Responsible Model Selection with Virny and VirnyView

Denys Herasymuk
Ukrainian Catholic University
Lviv, Ukraine
herasymuk@ucu.edu.ua

Falaah Arif Khan
New York University
New York, USA
fa2161@nyu.edu

Julia Stoyanovich
New York University
New York, USA
stoyanovich@nyu.edu

ABSTRACT

In this demonstration, we present a comprehensive software library for model auditing and responsible model selection, called Virny, along with an interactive tool called VirnyView. Our library is modular and extensible, it implements a rich set of performance and fairness metrics, including novel metrics that quantify and compare model stability and uncertainty, and enables performance analysis based on multiple sensitive attributes, and their intersections. The Virny library and the VirnyView tool are available at https://github.com/DataResponsibly/Virny and https://r-ai.co/VirnyView.

CCS CONCEPTS

• **Information systems** → **Data management systems**; • **Social and professional topics** → **Socio-technical systems**; • **Human-centered computing**;

KEYWORDS

data-centric AI, model selection, fairness, stability, robustness

ACM Reference Format:
Denys Herasymuk, Falaah Arif Khan, and Julia Stoyanovich. 2024. Responsible Model Selection with Virny and VirnyView. In *Companion of the 2024 International Conference on Management of Data (SIGMOD-Companion '24), June 9–15, 2024, Santiago, AA, Chile.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3626246.3654738

1 INTRODUCTION

Machine Learning (ML) models are being used to make decisions in increasingly critical domains. To determine whether models are production-ready, they must be comprehensively evaluated on a number of performance dimensions, not just accuracy. As a practical scenario, consider Ann, a data scientist working on a public policy task, such as to predict whether a low-income individual, not eligible for Medicare, has coverage from public health insurance (the ACSIncome task from [7]). Ann aims to develop an accurate, robust, and fair model, and so needs to assess multiple models from diverse hypothesis spaces and, further, to compare several fairness-enhancing interventions on the models. Since measuring only accuracy and fairness is not enough for building robust ML systems [3, 11, 14], each model involves at least three overall dimensions (correctness, stability, uncertainty) and three disparity dimensions evaluated on subgroups of interest (error disparity, stability disparity, uncertainty disparity). Adding to the complexity, these model dimensions exhibit trade-offs with one another [1]. Considering the multitude of model types, performance dimensions and trade-offs, Ann faces the challenge of *responsible model selection* — a task that appears insurmountable. In this demonstration, we present the comprehensive Virny[1] model profiling library, along with the interactive VirnyView[2] tool that profiles dataset properties related to protected groups, computes comprehensive *nutritional labels* [19] for individual models, compares multiple models according to multiple metrics, and guides Ann through model selection.

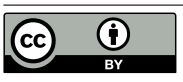
Contributions. In contrast to existing fairness software libraries [2, 18, 21] and model card generating frameworks [5, 17], our system stands out in four key aspects. Virny *(i)* facilitates the measurement of fairness, stability, and uncertainty metrics for a set of initialized models, both overall and broken down by user-defined subgroups of interest; *(ii)* enables data scientists to analyze performance using multiple sensitive attributes (including non-binary) and their intersections; *(iii)* offers diverse APIs for metric computation, designed to analyze multiple models in a single execution, assessing stability and uncertainty on correct and incorrect predictions broken down by protected groups, and testing models on multiple test sets, including in-domain and out-of-domain; *(iv)* implements streamlined flow design tailored for responsible model selection, reducing the complexity associated with numerous model types, performance dimensions, and data-centric and model-centric interventions.

2 OVERVIEW OF THE FUNCTIONALITY

2.1 Implementation of Virny

Virny is implemented in Python, and is designed based on three core principles: *(i)* facilitating easy extensibility of model analysis capabilities; *(ii)* ensuring compatibility with user-defined datasets and model types; and *(iii)* enabling simple composition of disparity metrics based on the context of use. The software framework decouples the process of model profiling into several stages, including *subgroup metric computation*, *disparity metric composition*, and *metric visualization*. This separation empowers data scientists with greater control and flexibility in using the library, both during model development and for post-deployment monitoring. Figure 1 shows the model analysis pipeline, with inputs shown in green, pipeline stages in blue, and per-stage model outputs in purple. We will describe each stage below, and will return to a discussion about the specific metrics supported by Virny in Section 2.2.

Inputs. Users provide three inputs: *base flow dataset*, *models config*, and *config yaml*. The *base flow dataset* is a custom object for the user's dataset that includes its descriptive attributes such as a target column, numerical columns, categorical columns, train and



SIGMOD-Companion '24, June 9–15, 2024, Santiago, AA, Chile

ACM ISBN 979-8-4007-0422-2/24/06
https://doi.org/10.1145/3626246.3654738

[1] https://github.com/DataResponsibly/Virny
[2] https://r-ai.co/VirnyView

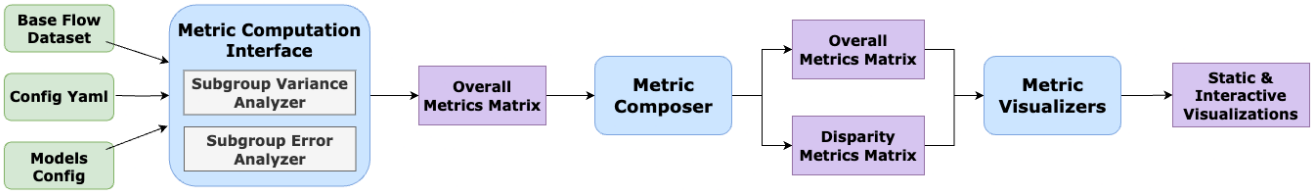


Figure 1: Responsible model selection in Virny: inputs shown in green, pipeline stages in blue, and per-stage outputs in purple.

test sets. The *models config* is a key-value mapping, where keys are model names and values are initialized models for analysis. Finally, the *config yaml* specifies configuration parameters.

Subgroup metric computation. Virny implements several interfaces for metric computation: an interface for multiple models, an interface for multiple test sets, and an interface for saving results into a database. The library incorporates a *Subgroup Variance Analyzer* and a *Subgroup Error Analyzer*, and it is easily extensible to encompass other analyzers. Once these analyzers finalize metric computation, their outputs are merged and returned as a pandas dataframe. Users have the option to specify a parameter for saving, allowing the metric dataframe to be stored on disk or in their database using the provided *df_writer* function.

The *Subgroup Error Analyzer* computes error metrics, including accuracy, F1, false-positive rate (FPR), and false-negative rate (FNR), on both the overall test set and on the subgroups of interest. The *Subgroup Variance Analyzer* computes stability and uncertainty metrics, both overall and for subgroups. To quantify estimator variance, we use bootstrapping [10]. We also compute a rich set of stability and uncertainty metrics, discussed in Section 2.2.

Disparity metric composition. The *Metric Composer* handles the second stage of the model profiling process, where it computes the error disparity, stability disparity, and uncertainty disparity metrics using the group-specific metrics computed in the previous stage. Users have the flexibility to compose additional metrics. For example, error disparity can be computed as the ratio of the Positive Rate on the privileged and disadvantaged subgroups.

Metric visualization. Virny provides two types of metric visualizers for building static and interactive visualizations: *Metric Visualizer* allows users to conveniently generate customized static visualizations for comprehensive metric analysis, while *Metric Interactive Visualizer* can be used to build an interactive web application that guides responsible model selection and generates nutritional labels for ML models. We use these visualizers to implement VirnyView.

2.2 Model performance metrics

Based on the influential work by Domingos [8], a model's error can be decomposed into (statistical) bias, variance, and noise. Accuracy metrics capture the extent of a model's bias, and have so far been used as the primary means of assessing model performance. However, these metrics fail to provide insights into model arbitrariness, which can be due to variance in its predictions, uncertainty in the data or in the model, or both. Therefore, Virny implements a wide range of accuracy, stability and uncertainty metrics.

Stability and uncertainty. Model stability is the property where small perturbations in the input cause proportionately small changes in the output. Several distinct approaches exist to quantify model stability, including the Bootstrap [10] and Jackknife [16]. These approaches construct probabilistic distributions for each test sample from the outputs of each trained model, rather than obtaining a single-point estimate. By computing measures of variation between the predictions of the ensemble of estimators for the same data point, we can quantify the instability of a single model fit on the full training set. This instability is expressed through such metrics as Label Stability [6], Jitter [15], and IQR (inter-quantile range of predictive variance), implemented in Virny.

Model uncertainty is related to how confident a model is in its predictions. Uncertainty can stem from two sources: the inherent randomness or variability in the data (*aleatoric*), or the lack of information, typically stemming from limitations in understanding the system or process being modeled (*epistemic*). Predictive variance is a common measure of epistemic uncertainty, whereas the expected entropy over all classes is a measure of aleatoric uncertainty [20].

Comparing performance across groups. It has been observed that ML models often exhibit unfairness in terms of selection rates or error distribution across groups (see, *e.g.*, [4, 9, 12, 13]), leading to the development of a number of *group fairness* metrics. Virny implements several such metrics, including disparate impact (parity in selection rates), accuracy parity, and parity in FPR and FNR, among others. Further, as we argued in prior work [1], popular fairness metrics in the ML literature do not explicitly capture whether the model is comparably arbitrary for different demographic groups. Virny computes several measures of stability-disparity and uncertainty-disparity to give a more comprehensive view of model fairness.

3 DEMONSTRATION SCENARIOS

We will start the demonstration with a brief overview of the Virny library. We will discuss the core principles behind the library, and will give a tour of the main architectural elements (see Section 2.1).

We will spend the bulk of the time with the interactive VirnyView application, implemented in Gradio, to present several *responsible model selection* scenarios based on three fair-ML benchmarks: ACS Income [7], ACS Public Coverage [7], and Law School [22]. For the demo, we pre-computed all metrics discussed in Section 2.2 based on 200 estimators, for each of the benchmarks.

Figure 2: VirnyView, Step 2 for ACS Income. The user selects two overall constraints C1&C2, and two disparity constraints C3&C4, and sets their min and max limits. Consequently, the bar chart shows the number of models that satisfy each constraint, all pairs of constraints, and all 4 constrains simultaneously. For ACS Income, only 4 out of 12 models satisfy all four constraints.

Users will be able to interact with any of the three benchmarks during the demo: select models to profile, specify sets of performance constraints, and conduct in-depth comparisons of performance of selected models based on various dimensions using intuitive visualizations with a clear color scheme and tolerance. Additionally, users will be able to break down the performance of a specific model concerning multiple protected groups and performance dimensions. We describe one such possible scenario below.

VirnyView consists of 6 visual components, each corresponding to a step in the responsible model selection pipeline. Users can interactively choose a specific combination of models, overall metrics, and disparity metrics across various model dimensions. This selection dynamically alters the visualization perspective.

Let us now step through these components to conduct *responsible model selection for ACS Income*, a dataset derived from US Census data and used to predict whether a person's income is $< \$50,000$ (label 0) or $>= \$50,000$ (label 1) [7]. For our demo, we used data from the US state of Georgia from 2018, and sub-sampled the data from 50,915 to 15,000 rows. We will showcase four model types and the DisparateImpactRemover fairness intervention, applied with repair levels of 0.0, 0.4, and 0.7 for each model type.

Step 1: Analyze demographic composition of the dataset. The application is structured from a high-level overview to a detailed examination. The upper screens provide general insights into the dataset and models, while the lower screens delve into the performance of individual models, broken down by protected groups. Thus, prior to delving into metric analysis, it is crucial to establish a comprehensive understanding of the proportions and base rates of the protected groups within the dataset. This information serves to elucidate potential disparities, for example, such as significant variations in overall accuracy and stability among different racial groups. In the ACS Income dataset, the overall base rate is 0.35, and there are no significant gaps in demographic composition: males vs. females (0.51 vs. 0.49) and white vs. black (0.68 vs. 0.32).

Step 2: Reduce the number of models to compare, based on overall and disparity metric constraints. Creating an accurate, robust, and fair model requires thorough validation of various model types, pre-processing techniques, and fairness interventions. However, the complexity arises when attempting to directly compare all models on a single plot or visualize every possible combination of the models. In this step, the user defines overall and disparity metric constraints to effectively narrow down the selection of models that meet these criteria. In this scenario for ACS Income dataset, we set Accuracy and Label Stability as overall constraints and Equalized Odds FNR and Label Stability Ratio as disparity constraints. Consequently, in Figure 2, we can see that only 4 out of 12 models satisfy all the four constraints (C1 & C2 & C3 & C4). This strategic reduction allows for a more detailed comparison of metrics, focusing on a manageable number of models in subsequent steps.

Steps 3-4: Compare models that satisfy all constraints using overall and disparity metric heatmaps. Figures 3 and 4 show overall and disparity heatmap comparisons of 4 models, respectively. Green signifies the most favorable model metric and red denotes the least favorable, compared to other models. Crucially, the color scheme takes into account that a score of 1.0 is considered best for Disparate Impact, while 0.0 is best for Equalized Odds FNR. Furthermore, users have the option to introduce a tolerance parameter to the comparison process. This means that if the discrepancy between metrics of different models falls below the tolerance threshold, these models are grouped together (*i.e.*, they are considered to be tied). This is beneficial when minor differences, such as 0.001%, can be considered negligible.

The overall heatmap for ACS Income (Figure 3) shows that Logistic Regression with repair level 0.7 has best performance. However, the disparity heatmap (Figure 4) shows a substantial drawback for Logistic Regression — high error disparity on Equalized Odds FNR for *sex* (0.043) and *race* (-0.086). Random Forest with repair level 0.4 shows the best performance based on disparity dimensions. In

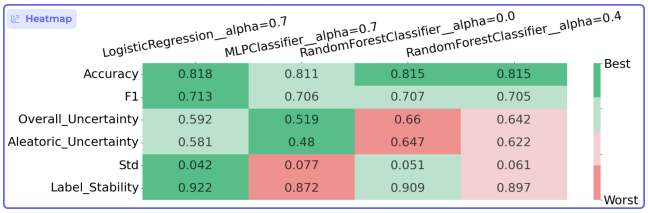


Figure 3: VirnyView Step 3: Overall heatmap with tolerance 0.005 for ACS Income and 4 models.

Figure 4: VirnyView Step 4: Disparity heatmap with tolerance 0.005 for ACS Income and 4 models.

subsequent steps, we will drill into the trade-offs between these models, and will generate comprehensive nutritional labels.

Step 5: Generate a nutritional label for the selected model. In the fifth step, users choose a particular model and a combination of overall and disparity metrics to generate a nutritional label, segmented by multiple protected groups and performance dimensions. The nutritional label includes bar charts for the overall and disparity metrics presented side-by-side that helps to find interesting insights between them. This graphical representation proves particularly effective in identifying performance gaps among binary or intersectional groups. For instance, the bar charts depicting the Logistic Regression with repair level 0.7 reveal notable disparities in Accuracy between racial dis and priv groups (0.871 vs. 0.792), and in F1 scores between race-based (0.747 vs. 0.701), sex-based groups (0.674 vs 0.739) groups. (See hosted demo for visualization.)

Step 6: Summarize performance of a selected model across different dimensions and protected groups. In our scenario (see hosted demo for visualization), the summary clearly displays differences between Logistic Regression (repair level 0.7) and Random Forest (repair level 0.4). The Logistic Regression excels across all overall and disparity dimensions, except for substantial error disparity in Equalized Odds FNR based on *race* (-0.086). On the other hand, while Random Forest has slightly inferior stability and uncertainty, it significantly outperforms the Logistic Regression in terms of error disparity. The decision between the two models depends on the specific context of use. If the application domain is highly sensitive to model uncertainty or instability, then Logistic Regression is preferable; otherwise, Random Forest may be a better choice.

This scenario underscores a clear trade-off among performance dimensions in practical settings, emphasizing the importance of measuring diverse overall and disparity performance dimensions, and showcasing the practical utility of Virny and VirnyView.

ACKNOWLEDGMENTS

This work was supported in part by NSF awards 1916505, 1922658, 2312930, and 2326193, by UL Research Institutes through the Center for Advancing Safety of Machine Intelligence, and by the NYU Center for Responsible AI.

REFERENCES

[1] Falaah Arif Khan, Denys Herasymuk, and Julia Stoyanovich. 2023. On Fairness and Stability: Is Estimator Variance a Friend or a Foe? *CoRR* abs/2302.04525 (2023). https://doi.org/10.48550/ARXIV.2302.04525 arXiv:2302.04525
[2] Rachel K. E. Bellamy et al. 2019. AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM J. Res. Dev.* 63, 4/5 (2019), 4:1–4:15. https://doi.org/10.1147/JRD.2019.2942287
[3] Irene Chen, Fredrik D Johansson, and David Sontag. 2018. Why is my classifier discriminatory? *Advances in neural information processing systems* 31 (2018).
[4] Alexandra Chouldechova. 2017. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data* 5, 2 (2017), 153–163. https://doi.org/10.1089/big.2016.0047
[5] Anamaria Crisan, Margaret Drouhard, Jesse Vig, and Nazneen Rajani. 2022. Interactive model cards: A human-centered approach to model documentation. In *ACM FAccT*. 427–439.
[6] Michael C. Darling and David J. Stracuzzi. 2018. Toward Uncertainty Quantification for Supervised Classification.
[7] Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. 2021. Retiring adult: New datasets for fair machine learning. *NeurIPS* 34 (2021), 6478–6490.
[8] Pedro Domingos. 2000. A Unifeid Bias-Variance Decomposition and its Applications. 231–238.
[9] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. 2015. Preserving statistical validity in adaptive data analysis. In *ACM TC*. 117–126.
[10] Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.
[11] A Feder Cooper, Solon Barocas, Christopher De Sa, and Siddhartha Sen. 2023. Variance, Self-Consistency, and Arbitrariness in Fair Classification. *arXiv e-prints* (2023), arXiv–2301.
[12] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and Removing Disparate Impact. In *ACM SIGKDD (KDD '15)*. 259–268. https://doi.org/10.1145/2783258.2783311
[13] Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2017. Inherent Trade-Offs in the Fair Determination of Risk Scores. In *ITCS (LIPIcs, Vol. 67)*. 43:1–43:23. https://doi.org/10.4230/LIPIcs.ITCS.2017.43
[14] Yan Li, Matthew Sperrin, Darren M Ashcroft, and Tjeerd Pieter Van Staa. 2020. Consistency of variety of machine learning and statistical models in predicting clinical risks of individual patients: longitudinal cohort study using cardiovascular disease as exemplar. *BMJ* 371 (2020).
[15] Huiting Liu, Avinesh P. V. S., Siddharth Patwardhan, Peter Grasch, and Sachin Agarwal. 2022. Model Stability with Continuous Data Updates. *CoRR* abs/2201.05692 (2022). arXiv:2201.05692 https://arxiv.org/abs/2201.05692
[16] Rupert G Miller. 1974. The jackknife-a review. *Biometrika* 61, 1 (1974), 1–15.
[17] Margaret Mitchell et al. 2019. Model cards for model reporting. In *ACM FAT*.
[18] Pedro Saleiro, Benedict Kuester, Loren Hinkson, Jesse London, Abby Stevens, Ari Anisfeld, Kit T Rodolfa, and Rayid Ghani. 2018. Aequitas: A bias and fairness audit toolkit. *arXiv preprint arXiv:1811.05577* (2018).
[19] Julia Stoyanovich and Bill Howe. 2019. Nutritional Labels for Data and Models. *IEEE Data Eng. Bull.* 42, 3 (2019).
[20] Anique Tahir, Lu Cheng, and Huan Liu. 2023. Fairness through Aleatoric Uncertainty. In *ACM CIKM*. 2372–2381. https://doi.org/10.1145/3583780.3614875
[21] Hilde J. P. Weerts, Miroslav Dudík, Richard Edgar, Adrin Jalali, Roman Lutz, and Michael Madaio. 2023. Fairlearn: Assessing and Improving Fairness of AI Systems. *J. Mach. Learn. Res.* 24 (2023). http://jmlr.org/papers/v24/23-0389.html
[22] Linda F Wightman. 1998. LSAC National Longitudinal Bar Passage Study. (1998).