducted by adjusting datasets, overlaps attribute angles, and thresholds.

**** These scikit-learn ML programs from [18] are integrated to DV 2.0.

DV 2.0 differs from other **lossless** multidimensional visualization software in various aspects. npGLC-Vis [20,21], a 2D non-paired General Line Coordinate (2D-NP-GLC) software, offers lossless visualization like DV 2.0, but lacks the same ability in classification.

This is because the npGLC-Vis software focuses solely on visualization of non-paired GLC, whereas DV 2.0 uses **Machine Learning** (ML) techniques to enhance its visualizations. npGLC-Vis also does not have a fully interactive user interface for its visualizations and instead requires end users to have some knowledge in programming to create visualizations.

Other software such as the "Three Experts" [22] are like DV 2.0 in that this software is designed to aid end users in understanding n-D data. "Three Experts" provides a highly intuitive user interface and many analytics beyond the base Principal Component Analysis (PCA),

Independent Component Analysis (ICA), and Multi-Dimensional Scaling (MDS) visualizations. Where DV 2.0 differs is that the "Three Experts" software visualization techniques are **not lossless**. Instead, the "Three Experts" operate by reducing the dimensions of data until they can be visualized with standard 2D means. DV 2.0's fully lossless methodology creates visualizations faithful to the original data.

Another software based on the lossless parallel coordinates [23] is comparable to DV 2.0 in its highly interactive nature but differs in its features. While [23] offers visualization techniques such as attribute reordering and statistical coloring with parallel coordinates, DV 2.0 offers comparable techniques.

For instance, it is common to order attributes by contribution in DV 2.0 by using each attributes angle weight to order the attributes from least to most contributing. This process can also be enhanced with Decision Trees (DT) to create highly interpretable visualizations. Additionally, while statistical coloring can be used to compare attributes to one another in [23], the angle weights in DV 2.0 can be used similarly.

XDAT [24] and Pandas PC plot method [25] are an alternative software based on the lossless parallel coordinates (PC). DV 2.0 differs from them by offering many classification and regression features outside of the visualizations provided by these systems. Also, they are limited to PC while DV 2.0 includes PC, GLC-L, and DSC2.

Figs. 4, 5, and 6 shows these visualizations techniques with the Wisconsin Breast Cancer dataset.

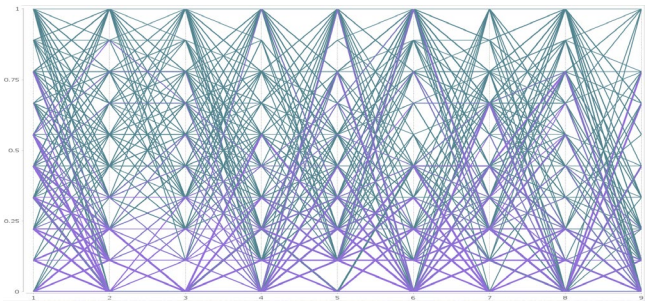

Fig. 4. Visualization using DV 2.0 in Paralel Coodinates.

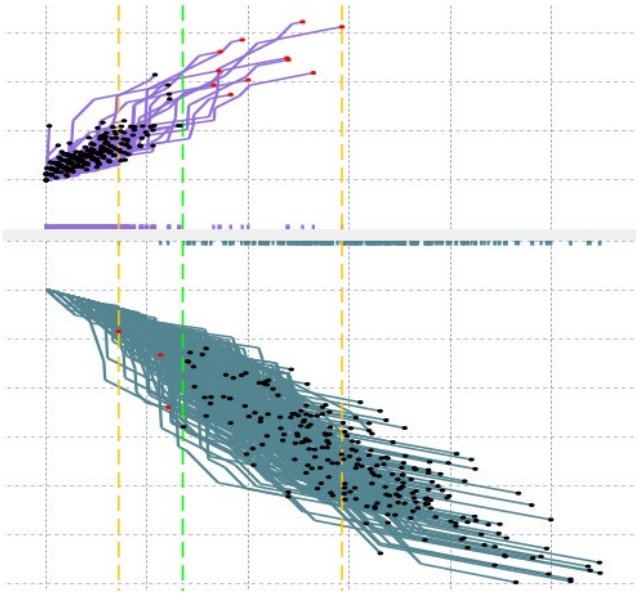

Fig. 5. Visualization using DV 2.0 in GLC-L.

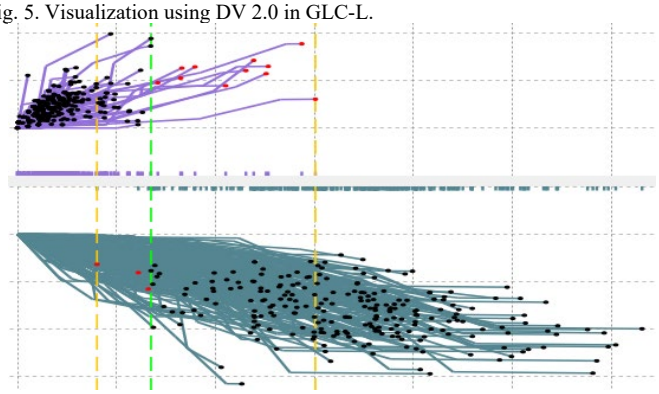

Fig. 6. Visualization using DV 2.0 in DSC2.

Visus [4] is a software which seeks to provide end users a way to enhance Automatic Machine Learning (AutoML) methods with Human Guided Machine Learning (HGML). DV 2.0 differs from Visus in that instead of focusing solely on the human guided part of machine learning task formulation (ML) like selecting predictor and target attributes by analyzing partial (pair) dependencies graphically, DV 2.0 seeks instead to offer end users a way to visually interpret and analyze both input n-D data fully and the **result** of ML systems.

Visus does not offer such detailed and lossless visualization techniques and instead offers end users a way to understand AutoML systems by allowing them to guide the formation of such systems by interactively formulating prediction problems, performing data augmentation, and exploring and comparing models.

Another visualization software is DataShiftExplorer [26]. It is a visual analytics technique which visualizes the data shift between two multidimensional distributions. DataShiftExplorer can have a different form. Typically, it is required to observe data over a period to see the dynamics of the data change. However, for many datasets this is not possible. For example, for geological data waiting for new data will require centuries and millennia.

For disease distribution data we also often cannot wait because of the potential harm. Therefore, in such situations the practical approach is to use available data and simulate with that data shift. This means selecting a subset of data for example around the tails or middle of the distribution or around the area where the data are heavily overlapped. This is the exact opportunity DV 2.0 provides with the overlap area selection slider.

So, the difference between DataShiftExplorer and DV 2.0 is that DataShiftExplorer assumes two independent data sets are given and can be compared whereas in DV 2.0 we are splitting existing data sets to subsets and analyze the accuracy of models built on the different subsets.

Lastly, XAutoML [6] and PipelineProfiler [5] are visualization software designed to explain black box Automatic Machine Learning (AutoML) methods. Both software systems visualize and explain AutoML pipeline systems by creating visualizations which compare different Machine Learning (ML) pipelines against one another to see which perform best. Each of these ML pipelines are also visually presented to show how they achieve their classifications.

In comparison, DV 2.0 forgoes explaining exactly how ML models reach their classifications and instead visually shows how data looks when visualized with the weights given by a ML model. Along with its interactive nature, DV 2.0 allows end users to completely understand and control any ML model visualization.

III. Features

A. DV 2.0 Classification

Two class classification is a primary feature of DV 2.0. After loading a dataset into DV 2.0 and selecting the classification option, Linear Discriminant Analysis (LDA) is automatically performed to compute the optimal coefficient for each attribute in the dataset. These coefficients are then turned into angles for each attribute in a GLC-L visualization. Then, a horizontal linear separation threshold is automatically computed to help classify the data on a single dimension.

At this point a user can begin interacting with the visualization. Both the attribute *angles* and the separation *threshold* can be adjusted by the user to *guide* classifications. This gives domain experts a chance to bring deep domain knowledge into the ML model creation process.

Otherwise, angles can be *automatically* optimized by a search algorithm. Data a*ttributes* can also be reordered and rearranged to enhance the visualization. This can be done automatically through a Decision Tree (DT) or angle contribution, or manually. Fig. 7 shows a pipeline of how a user would interact with this classification feature.

Many datasets are not linearly separable and therefore require non-linear classifiers. DV 2.0 offers additional non-linear features to enhance the accuracy of classifications on such datasets. First, by using Support Vector Machines (SVM) DV 2.0 offers a data altering algorithm called GLC non-Linear (GLC-nL). This algorithm transforms a given dataset using SVM support vectors to create a new non-linear version of the dataset. By doing this we can visualize a general weighted sum of functions in GLC-L using SVM kernels. This creates a more accurate visualization on not linearly separable data.

Second, DV 2.0 offers hyperblock creation as a highly interpretable non-linear alternative to GLC-nL. For a given dataset hyperblocks are created with the algorithm GLC Hyperblock Rules Linear (GLC-HBRL) [15]. These hyperblocks are built upon the original Linear Discriminant Function (LDF) created by the initial GLC-L visualization.

Any correct classifications from the original LDF will be maintained in the hyperblocks and any misclassifications will either be rectified or remain unchanged. This enhances user trust in the model by providing two different interpretable classifiers for the data.

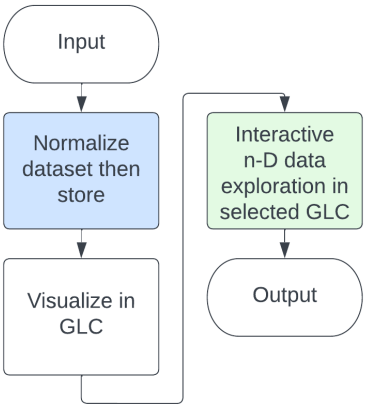

Fig. 7. DV 2.0 pipeline for a user-guided interactive classification (Blue – automatic process, Green – interactive process)

Additionally, DV 2.0 offers classifications for datasets with three or more classes. This is done by combining certain classes together to create a super-class while keeping one target class the same.

This is necessary due to the limitations of DV 2.0's single dimension horizontal linear separation threshold. In multiclass classification problems it is often that all classes can be separated but only on different dimensions. DV 2.0's class combination methodology counteracts this problem.

Once a super-class is created with all classes besides the target class, DV 2.0 can simply create a two-class classifier for the target and super-class. This momentarily simplifies the multiclass problem into a two-class problem. Then, once an ideal classifier is created for the super-class and the target class, the super-class can be simplified by removing one class from the super class.

The classifier can then be saved, and a new classifier can be created for the simplified super-class. This process can be repeated until only two classes remain. Thus, DV 2.0 can create a series of classifiers to solve multiclass classification problems. Fig. 8 shows a pipeline of how a user would interact with this classification feature.

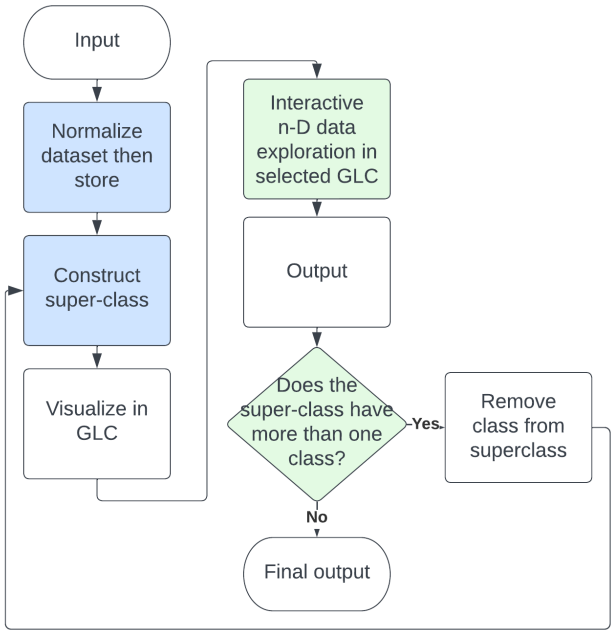

Fig. 8. DV 2.0 pipeline for a user-guided interactive multiclass classification (Blue – automatic process, Green – interactive process)

B. DV 2.0 Regression

A secondary feature for DV 2.0 is regression. After loading a dataset into DV 2.0 and selecting the regression option, Linear Regression (LR) is automatically performed to compute the optimal coefficient for each attribute in the dataset.

These coefficients are then transformed into angles for each attribute in a GLC-L visualization. Then, the values predicted by the visualization will be projected onto the x-axis where an additional point **A** will denote the accuracy of the prediction by showing the error of the prediction through distance.

The closer **A** is vertically to the prediction, the closer the GLC-L prediction was to the original value. If the GLC-L prediction is perfect, then the green squares will lie on top of the GLC-L prediction. An example of this can be seen in Fig. 9 where the purple squares denote the GLC-L predictions, and the green squares denote the prediction accuracies **A**. GLC-L predictions with multiple green squares below them are overlapped with other GLC-L predictions.

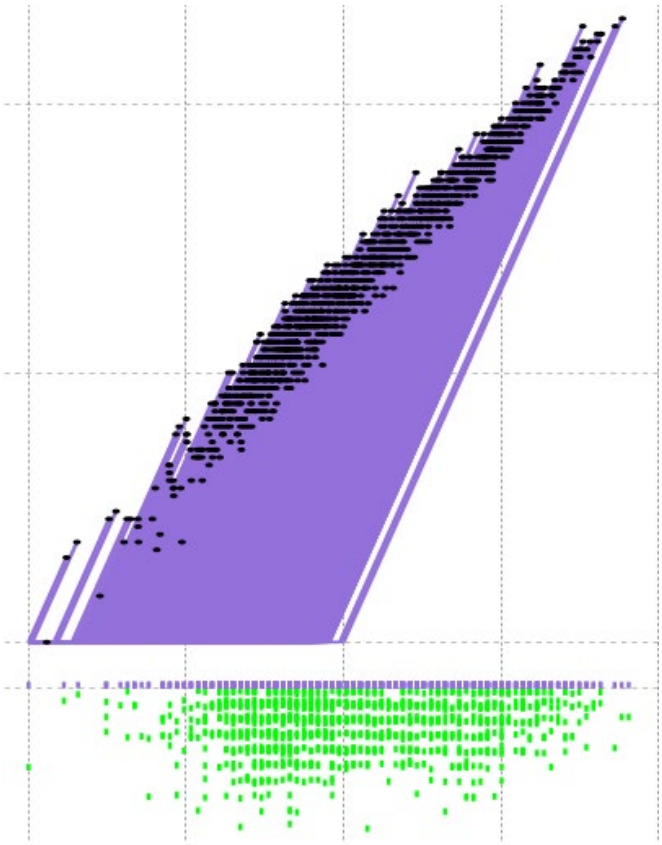

Fig. 9. DV 2.0 Regression

IV. Examples

In the previous sections the importance and features of DV 2.0 were discussed, in this section a brief walkthrough of DV 2.0 will be done with the Wisconsin Breast Cancer (WBC) dataset. This dataset has nine attributes and two classes: benign and malignant. The benign class has 444 cases, and the malignant class has 239 cases.

This dataset can be seen visualized in Fig. 10 with 98.1% accuracy after going through the pipeline described in Fig. 7. Then, by reordering the attributes by contribution, a more interpretable visualization can be seen in Fig. 11, which zooms Fig. 10.

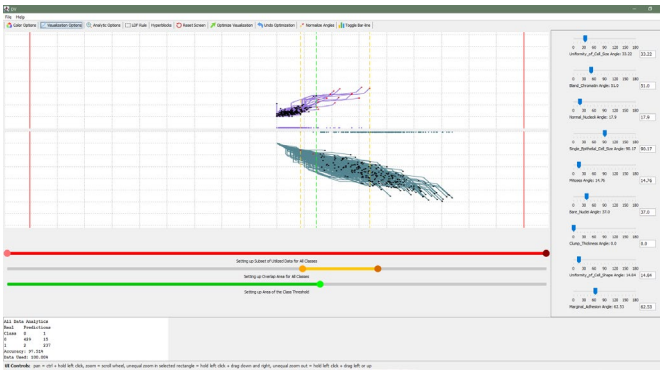

Fig. 10. WBC dataset visualized with 98.1% with GLC-L in DV 2.0. Sliders to adjust the attribute angles can be seen on the right and sliders to adjust the threshold can be seen underneath the visualization. Other visualization options can be seen on a toolbar above the visualization.

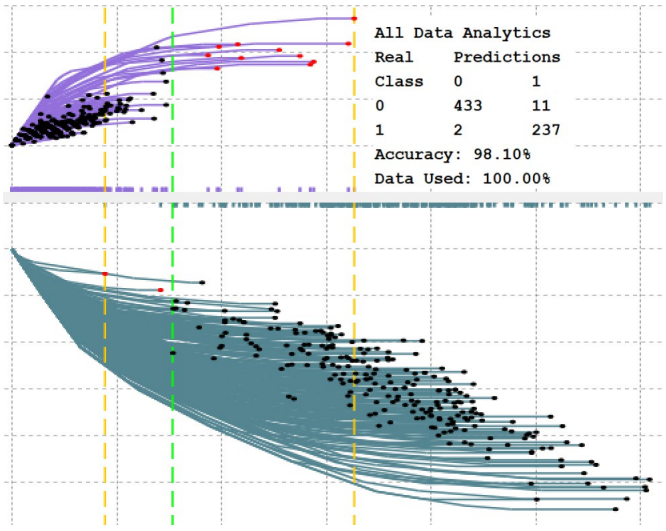

Fig. 11. WBC dataset visualized with 98.1% with GLC-L in DV 2.0 after angle and threshold optimization. Attributes reordered by contribution.

In this case it is easy to see that the first attribute is vertical and contributes nothing to the classification while the last attribute is horizontal and contributes the most to the classification.

Then, to further enhance the interpretability of the model, hyperblocks can be created by using the hyperblocks button located on the toolbar at the top of the window. In this instance, the hyperblocks created from the GLC-HBRL algorithm match up precisely with the cases classified correctly and incorrectly the GLC-L algorithm.

All cases that were classified correctly by GLC-L are also classified correctly by the hyperblocks and all cases that were misclassified by GLC-L are either classified incorrectly by the hyperblocks or have been given their own hyperblock for independent analysis.

The bounds of each hyperblock give a rule to which the linear discriminant function created by the original GLC-L visualization can be interpreted. These hyperblocks can be seen in Fig. 12 in parallel coordinates. Additionally, Fig. 13 shows the first three hyperblocks zoomed and Fig. 14 shows an alternative GLC-L visualization of the first three hyperblocks.

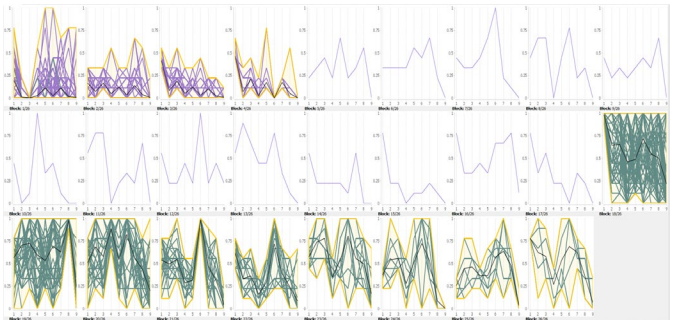

Fig. 12. Hyperblocks for WBC dataset in PC.

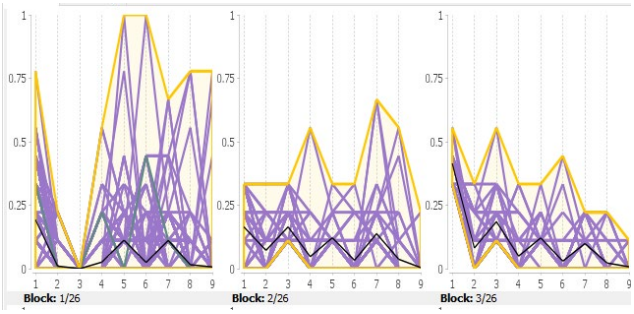

Fig. 13. First three hyperblocks from Fig. 12 zoomed.

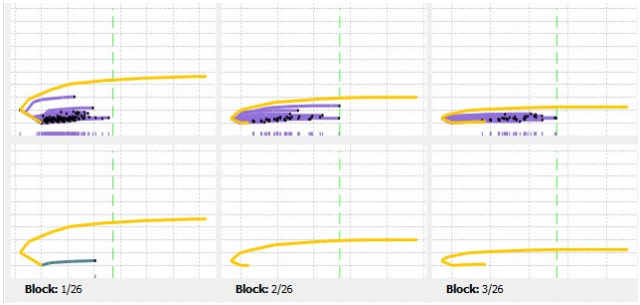

Fig. 14. Hyperblocks from Fig. 13 visualized in GLC-L

Using both the hyperblocks and LDF new cases not part of the existing cancer data can be picked up. Then, a hyperblock where this case is located can be identified, and the LDF for this case can be computed. This new case will then get a predictive explanation by both the HBs and LDF. Confirmation from both the HBs and GLC-L visualization will add user confidence in this predictive explanation.

V. Conclusion

Human-guided machine learning can enhance data exploration, model discovery, and model interpretability. It allows subject matter experts to define modeling problems and drive the development process. However, current human-guided machine learning systems are not able to fully realize this goal. Lossy and unexplainable visualizations severely hinder multidimensional exploration. New techniques such as Visual Knowledge Discovery are needed to fill this void.

DV 2.0 uses Visual Knowledge Discovery techniques to provide a wide range of explainable and interactive visualization methods. These methods contribute to interpretable and human-guided machine learning by allowing end-users to guide the development of models. In the future DV 2.0 will be expanded to create more general rules for classification and allow for larger datasets to be classified.

VI. References

[1] Elliott J. Data-Driven Discovery of Models (D3M), DARPA, 2021, https://www.darpa.mil/program/data-driven-discovery-of-models.
[2] Data-Driven Discovery of Models, 2022, ttps://datadrivendiscovery.org/
[3] AutoML: CMU-TA2, 2023, https://gitlab.com/sray/cmu-ta2
[4] Santos A, Castelo S, Felix C, Ono JP, Yu B, Hong SR, Silva CT, Bertini E, Freire J. Visus: An interactive system for automatic machine learning model building and curation. In: Proceedings of the Workshop on Human-In-the-Loop Data Analytics 2019 Jul 5 (pp. 1-7).
[5] Ono JP, Castelo S, Lopez R, Bertini E, Freire J, Silva C. Pipelineprofiler: A visual analytics tool for the exploration of automl pipelines. IEEE Transactions on Visualization and Computer Graphics. 2020, 3;27(2):390-400. https://doi.org/10.48550/ARXIV.2005.00160
[6] Zöller MA, Titov W, Schlegel T, Huber MF. XAutoML: A Visual Analytics Tool for Establishing Trust in Automated Machine Learning. arXiv preprint arXiv:2202.11954. 2022 Feb 24.
[7] D3M Developer Documentation, 2022, https://docs.datadrivendiscovery.org/devel/
[8] TwoRavens platform, 2020, http://2ra.vn/
[9] Kovalerchuk, B., Visual Knowledge Discovery and Machine Learning., Springer Nature, 2018, 317 p.
[10] Kovalerchuk, B., Nazemi, K., Andonie, R., Datia, N., Bannissi E. (Eds), Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery, Springer, 2022
[11] Kovalerchuk, B., Phan J., Full interpretable machine learning in 2D with inline coordinates. 25th International Conf. Information Visualisation IV-2021,2021, Vol. 1, pp. 189-196, IEEE, arXiv:2106.07568.
[12] McDonald R, Kovalerchuk B. Non-linear Visual Knowledge Discovery with Elliptic Paired Coordinates. In: Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery 2022, (pp. 141-172). Springer, Cham.
[13] Wagle SN, Kovalerchuk B. Self-service Data Classification Using Interactive Visualization and Interpretable Machine Learning. In: Integrating Artificial Intelligence and Visualization for Visual Knowledge Discovery 2022 (pp. 101-139). Springer, Cham.
[14] Recaido C., Kovalerchuk B., Interpretable Machine Learning for Self-Service High-Risk Decision-Making, in: 26th International Conference Information Visualisation, 2022, pp. 322–329, IEEE, arXiv:2205.04032.
[15] Huber L. Kovalerchuk B., Recaido C. Visual Knowledge Discovery with General Line Coordinates, CWU Techincal report, 2023.
[16] Kovalerchuk B., Gharawi A., Decreasing Occlusion and Increasing Explanation in Interactive Visual Knowledge Discovery, In: S. Yamamoto and H. Mori (Eds.) Human Interface and the Management of Information. Interaction, Visualization, and Analytics, LNCS 10904, pp. 505–526, 2018, Springer.
[17] Worland, A, Wagle S, Kovalerchuk B. Visualization of Decision Trees based on General Line Coordinates to Support Explainable Models, in: 26th International Conference Information Visualisation, 2022, pp. 351–358, IEEE, arXiv:2205.04035.
[18] Tanagra_machine_learning software system, Wikipedia, https://en.wikipedia.org/wiki/Tanagra_(machine_learning)
[19] Scikit-learn, https://scikit-learn.org/stable/index.html
[20] Antonini, A. S., Luque, L., Ganuza, M. L., & Castro, S. M. (2022). Toward a taxonomy for 2D non-paired General Line Coordinates: A comprehensive survey. International Journal of Data Science and Analytics, 15(2), 133–158. https://doi.org/10.1007/s41060-022-00361-w
[21] Luque, L. E., Ganuza, M. L., Antonini, A. S., & Castro, S. M. (2021). npGLC-Vis Library for Multidimensional Data Visualization. In Cloud computing, Big Data & Emerging Topics: 9th conference, JCC-BD&ET, La Plata, Argentina, 2021, Proceedings (pp. 188–202). Springer.
[22] Albrecht, G., & Pang, A. (2016). Interactive high-dimensional data analysis using the “three experts.” Electronic Imaging, 28(1), 1–10. https://doi.org/10.2352/issn.2470-1173.2016.1.vda-492
[23] Glendenning, K., Wischgoll, T., Harris, J., Vickery, R., & Blaha, L. (2016). Parameter space visualization for large-scale datasets using parallel coordinate plots. Journal of Imaging Science and Technology, 60(1). https://doi.org/10.2352/j.imagingsci.technol.2016.60.1.010406
[24] XDAT - a free parallel coordinates software. 2023, https://www.xdat.org/
[25] Pandas.plotting.parallel_coordinates.https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.plotting.parallel_coordinates.html
[26] Schneider, B., Keim, D., & El-Assady, M. (2020). DataShiftExplorer: Visualizing and comparing change in multidimensional data for supervised learning. Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. https://doi.org/10.5220/0008940801410148