Interpretable Machine Learning: Methods and Challenges

Oscar Gomez

Agenda

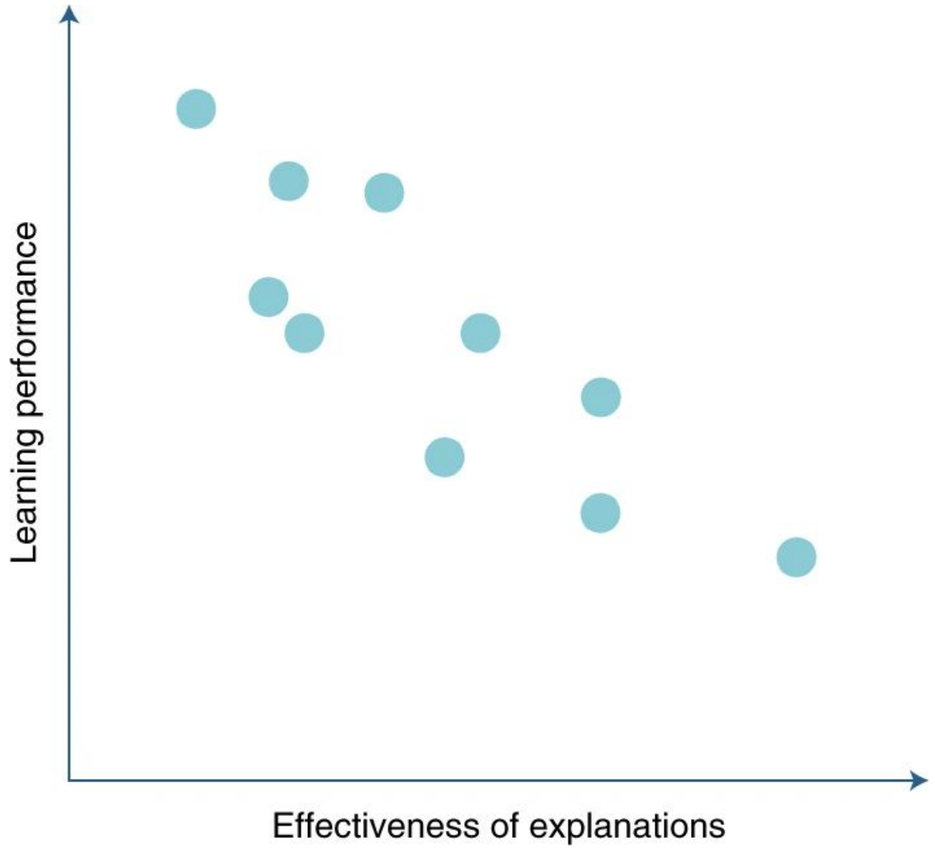
1. Introduction and motivation
2. Overview of Methods
3. Local Feature Importance Methods
    a. LIME
    b. SHAP
4. Counterfactual Methods
    a. DiCE
5. Interactive Tools
    a. FICO xML Challenge
    b. ViCE
    c. AdViCE
6. Challenges
7. Resources

Need for interpretability

- Simple accuracy measures often fail to describe deeper flaws such as hidden biases and false generalizations.
- Accountability, regulation (GDPR right to explanation)
- High stakes settings - can ML help make better decisions?
- Aid in detecting errors / model debugging
  - Choosing between models
- High level of variability between
  - Types of data
  - Involved actors

Interpretability / Accuracy tradeoff

- Simple models do not have the predictive power of more complex ones



https://www.nature.com/articles/s42256-019-0048-x/figures/1

Overview of Methods / Taxonomy

White vs Black boxes

- White-box models are those intrinsically interpretable models, where the logic of making a decision is transparent and intelligible.
  - Decision trees, linear regression

- Black-box models tend to have complex structures and are hard to understand
  - Deep Neural Networks, Ensemble Models
  - Post-Hoc techniques

Global vs Local Explanations

- Local explanations try to explain how a decision is made for a specific instance.
  - LIME and SHAP (weight for each feature), counterfactuals

- Global explanation methods refer to showing the overall logical structure of a model.

Overview of Methods

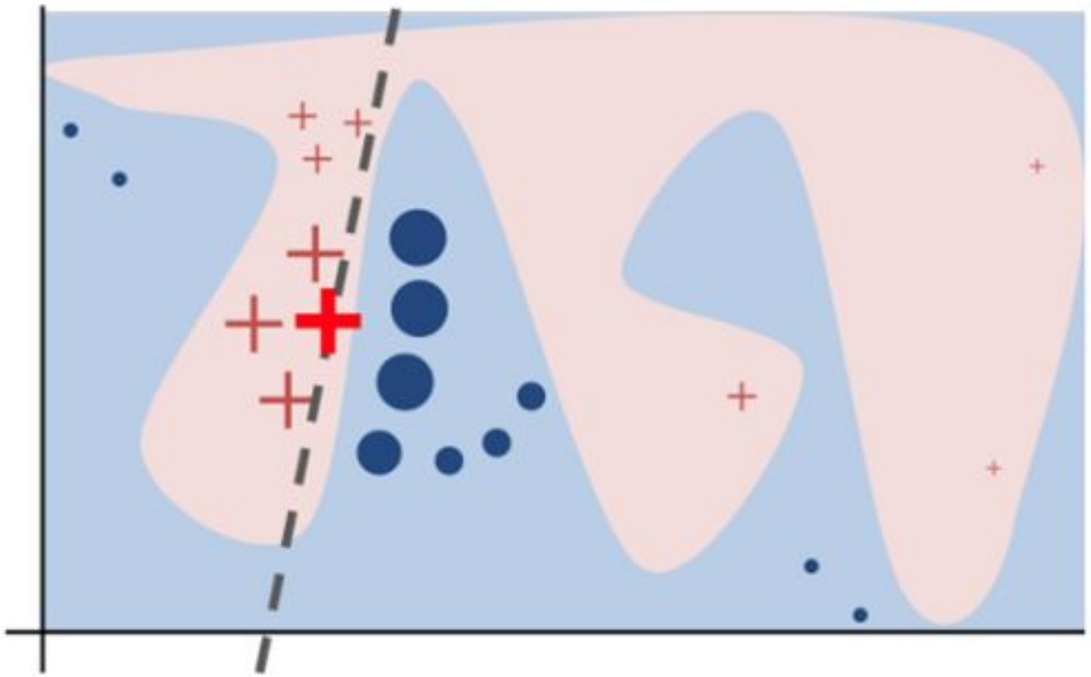
Interpretable Models

- Decision trees, linear regression, decision rules, GLMs.
  - Still might be hard to interpret

Model agnostic (Post-hoc)

- Partial Dependence Plots (PDP)
  - Shows the marginal effect one or two features have on the predicted outcome of a machine learning model
- Global surrogate model
  - Interpretable model that is trained to approximate the predictions of a black box model
- Local feature importance
  - Gives relative importance magnitudes of features around a desired point
- Counterfactuals
  - Provide sets of changes that alter the model's prediction

LIME

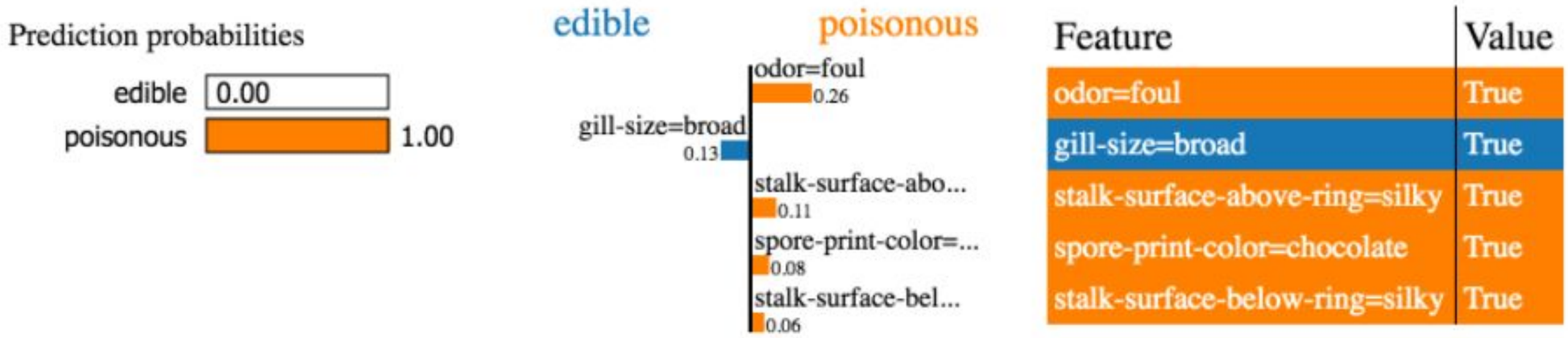


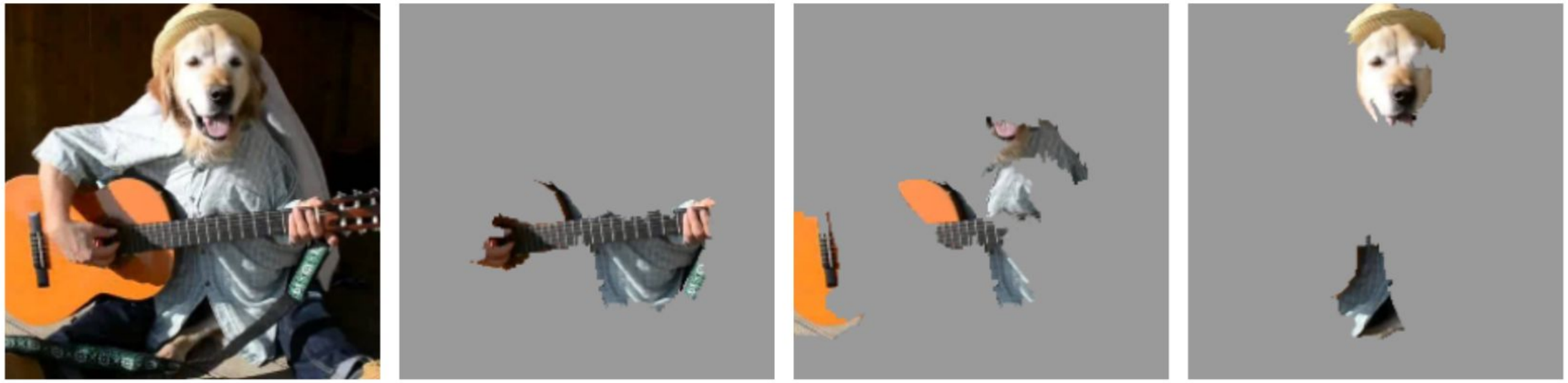
$$\text{explanation}(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

- Generates a new dataset consisting of permuted samples and the corresponding predictions of the black box model.

- Trains an interpretable model, which is weighted by the proximity of the sampled instances to the instance of interest

LIME

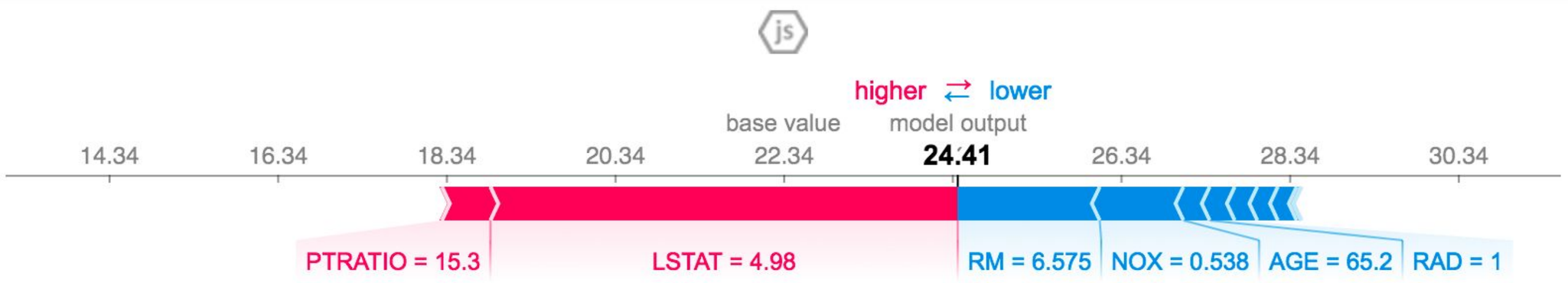




(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

https://github.com/marcotcr/lime

SHAP



- Generalizes LIME, unifies it with Shapley values from coalitional game theory
- Considers effects of the feature values as they move the prediction away from the mean.
- Satisfies desirable properties (only feature attribution method that does so)
  - Consistency, missingness, local accuracy
- Intractable to compute exactly, uses perturbation methods
  - Efficient and more accurate methods available for some models like Trees
- Has been widely adopted in practice

https://github.com/slundberg/shap

Counterfactual Methods

- Describes the smallest change to the feature values that changes the prediction to a predefined output.
- Provide complete fidelity to the underlying model
- Only require query access
- General approach by Wachter et. al (2017)

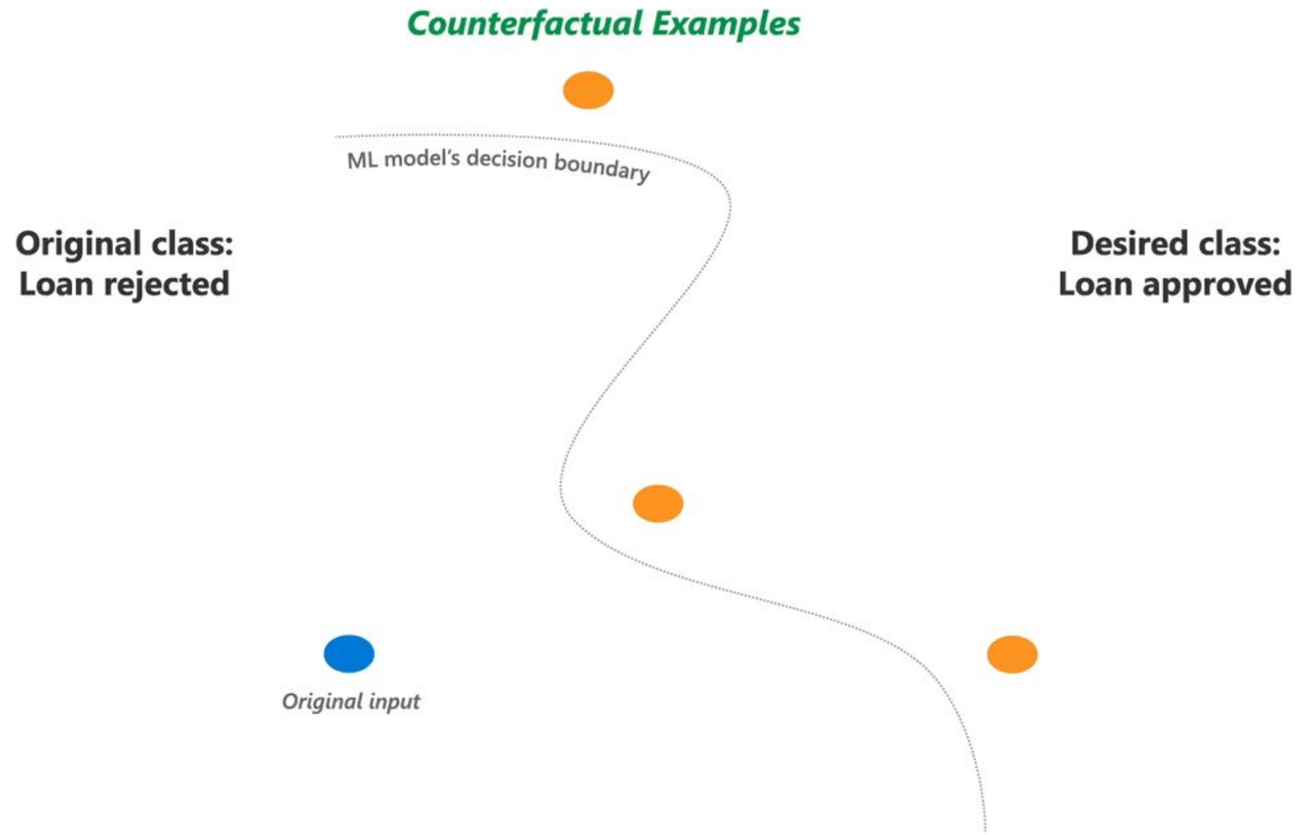
$$L(x, x', y', \lambda) = \lambda \cdot (\hat{f}(x') - y')^2 + d(x, x')$$

- Are never really observed in real life.

DiCE: Diverse Counterfactual Explanations

- Method for generating numerous diverse counterfactuals, which takes into account usefulness and relative ease.

- Ramaravind K. Mothilal, Amit Sharma, Chenhao Tan



https://www.microsoft.com/en-us/research/blog/open-source-library-provides-explanation-for-machine-learning-through-diverse-counterfactuals/

Objectives

- Framework for generating and evaluating counterfactual explanations.
  - Based on **determinantal point processes**
  - Properties: **feasibility, diversity**

- Provide **metrics** for evaluating and comparing counterfactual explanations
  - Among sets of counterfactuals
  - Against other explanation methods (LIME)

- Demonstrate effectiveness with **experiments** on multiple datasets.

Background

- Explanations based on key features or **feature importance** don't help people decide what to do next.
  - Many based on **proxy** models - "lie" because of tradeoff between interpretability and truthfulness
    - Ex: LIME, SHAP, LORE, Global Surrogates
  - Explanations through **visualization** (image highlights or activations in CNNs) difficult in scenarios that are not inherently visual.
  - Counterfactual explanations **provide truthfulness** to the model
    - Are human interpretable by letting user explore **what-if scenarios**
- CF can be useful to:
  - End-users
  - model builders, fairness evaluators (debugging biases)

Desired Properties of Counterfactuals

Actionability, validity

Diversity
- Should provide a set of examples as an example-based decision support system
- Generate any number of CF examples for an input

Proximity

Follow causal laws of human society

Support user-provided inputs
- Custom weights for features
- Constraints on perturbations

Assumptions

- ML model remains relatively **static**
- **Binary** classification for **differentiable models**
- CF does **not** have **causal knowledge** of features they modify
  - Perturbing features independently can lead to infeasible examples
    - Ex: Obtaining a higher degree without aging
- Work based on formulation by Watcher et. al.

$$\boldsymbol{c} = \arg\min_{\boldsymbol{c}} \text{yloss}(f(\boldsymbol{c}), y) + |\boldsymbol{x} - \boldsymbol{c}|, \quad (1)$$

where the first part (yloss) pushes the counterfactual $\boldsymbol{c}$ towards a different prediction than the original instance, and the second part keeps the counterfactual close to the original instance.

Terminology

ML model (f)	The trained model obtained from the training data.
Original input ($\boldsymbol{x}$)	The feature vector associated with an instance of interest that receives an unfavorable decision from the ML model.
Original outcome	The prediction of the original input from the trained model, usually corresponding to the undesired class.
Original outcome class	The undesired class.
Counterfactual example ($\boldsymbol{c}_i$)	An instance (and its feature vector) close to the original input that would have received a favorable decision from the ML model.
CF class	The desired class.

Table 1: Terminology used throughout the paper.

Diversity and Feasibility

$$dpp\_diversity = det(\mathbf{K}), \tag{2}$$

where $\mathbf{K}_{i,j} = \frac{1}{1+dist(\boldsymbol{c}_i, \boldsymbol{c}_j)}$ and $dist(\boldsymbol{c}_i, \boldsymbol{c}_j)$ denotes a distance metric between the two counterfactual examples.

$$Proximity := -\frac{1}{k}\sum_{i=1}^{k} dist(\boldsymbol{c}_i, \boldsymbol{x}).$$

- **Sparsity** handled after the CF generation
- User constraints:
  - Feasible **ranges** for each feature
  - Variables that cannot be **changed**

Optimization

$$C(\boldsymbol{x}) = \underset{\boldsymbol{c}_1,\ldots,\boldsymbol{c}_k}{\arg\min} \frac{1}{k} \sum_{i=1}^{k} \text{yloss}(f(\boldsymbol{c}_i), y) + \frac{\lambda_1}{k} \sum_{i=1}^{k} dist(\boldsymbol{c}_i, \boldsymbol{x}) - \lambda_2 \, \text{dpp\_diversity}(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k) \quad (4)$$

- Combined loss function
  - Non-convex (can't always achieve f(c) = y)

- Optimized with gradient descent
  - 5000 steps max.
  - c_i initialized randomly

Implementation

YLoss

$$hinge\_yloss = max(0, 1 - z * logit(f(\boldsymbol{c}))),$$

where z is -1 when $y = 0$ and 1 when $y = 1$, and $logit(f(\boldsymbol{c}))$ is the unscaled output from the ML model (e.g., final logits that enter a softmax layer for making predictions in a neural network).

Distance Functions

$$\text{dist\_cont}(\boldsymbol{c}, \boldsymbol{x}) = \frac{1}{d_{cont}} \sum_{p=1}^{d_{cont}} \frac{|\boldsymbol{c}^p - \boldsymbol{x}^p|}{MAD_p}, \qquad \text{dist\_cat}(\boldsymbol{c}, \boldsymbol{x}) = \frac{1}{d_{cat}} \sum_{p=1}^{d_{cat}} I(\boldsymbol{c}^p \neq \boldsymbol{x}^p),$$

Evaluation

Validity

$$\%\text{Valid-CFs} = \frac{|\{\text{unique instances in } C \text{ s.t. } f(c) > 0.5\}|}{k}$$

Proximity

$$\text{Continuous-Proximity} : -\frac{1}{k}\sum_{i=1}^{k} \text{dist\_cont}(\boldsymbol{c}_i, \boldsymbol{x}), \qquad (7)$$

$$\text{Categorical-Proximity} : 1 - \frac{1}{k}\sum_{i=1}^{k} \text{dist\_cat}(\boldsymbol{c}_i, \boldsymbol{x}), \qquad (8)$$

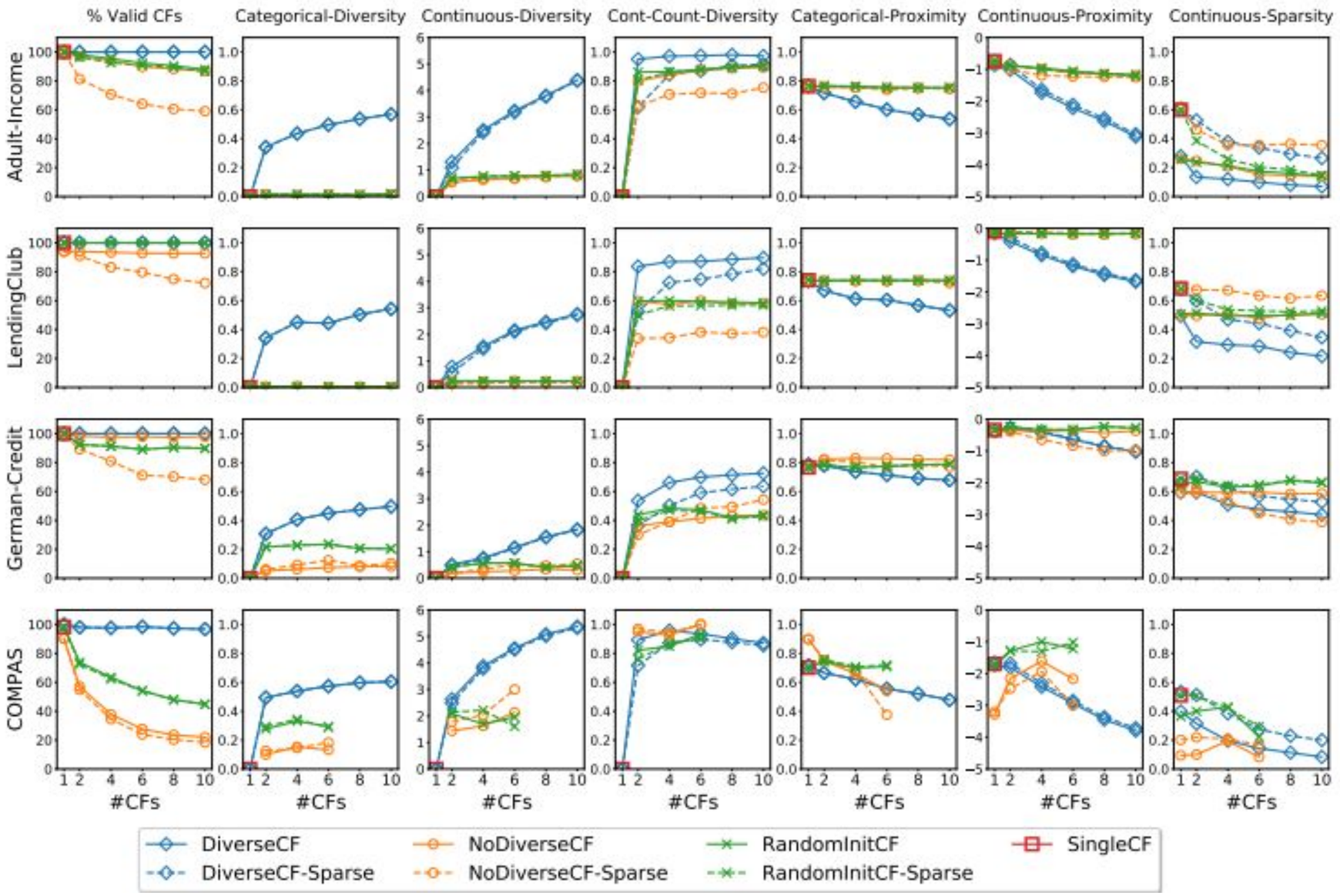
Sparsity

$$\text{Sparsity} : 1 - \frac{1}{kd}\sum_{i=1}^{k}\sum_{l=1}^{d} 1_{[\boldsymbol{c}_i^l \neq \boldsymbol{x}_i^l]}$$

Diversity

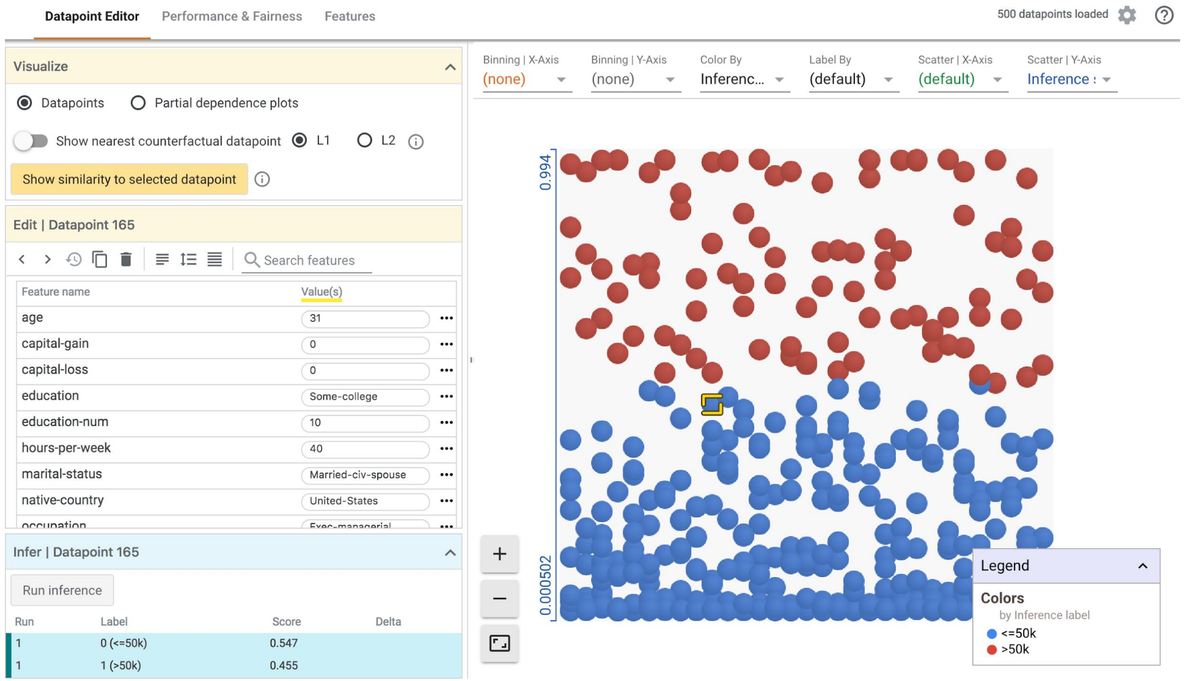
$$\text{Diversity} : \Delta = \frac{1}{C_k^2}\sum_{i=1}^{k-1}\sum_{j=i+1}^{k} \text{dist}(\boldsymbol{c}_i, \boldsymbol{c}_j),$$

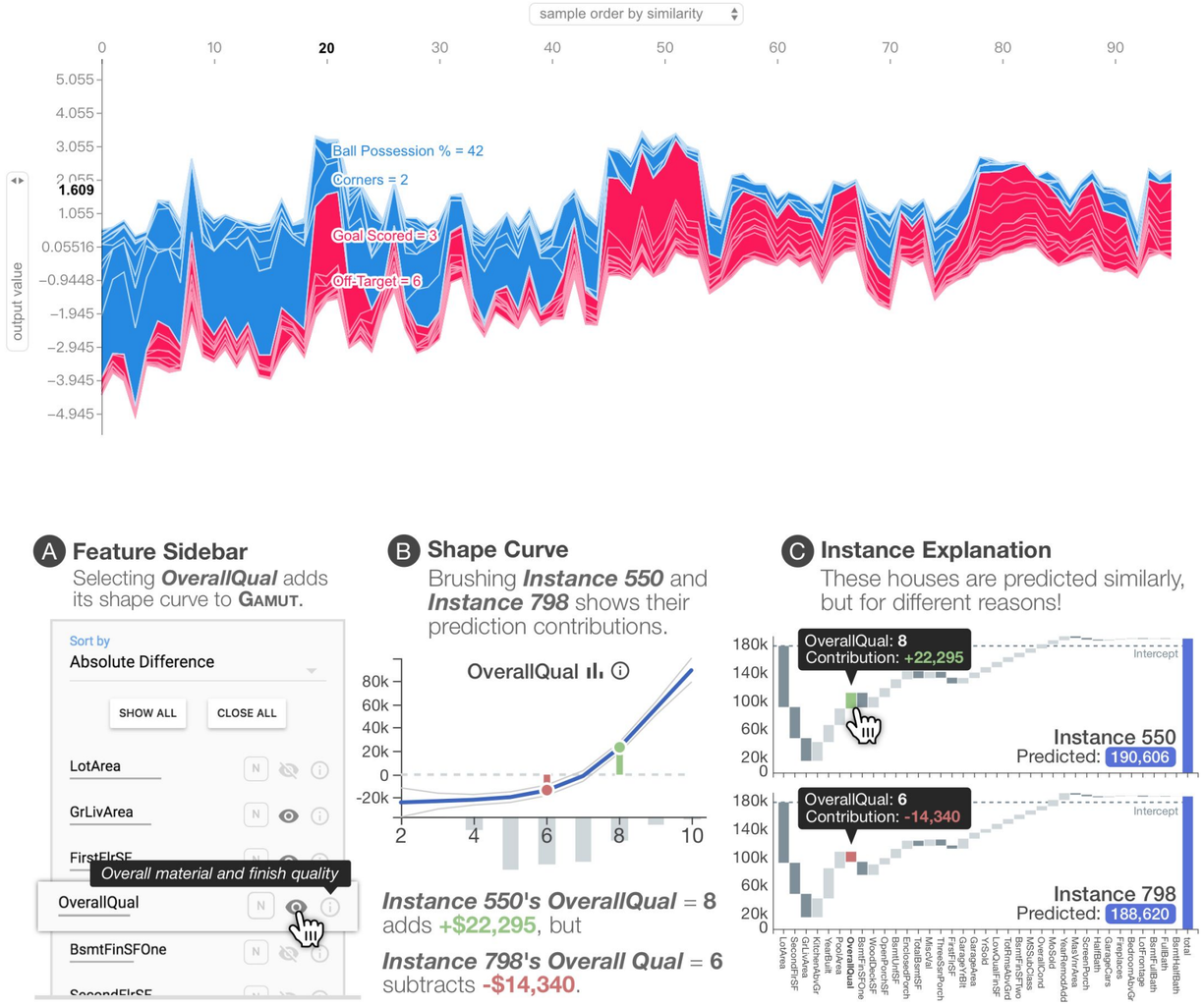
Results


% Valid CFs
Categorical-Diversity
Continuous-Diversity
Cont-Count-Diversity
Categorical-Proximity
Continuous-Proximity
Continuous-Sparsity
Adult-Income
LendingClub
German-Credit
COMPAS
#CFs
DiverseCF
DiverseCF-Sparse
NoDiverseCF
NoDiverseCF-Sparse
RandomInitCF
RandomInitCF-Sparse
SingleCF

Interactive Tools

- **Visualization** and **interactive** tools have been increasingly used to support **understanding**, **debugging**, **verification**, and **refinement** of ML models.

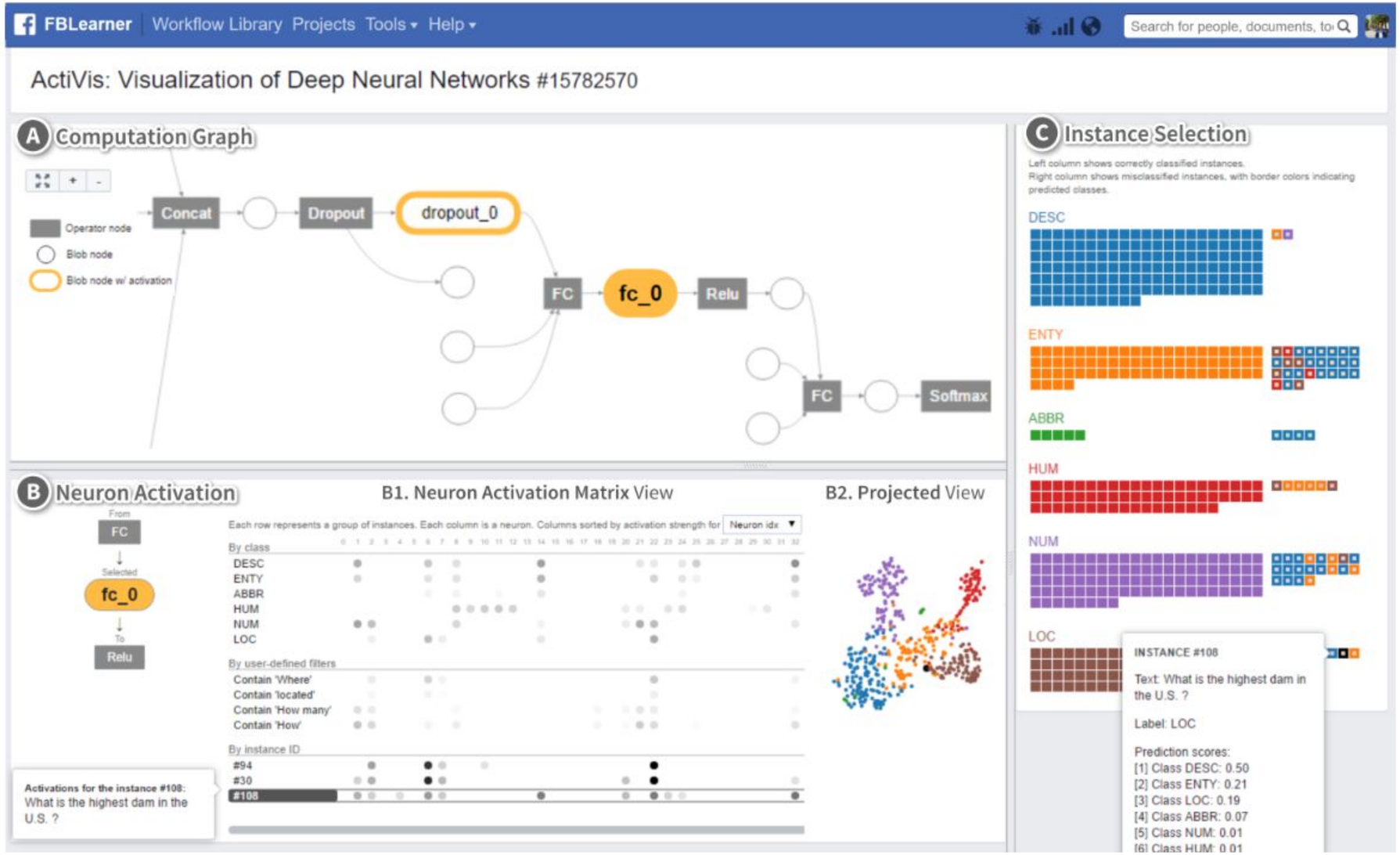




Interactive Tools

- Visualizations of **feature importance**
- Google **What-If Tool**
- Gamut: A Design Probe to Understand How Data Scientists Understand Machine Learning Models
- Many others

Interactive Tools

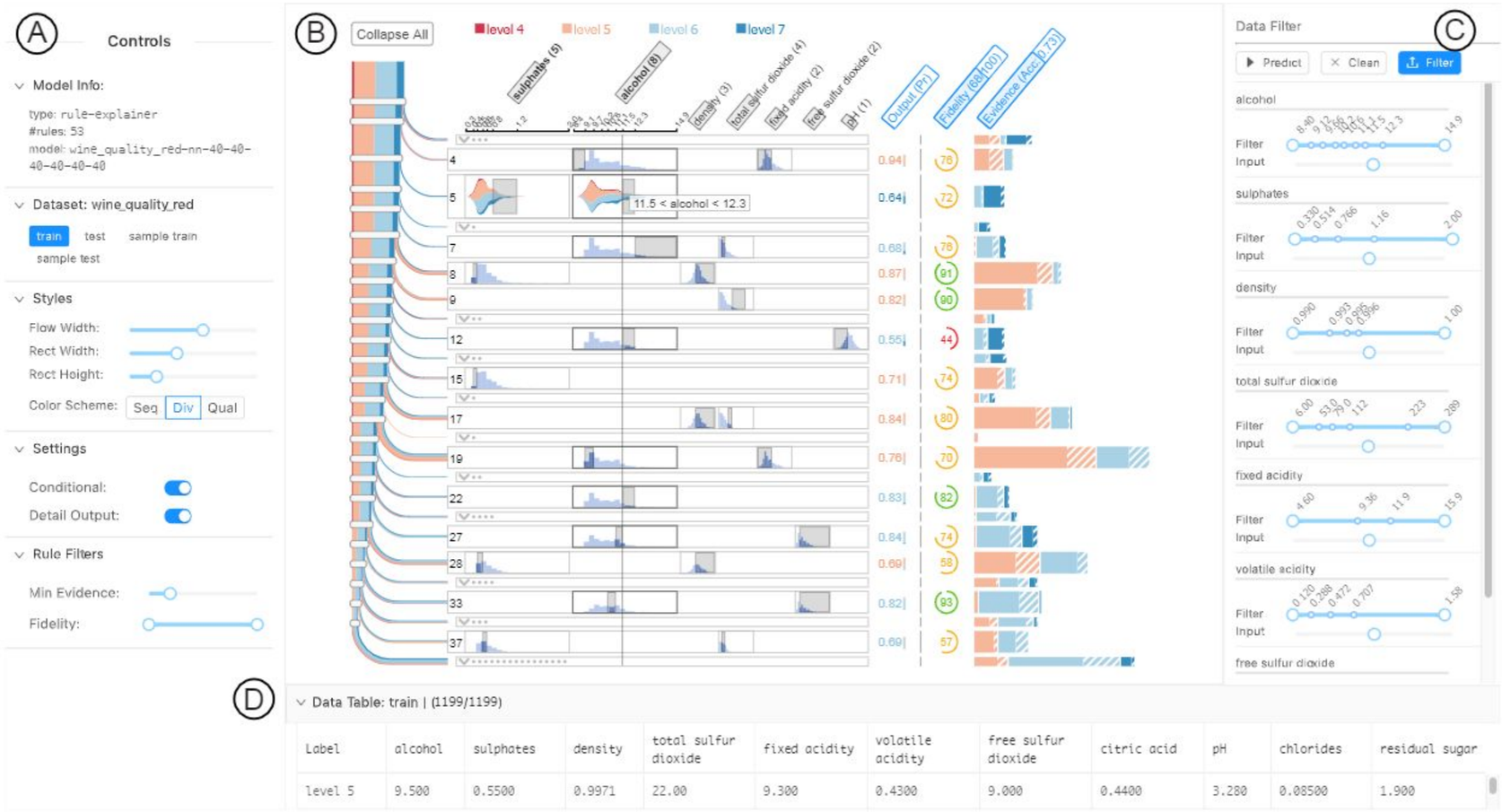


https://minsuk.com/research/papers/kahng-activis-vast2017.pdf

Interactive Tools

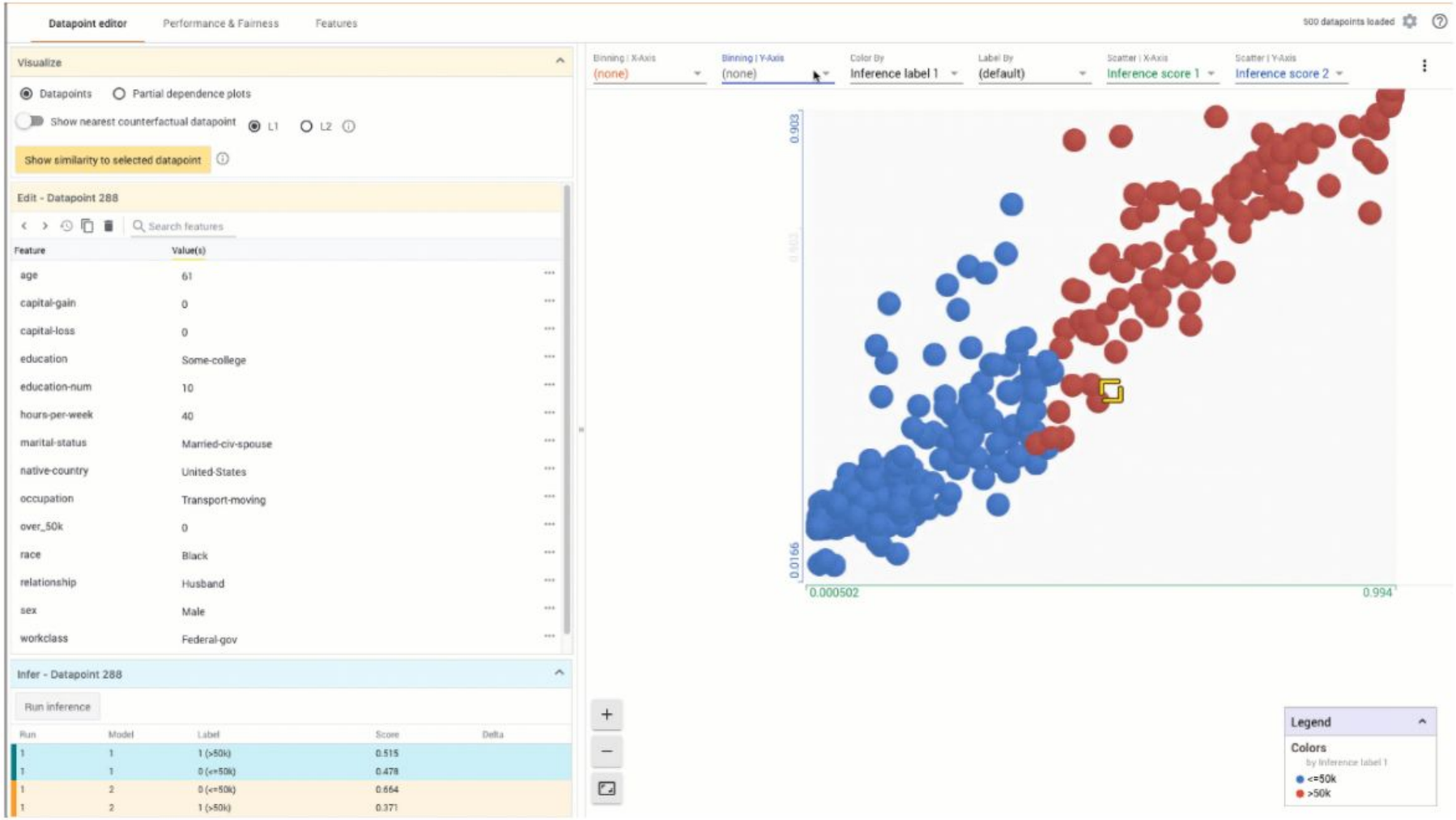
RuleMatrix: Visualizing and Understanding Classifiers with Rules

Yao Ming, Huamin Qu, *Member, IEEE*, and Enrico Bertini, *Member, IEEE*



/https://arxiv.org/pdf/1807.06228.pdf

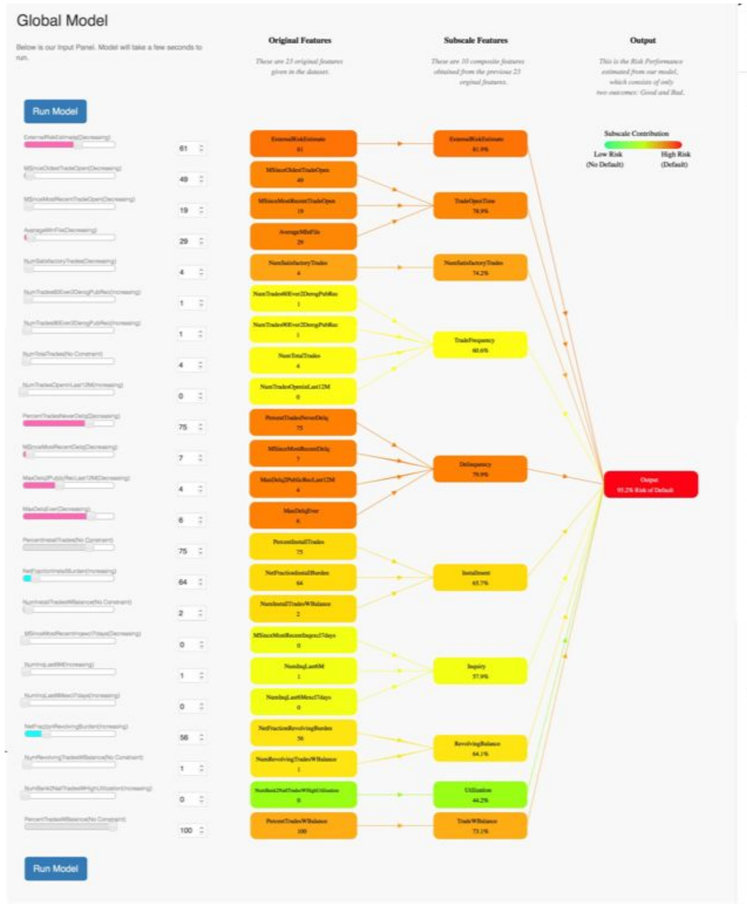
What-If



https://pair-code.github.io/what-if-tool/

FICO xML Challenge

- Participants were challenged to create machine learning models with both high accuracy and explainability using a real-world dataset provided by FICO.
- Empirical evaluation method that considered how useful explanations are for a data scientist with the domain knowledge in the absence of model prediction.



Consistent Rule-based Explanations

The system solves an optimization problem (using Gurobi(c)) to compute the smallest set of rules that guarantees identical prediction by our global model.

For all 594 people whose:

- ExternalRiskEstimate is 63 or less

  and

- AverageMInFile is 48 or less

all of them were predicted to default.

https://community.fico.com/s/blog-post/a5Q2E0000001czyUAA/fico1670

FICO xML Challenge
Approach

Client focused solution

- Useful feedback to clients
- Reasons for decision
- Suggestions for improvement / warnings

Visual Interface

- Aggregation / exploration of individual explanations
- Customizable screen

Dataset

- Cleaned version of the FICO dataset. 10459 data points, 23 predictor features and 1 target feature.
- Anonymized Home Equity Line of Credit applications.
- Target is Risk Performance, indicates if the customer paid the credit as established.
- Given with monotonicity constraints and special values.

Features	Description	Monotonicity Constraint
ExternalRiskEstimate	Consolidated version of risk markers	Monotonically Decreasing
MSinceOldestTradeOpen	Months Since Oldest Trade Open	Monotonically Decreasing
MSinceMostRecentTradeOpen	Months Since Most Recent Trade Open	Monotonically Decreasing
AverageMInFile	Average Months in File	Monotonically Decreasing
NumSatisfactoryTrades	Number Satisfactory Trades	Monotonically Decreasing
NumTrades60Ever2DerogPubRec	Number Trades 60+ Ever	Monotonically Increasing
NumTrades90Ever2DerogPubRec	Number Trades 90+ Ever	Monotonically Increasing
PercentTradesNeverDelq	Percent Trades Never Delinquent	Monotonically Decreasing
MSinceMostRecentDelq	Months Since Most Recent Delinquency	Monotonically Decreasing
MaxDelq2PublicRecLast12M	Max Delq/Public Records Last 12 Months. See tab "MaxDelq" for each (	Values 0-7 are monotonically decreasing
MaxDelqEver	Max Delinquency Ever. See tab "MaxDelq" for each category	Values 2-8 are monotonically decreasing
NumTotalTrades	Number of Total Trades (total number of credit accounts)	No constraint
NumTradesOpeninLast12M	Number of Trades Open in Last 12 Months	Monotonically Increasing
PercentInstallTrades	Percent Installment Trades	No constraint
MSinceMostRecentInqexcl7days	Months Since Most Recent Inq excl 7days	Monotonically Decreasing
NumInqLast6M	Number of Inq Last 6 Months	Monotonically Increasing
NumInqLast6Mexcl7days	Number of Inq Last 6 Months excl 7days. Excluding the last 7 days remo	Monotonically Increasing
NetFractionRevolvingBurden	Net Fraction Revolving Burden. This is revolving balance divided by cred	Monotonically Increasing
NetFractionInstallBurden	Net Fraction Installment Burden. This is installment balance divided by (	Monotonically Increasing
NumRevolvingTradesWBalance	Number Revolving Trades with Balance	No constraint
NumInstallTradesWBalance	Number Installment Trades with Balance	No constraint
NumBank2NatlTradesWHighUtilization	Number Bank/Natl Trades w high utilization ratio	Monotonically Increasing
PercentTradesWBalance	Percent Trades with Balance	No constraint

Legend

Categorical features

Features with -8 special value

Features with -7 and -8 special value

Machine Learning Model
Training & Pre-processing

Pre-processing Data

- Omit redundant data:
  - Samples with all the fields with -9 value (not investigated or not found)
- Linear Regression:
  - Samples with -9 values for External Risk Estimate
- k-NN Imputation:
  - Samples with -8 values (no usable / valid accounts)
- Approximation:
  - Samples with -7 values (condition not met)
- Standardization of categorical values

Model

- SVM (Linear Kernel)
- Test accuracy:
  - ~68% before pre-processing
  - ~74.8% after processing

Minimal Set of Changes

- Suggest the fewest changes to flip a decision.
- Greedy procedure that optimizes the change in the model's prediction at each step.
- Similar to Martens et. al (2014), explaining why a document was or was not classified as a particular class.

Key Features

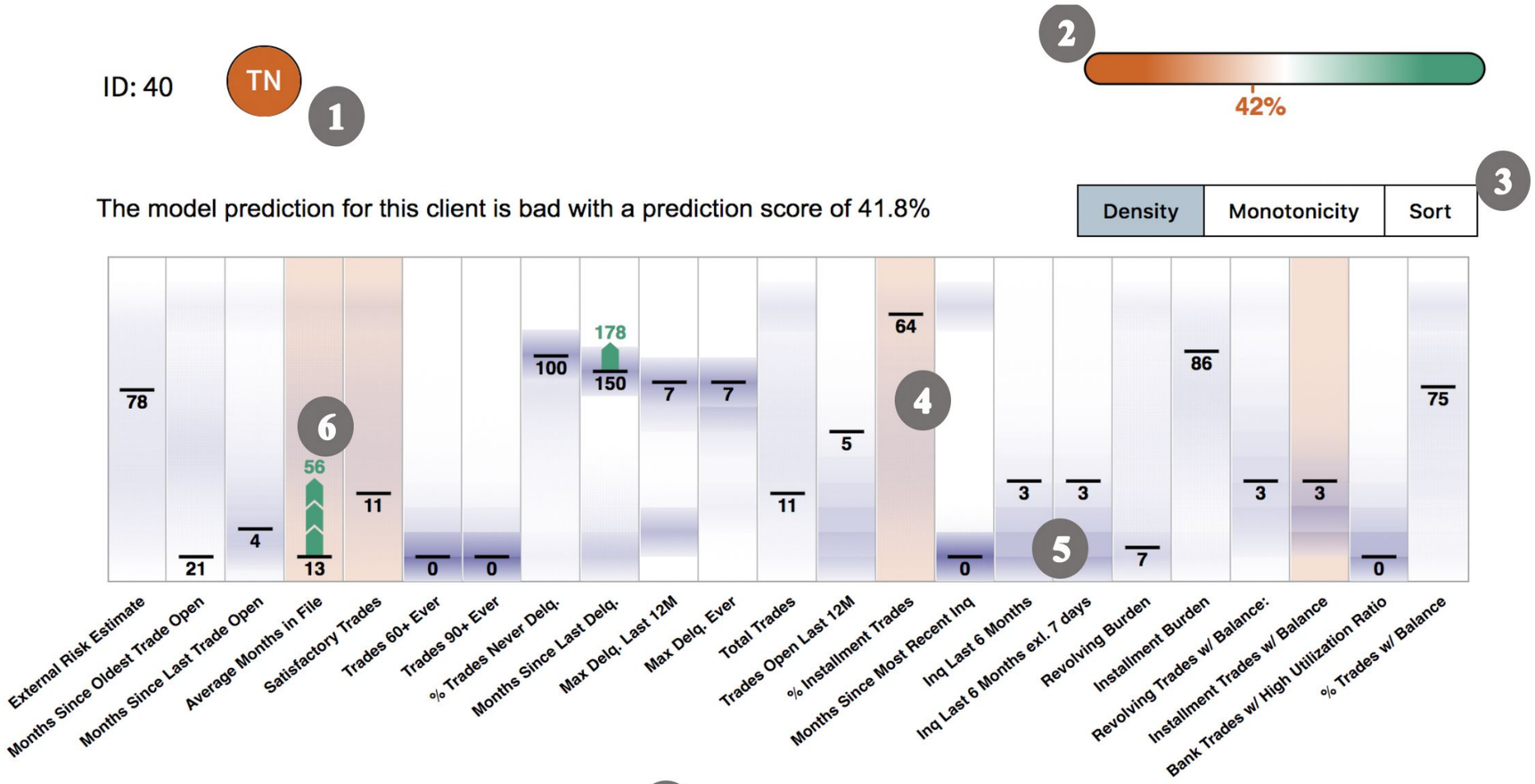
- Based on Anchors by Ribeiro et. al (2018)
- Systematically perturbing a sample instance and measuring the resistance to change against a predetermined threshold.
- Highlighting the features that are of paramount importance for the model.
- Fixing one feature at a time and perturbing all the other columns by their respective Gaussians
- To add a dimension to the visualization a density estimation was performed to highlight the data distribution.

Data Discretization

- Distribute numerical features into ten bins.
- Range of two standard deviations below the mean to two above it.

Individual Explanation
Client Overview

1. Classification correctness
2. Model's percentage prediction
3. Buttons that allow modifying the display
4. Highlights a key feature for this decision
5. Shows the density distribution
6. Minimum changes needed to reverse the decision
7. Text version of the explanation



http://oscargomezg.pythonanywhere.com/intro

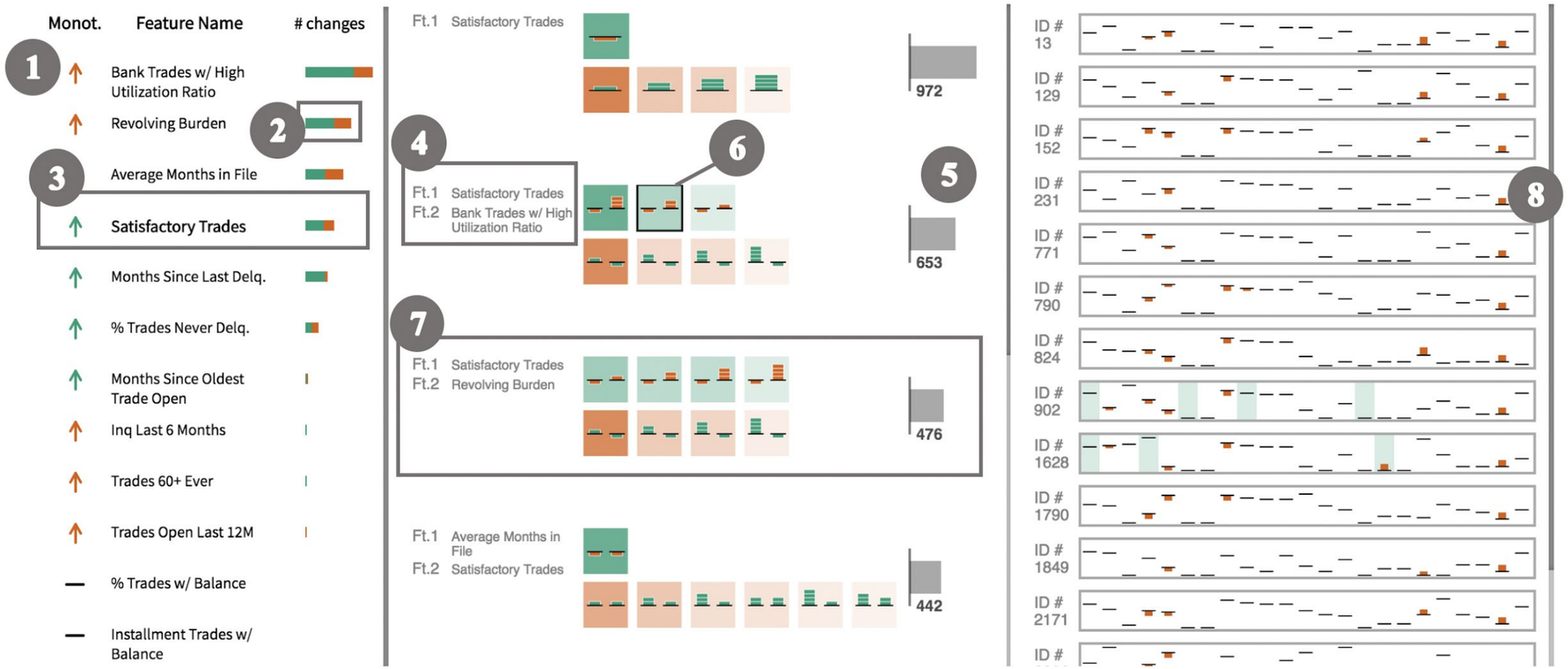
Global Explanation
Key Features

1. Monotonicity of the feature.
2. Number of samples where this feature is key
3. Selected feature(s)
4. Combination of features used for explanation
5. Total number of samples with these changes
6. All samples where such combination of changes is present
7. Set of samples explained by 4)
8. Miniature individual explanation

http://oscargomezg.pythonanywhere.com/intro

Global Explanation
Necessary Changes

1. Monotonicity of the feature
2. Number of samples where this feature is key
3. Selected feature(s)
4. Combination of features used for explanation
5. Total number of samples with these changes
6. All samples where such combination of changes is present
7. Set of samples explained by 4)
8. Miniature individual explanation



http://oscargomezg.pythonanywhere.com/intro

ViCE: Visual Explanations for Machine Learning Models

Oscar Gomez, Steffen Holter, Jun Yuan, Enrico Bertini

جامعة نيويورك أبوظبي
NYU | ABU DHABI

VIDA VISUALIZATION IMAGING AND DATA ANALYSIS CENTER

Introduction

ViCE

- A novel design for an explainable machine learning visual analytics tool.

- End-user: the client-facing person trying to better understand predictions made by the model. This could include doctors inferring why a patient is predicted as high risk for diabetes or admissions officers looking into why a particular candidate was rejected.

Introduction

Counterfactuals

- New algorithm for calculating counterfactuals that is not limited to binary variables and is intended for use with tabular numerical data.
- First visual interface that is able to display these explanations effectively and coherently.
- Supplemented with functionality that contextualizes the targeted sample with regards to the rest of the dataset.
- Interface does not only clarify the model's decision but can also be used to pinpoint bias and undesired behaviour.

Usage

- Each explanation provides actionable suggestions that can help adjust the model's prediction. For example, it could be used by a loan-officer looking to get a previously rejected application approved.

Goals

Overall

- Support understanding of individual predictions through counterfactual explanations and to provide an intuitive visual representation for them.

- What is the minimal set of changes that is required to change the prediction?
  i. Which features need to change?
  ii. Extent to which they have to change?

Questions

- **Q1:** How do the values of the instance compare to those across the rest of the dataset?

- **Q2:** Which features have the most considerable effect on the model's prediction?

- **Q3:** Are there changes that could alter the model's current prediction?

- **Q4:** Is it possible to change only a subset of *actionable* features to change the model's prediction?

Counterfactual Algorithm

Overall

- Find the minimal set of changes needed to change the model's output.
- Simple heuristic greedy search.
- Two *constraints* that ensure the explanation is interpretable and feasible.

Pre-processing

- Entire dataset is discretized by fitting a Gaussian on each of the features and splitting the values into **n** bins such that the middle **n-2** capture four standard deviations from the mean, and the extremal bins capture data points beyond this.

Counterfactual Algorithm

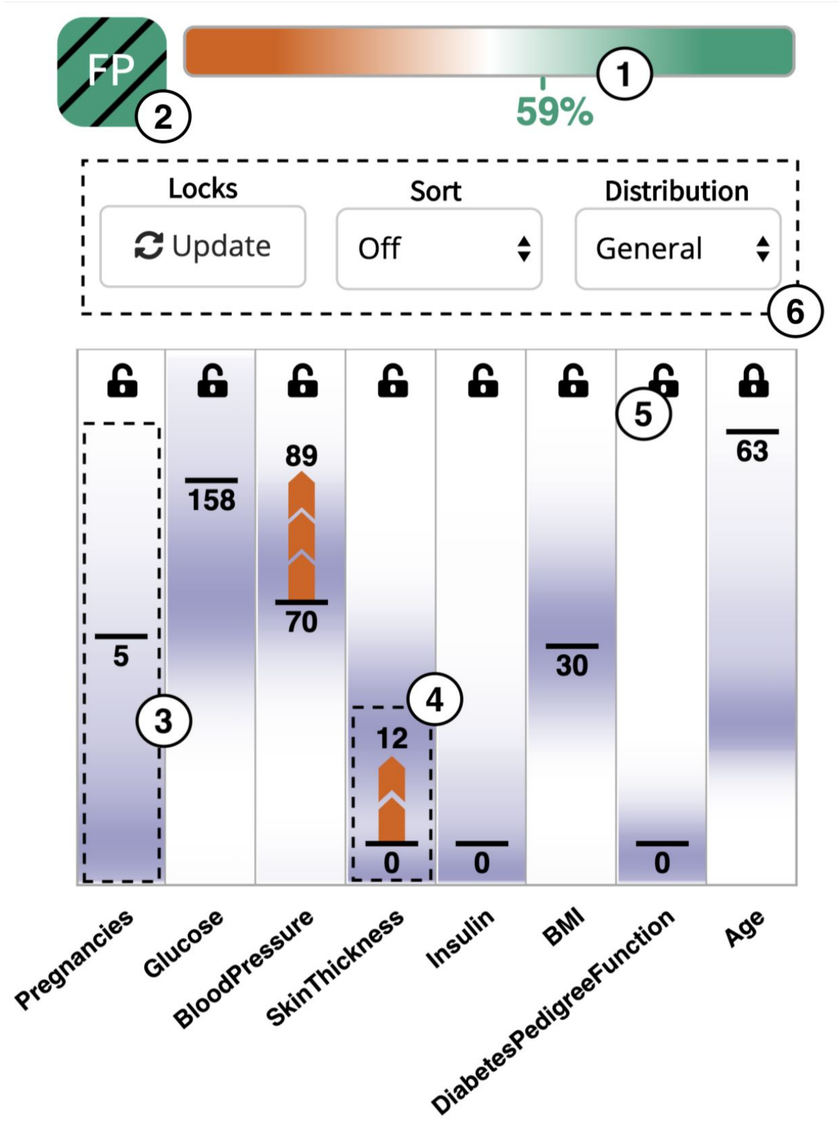
Iterations

- Greedily move feature values across the bins until the predicted class is changed, or until the pre-defined constraints are reached.
  - No more than **w** features are changed in a single explanation and no feature value is moved across more than **l** bins.

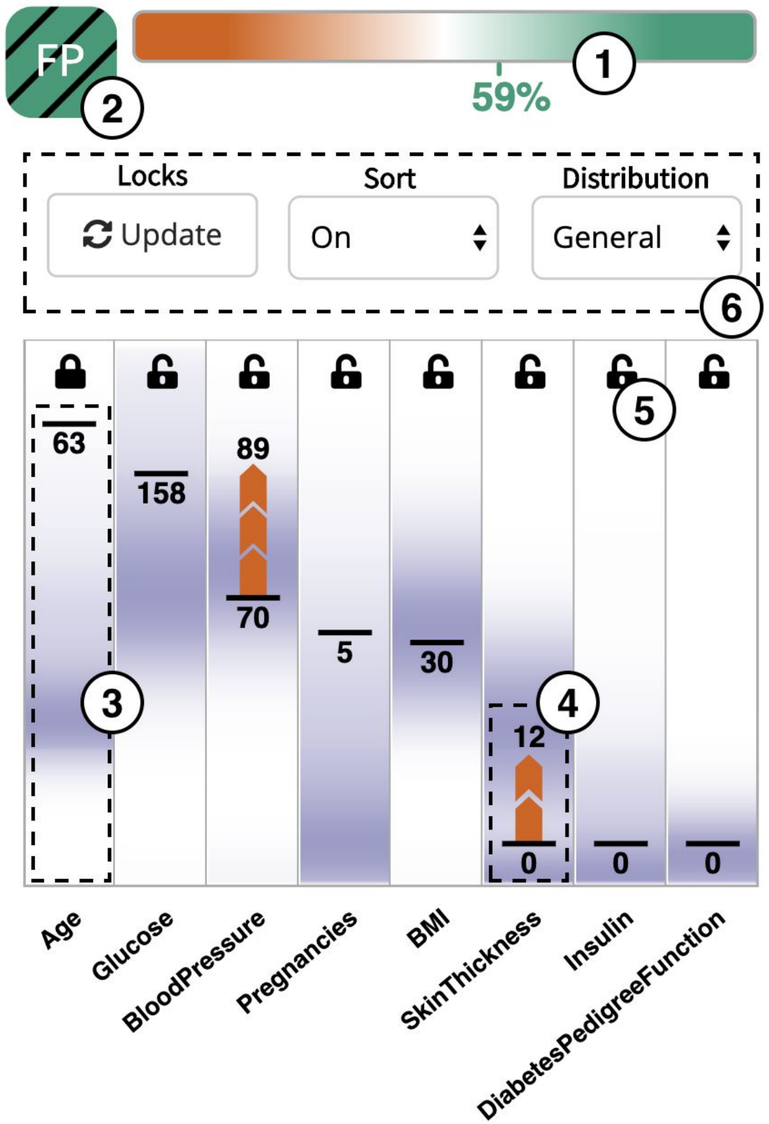
1. Starts with the original feature values, given an arbitrary set of unlocked features which can be acted upon.
2. Independently moves the value in each of the unlocked features to the bins above and below the current one and chooses the one eliciting the largest change in the model's output.
3. Take the maximum change across all the unlocked features and uses this as the input for the next iteration.

Visual Interface

1. Predicted probability
2. Classification correctness
3. Frequency density distribution and feature values
4. Counterfactual explanation
5. Locking functionality
6. Lock, sort, and distribution toggles



Unsorted



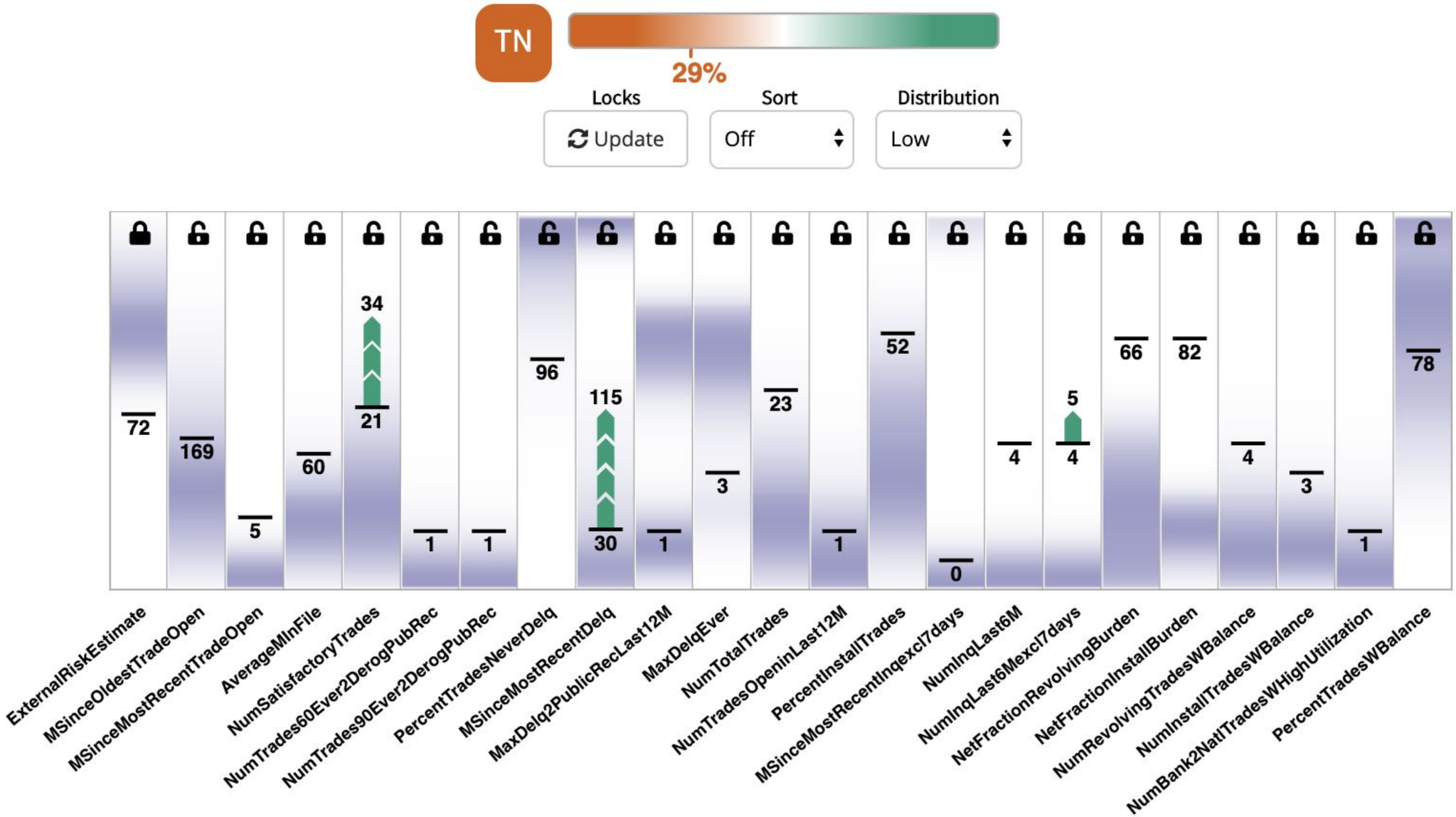
Sorted

https://dl.acm.org/doi/abs/10.1145/3377325.3377536

Implementation

- Flask web application with the back-end running on Python.
- Visualisations are created using D3 and JavaScript.
- Can accept any binary classification dataset in a CSV format.
- Default SVM model is trained with scikit-learn, however, the program also accommodates custom input models as long as probability prediction methods are provided.
- Data is processed in real time to accommodate customized end-user inputs.
- Split feature values across **n=10** bins and set **w = l = 5** for the algorithm constraints.

Case Study



https://dl.acm.org/doi/abs/10.1145/3377325.3377536

Limitations

- Algorithm cannot effectively handle categorical features.
  - Presetting a search path or performing a brute force analysis of features that are known to be categorical.

- Generating counterfactuals does not currently extend to multi-class classification and only works with binary target variables.
  - Aim to accommodate multiple class datasets to improve the tool's versatility.

- Intended for tabular numerical data and is therefore not suitable for other contexts such as image or text classification.

- Visualization can realistically display a maximum of around 30 features.
  - Larger datasets can be accommodated by utilizing the sorting feature and only displaying the top **k** features or those that are part of the counterfactual.

Future Work

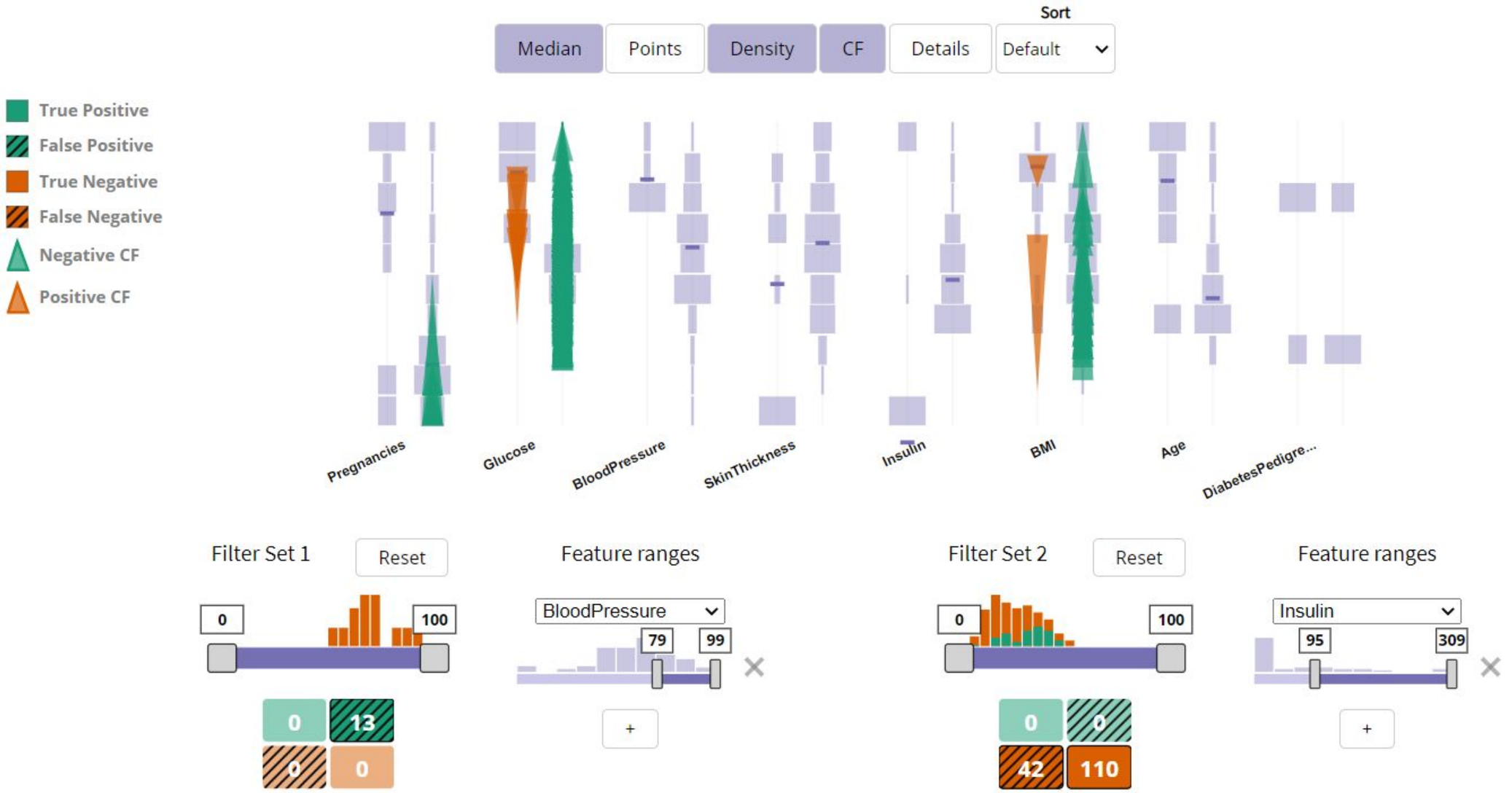
- Introduce increased interactivity for the UI.
  - Adding an option to view the impact of custom changes inputted by the user.
  - Integrate additional explanation methods will be integrated.
    - Customizing the sorting functionality to order the features according to their local importance magnitudes

- Extending the tool to a global scale through the aggregation of instance explanations could further increase its usefulness for model developers.

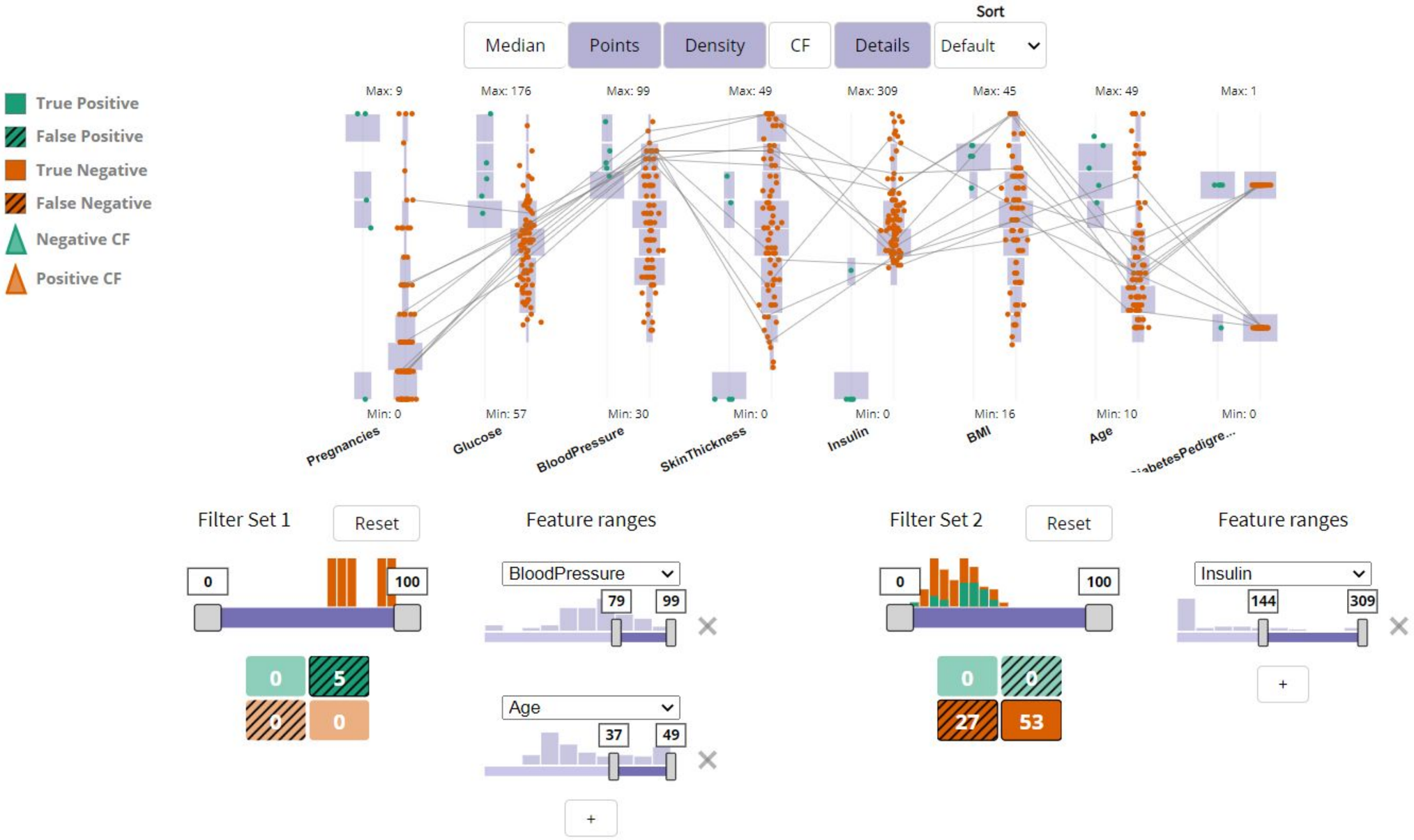
Conclusions

- *ViCE* – a novel way for the end-user to gain insight into model predictions through counterfactual explanations.
- For each sample the minimal set of changes needed to alter the decision was shown.
- Interacting with the interface allows customizing the explanation according to the user's requirements.
- First in visualising counterfactuals for non-binary data.
- Modular black-box based nature of the tool allows for a seamless integration of continued improvements
  - Including different methods to generate counterfactuals
  - Providing users with a set of alternatives to the displayed counterfactual explanation.

AdViCE: Aggregated Visual Counterfactual Explanations



AdViCE: Aggregated Visual Counterfactual Explanations



Challenges

- Fooling perturbation based explanations (H. Lakkaraju, 2020)
  - Manipulating user trust, high degrees of freedom in methods like LIME
  - Even methods with very high fidelity can be unstable and unreliable [Rudin 2019, Lipton 2017, Ghorbani 2019]
  - Saliency maps vulnerable to adversarial attacks
  - Not necessarily causal and are counterfactual - need to be communicated to end user

- Is the accuracy/interpretability tradeoff real? (C. Rudin, 2019)
  - Rashomon Effect
  - Define interpretability for specific domains
  - Stop explaining in high stakes domains

- Lack of evaluation at scale in real world scenarios (R. Ghani, 2020)
  - Methods usually tested on standard, common datasets

- Problems with “Mathiness” and “Language” (Z. Lipton, 2019)
  - Weakness in arguments hidden in theorems, weakness in theorems hide in text
  - Inflating simple technical concepts

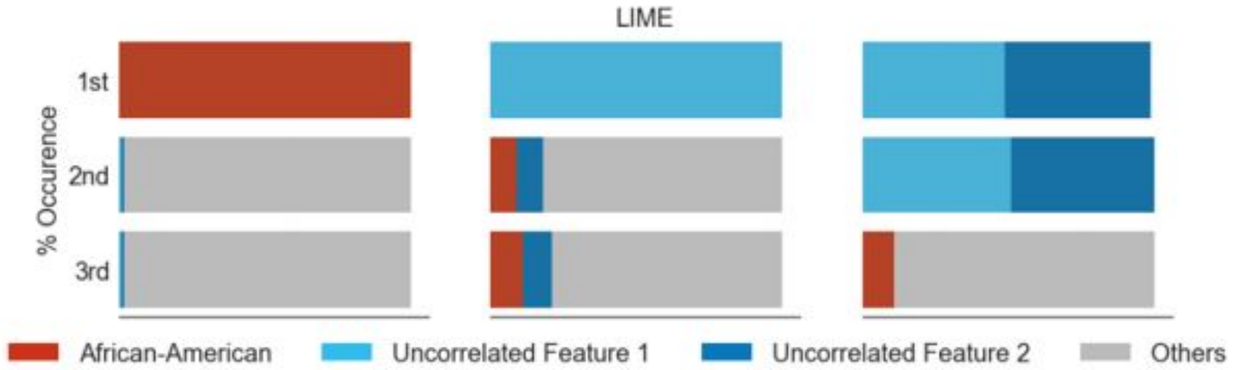
Challenges

Fooling perturbation based methods

- Can effectively mask the discriminatory biases of any black box classifier

- Exploit that perturbed samples are OOD (out of distribution)
  - Build a classifier that is biased on in-sample data points and unbiased on OOD

$$e(x) = \begin{cases} f(x), & \text{if } x \in \mathcal{X}_{dist} \\ \psi(x), & \text{otherwise} \end{cases}$$

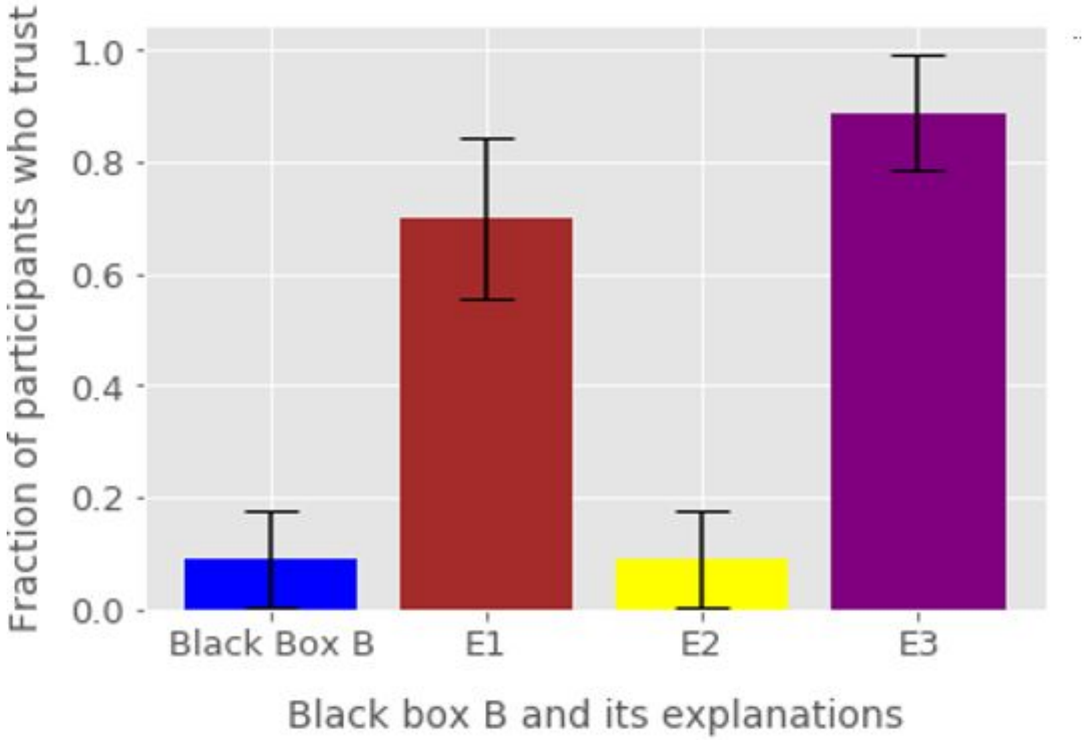


https://interpretable-ml-class.github.io/

Challenges

Fooling perturbation based methods

- Experts are 9.8 times more likely to trust the black box if they see an “agreeable” explanation



https://interpretable-ml-class.github.io/

Resources

Toolkit	Data Explanations	Directly Interpretable	Local Post-Hoc	Global Post-Hoc	Persona-Specific Explanations	Metrics
AIX360	✓	✓	✓	✓	✓	✓
Alibi [1]			✓			
Skater [7]		✓	✓	✓		
H2O [4]		✓	✓	✓		
InterpretML [6]		✓	✓	✓		
EthicalML-XAI [3]				✓		
DALEX [2]			✓	✓		
tf-explain [8]			✓	✓		
iNNvestigate [5]			✓			

Table 1: Comparison of AI explainability toolkits.

https://arxiv.org/pdf/1909.03012.pdf

Resources

- Interpretable Machine Learning Book
- Interpretability and Explainability in Machine Learning
- Fairness and machine learning Limitations and Opportunities
- Repository of machine learning interpretability resources
- AI Explainability 360
- VISxAI Workshop
- h2oai/mli-resources
- Many, many others...