AReS: An AutoML Regression Service for Data Analytics and Novel Data-centric Visualizations

Conference Paper · August 2023

CITATIONS
0

READS
9

7 authors, including:

Sam Johnson
Indiana University Bloomington
3 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Parichit Sharma
Indiana University Bloomington
9 PUBLICATIONS **19** CITATIONS
SEE PROFILE

Hasan Kurban
Texas A&M University at Qatar
38 PUBLICATIONS **206** CITATIONS
SEE PROFILE

Mehmet Dalkilic
Indiana University Bloomington
91 PUBLICATIONS **706** CITATIONS
SEE PROFILE

AReS: An AutoML Regression Service for Data Analytics and Novel Data-centric Visualizations

Josh Elms[1], Sam Johnson[1], Madhavan K R[1], Keerthana Sugasi[1], Parichit Sharma[1], Hasan Kurban[3,1,2], and Mehmet M. Dalkilic[1,2]

[1]Computer Science Department, Indiana University, Bloomington, Indiana, IN, USA
[2]Data Science Program, Indiana University, Bloomington, Indiana, IN, USA
[3]Electrical and Computer Engineering, Texas A&M University at Qatar, Doha, Qatar
Email: {jmelms, sj110, madhkr, ksugasi, parishar, hakurban, dalkilic}@iu.edu

***Abstract*—While machine learning (ML) use has become prevalent across most domains, there is a growing gap between programmers and non-programmers in their use of ML. Indeed, choosing the best models, applying the models, and verifying their quality is out of reach for individuals who rely on this kind of quantitative analysis but have limited programming experience–particularly those in the natural and social sciences. Automatic ML (AutoML) is supposed to be the stopgap giving non-programmers the ability to fully use ML, but in practice, these tools fall short. In response to this challenge, we built a data-centric machine learning web service we call "AReS" that both simplifies and streamlines the entire ML pipeline. AReS at its simplest requires only data. It chooses among dozens of diverse regression algorithms, picking the best. AReS gives both symbolic and visual assessments of the model's performance through novel data-centric visualizations that provide insight into the data itself, both individual points and collections. To validate AReS, two cases using real-world Kaggle competitions (kaggle.com) are studied with AReS' default settings. AReS delivers competitive results in both but is among the best results in one. This paper's novel web service, AReS, can be accessed at https://dalkilic.luddy.indiana.edu/.**

1. Introduction

The rapid growth of machine learning (ML) algorithms coincides with their appearance in many diverse, unrelated domains *e.g.*, sports, healthcare, finance, transportation, entertainment, policy, agriculture, and service industries [1]. Unfortunately, ML's ease-of-use has not kept up. For those outside of computer science and a few related groups, using ML is nearly impossible. The recent disruption of generative AI shows that a "smart" intermediary can give more power to non-experts by removing a problem's technical details and replacing them with general tasks. This paradigm is not confined to simple search. Indeed, most research done in the natural and social sciences is conducted by non-computer scientists who need computation–in particular, modeling with its accompanying analytics. Why not have a service driven by AI that helps scientists effectively use computation for analysis without having to program?

The most ubiquitous model is linear regression; there is certainly no quantitative analysis in the natural and social sciences that does not rely heavily on it [2] [3] [4]. While several tools purport to be either no-code or low-code automatic machine learning (AutoML) solutions, they are still more or less oriented towards technical individuals, or as Google describes them on their AutoML webpage: "developers with limited machine learning experience" [5]. Popular AutoML software includes Google's Cloud AutoML [5], Microsoft's Azure Machine Learning [6], and Amazon Web Service's AutoML Solutions [7]. Each offers a comprehensive AutoML tool designed to handle data ingestion, basic data cleaning, and modeling services to build, tune, and deploy ML models. Amazon even promotes an open-source framework called AutoGluon, a Python package offering solutions for modeling. While each of these tools is simple and powerful compared to interfacing with the libraries directly, they all require programming and are, therefore, inaccessible to most of the individuals to whom they are offered.

Scientists have had to rely on spreadsheets. This versatile tool allows for keeping track of data, building simple functions, and performing a few standard statistical tests. Although a full programming language exists for the most popular, Microsoft Excel, the environment is not suited for modern machine learning. Excel provides no tools for either classification or clustering. Only one kind of regression is offered: linear. Analysis of this regression is limited to a few standard error metrics such as Mean Squared Error (MSE) or Mean Absolute Error (MAE). On their own, these numbers do little to aid in the interpretation of model performance. For example, an MSE of 5.7 does not inform the user which region of the data is poorly captured by the model, whether the model could be improved by adding more data, or whether the data is linear. The only alternative is polynomial regression over 2D data. Clearly, this is out of touch with the current state of the art, but with a growing programming gap, scientists have no other choice. The other reason to

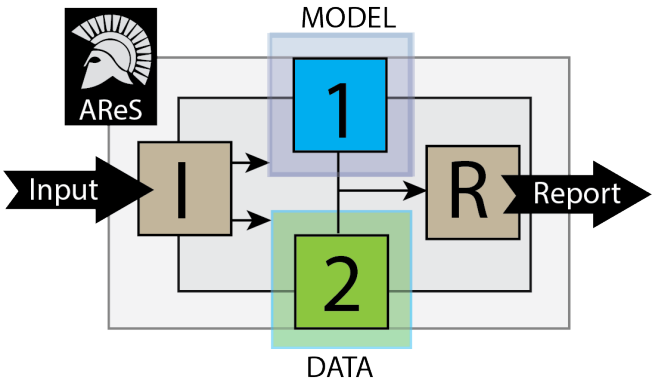

Figure 1. High-level view of AReS. After verifying the input (I), AReS sends the information to two pipelines: (1) for model building, testing, and selection, and (2) for data-centric exploration and visualization. The results of both pipelines are delivered to the report generator (R).

use spreadsheets is their visualization capability. Indeed, every research paper which is not purely formal employs analytics and visualization as a critical part of its argument. But just as spreadsheets are limited in what can be done with their built-in functions, their visualization options consist of only simple 2D plots with some 3D elements. There are no options for visualizing high-dimensional data, much less along with its model.

A very recent term describing the importance of data in ML is "data-centric." Instead of laboring to fine-tune a particular model (usually by only a marginal amount), the focus is squarely on data: improving quality, expressiveness, management strategy, etc [8]. As we move toward a system for scientists, we prioritize its data-centric tenets of data quality, analysis of regional model performance, and interpretable visualizations.

To this end, we have built an elegant web service for scientists to model and explore their data for a multitude of available regression algorithms. We employ a data-centric perspective by visualizing important aspects of the data itself that we expect will be actionable. We designed an automatic pipeline to inspect the data, create training and test data, train and select ML models, score the models to show the best performing, and ultimately generate a human-readable report. We allow for some human-in-the-loop, but our aim is to aid non-expert scientists who require more than a single regression with a simple line plot.

The core components of our web service are a user-friendly interface for data ingestion, a selection of state-of-the-art ML algorithms, an automated model training routine, and a novel model evaluation and visualization framework. The service is designed to scale with the size and complexity of the data. The system only requires a modicum of information, for example, names of chosen regressors. In this paper, we discuss the underlying principles and design of our data-centric machine learning web service, and we highlight its key advantages, along with several innovative data-centric visualizations. We then present a series of case studies demonstrating both its effectiveness and versatility using real-world problems. The end-product is a report that gives textual and visual information about the data, model, and performance. We call this service "AReS" (**A**utoML **Re**gression **S**ervice); AReS can be accessed at https://dalkilic.luddy.indiana.edu/.

2. Web Service

AReS is available to users through a web application. Users configure a pipeline that reflects different expertise and requirements. Parameters for the pipeline include regressors, cross-validation settings, evaluation metrics, and the model ranking mechanism. The parameters can be initialized in two ways: the default form is useful for non-expert users and an advanced form gives users fine-grained control over the pipeline's execution. AReS' default setting requires only the dataset, and the pipeline's other parameters are intelligently selected. In the advanced mode, AReS allows for modifying all of the pipeline's parameters.

When the user submits a request, a request ID is generated and AReS executes asynchronously. The user should expect processing times to vary according to the size of their data; requests with datasets approaching 10,000 rows by 100 columns may take as long as 48 hours. The user can return to the site after the run is complete and enter their request ID to view a data visualization dashboard (VDB). The VDB is implemented using Apache ECharts [9] and Plotly [10]: open-source data visualization libraries for creating charts and graphs in web applications. AReS' interface is configurable, giving the user control over the regressors, metrics, and visualizations. Additionally, in the Point Predictability and Region Predictability plots to be introduced later, the user can adjust how much data is shown on the plot based on predictability. High-quality plot images, suitable for publication or presentations, can be saved to the user's desktop.

AReS is designed using a cloud-native architecture with the latest software stack to meet current software development standards. The cloud-native architecture enables the package to be deployed to any existing cloud platform, giving it the flexibility to scale based on the workload. However, this service is intended for use on open-source problems; as such, no safeguards are in place to ensure data privacy. We supply a link for our GitHub repository in Supplementary Materials to allow interested individuals to utilize our novel pipeline for their own purposes.

3. AReS: A Detailed Look

Our system leverages a novel framework first presented in [11]. Fig. 2 shows AReS in more detail (see Fig. 1 for a general description). The pipeline begins when a user submits a form, called a request, that includes a dataset and several parameters for the pipeline. Upon reception of the request, the dataset undergoes validation, including confirmation of the following attributes: dataset contains only numeric data, does not exceed 10,000 rows, and does

Class	Regressor
Linear Models	LinearRegression, Ridge, Lasso, ElasticNet, Lars, LassoLars, LassoLarsIC, BayesianRidge, OrthogonalMatchingPursuit, ARDRegression, TransformedTargetRegressor
Generalized Linear Models	TweedieRegressor, GammaRegressor, PoissonRegressor
Linear Models Robust to Outliers	HuberRegressor, TheilSenRegressor, RANSACRegressor
Support Vector Machines	NuSVR, SVR, LinearSVR
Nearest Neighbor Models	KNeighborsRegressor, RadiusNeighborsRegressor
Tree-Based Models	DecisionTreeRegressor, BaggingRegressor, ExtraTreesRegressor, ExtraTreeRegressor, RandomForestRegressor, AdaBoostRegressor, HistGradientBoostingRegressor, GradientBoostingRegressor
Miscellaneous	MLPRegressor, PassiveAggressiveRegressor, PLSRegression

Table 1. Groupings of scikit-learn regressors selected for the final AutoML pipeline.

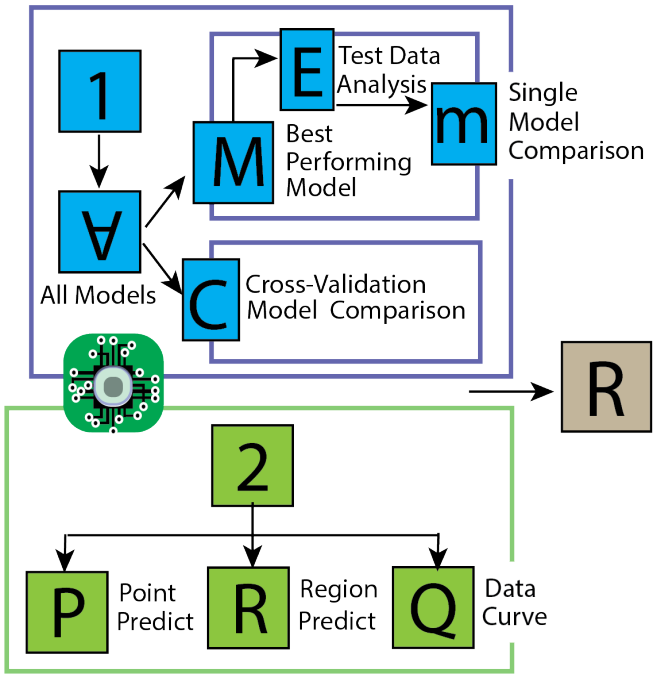

Figure 2. A more detailed look at the AReS pipeline (see Fig. 1). (1) is a user request. (∀) is the training of all models, from which (M) is selected. (E), (C), and (m) are then produced and delivered in a report (R). Workflow (2) is activated and figures (P), (R), and (Q) are generated and added to the report.

not exceed 100 columns. All variables are standardized by subtracting the mean and scaling to unit variance. This is necessary to eliminate the effect of variable scale on regressors that rely on distance calculations, such as k-Nearest Neighbors (k-NN), or Support Vector Machine (SVM). None of the regressors used in the pipeline allow training data to have missing values. If a user submits a dataset with missing values, our pipeline performs missing value imputation using the k-NN method with parameters $k = 5$ and Euclidean distance. The data is then shuffled and split into one training set and one test set. The proportion of data allocated to the test set is determined by the user and defaults to 0.1 if unspecified.

Currently, scikit-learn implements 55 regressors. We hand-selected 33, removing those that either consistently performed poorly in tests, were relatively slow to train, or performed multi-output regression or cross-validation. These 33 selected regressors are displayed in Table 1 along with their assigned groupings. The user can select from among these regressors when they submit a request.

To train models, AReS performs k-Fold Cross-Validation training (CV). AReS splits the training data into the k evenly-sized folds specified by the user, then for each regressor, AReS fits k models and uses each model to predict the labels of the holdout fold. Each metric chosen by the user is computed for these predictions. The best-performing of the k models trained is selected according to the model-ranking metric specified by the user. Each of these models is then evaluated on the test set held out at the beginning of the pipeline. After training and evaluating each model, several additional processes are triggered which generate traditional and data-centric visualizations that can aid the user in making informed decisions from their data.

4. Data-Centric Visualization Techniques

Traditional ML visualizations are intended to aid in understanding the model and data. In addition to producing these standard plots, AReS provides a set of novel data-centric visualizations that can help scientists better understand their data. For example, they might develop an insight into how their data can be better curated to create more impactful models. More complete descriptions and examples can be found in Supplementary Materials.

4.1. Quantity Curve

Learning curves are used as a diagnostic tool when training neural networks. This idea led us to design and implement what we call the "Quantity Curve". In contrast to a learning curve, which plots error as a function of the number of training iterations, the Quantity Curve plots error as a function of the amount of training data provided to

the model. The purpose of showing error as a function of training data is to determine when the model is overfitting; as test error begins to increase, the likelihood of overfitting increases as well. The tail-end behavior of a plot can be used to mitigate overfitting. If the training and test error have not converged and test error has a negative slope, then providing the model with more data is likely to reduce the prediction error.

Algorithm 1 shows the steps for building a Quantity Curve plot for a single regressor. The process begins by partitioning the data into 11 folds and then training a regressor on each fold. For each regressor, training set error is measured. The test set error is then computed from predictions for the target attribute of the holdout fold. The inclusion of CV in the algorithm prevents the initial partitioning of the dataset from introducing bias. The plot shows, for each value on the abscissa, the average error for a model trained on that percentage of the data. For example, $k = 1$ corresponds to training on 10% of the data, $k = 2$ corresponds to training on 20% of the data, and so on. We repeat this process for each regressor selected by the user. Figure 3 is provided as an example of a Quantity Curve graph from our web service.

4.2. Point Predictability

Data is akin to a sand dune–individual grains at once comprise it and disappear into it. Each interpretation exhibits different properties, and AReS provides both of these useful perspectives. For each point, AReS assesses its "predictability" by examining the model's error on the datum. To begin the construction of this plot, model training is conducted via k-fold CV. Then, AReS predicts the target variable of the original dataset using each of the k models and records the predicted target value for each data point. The Mean

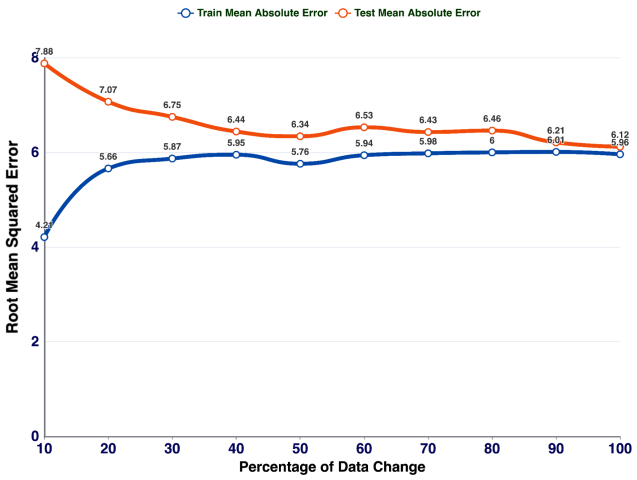

Figure 3. An example Quantity Curve displaying Root Mean Squared Error for the AdaBoost regressor over training and test sets of the Concrete Compressive Strength (CCS) Dataset [12]. The training and test error converge for this regressor trained on 100% of the training data; this indicates that training on more data will yield only marginal improvements to the model.

Algorithm 1 Quantity Curve Algorithm

Data: $X =$ **Predictor Attributes**, $Y =$ **Target Attribute**
Result: $\alpha =$ **Training Errors**, $\beta =$ **Test Errors**

```
α ← [ ] {Empty list}
β ← [ ]
K ← 11
X' ← X randomly partitioned into K folds
Y' ← Y partitioned into same K folds
for i ← 1 to K do
    a ← {X'_j | j ≠ i}
    b ← X'_i
    c ← {Y'_j | j ≠ i}
    d ← Y'_i
    for p ← 1 to K − 1 do
        a' ← a_0 to a_p
        c' ← c_0 to c_p
        m ← train regressor over a' {m is a fitted model}
        z_tr ← m predict over a'
        z_te ← m predict over b
        Append(α,dissim(z_tr, c')) {any error metric}
        Append(β,dissim(z_te, d))
    end for
end for
```

Absolute Percentage Error (MAPE) is determined from the ground-truth target value for each data point $x_1, x_2, \ldots x_n$:

$$\text{MAPE}(x_i) \quad = \quad \frac{1}{k} \sum_{j=1}^{k} \frac{|m_j(x_i) - y_i|}{y_i} \qquad (1)$$

where $m_1, m_2, \ldots m_j \ldots m_k$ are the models generated by k-fold CV and $m_j(x_i)$ is a prediction for x_i using model m_j. Each data point with a MAPE value $v \in [0, 1]$ is assigned a color on an even gradient from blue to red. Points where $v \approx 0$ have high predictability and are colored blue. $v \approx 1$ represents predictions more than double or less than half of their ground-truth value and are colored red. In other words, red indicates low predictability, which perhaps indicates noise. Sample Point Predictability plots are shown in Figures 4 and 6.

AReS produces the visualization by first applying a dimensionality reduction technique and then plotting in either 2D or 3D. Data reduction is done using either principal component analysis (PCA) or an innovative new technique, t-Distributed Stochastic Neighbor Embedding (t-SNE). PCA can be used to compute a projection of the data points that preserves a meaningful portion of the variance in the data [13]. The t-SNE algorithm is able to extract non-linear relationships between features, making it a powerful tool for data visualization in 2D and 3D space [14]; this has spurred its widespread use in the natural sciences.

AReS' data-centric Point Predictability plot provides information unavailable from other visualizations. For example, consider the case of one low-predictability data point nestled within a group of high-predictability data points; this likely indicates an outlier. As another example, the presence of a collection of low-predictability data points

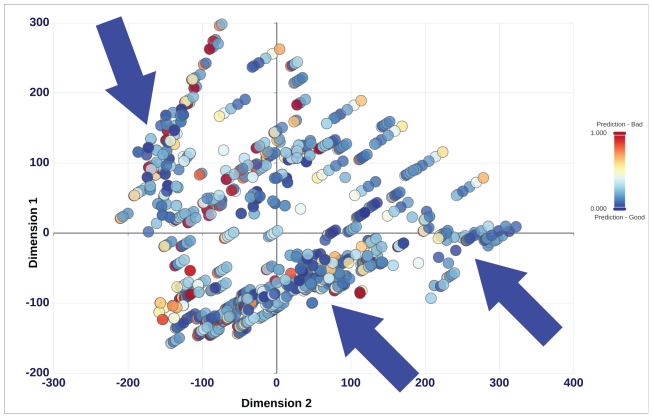

Figure 4. An example Point Predictability graph created using 2D PCA and Linear Regression on the CCS dataset. The arrows point to regions of high predictability.

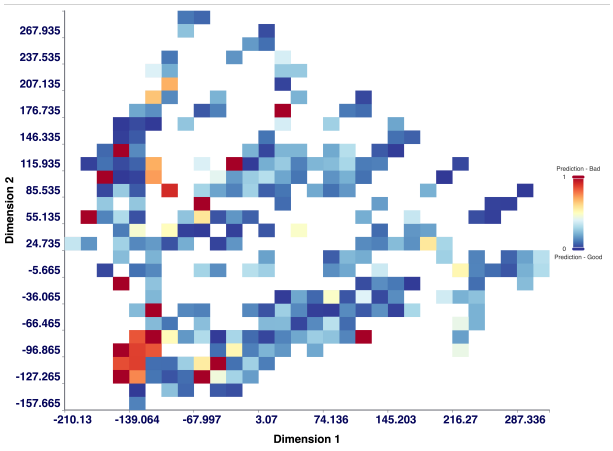

Figure 5. A sample Region Predictability plot using PCA for SVR on the CCS dataset. This plot exposes potential bias in the model for data regions in the lower extremes of both dimensions.

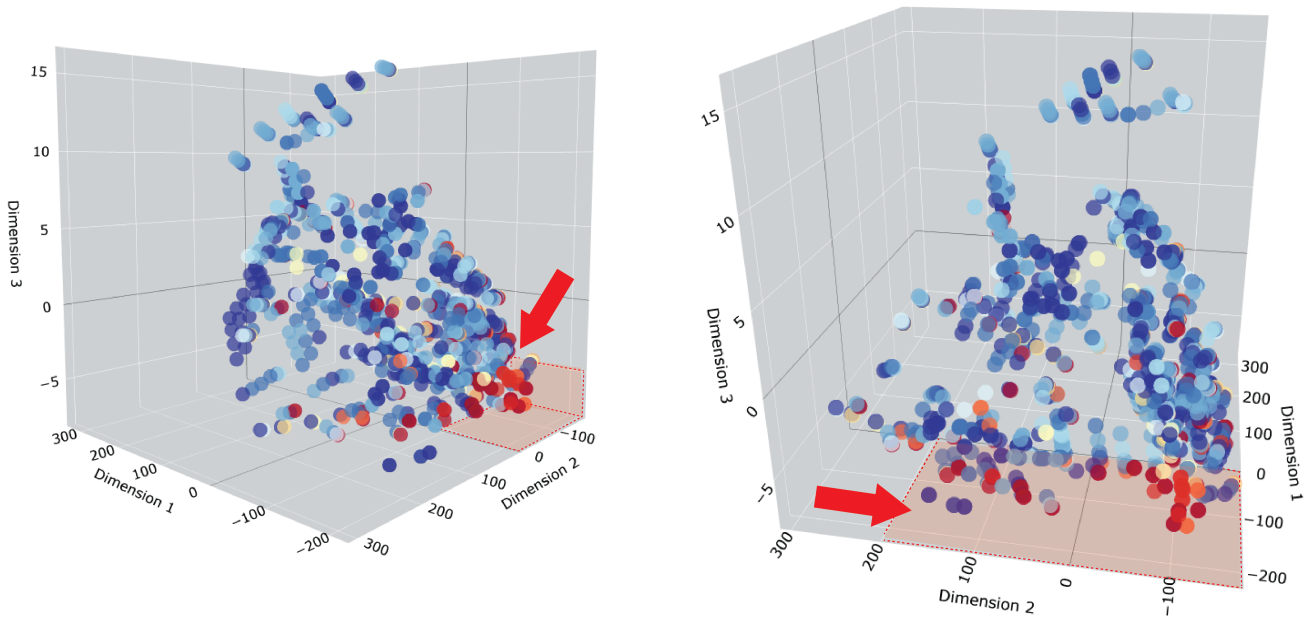

Figure 6. Two views of an example Point Predictability graph created using 3D PCA for SVR on the CCS dataset. The user may notice a clump of low-predictability data points at the lower extremes of each dimension. To improve model accuracy, the user might consider collecting more of these data points and retraining the model.

on the fringes might indicate a number of problems *e.g.*, equivocation (w.r.t. the model), machine limitations in the case of a physical experiment, or noise. With AReS' Point Prediction, users have a detailed look into the data that has no parallel among traditional visualizations.

4.3. Region Predictability

AReS' Region Predictability provides insight into collections of data. Using regular rectangular regions, AReS first partitions the data in a 2D space. AReS then colors each region according to the average predictability of the points in that region. A sample Region Predictability plot is shown in Fig 5.

When applying PCA to a dataset, a linear combination of the feature variables is produced. This enables unseen data points from the same population to be mapped onto the principal-component space *ad hoc*. AReS generates a plot for each specified regressor so that the user is able to compare predictability for each data point across regression

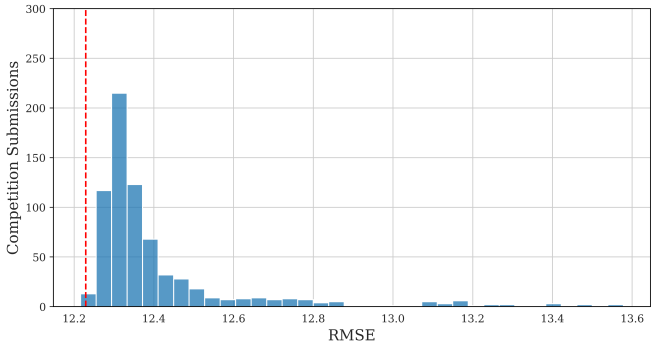

Figure 7. A histogram of the top $\sim$ 90% of performers in a Kaggle competition for predicting concrete strength. The best-performing model produced by our AutoML pipeline is shown as a dashed red line.

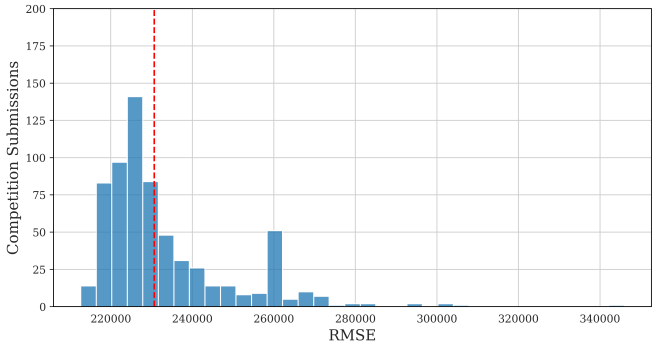

Figure 8. A histogram of the top $\sim$ 90% of competitors in a Kaggle competition for predicting Paris house prices. The best-performing model produced by our AutoML pipeline is shown as a dashed red line.

algorithms. An outcome is that the user may be able to reason about which set of regressors will best predict new, unlabeled data without generating any additional models. Another outcome is that regions can be scrutinized for potential improvements. For example, failure of a model to make accurate predictions for a particular region may indicate bias [15]. It is likely that data sharing a region in the principal-component space are similar on one or more attributes in the original space. We point out that certain ethical questions arise [16]. If the similar feature(s) are sensitive groups, such as race or sex, this bias will likely cause the model to produce spurious results [17] [18] [19].

5. Case Studies

To assess AReS' usefulness, we present two case studies. In these studies, we chose two regression datasets that have recently been featured in a Kaggle (kaggle.com) competition. In these competitions, users attempt to train the best model according to a predetermined performance metric. Goodness-of-fit for regression tasks is almost always measured with Root Mean Squared Error (RMSE). Kaggle provides labels for only the training partition of the dataset, so we used only the training set in our AutoML pipeline. This means that our performance figures were computed for a different test set than the other competitors. This is not to say that we are advantaged; our models are trained on less data than those submitted to the Kaggle competition and the observed RMSE values are therefore likely to be conservative estimates. **Case I.** is a synthetic concrete strength dataset generated by a deep learning model trained on the CCS dataset [12] [20]. The dataset has 5,407 instances and nine attributes. There were 767 submissions for this competition, and the RMSE of the top $\sim$ 90% of competitors is displayed in Figure 7. AReS' best performing model, a Gradient-Boosting Regressor, predicted the holdout set with an RMSE of 12.922 and is shown with a red dotted line. This score puts AReS among the very best competitors, despite it requiring neither fine-tuning nor domain knowledge. **Case II.** is the Paris Housing Price dataset [21]. This dataset has 22,730 instances and 17 numerical columns. AReS again determines that the best-performing regressor is Gradient Boosting. This regressor produced an RMSE value of 230,678.0871 and is comparable to the majority of submissions. Of the 705 submissions, the top $\sim$ 90% are visualized in the histogram in Figure 8 along with AReS' performance. Our service performs approximately as well as the median Kaggle competitors in this contest but with almost no user input. We intend to include data engineering and hyperparameter optimization in future versions of the service, which should enable the service to perform exceedingly well in this type of competition.

6. Summary and Conclusions

In this paper, we presented a data-centric machine learning web service called "AReS" for non-experts that automates and streamlines the ML pipeline, from model training to model evaluation. By providing a user-friendly interface and incorporating advanced modeling techniques, AReS enables users who are not programmers to employ current ML algorithms and advanced visualizations. AReS removes roadblocks common in other AutoML systems. Additionally, AReS provides novel data-centric analysis on the data, which elucidates a model's performance.

Future work includes further testing of AReS, expansion to accommodate new domains, and incorporation of advanced ML techniques, such as hyperparameter tuning, to enhance model performance. As we refine the visualization algorithms, the service will be able to accommodate larger datasets and datasets with non-numeric attributes. With regards to data reduction, we will enable test points to be overlaid on the Region Predictability plot. As a result, a user may be able to infer the potential predictability of an unlabeled data point by observing the predictability of its region in the principal-component space. Missing and spurious values will also be addressed, though this problem is among the most difficult. Lastly, in future versions of AReS we intend to give users the ability to predict over new data using the best-performing models from our service.

References

[1] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[2] R. J. Leatherbarrow, "Using linear and non-linear regression to fit biochemical data," *Trends in Biochemical Sciences*, vol. 15, no. 12, pp. 455–458, 1990. [Online]. Available: https://www.sciencedirect.com/science/article/pii/096800049090295M

[3] E. D. Feigelson and G. J. Babu, "Linear Regression in Astronomy. II." , vol. 397, p. 55, Sep. 1992.

[4] Y. Petscher and J. A. R. Logan, "Quantile regression in the study of developmental sciences," *Child Development*, vol. 85, no. 3, pp. 861–881, 2014. [Online]. Available: https://srcd.onlinelibrary.wiley.com/doi/abs/10.1111/cdev.12190

[5] "Automl," [Accessed: April 30, 2023]. [Online]. Available: https://cloud.google.com/automl

[6] "What is automated machine learning (automl)?" [Accessed: April 30, 2023]. [Online]. Available: https://learn.microsoft.com/en-us/azure/machine-learning/concept-automated-ml?view=azureml-api-2

[7] "Aws automl solutions: Built-in automl across the aws ml stack," [Accessed: April 30, 2023]. [Online]. Available: https://aws.amazon.com/machine-learning/automl/

[8] A. Ng, "A chat with andrew on mlops: From model-centric to data-centric ai," *URL: https://youtu. be/06-AZXmwHjo*, 2021.

[9] D. Li, H. Mei, Y. Shen, S. Su, W. Zhang, J. Wang, M. Zu, and W. Chen, "Echarts: a declarative framework for rapid construction of web-based visualization," *Visual Informatics*, vol. 2, no. 2, pp. 136–146, 2018.

[10] P. T. Inc. (2015) Collaborative data science. Montreal, QC. [Online]. Available: https://plot.ly

[11] H. Kurban, M. Kurban, and M. M. Dalkilic, "Rapidly predicting kohn–sham total energy using data-centric ai," *Scientific Reports*, vol. 12, no. 1, p. 14403, 2022.

[12] I.-C. Yeh, "Modeling of strength of high-performance concrete using artificial neural networks." cement and concrete research, 28(12), 1797-1808," *Cement and Concrete Research*, vol. 28, pp. 1797–1808, 12 1998.

[13] H. Hotelling, "Simplified calculation of principal components," *Psychometrika*, vol. 1, pp. 27–35, 1936.

[14] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html

[15] S. Alelyani, "Detection and evaluation of machine learning bias," *Applied Sciences*, vol. 11, no. 14, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/14/6271

[16] M. Kearns and A. Roth, *The Ethical Algorithm: The Science of Socially Aware Algorithm Design*. USA: Oxford University Press, Inc., 2019.

[17] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, ser. Proceedings of Machine Learning Research, S. A. Friedler and C. Wilson, Eds., vol. 81. PMLR, 23–24 Feb 2018, pp. 77–91. [Online]. Available: https://proceedings.mlr.press/v81/buolamwini18a.html

[18] B. Lwowski and A. Rios, "The risk of racial bias while tracking influenza-related content on social media using machine learning," *Journal of the American Medical Informatics Association*, vol. 28, no. 4, pp. 839–849, 01 2021. [Online]. Available: https://doi.org/10.1093/jamia/ocaa326

[19] M. Samorani, S. L. Harris, L. G. Blount, H. Lu, and M. A. Santoro, "Overbooked and overlooked: machine learning and racial bias in medical appointment scheduling," *Manufacturing & Service Operations Management*, vol. 24, no. 6, pp. 2825–2842, 2022.

[20] W. R. Ashley Chow, inversion, "Regression with a tabular concrete strength dataset," 2023, [Accessed: May 24, 2023]. [Online]. Available: https://kaggle.com/competitions/playground-series-s3e9

[21] A. Chow, Inversion, and W. Reade, "Regression with a tabular paris housing price dataset," 2023, [Accessed: May 24, 2023]. [Online]. Available: https://kaggle.com/competitions/playground-series-s3e6

7. Supplementary Materials

This paper's novel web service, AReS, can be accessed at https://dalkilic.luddy.indiana.edu/. A complete tutorial on the usage of AReS is located at https://dalkilic.luddy.indiana.edu/walkthrough. The data and code used throughout this paper can be found at https://github.com/Joshua-Elms/AReS-whitepaper-supplementary-materials.