Budapest-ML

# StickyLand

## Break the Linear Presentation of Computational Notebooks

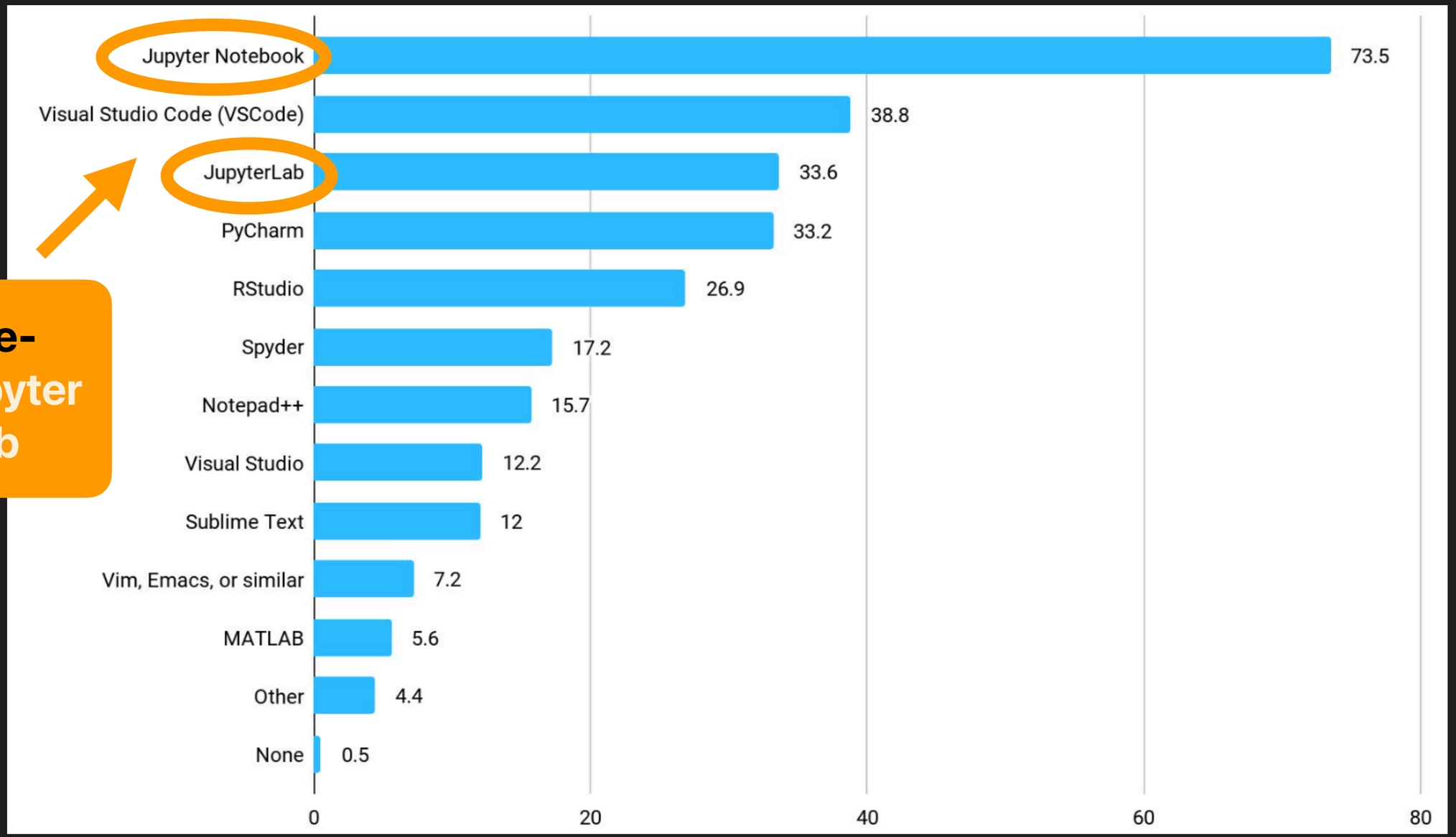

**Jay Wang**

**Katie Dai**

**Keith Edwards**

Georgia Tech

# Programming Environment Popularity: Kaggle Survey (2021)




More than **three-quarters** use **Jupyter Notebook/Lab**

https://www.kaggle.com/kaggle-survey-2021

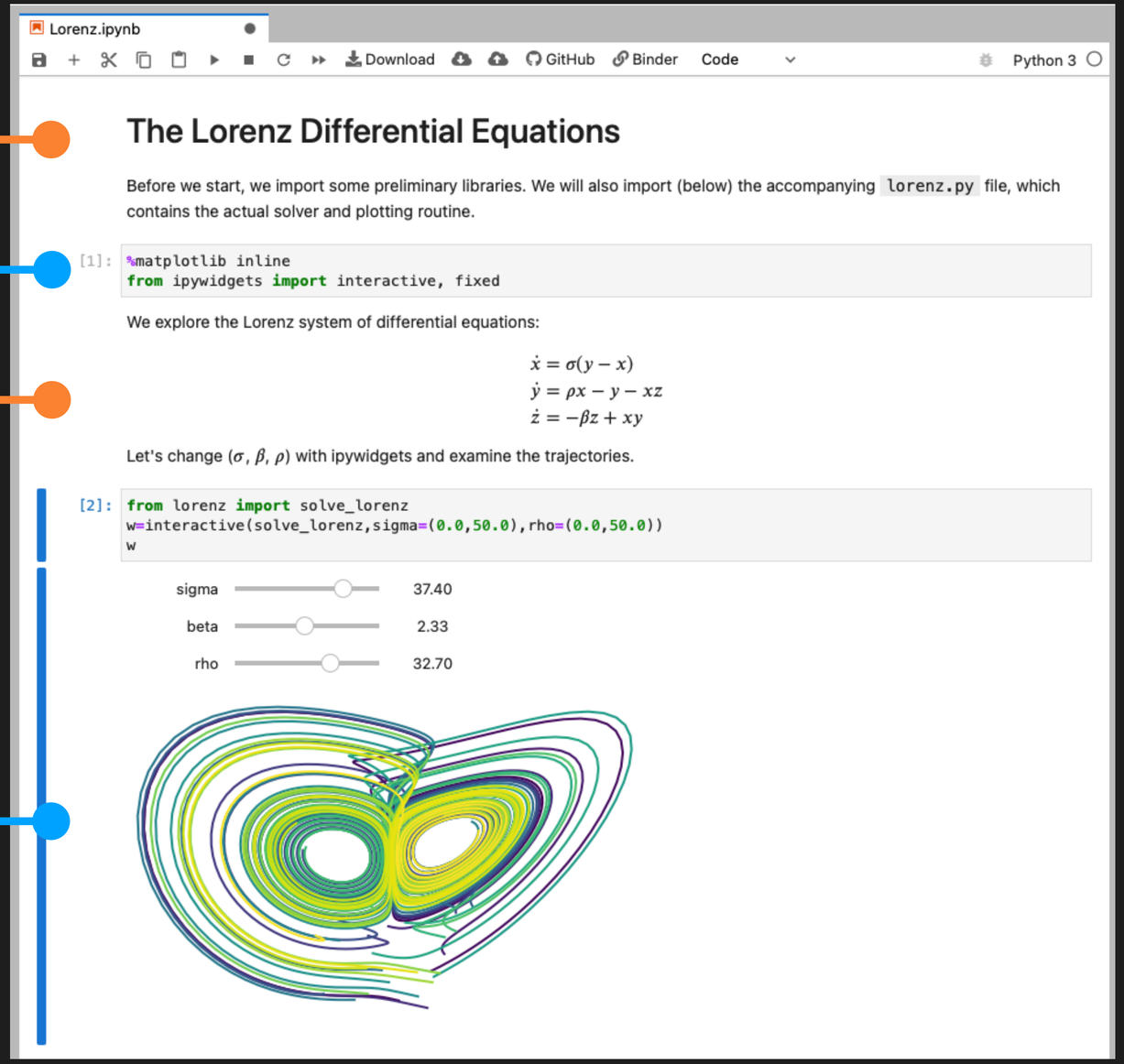
# What are Computational Notebooks?

Markdown Text

Code

Markdown Text
(LaTeX)

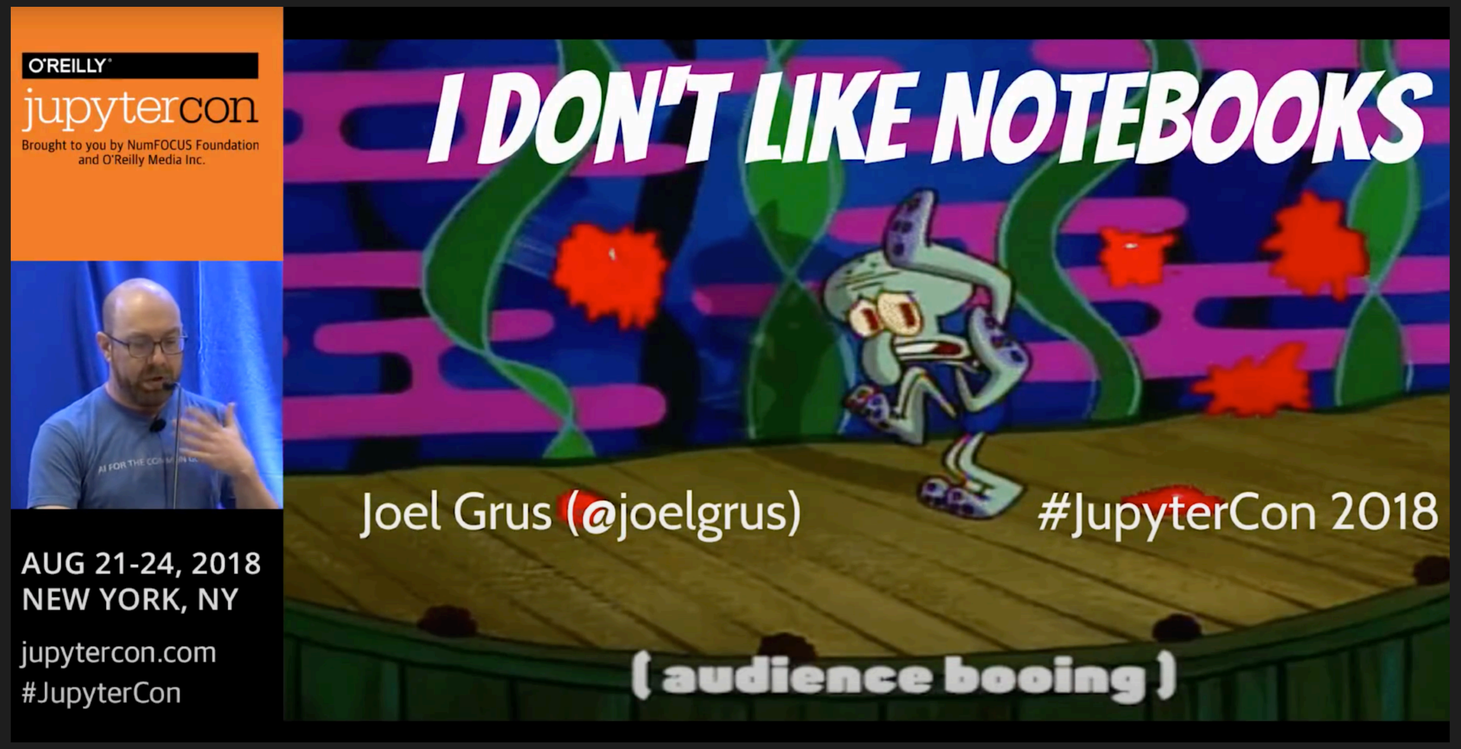
Code Output
(Interactive Visualization)



# Are Computational Notebooks Perfect?

*Nope! Many people actually hate notebooks!*

- Code Organization
  Data analysis is not linear
- Execution History
  Lose track of the code states
- Not Product-ready
  Need clean-up & packaging for deployment

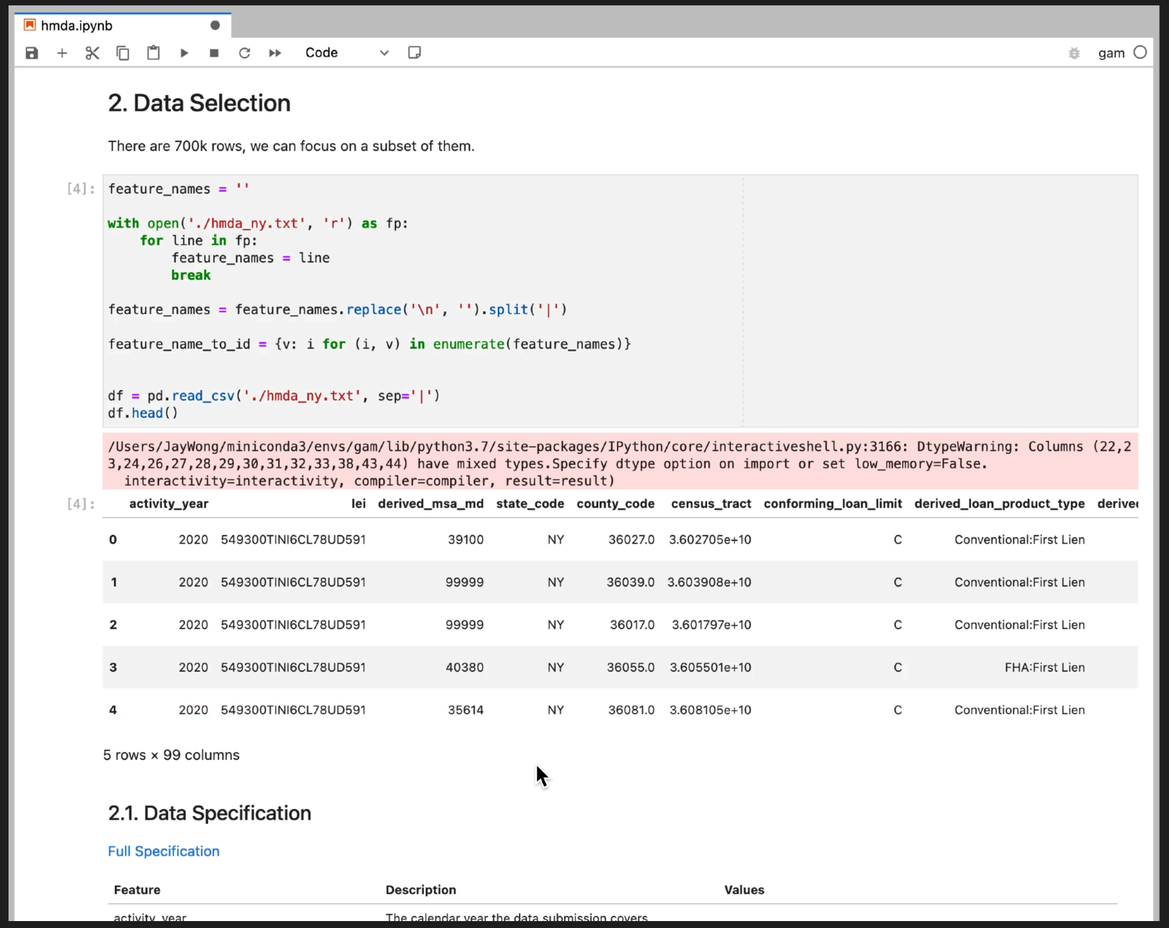



Chattopadhyay, Souti, et al. "What's Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities." CHI 2020

# Are Computational Notebooks Perfect?

*Nope! Many people actually hate notebooks!*

- Code Organization
  Data analysis is not linear
- Execution History
  Lose track of the code states
- Not Product-ready
  Need clean-up & packaging for deployment




Chattopadhyay, Souti, et al. "What's Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities." CHI 2020

# Design for Better Code Organization

## Managing Messes in Computational Notebooks

**Andrew Head**
UC Berkeley
andrewhead@berkeley.edu

**Fred Hohman**
Georgia Institute of Technology
fredhohman@gatech.edu

**Titus Barik**
Microsoft
titus.barik@microsoft.com

**Steven M. Drucker**
Microsoft Research
sdrucker@microsoft.com

**Robert DeLine**
Microsoft Research
rob.deline@microsoft.com

### ABSTRACT

Data analysts use computational notebooks to write code for analyzing and visualizing data. Notebooks help analysts iteratively write analysis code by letting them interleave code with output, and selectively execute cells. However, as analysis progresses, analysts leave behind old code and outputs, and overwrite important code, producing cluttered and inconsistent notebooks. This paper introduces code gathering tools, extensions to computational notebooks that help analysts find, clean, recover, and compare versions of code in cluttered, inconsistent notebooks. The tools archive all versions of code outputs, allowing analysts to review these versions and recover the subsets of code that produced them. These subsets can serve as succinct summaries of analysis activity or starting points for new analyses. In a qualitative usability study, 12 professional analysts found the tools useful for cleaning notebooks and writing analysis code, and discovered new ways to use them, like generating personal documentation and lightweight versioning.

### CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; • **Software and its engineering** → Development frameworks and environments.

### KEYWORDS

Computational notebooks; messes; clutter; inconsistency; exploratory programming; code history; program slicing



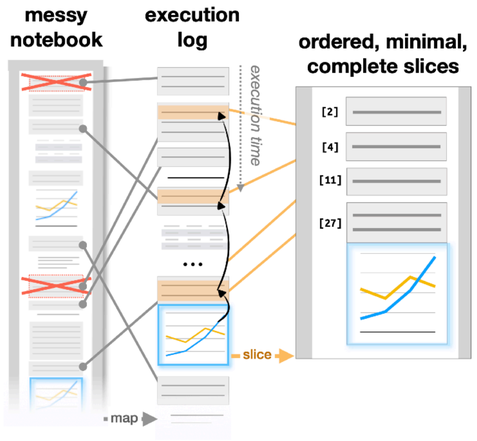



**Figure 1: Code gathering tools help analysts manage programming messes in computational notebooks.** The tools map selected results (e.g., outputs, charts, tables) in a notebook to the ordered, minimal subsets or "slices" of code that produced them. With these slices, the tools help analysts clean their notebooks, browse versions of results, and discover provenance of results.

**ACM Reference Format:**
Andrew Head, Fred Hohman, Titus Barik, Steven M. Drucker, and Robert DeLine. 2019. Managing Messes in Computational Notebooks. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland UK.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3290605.3300500

### 1 INTRODUCTION

Data analysts often engage in "exploratory programming"

## Fork It: Supporting Stateful Alternatives in Computational Notebooks

Nathaniel Weinman
nweinman@berkeley.edu
Microsoft Research/UC Berkeley

Titus Barik
titus.barik@microsoft.com
Microsoft Research

Steven M. Drucker
sdrucker@microsoft.com
Microsoft Research

Robert DeLine
rob.deline@microsoft.com
Microsoft Research

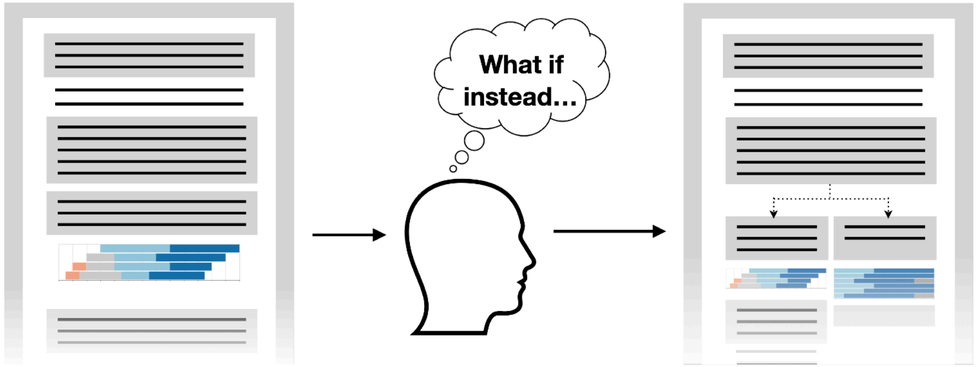



**Figure 1: A user makes progress in a notebook (left) to test a hypothesis. They then wonder how an alternative approach may have worked out (middle). They fork their existing code in the middle of their notebook, starting their new exploration next to the old (right).**

### ABSTRACT

Computational notebooks, which seamlessly interleave code with results, have become a popular tool for data scientists due to the iterative nature of exploratory tasks. However, notebooks provide a single execution state for users to manipulate through creating and manipulating variables. When exploring alternatives, data scientists must carefully create many-step manipulations in visually distant cells.

We conducted formative interviews with 6 professional data scientists, motivating design principles behind exposing multiple states. We introduce forking — creating a new interpreter session — and backtracking — navigating through previous states. We implement these interactions as an extension to notebooks that help data scientists more directly express and navigate through decision points a single notebook. In a qualitative evaluation, 11 professional data scientists found the tool would be useful for exploring alternatives and debugging code to create a predictive model. Their insights highlight further challenges to scaling this functionality.

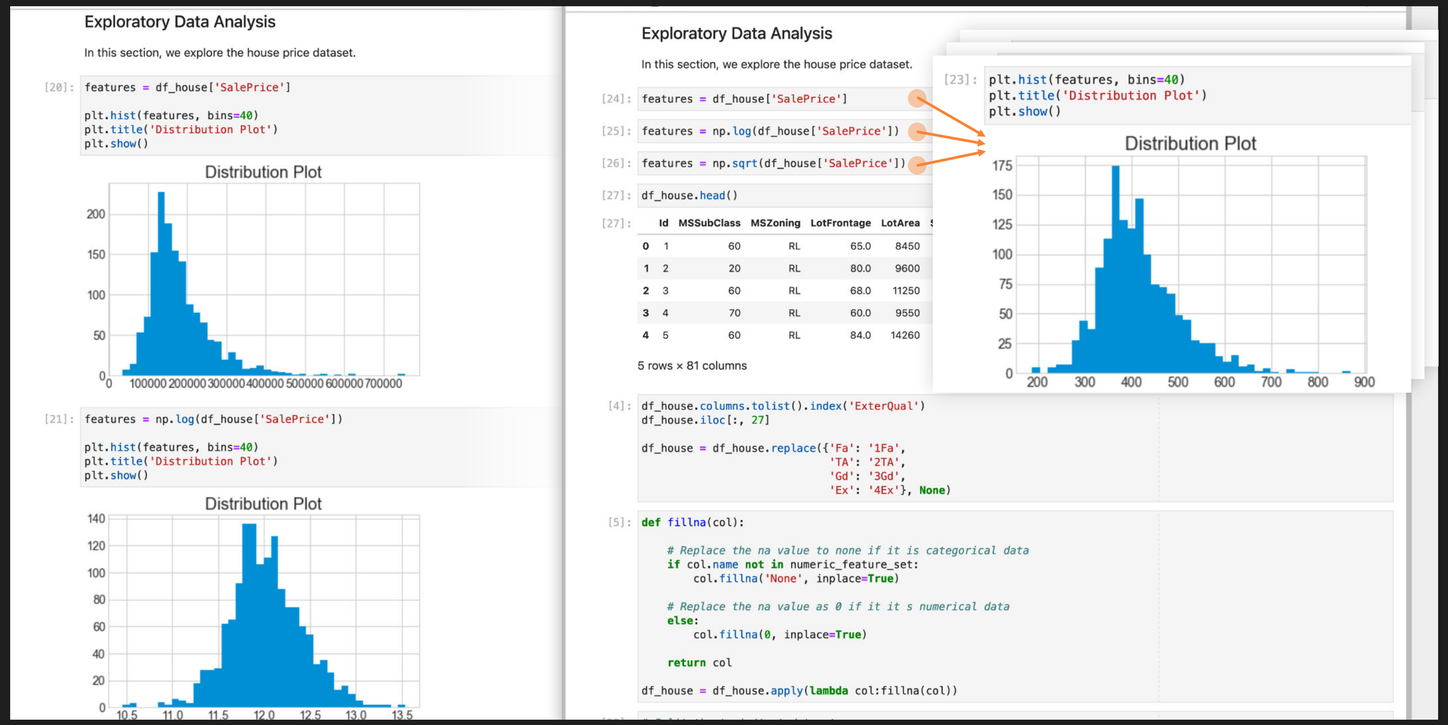
### CCS CONCEPTS

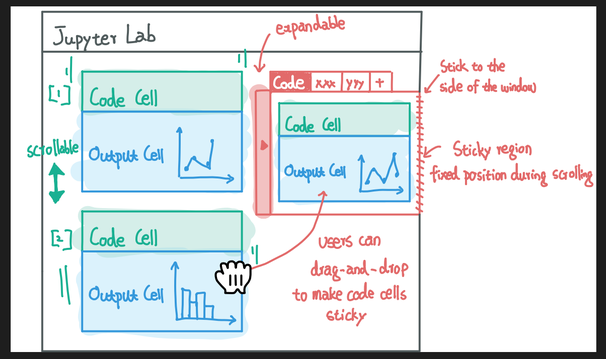
• **Software and its engineering** → **Development frameworks and environments.**

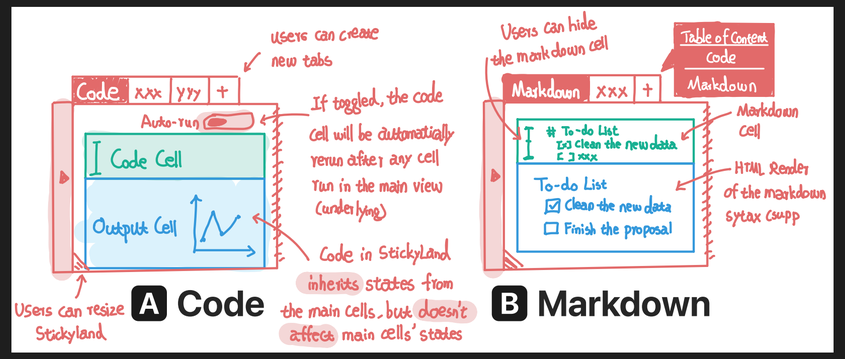
### KEYWORDS

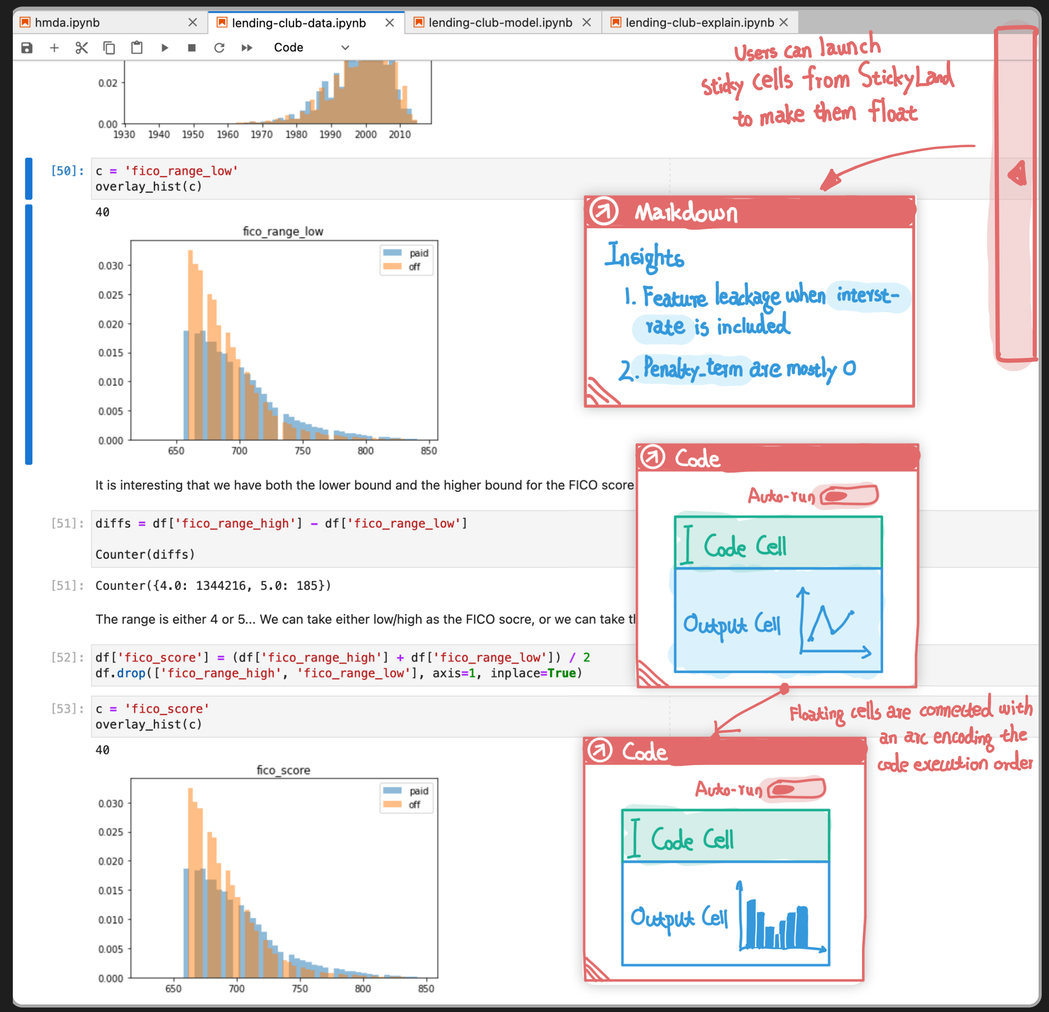
Alternatives, computational notebooks, exploratory programming, code history

# What if... Sticky Cells?



# StickyLand

Break the Linear Presentation of Computational Notebooks

bit.ly/stickyland

```
pip install stickyland
```

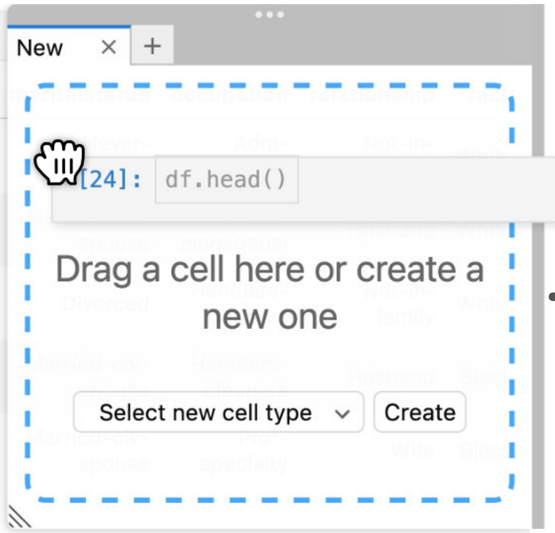
## A Notebook Cell

```
[24]: df.head()
```

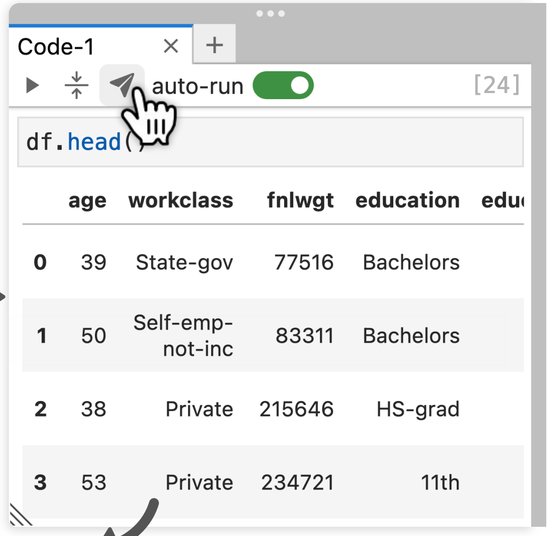
[24]:

| | age | workclass | fnlwgt | education | educationnum |
|---|---|---|---|---|---|
| 0 | 39 | State-gov | 77516 | Bachelors | 13 |
| 1 | 50 | Self-emp-not-inc | 83311 | Bachelors | 13 |
| 2 | 38 | Private | 215646 | HS-grad | 9 |
| 3 | 53 | Private | 234721 | 11th | 7 |
| 4 | 28 | Private | 338409 | Bachelors | 13 |

## B Sticky Dock




## C Sticky Cell




| | age | workclass | fnlwgt | education | edu |
|---|---|---|---|---|---|
| 0 | 39 | State-gov | 77516 | Bachelors | |
| 1 | 50 | Self-emp-not-inc | 83311 | Bachelors | |
| 2 | 38 | Private | 215646 | HS-grad | |
| 3 | 53 | Private | 234721 | 11th | |

## D Floating Cell

Code-1

auto-run [24]

```
df.head()
```

| | age | workclass | fnlwgt | education | educationnum | maritalstatus | occupation | relationship | race | gender | capitalgain | capitalloss | hoursperweek | nativecountry | income |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 39 | State-gov | 77516 | Bachelors | 13 | Never-married | Adm-clerical | Not-in-family | White | Male | 2174 | 0 | 40 | United-States | <=50K |
| 1 | 50 | Self-emp-not-inc | 83311 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 0 | 0 | 13 | United-States | <=50K |
| 2 | 38 | Private | 215646 | HS-grad | 9 | Divorced | Handlers-cleaners | Not-in-family | White | Male | 0 | 0 | 40 | United-States | <=50K |

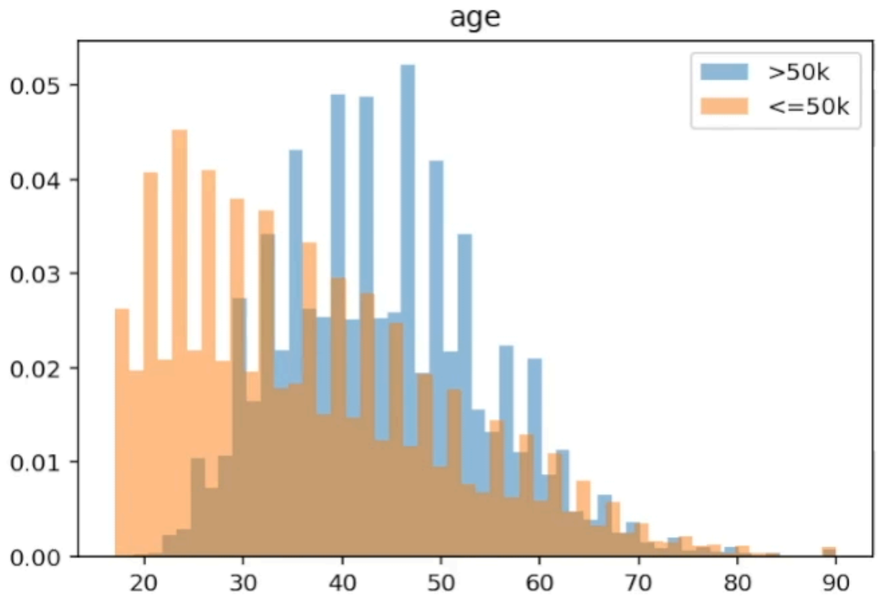
## 1.2. Data Engineering

In this section, we delete or transform some features before training the binary classifier.

[141]:
```
intersted_feature = 'age'
```

[143]:
```
overlay_hist(df, intersted_feature)
```

```
Num of unique values:  73
```




The distribution difference between these two groups on age is quite significant.

[26]:
```
overlay_hist(df, 'workclass')
```

```
Num of unique values:  9
```

image-augmentation.ipynb

Code

Python 3 (ipykernel)

```
[ ]: def rotate(image):
         """Rotate the image"""
         aug = iaa.Affine(rotate=(-10, -9))
         image_aug = aug(image=image)
         return image_aug

     def add_noise(image):
         """Add random noise on the image"""
         aug = iaa.CoarseDropout(0.02, size_percent=0.5)
         image_aug = aug(image=image)
         return image_aug

     def corrupt(image):
         """Corrupt the image"""
         aug = iaa.MultiplyHueAndSaturation(mul_hue=4)
         image = aug(image=image)
         aug = iaa.MultiplyHueAndSaturation(mul_hue=4)
         image = aug(image=image)
         return image
```

```
[ ]: image = load_random_image(dataset)
     plt.imshow(image);
```

```
[ ]: image = rotate(image)
     plt.imshow(image);
```

```
[ ]: image = add_noise(image)
     plt.imshow(image);
```

New

Drag a cell here or create a new one

Select new cell type

Create

example-ml.ipynb

Markdown

Python 3 (ipykernel)

# Binary Classification on Adult Dataset

In this notebook, we explore the Adult dataset and train a simple binary classifier.

[7]:
```
import numpy as np
import pandas as pd
import gamchanger as gc

from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from collections import Counter
from interpret import show
from interpret.glassbox import ExplainableBoostingClassifier

%config InlineBackend.figure_format = 'retina'
```

[2]:
```
df = pd.read_csv(
    'https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data',
    sep=', ',
    engine='python',
    header=None
)

column_names = [
    'Age', 'WorkClass', 'fnlwgt', 'Education', 'EducationNum',
    'MaritalStatus', 'Occupation', 'Relationship', 'Race', 'Gender',
    'CapitalGain', 'CapitalLoss', 'HoursPerWeek', 'NativeCountry', 'Income'
]
df.columns = [n.lower() for n in column_names]

df.shape
```

[2]:
```
(32561, 15)
```

[3]:
```
df['target'] = [0 if l else 1 for l in (df['income'] == '<=50K')]
```

[4]:
```
train_cols = df.columns[0:-2]
label = df.columns[-1]
x = df[train_cols]
y = df['target']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2)
```

[5]:
```
ebm = ExplainableBoostingClassifier(n_jobs=-1)
ebm.fit(x_train, y_train)
```

[5]:
```
ExplainableBoostingClassifier(feature_names=['age', 'workclass', 'fnlwgt',
                                             'education', 'educationnum',
                                             'maritalstatus', 'occupation',
                                             'relationship', 'race', 'gender',
                                             'capitalgain', 'capitalloss',
                                             'hoursperweek', 'nativecountry',
                                             'relationship x hoursperweek',
                                             'age x relationship',
                                             'age x capitalloss',
                                             'maritalstatus x hoursperweek',
                                             'educationnum x occupation',
                                             'age x fnlwgt'...
                              feature_types=['continuous', 'categorical',
                                             'continuous', 'categorical',
                                             'continuous', 'categorical',
                                             'categorical', 'categorical',
                                             'categorical', 'categorical',
```

New

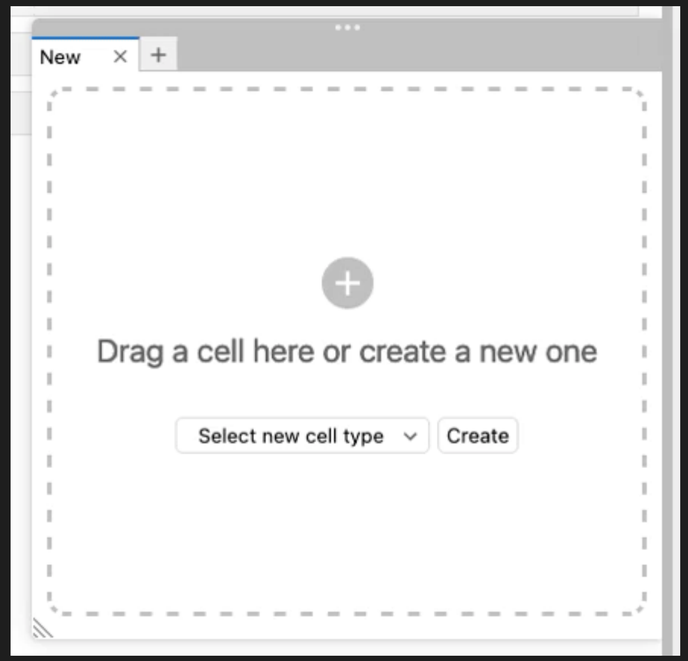
Drag a cell here or create a new one

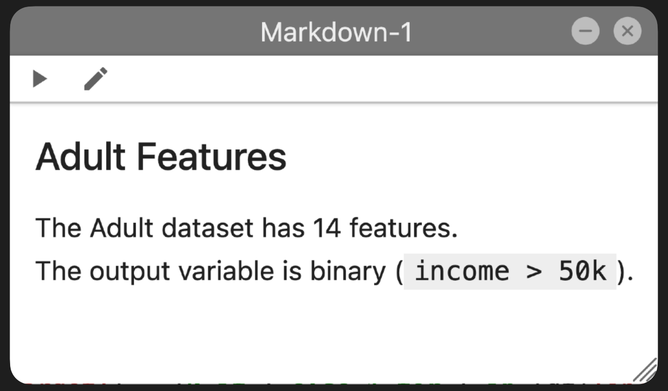
Select new cell type

Create

# How Did We Implement It?

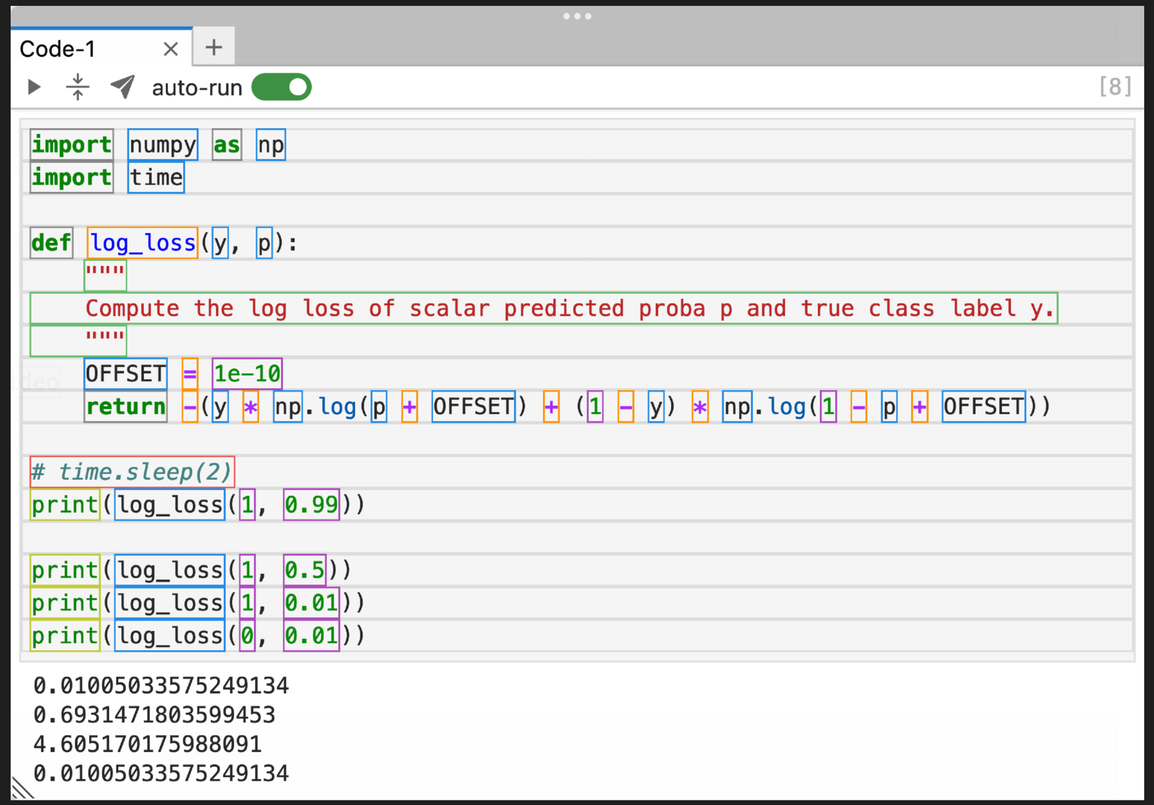
- Develop from Scratch
  - TypeScript + CSS + Python glue
  - CodeMirror for Text Editor
- JupyterLab API
  - Run code, render markdown
  - Listen to events
- Native Look
  - CSS to mimic JupyterLab style



# Reflection: UI Toolkits are Helpful

- Text Editor is Not that Simple
  - Grid of mono-space characters
  - Cursor is a blinking box
- Numerous Edge Cases
  - Arrow key to navigate
  - Command + arrow key
  - Scrolling, resizing



# Future Work

1. **Topological** Auto-run of Sticky Cells
   Only run when referenced values changed in other cells

2. **Separate Code State** in StickyLand
   Throw-away experiment without inferencing the notebook

3. **Generalize** to Other Computational Notebooks
   Google Colab, VSCode Notebook, Azure Notebook, etc.

4. **Evaluation** of Usability and Utility
   How do people use StickyLand? Is it helpful?

Markdown-1

# ML Error Analysis Dashboard

In this dashboard, we interactively diagnose a binary classification model trained on the Adult dataset

A

Code-1

auto-run [46]

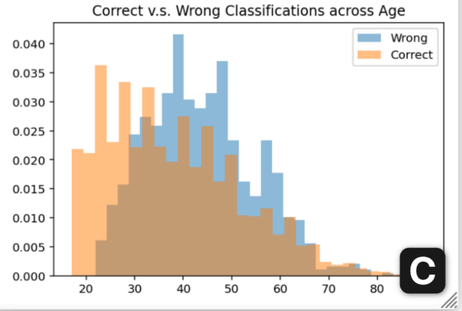
```
# Step 1: Specify our interested feature

# Let's explore age feature first
cur_feature = 'age'
```

B

Code-2

auto-run [50]

```
draw_error_hist(cur_feature)
```




C

Code-3

auto-run [49]

```
# Step 2: Specify a select query

# Here we focus on mis-classified samples
# who are older than 35 years

interseted_index = np.where(
    np.logical_and(
        x_test[cur_feature] >= 35,
        y_test != y_pred
    )
)[0].tolist()

log = 'The query has {} results.'
print(log.format(len(interseted_index)))
```

```
The query has 678 results.
```

D

Code-4

auto-run [51]

```
x_test.iloc[interseted_index, :].head()
```

| | age | workclass | fnlwgt | education | educationnum | maritalstatus | occupation | relationship | race | gender |
|---|---|---|---|---|---|---|---|---|---|---|
| **20132** | 48 | Federal-gov | 215389 | HS-grad | 9 | Married-civ-spouse | Transport-moving | Husband | Black | Male |
| **13680** | 57 | Private | 31532 | Bachelors | 13 | Married-civ-spouse | Transport-moving | Husband | White | Male |
| **30314** | 59 | Self-emp-inc | 349910 | Assoc-acdm | 12 | Married-civ-spouse | Craft-repair | Husband | White | Male |
| **28622** | 57 | Private | 201991 | Bachelors | 13 | Married-civ-spouse | Prof-specialty | Husband | White | Male |
| **18927** | 35 | Private | 322171 | Some-college | 10 | Married-civ-spouse | Other-service | Wife | White | Male |

E

Code-5

auto-run [52]

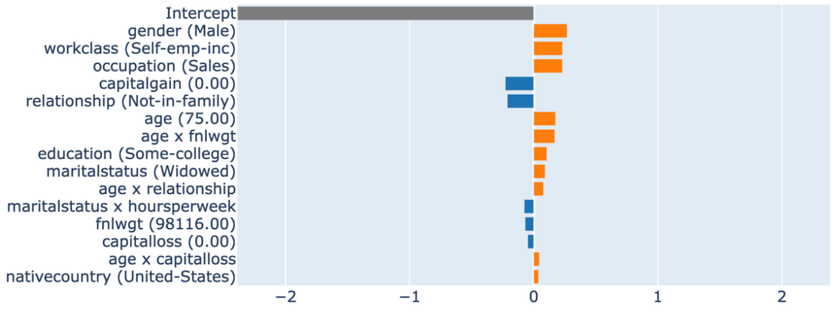
```
ebm_local = ebm.explain_local(x_test.to_numpy()[interseted_index, :],
                              y_test.to_numpy()[interseted_index],
                              name='EBM')

show(ebm_local)
```

Select Component to Graph

231 : Predicted (0.0) | PrScore (0.824) | Actual (1.0)

EBM [231]

Predicted (0): 0.824 | Actual (1): 0.176

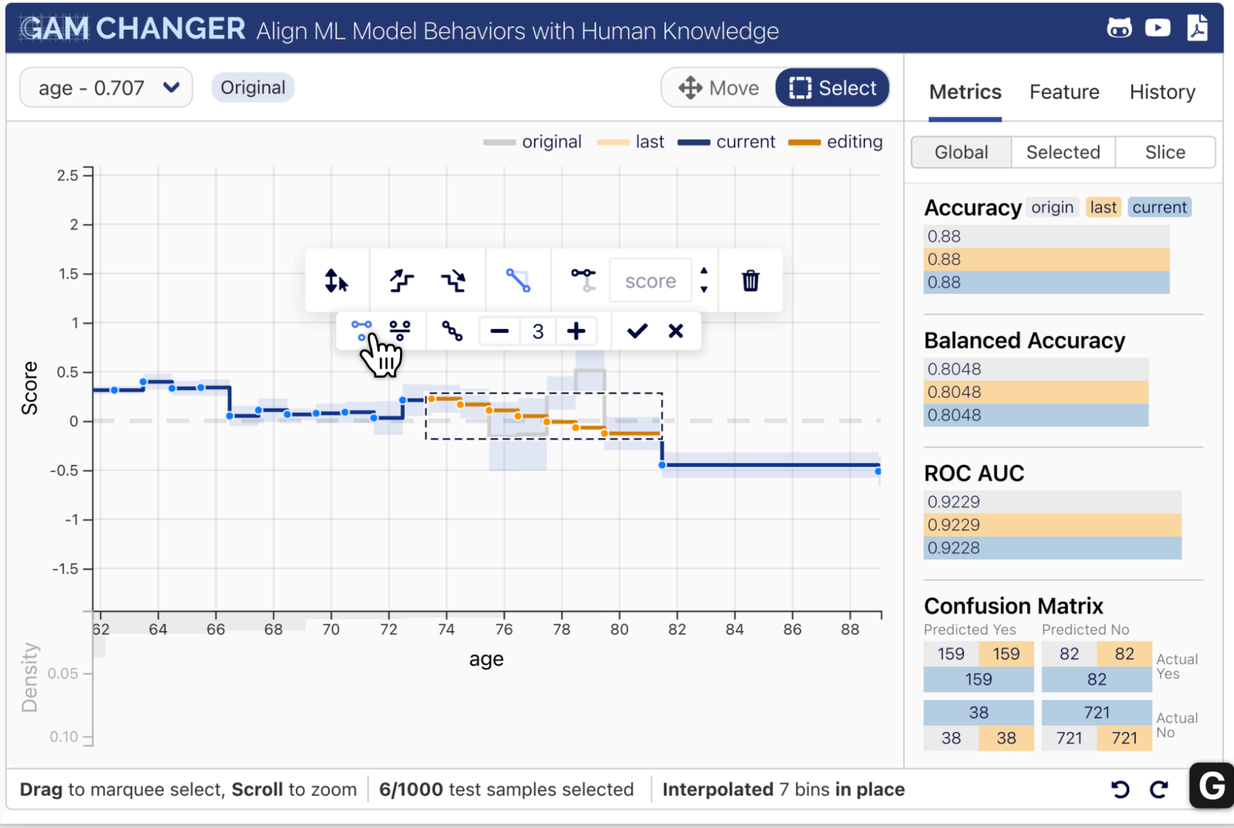



F

Code-6

auto-run [45]

```
gc.visualize(ebm, x_train[:1000], y_train[:1000])
```




G

Align ML Model Behaviors with Human Users' Knowledge

# GAM CHANGER

Jay Wang
Georgia Tech

Alex Kale
University of Washington

Harsha Nori
Microsoft

Peter Stella
NYU Langone Health

Mark Nunnally
NYU Langone Health

Polo Chau
Georgia Tech

Mickey Vorvoreanu
Microsoft Research

Jenn Wortman Vaughan
Microsoft Research

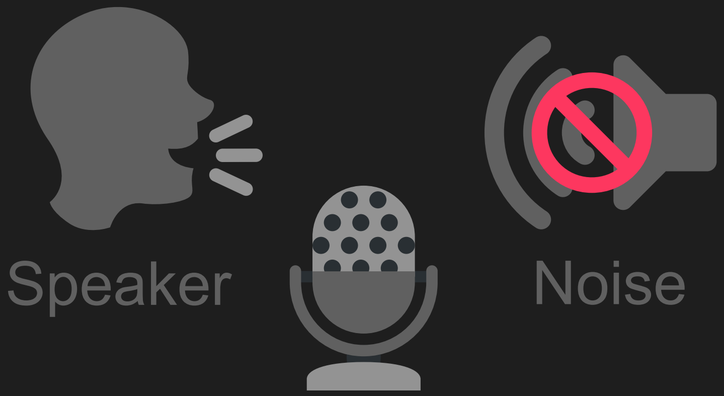
Rich Caruana
Microsoft Research

# ML is Ubiquitous

ML technology is in our everyday life, including this talk


Natural Language

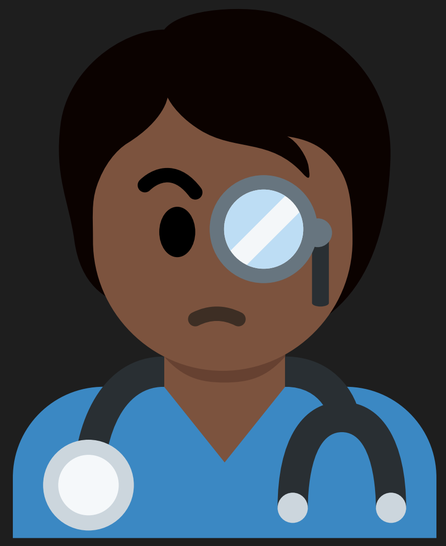
Computer Vision




Audio Signal

# Not so ubiquitous in high-stake domains

Why?



I don’t know

I won’t use you

# Intelligible Machine Learning

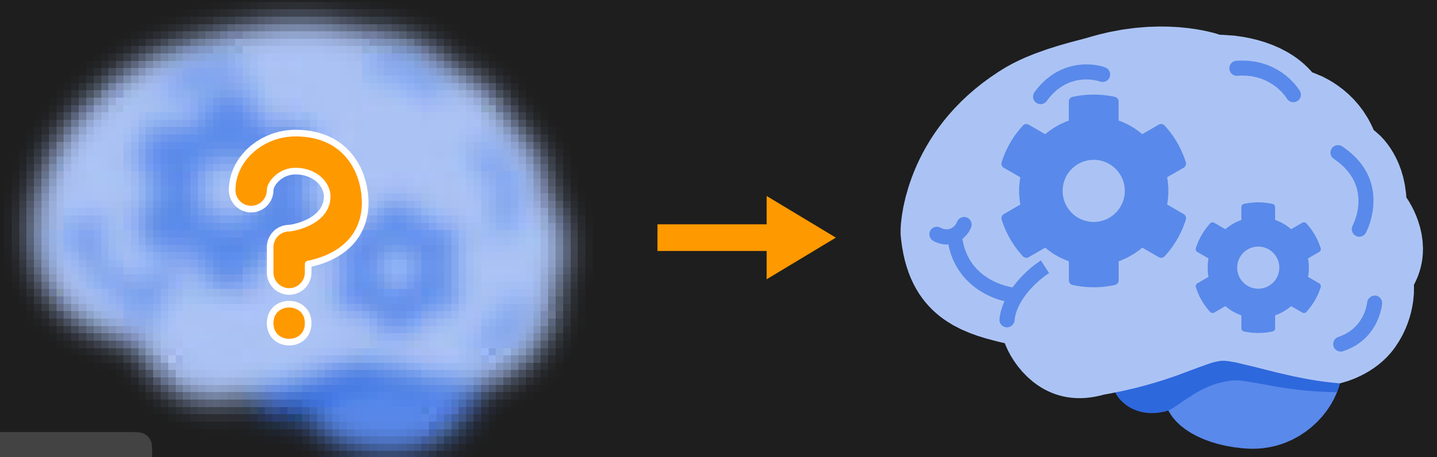
Explainable Boosting Machine

Visual Analytics

Distillation

Saliency Map

Prototypes



TCAV

Feature Visualization

LIME

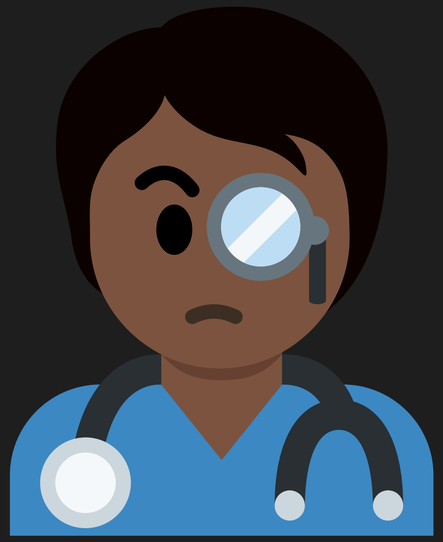
InterpretML

Counterfactual

SHAP

# "New problems" after interpretability

Why?



Ah, because the age...

Also, the height...

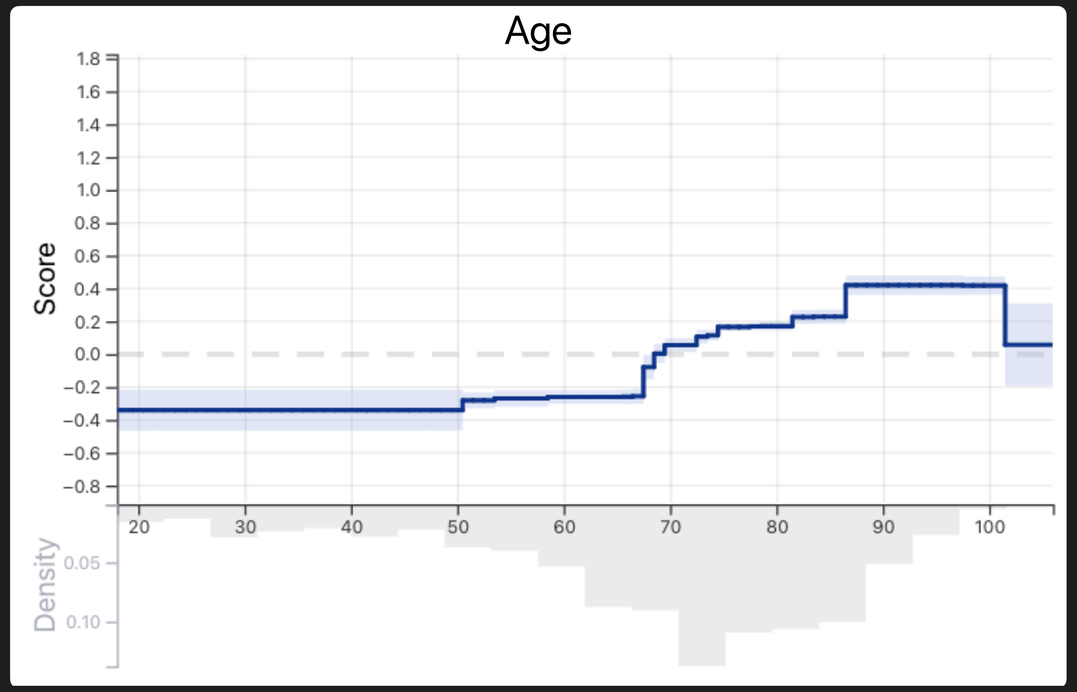
Height? Not what I expected

# Medical Data Example

## Explainable Boosting Machine (EBM)

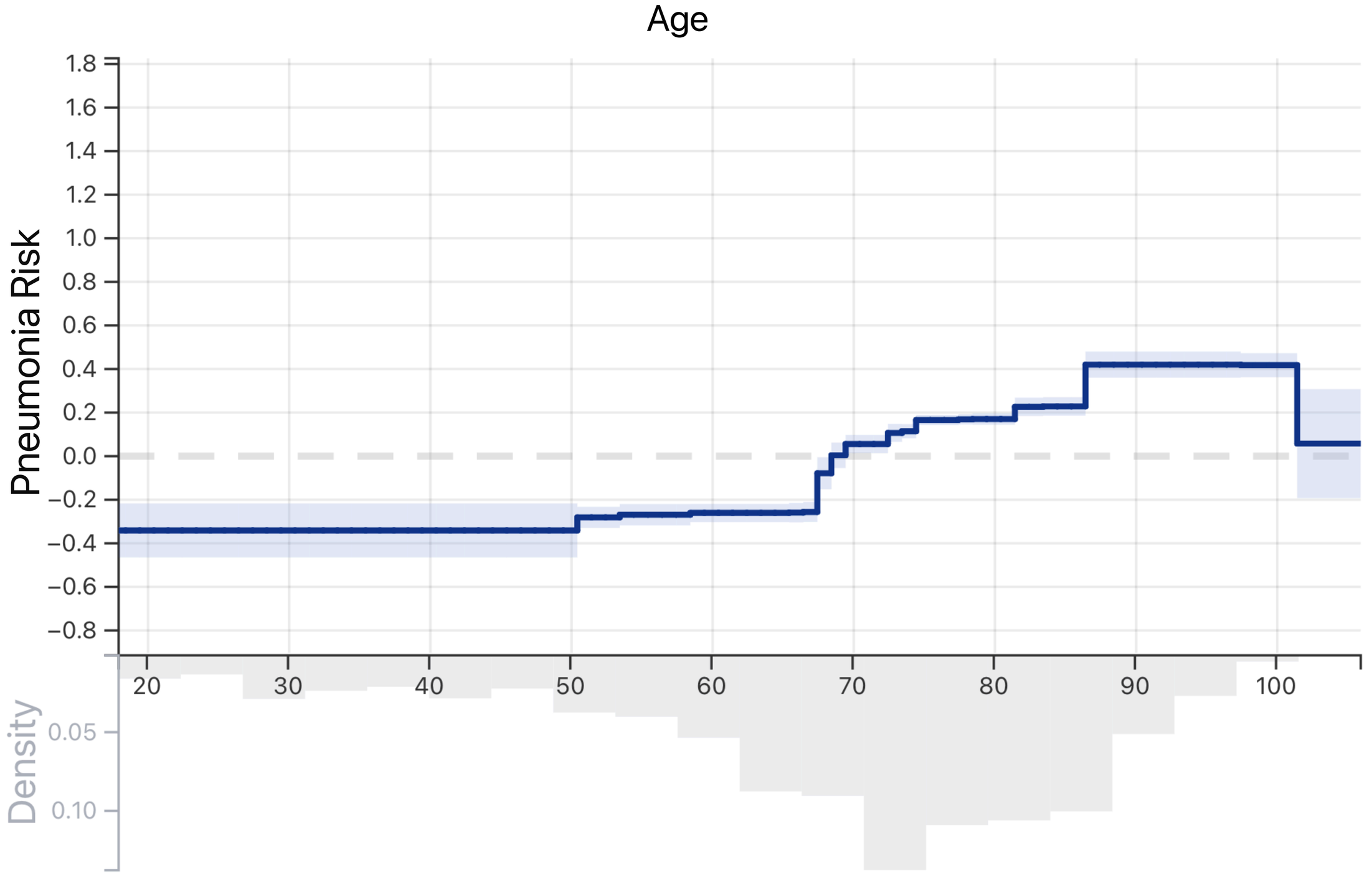
- Generalized additive model (GAM)
- Glass-box model
- Easy to understand

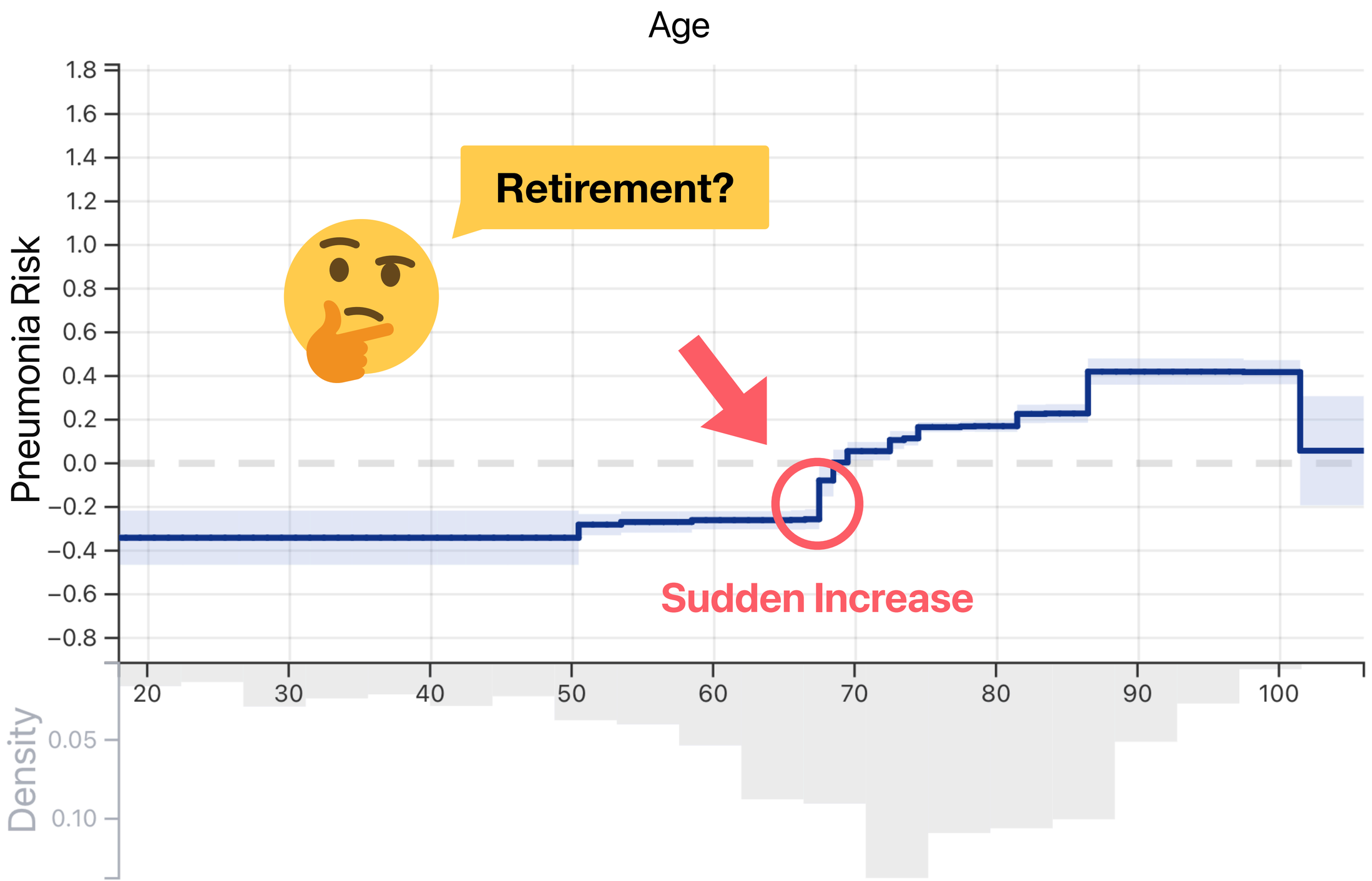
$$y = \beta_0 + f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n)$$

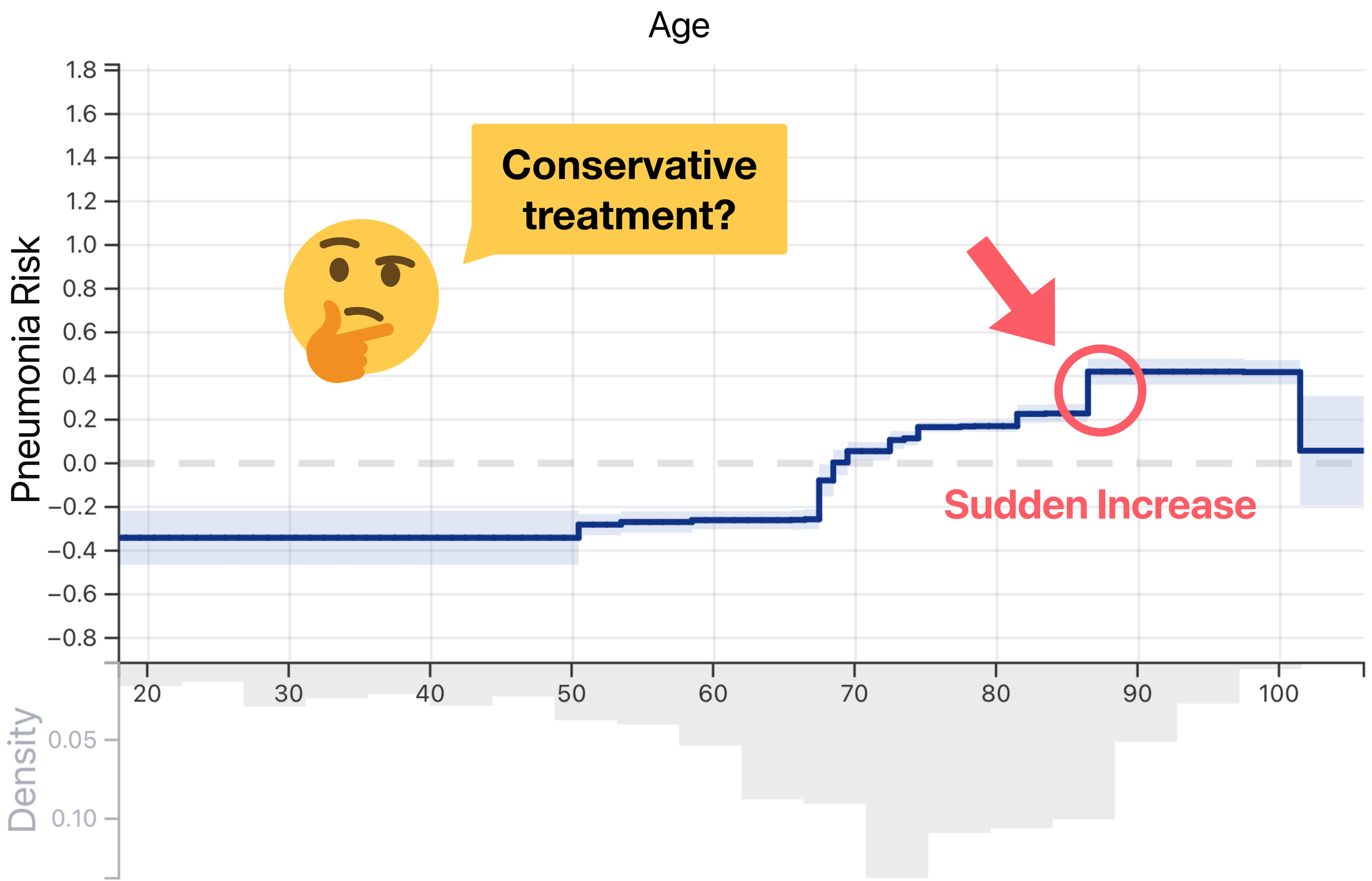


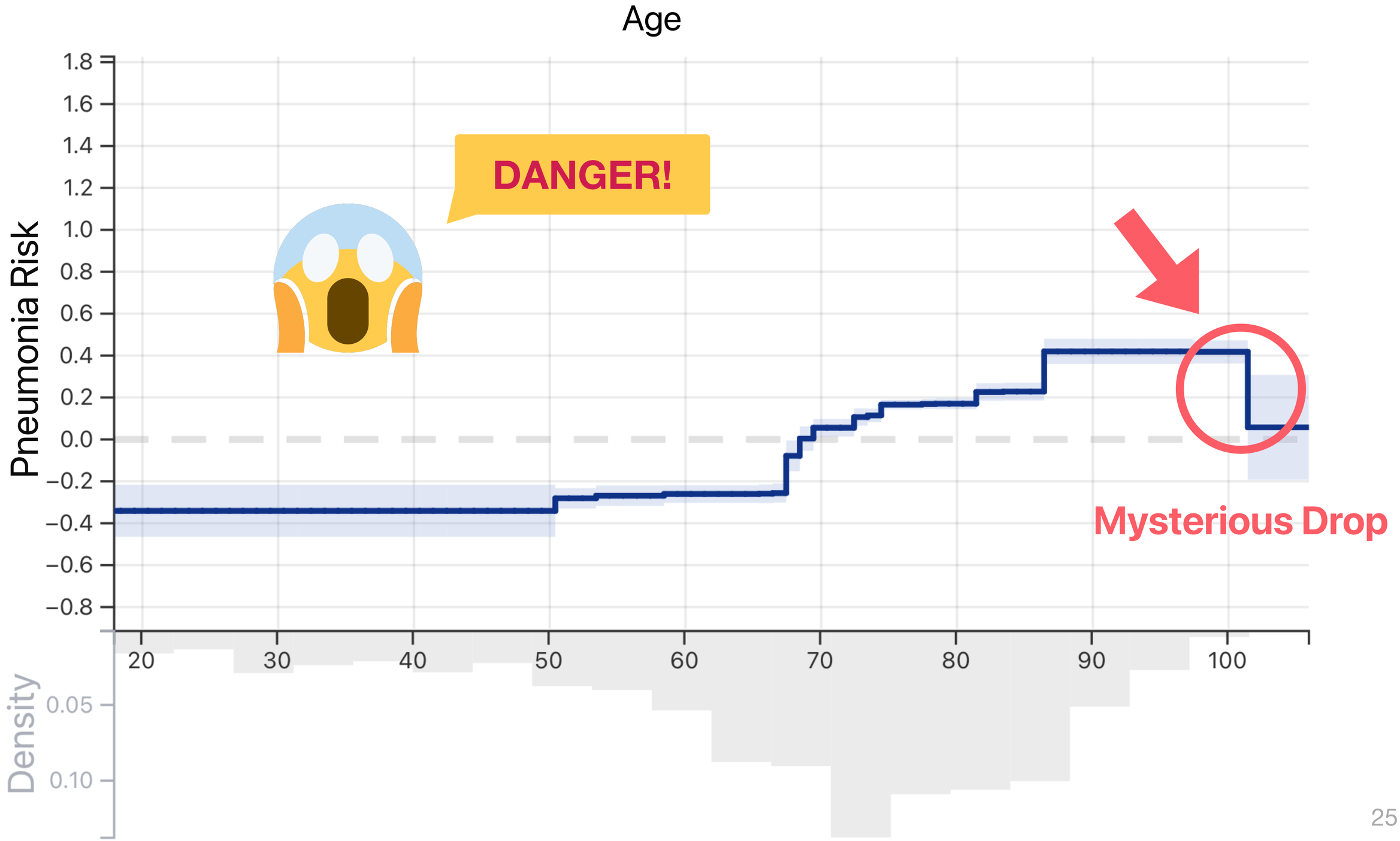

## Pneumonia mortality prediction

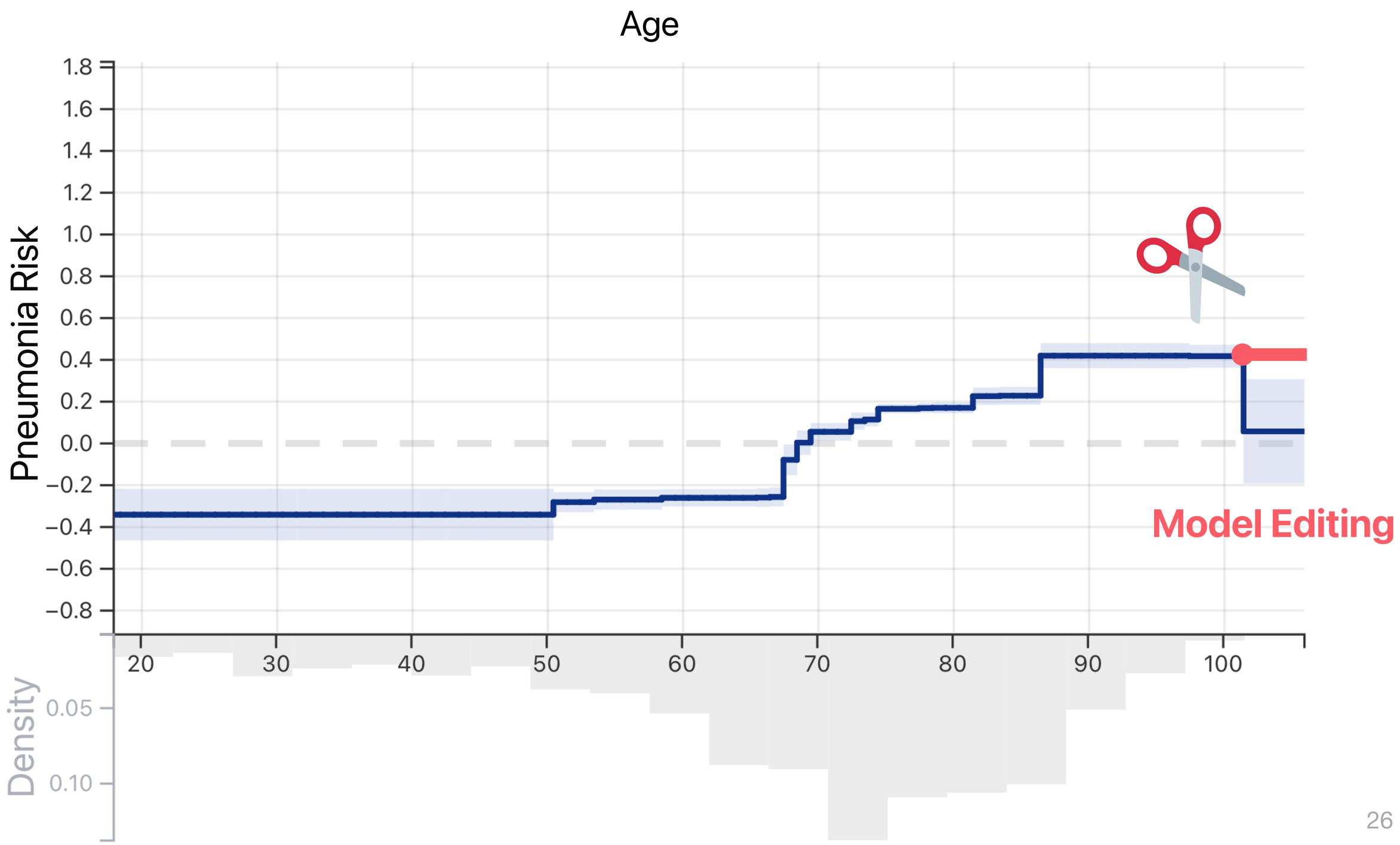
- Continuous, categorical, interaction

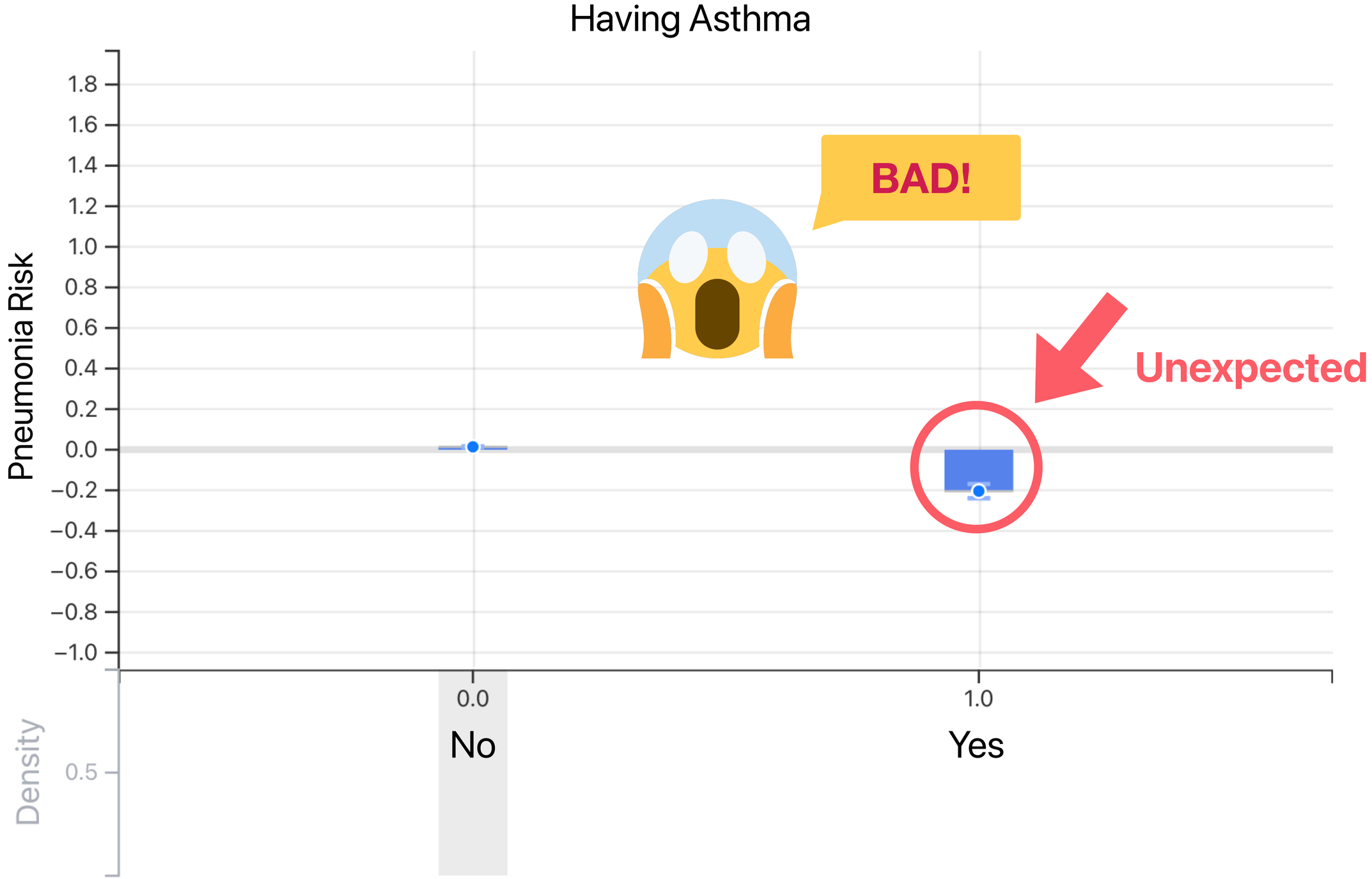

Age
Pneumonia Risk
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
20
30
40
50
60
70
80
90
100
Density
0.05
0.10

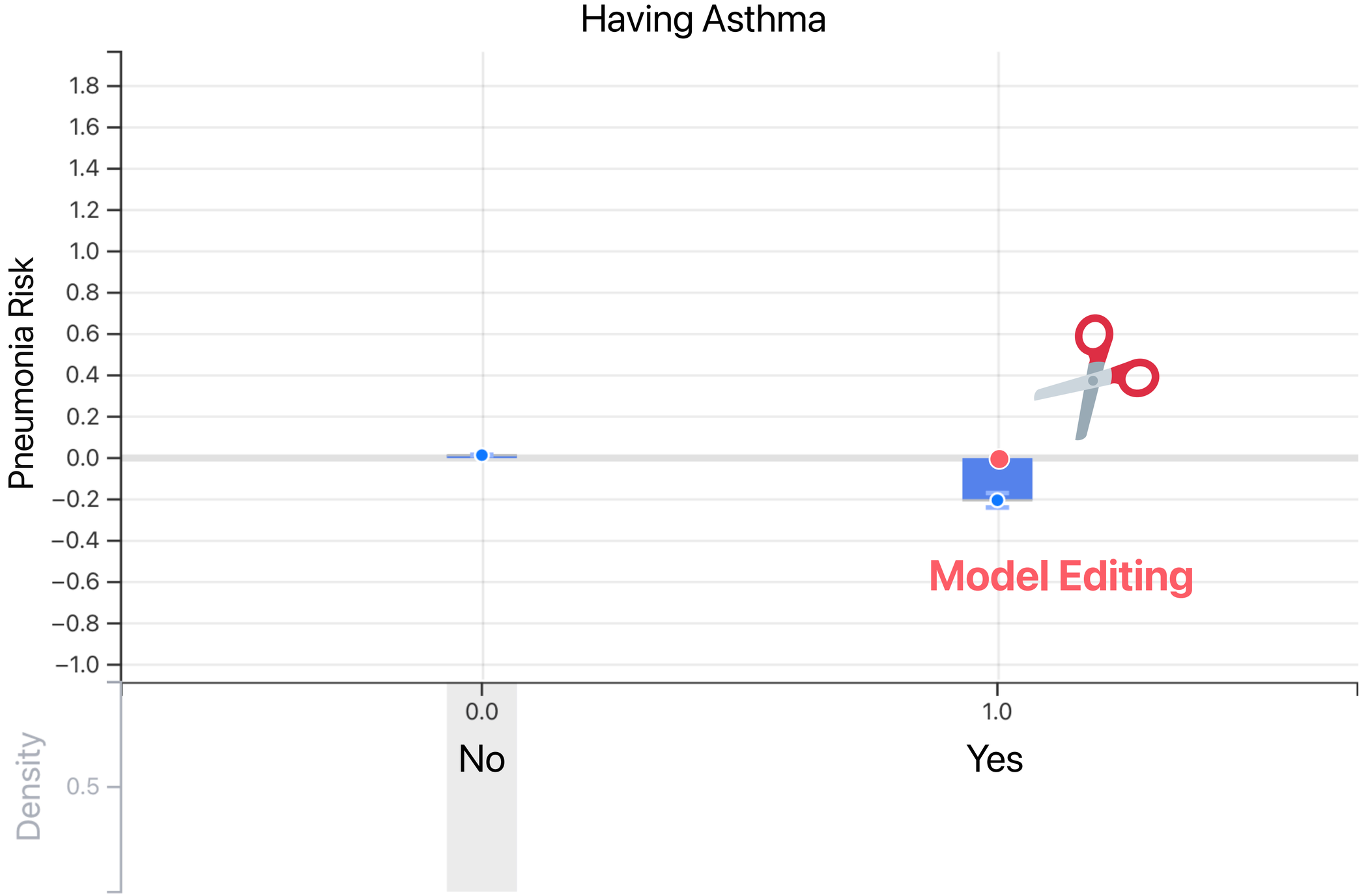

Age
Pneumonia Risk
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
Retirement?
Sudden Increase
20
30
40
50
60
70
80
90
100
Density
0.05
0.10


Age
Pneumonia Risk
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
Conservative treatment?
Sudden Increase
20
30
40
50
60
70
80
90
100
Density
0.05
0.10


Age
Pneumonia Risk
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
DANGER!
Mysterious Drop
20
30
40
50
60
70
80
90
100
Density
0.05
0.10


Age
Pneumonia Risk
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
20
30
40
50
60
70
80
90
100
Density
0.05
0.10
Model Editing


Having Asthma
Pneumonia Risk
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
BAD!
Unexpected
0.0
No
1.0
Yes
Density
0.5


Having Asthma
Pneumonia Risk
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
Model Editing
0.0
No
1.0
Yes
Density
0.5

# Real Needs for Model Editing

**Fix undesirable behaviors**

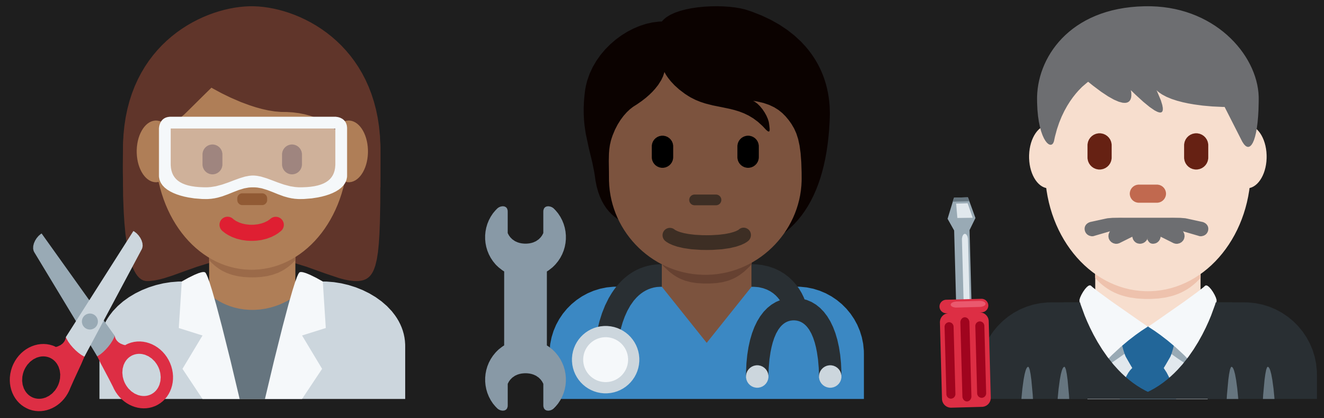
Higher age should have higher risk

**Remedy mistakes in the dataset**

Outliers, missing values, wrong data

**Fairness and Bias**

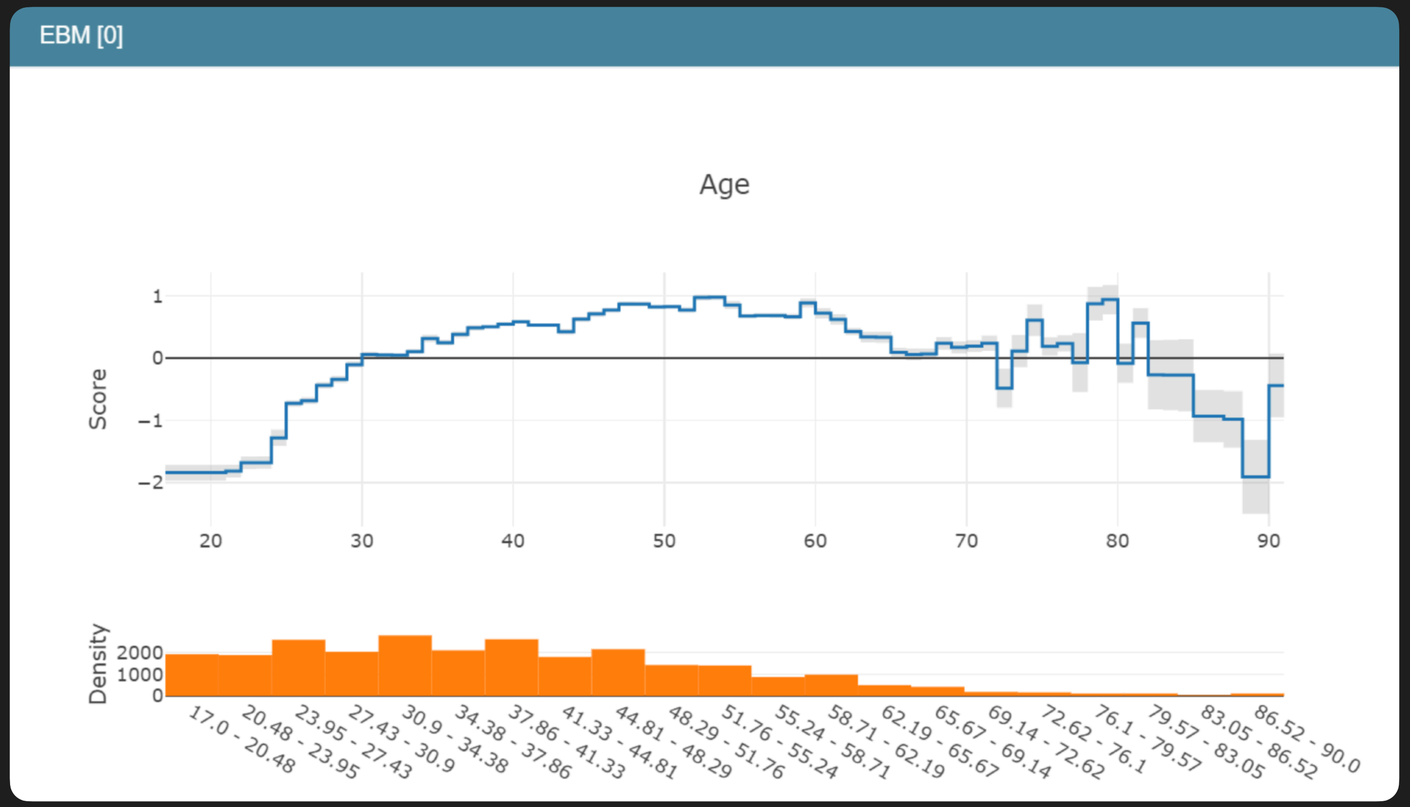
Change effects of protected attributes

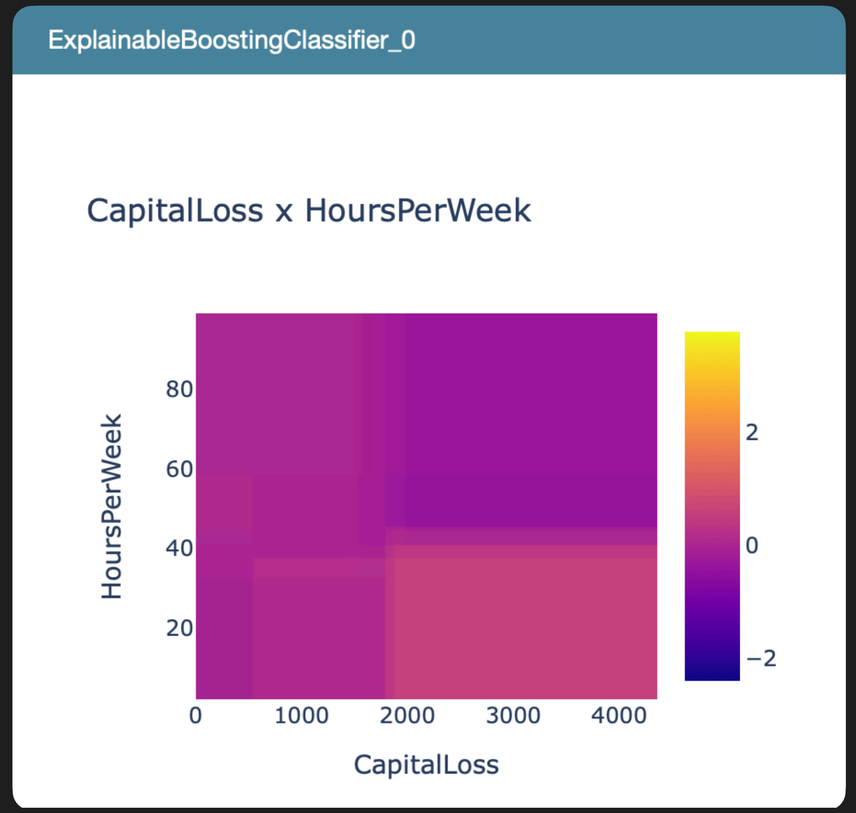


**Regulatory Compliance**

Enforce monotonicity required by law

# Visualization Tools for ML

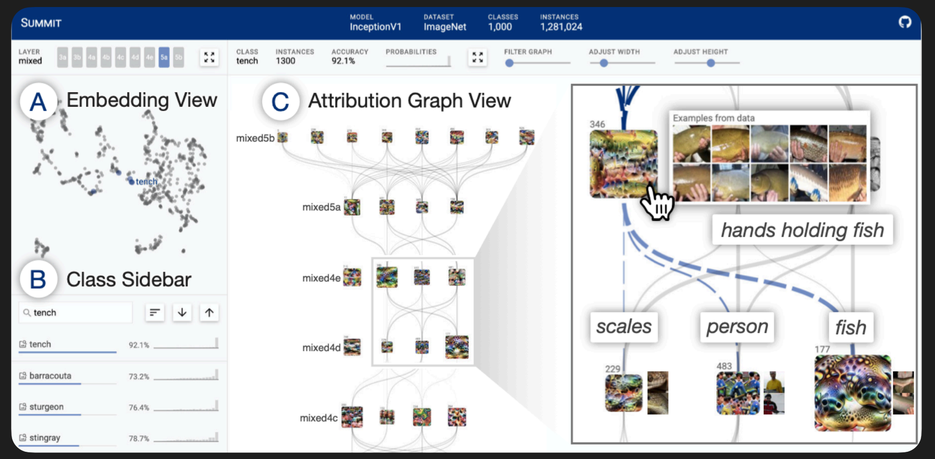


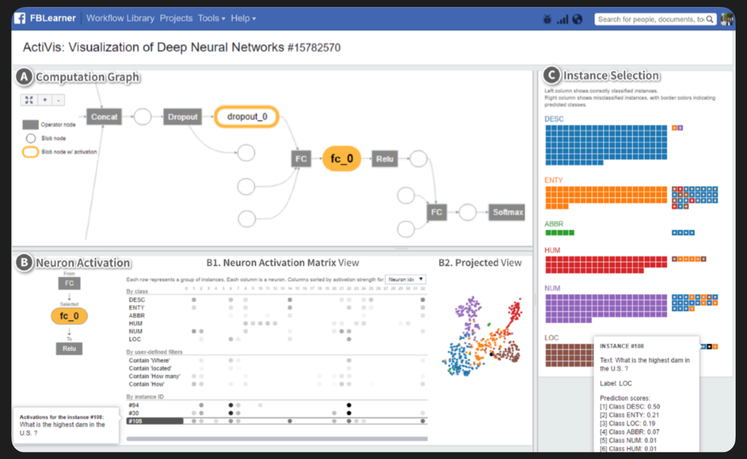

Nori, H., Jenkins, S., Koch, P., & Caruana, R. (2019). Interpretml: A unified framework for machine learning interpretability. arXiv preprint arXiv:1909.09223.

# Visualization Tools for ML

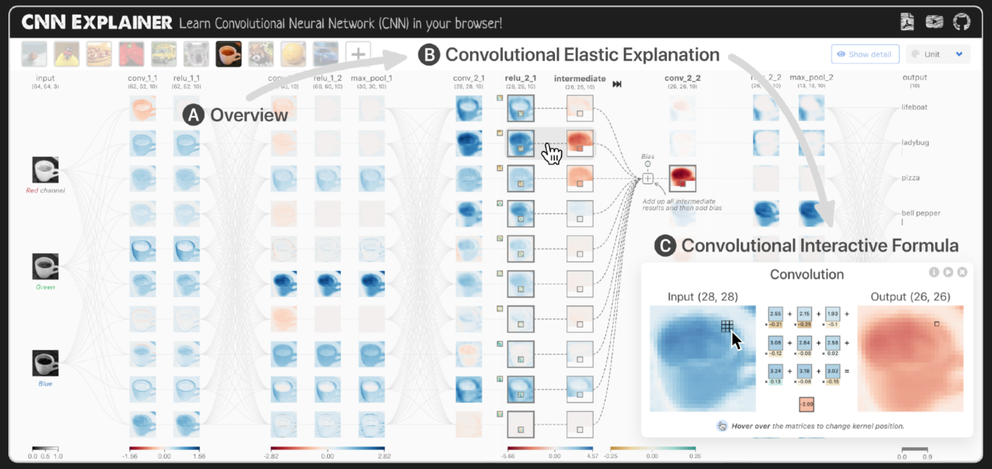
[Gamut Hohman19]



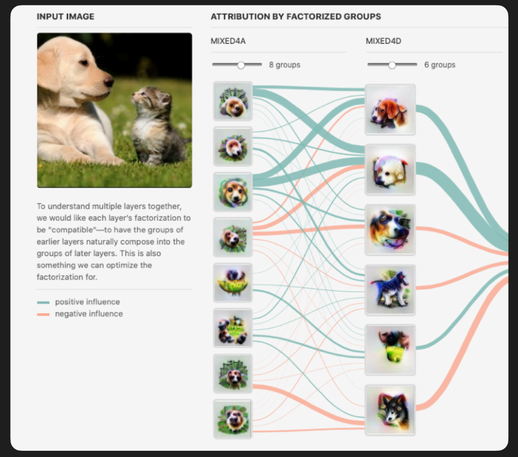

[Summit Hohman19]



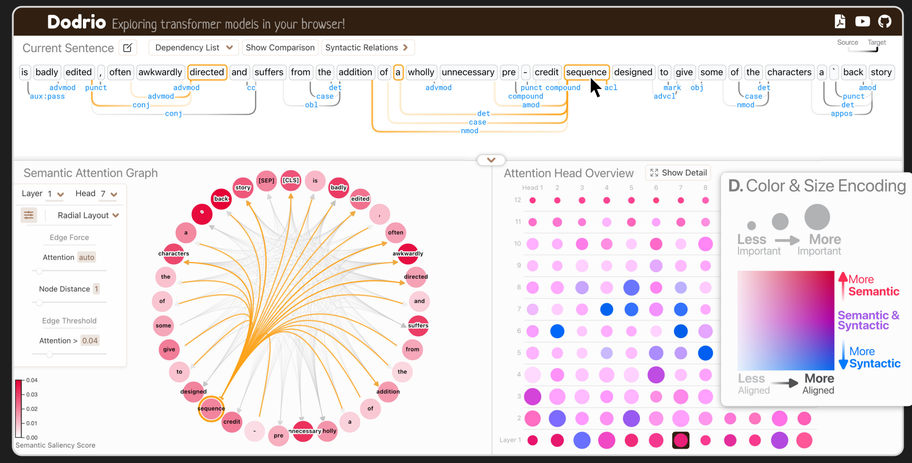

[ActiVis Kahng17]




[CNN Explainer Wang20]




[FeatureVis Olah17]



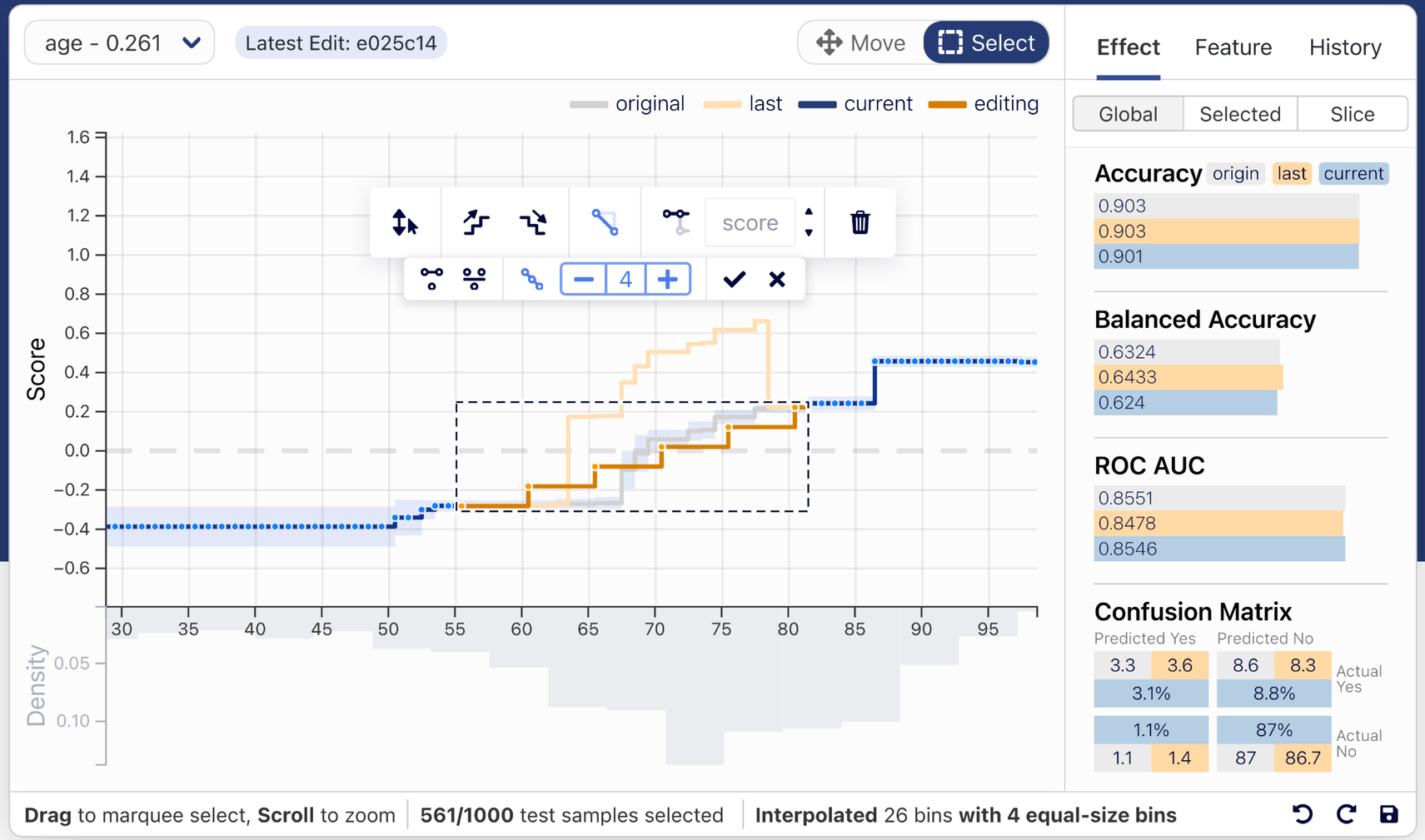

[Dodrio Wang21]

# GAM CHANGER

Align ML Model Behaviors with Human Knowledge

# GAM CHANGER

bit.ly/demo-msr




Code

Video

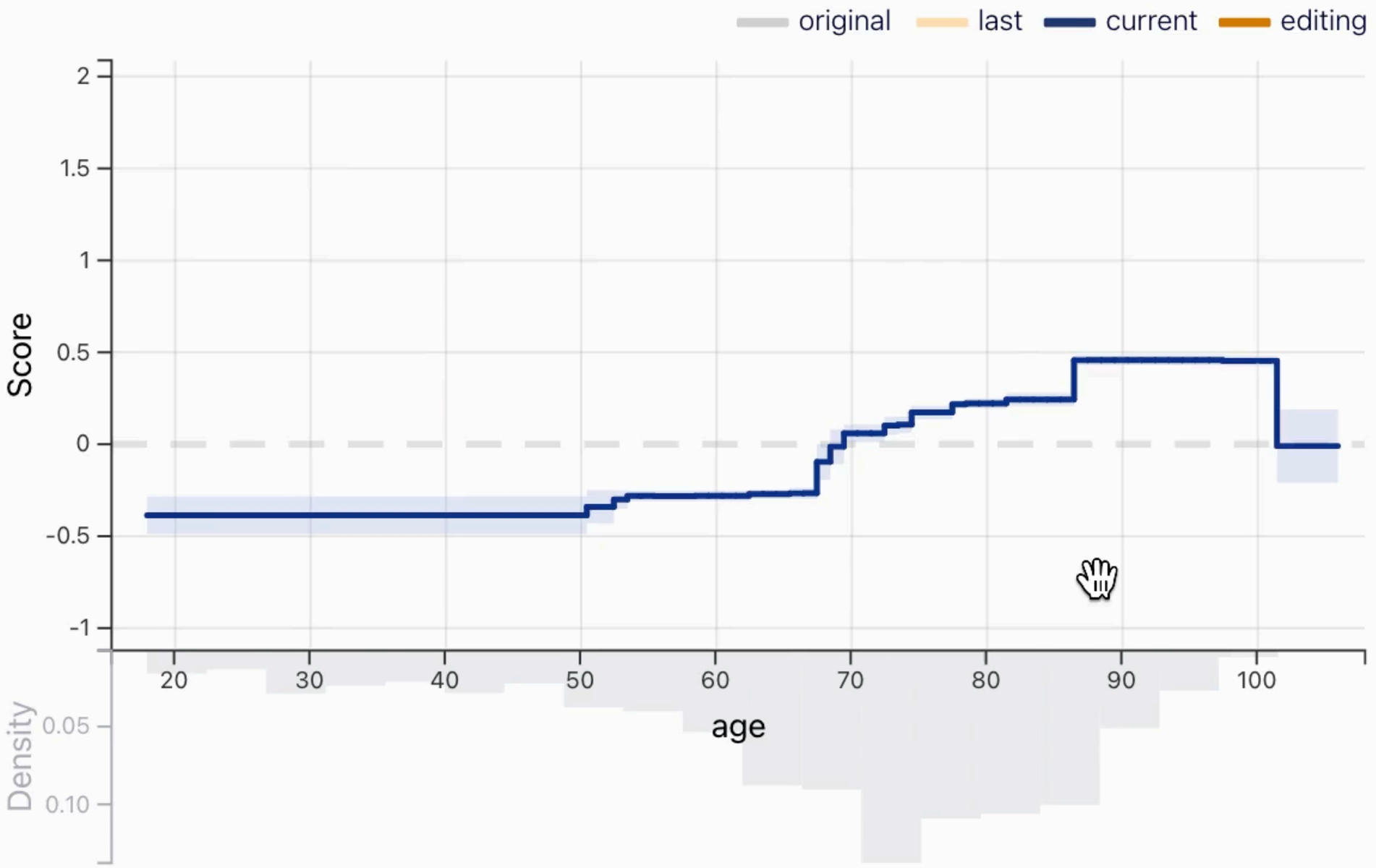
Paper

Choose a model: **PNEUMONIA MORTALITY** CENSUS INCOME MY MODEL

age - 0.261 | Original

Move | Select

Metrics | Feature | History




Global | Selected | Slice

## Accuracy

origin | last | current

0.9038
NA
0.9038

## Balanced Accuracy

0.606
NA
0.606

## ROC AUC

0.8758
NA
0.8758

## Confusion Matrix

| | Predicted Yes | | Predicted No | | |
|---|---|---|---|---|---|
| Actual Yes | 124 | NA | 432 | NA | |
| | 124 | | 432 | | |
| Actual No | 49 | | 4395 | | |
| | 49 | NA | 4395 | NA | |

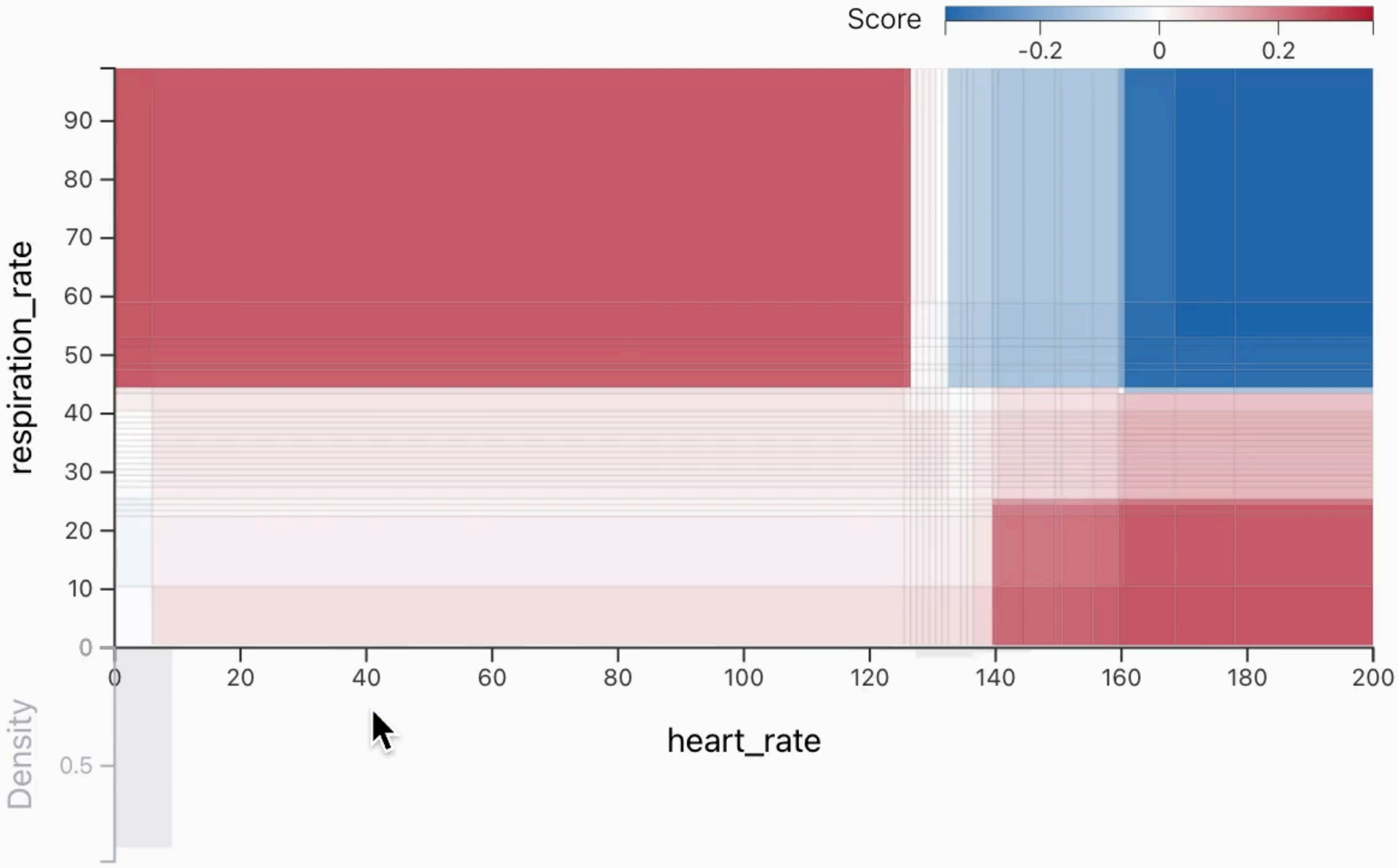
**Drag** to pan view, **Scroll** to zoom | **0/5000** test samples selected

heart_rate x respiration_rate - 0.018

Latest Edit: fb867b9

Move | Select

Metrics | Feature | History




Global | Selected | Slice

## Accuracy

origin | last | current

0.9038

NA

0.9038

## Balanced Accuracy

0.606

NA

0.606

## ROC AUC

0.8758

NA

0.8758

## Confusion Matrix

| | Predicted Yes | Predicted No | |
|---|---|---|---|
| Actual Yes | 124 NA<br>124 | 432 NA<br>432 | |
| Actual No | 49<br>49 NA | 4395<br>4395 NA | |

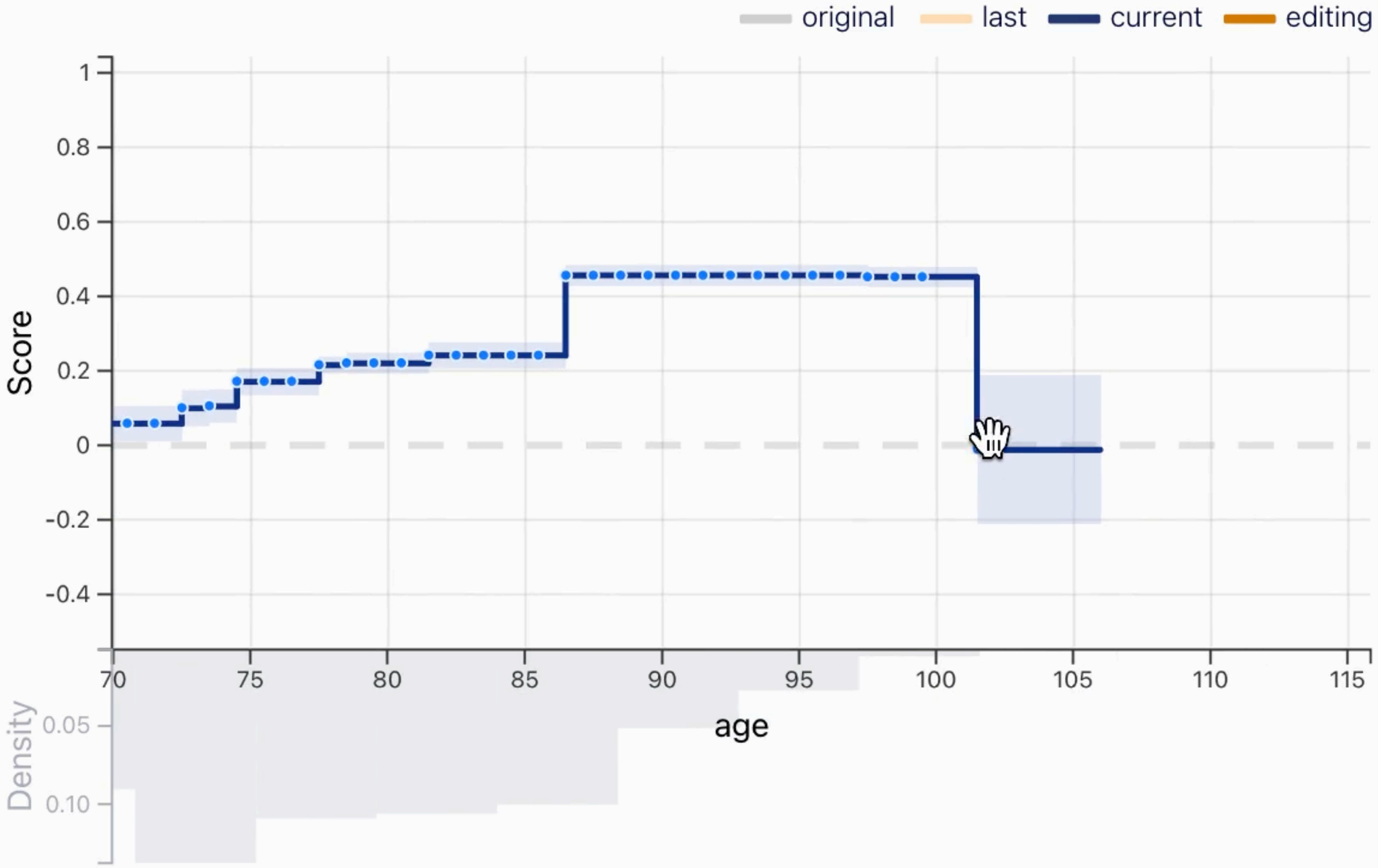
**Drag** to pan view, **Scroll** to zoom | **0/5000** test samples selected

age - 0.261

Latest Edit: 96d26af

Move | Select

Metrics | Feature | History




Global | Selected | Slice

## Accuracy

origin | last | current

0.9038

NA

0.9038

## Balanced Accuracy

0.606

NA

0.606

## ROC AUC

0.8758

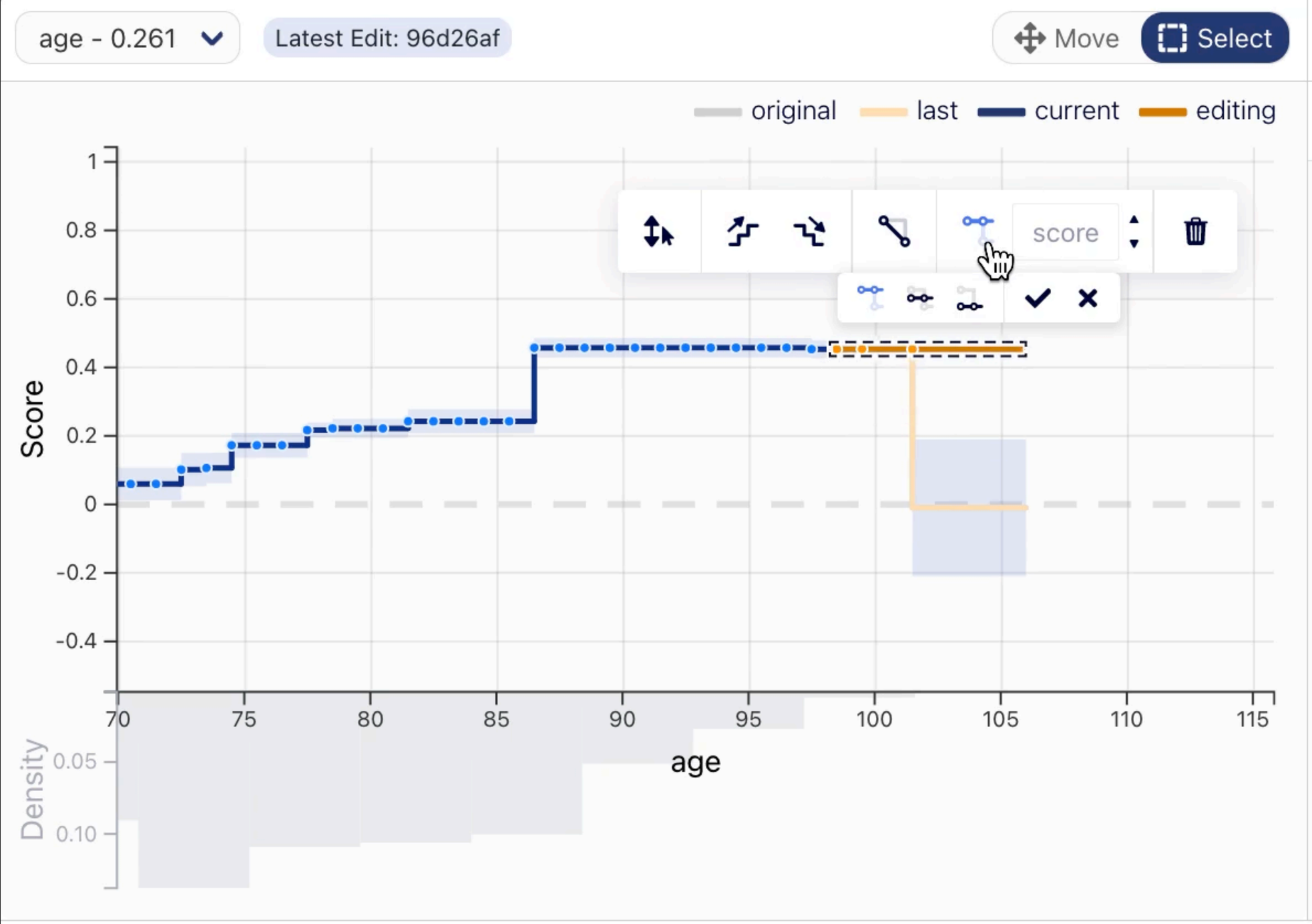
NA

0.8758

## Confusion Matrix

| | Predicted Yes | | Predicted No | | |
|---|---|---|---|---|---|
| | 124 | NA | 432 | NA | Actual Yes |
| | 124 | | 432 | | |
| | 49 | | 4395 | | Actual No |
| | 49 | NA | 4395 | NA | |

**Drag** to pan view, **Scroll** to zoom | **0/5000** test samples selected




Metrics | Feature | History

Global | Selected | Slice

## Accuracy

origin | last | current

0.9038
0.9038
0.9034

## Balanced Accuracy

0.606
0.606
0.6058

## ROC AUC

0.8758
0.8758
0.8755

## Confusion Matrix

| | Predicted Yes | | Predicted No | | |
|---|---|---|---|---|---|
| | 124 | 124 | 432 | 432 | Actual Yes |
| | 124 | | 432 | | |
| | 51 | | 4393 | | Actual No |
| | 49 | 49 | 4395 | 4395 | |

**Drag** to marquee select, **Scroll** to zoom | **28/5000** test samples selected | Set scores of 3 bins to **0.4533**

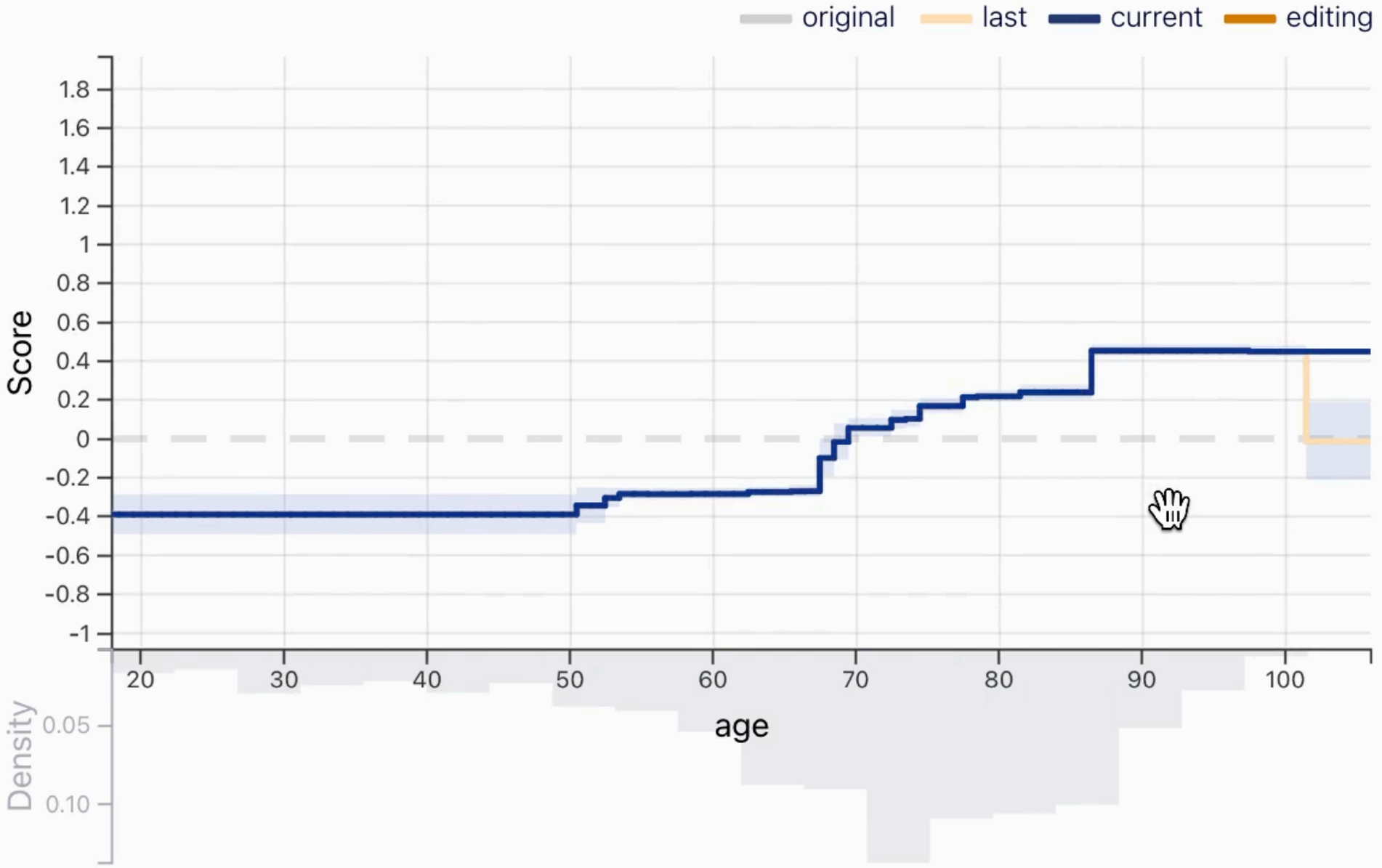

age - 0.261
Latest Edit: 96d26af
Move
Select
Metrics
Feature
History
original
last
current
editing
Continuous
Categorical
Frequency Distributions
Sorted by correlation ↓
all
select
score
Score
age
Density
altered_mental_status
gender: 1.0
chronic_lung_disease
congestive_heart_failure
renal_failure
wheezing
admitted_from_nursing_home
diabetes_mellitus
cerebrovascular_disease
heart_murmur
Drag to marquee select, Scroll to zoom
28/5000 test samples selected
Set scores of 3 bins to 0.4533

age - 0.261

Latest Edit: 9255aa6

Move | Select

Metrics | Feature | History




Global | Selected | Slice

## Accuracy

origin | last | current

0.9038
0.9038
0.9034

## Balanced Accuracy

0.606
0.606
0.6058

## ROC AUC

0.8758
0.8758
0.8755

## Confusion Matrix

| | Predicted Yes | Predicted No | |
|---|---|---|---|
| Actual Yes | 124 (124) | 432 (432) | 124 / 432 |
| Actual No | 51 (49) | 4393 (4395) | 49 / 4395 |

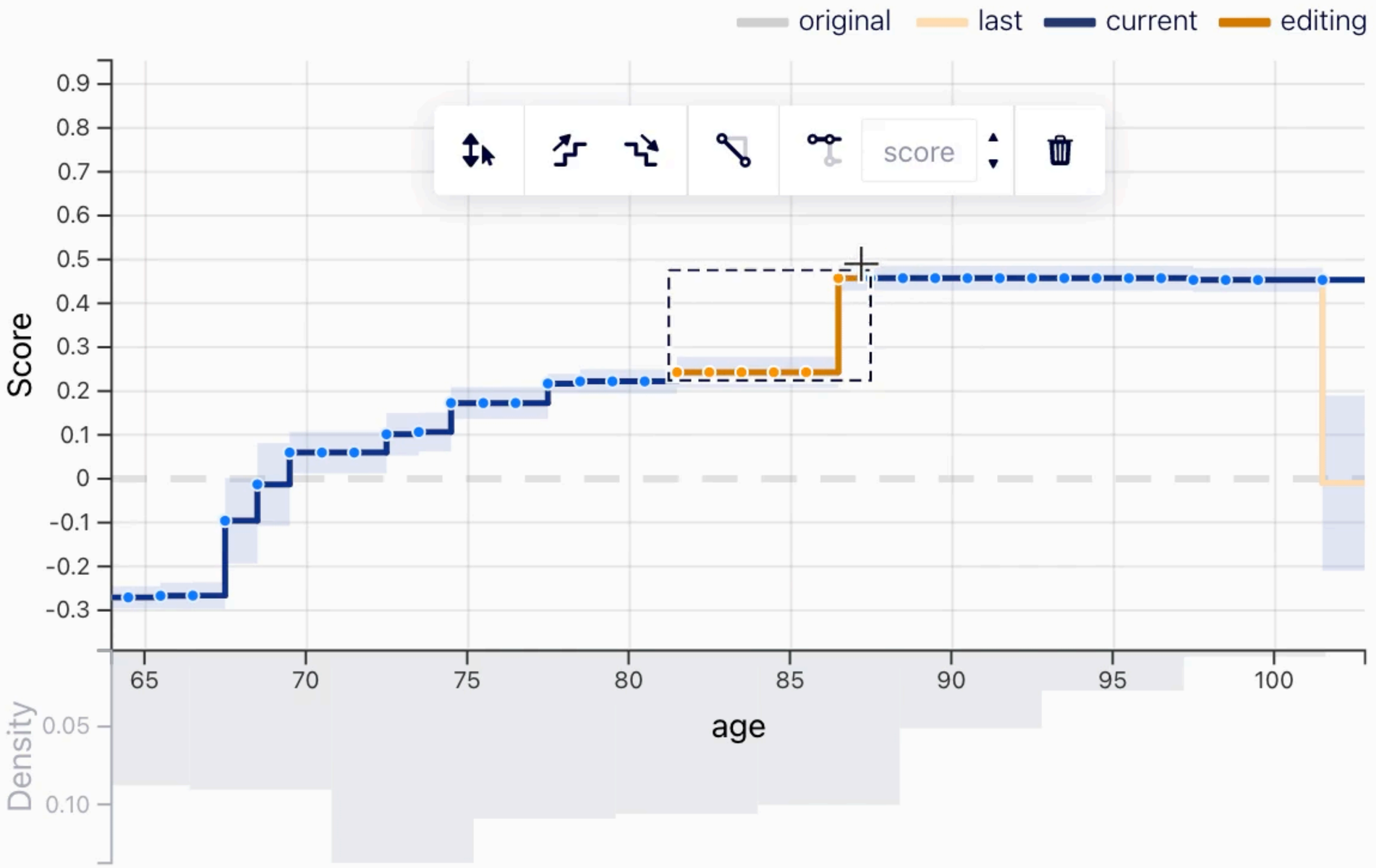
**Drag** to pan view, **Scroll** to zoom | **0/5000** test samples selected

age - 0.261

Latest Edit: 9255aa6

Move | Select

Metrics | Feature | History




Global | Selected | Slice

## Accuracy

origin | last | current

0.9038
0.9038
0.9034

## Balanced Accuracy

0.606
0.606
0.6058

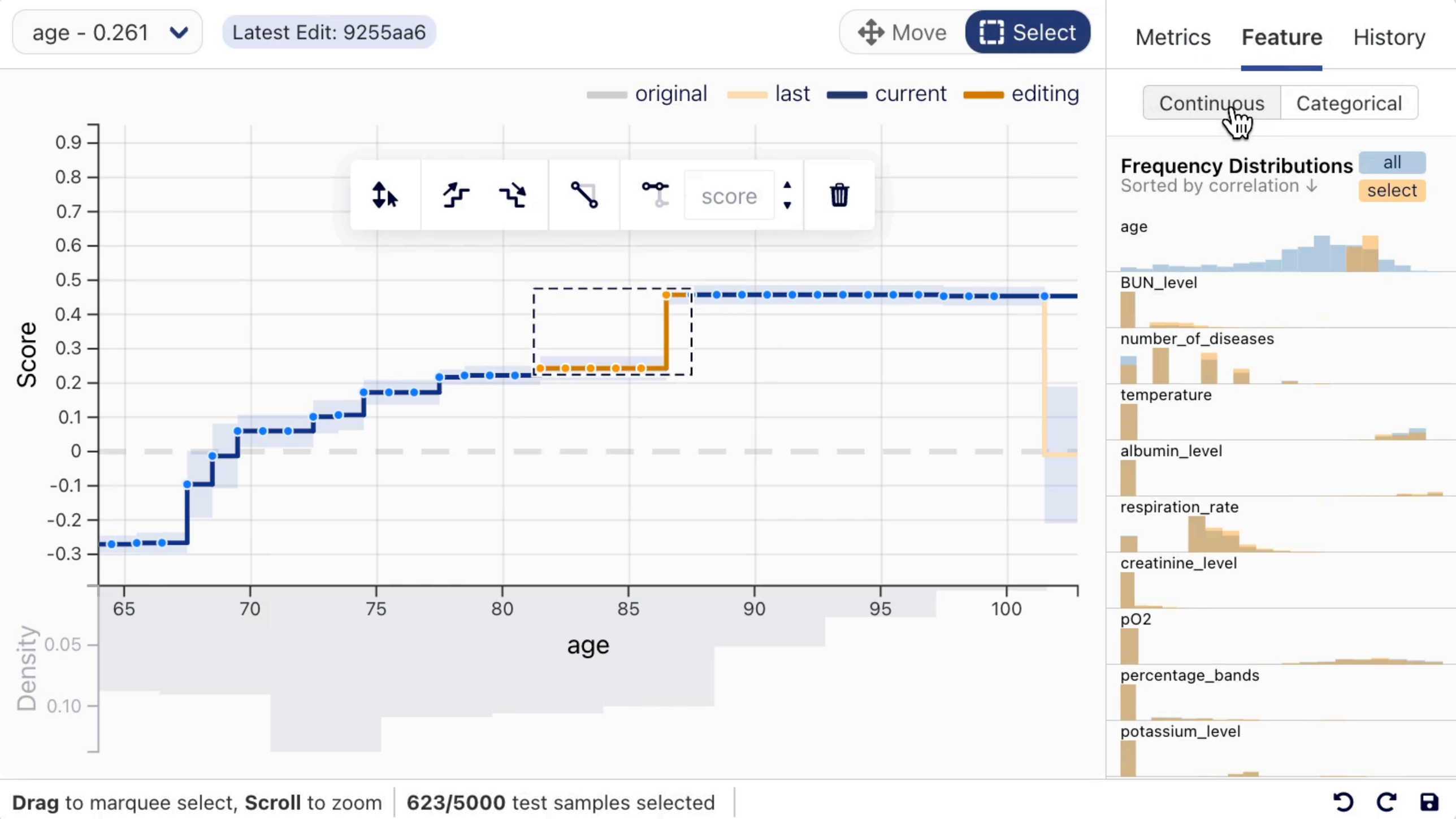
## ROC AUC

0.8758
0.8758
0.8755

## Confusion Matrix

| | Predicted Yes | | Predicted No | | |
|---|---|---|---|---|---|
| | 124 | 124 | 432 | 432 | Actual Yes |
| | 124 | | 432 | | |
| | 51 | | 4393 | | Actual No |
| | 49 | 49 | 4395 | 4395 | |

**Drag** to marquee select, **Scroll** to zoom | **623/5000** test samples selected

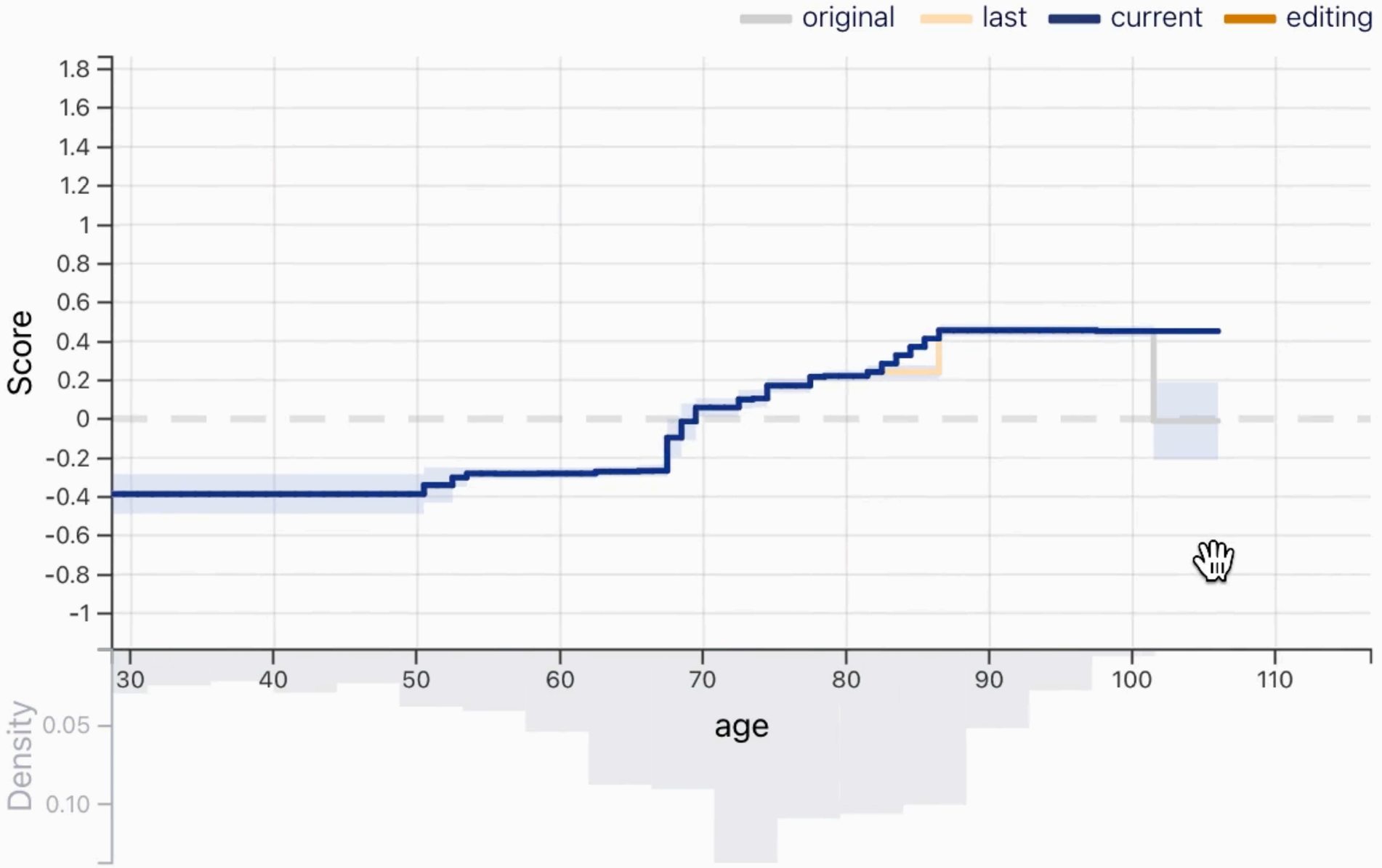

age - 0.261
Latest Edit: 9255aa6
Move
Select
Metrics
Feature
History
original
last
current
editing
Continuous
Categorical
Frequency Distributions
Sorted by correlation ↓
all
select
score
Score
age
Density
BUN_level
number_of_diseases
temperature
albumin_level
respiration_rate
creatinine_level
pO2
percentage_bands
potassium_level
Drag to marquee select, Scroll to zoom
623/5000 test samples selected

age - 0.261

Latest Edit: 897f310

Move | Select

Metrics | Feature | History




Global | Selected | Slice

## Accuracy

origin last current

0.9038
0.9034
0.9036

## Balanced Accuracy

0.606
0.6058
0.6067

## ROC AUC

0.8758
0.8755
0.8756

## Confusion Matrix

| | Predicted Yes | | Predicted No | | |
|---|---|---|---|---|---|
| Actual Yes | 124 | 124 | 432 | 432 | |
| | 125 | | 431 | | |
| Actual No | 51 | | 4393 | | |
| | 49 | 51 | 4395 | 4393 | |

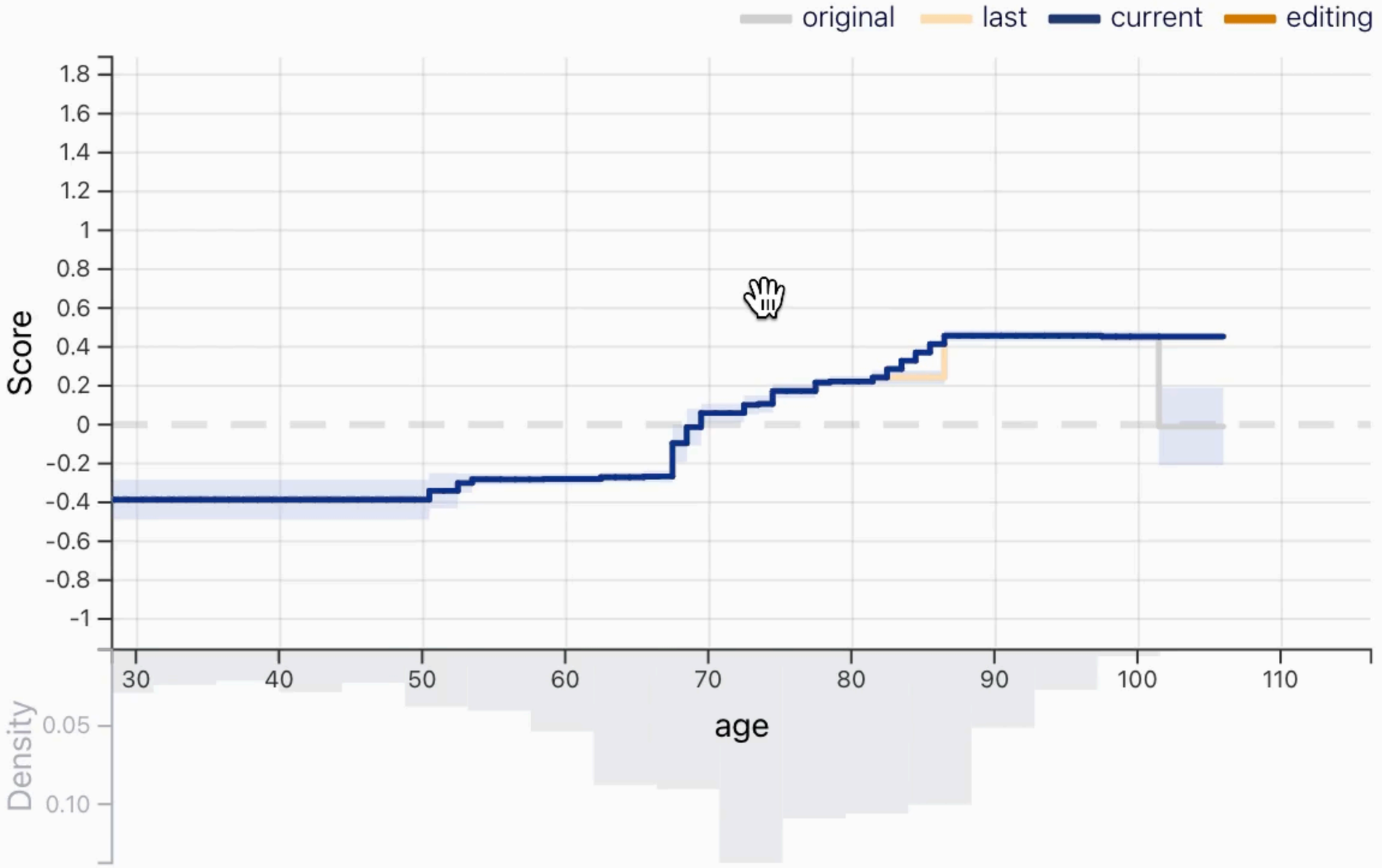
**Drag** to pan view, **Scroll** to zoom | **0/5000** test samples selected

age - 0.261

Latest Edit: 897f310

Move | Select

Metrics | Feature | History




Global | Selected | Slice

## Accuracy

origin last current

0.9038
0.9034
0.9036

## Balanced Accuracy

0.606
0.6058
0.6067

## ROC AUC

0.8758
0.8755
0.8756

## Confusion Matrix

| | Predicted Yes | | Predicted No | | |
|---|---|---|---|---|---|
| Actual Yes | 124 | 124 | 432 | 432 | |
| | 125 | | 431 | | |
| Actual No | 51 | | 4393 | | |
| | 49 | 51 | 4395 | 4393 | |

**Drag** to pan view, **Scroll** to zoom | **0/5000** test samples selected


asthma - 0.027
Latest Edit: 42087fa
Move
Select
Metrics
Feature
History
original
last
current
editing
Continuous
Categorical
Frequency Distributions
Sorted by correlation ↓
all
select
number_of_diseases
age
BUN_level
pO2
albumin_level
creatinine_level
temperature
hematocrit
WBC_count
respiration_rate
score
Score
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
0.0
1.0
asthma
Density
0.5
Drag to pan view, Scroll to zoom
361/5000 test samples selected


asthma - 0.027
Latest Edit: 42087fa
Move
Select
Metrics
Feature
History
original
last
current
editing
Continuous
Categorical
Frequency Distributions
Sorted by correlation ↓
all
select
asthma
wheezing
chronic_lung_disease
gender
altered_mental_status
admitted_from_nursing_home
cerebrovascular_disease
cancer
liver_function_tests
congestive_heart_failure
score
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.2
Score
0.0
1.0
asthma
0.5
Density
Drag to pan view, Scroll to zoom
361/5000 test samples selected

# Design Principles for Model Editing

**P1** **Diverse Editing Methods**
Different editing technique for different scenarios

**P2** **Communicating Feedback**
Responsible editing & mitigate editing bias

**P3** **Reversibility and Documentation**
Model version control & disposable experiment

**P4** **Low Barrier to Entry**
Empower everyone to exercise human agency

# P1 Diverse Editing Methods

## [Monotonically Increase]

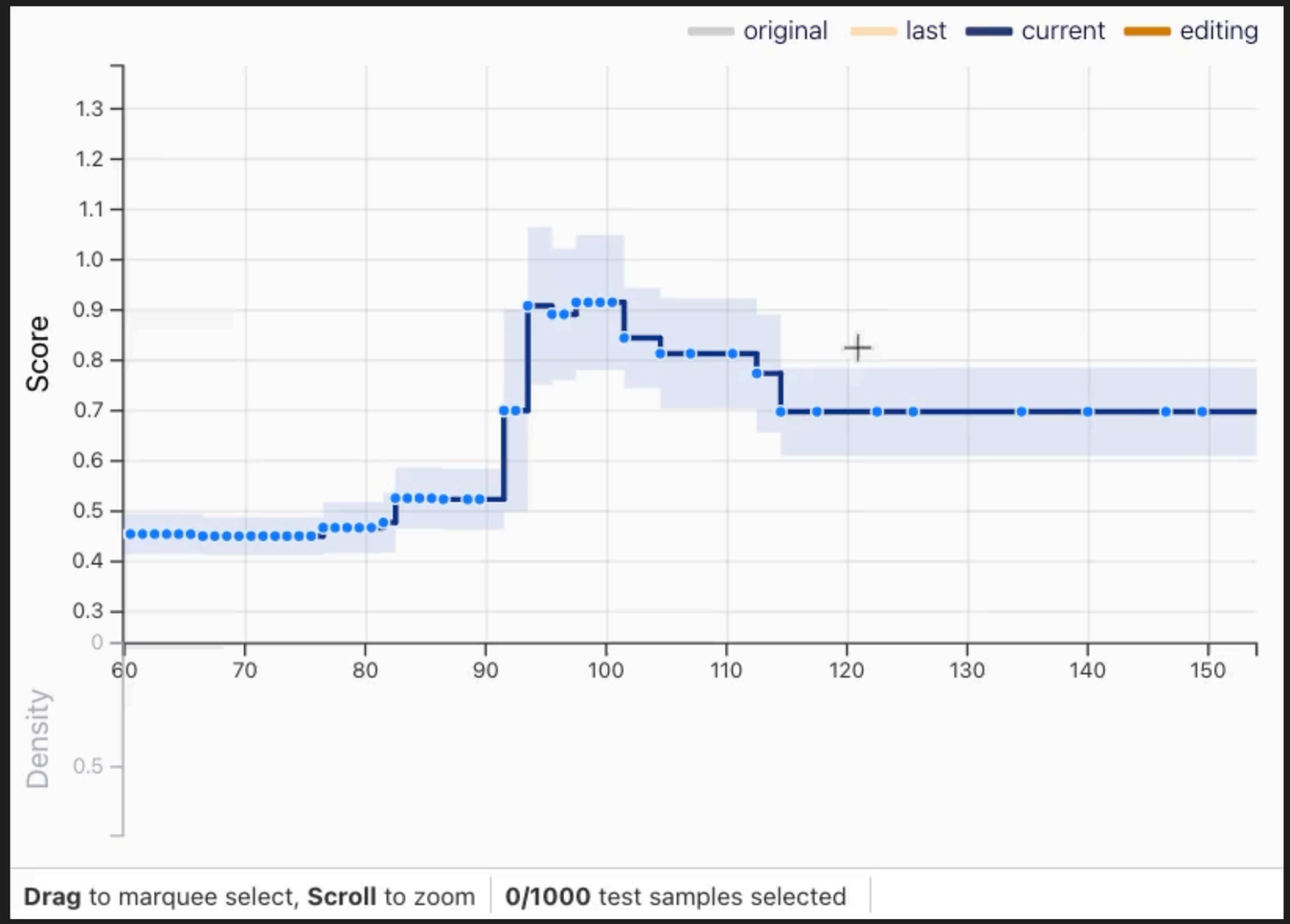
Enforce monotonicity
Make the curve smoother

## [Interpolate in-place]

Remove bumps
Enforce linearity

## [Align to the left]

Nullify feature effect
Broadcast effect



# P2 Communicating Feedback

## Model Performance

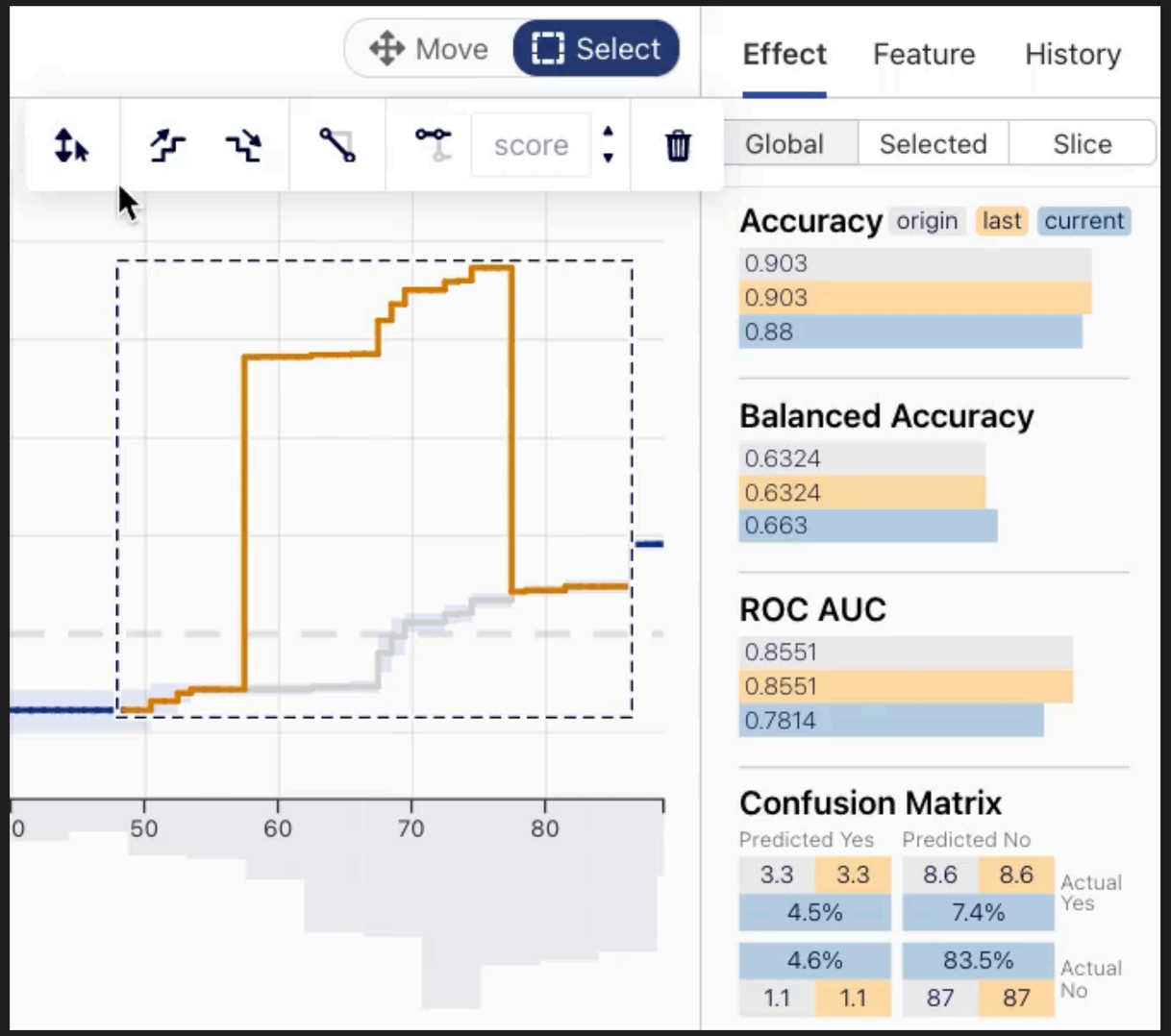
How would it affect the accuracy?

How would it affect my protected group?

## Feature Correlations

Other proxy features?

Features need to edit next?



# P2 Communicating Feedback

## Model Performance

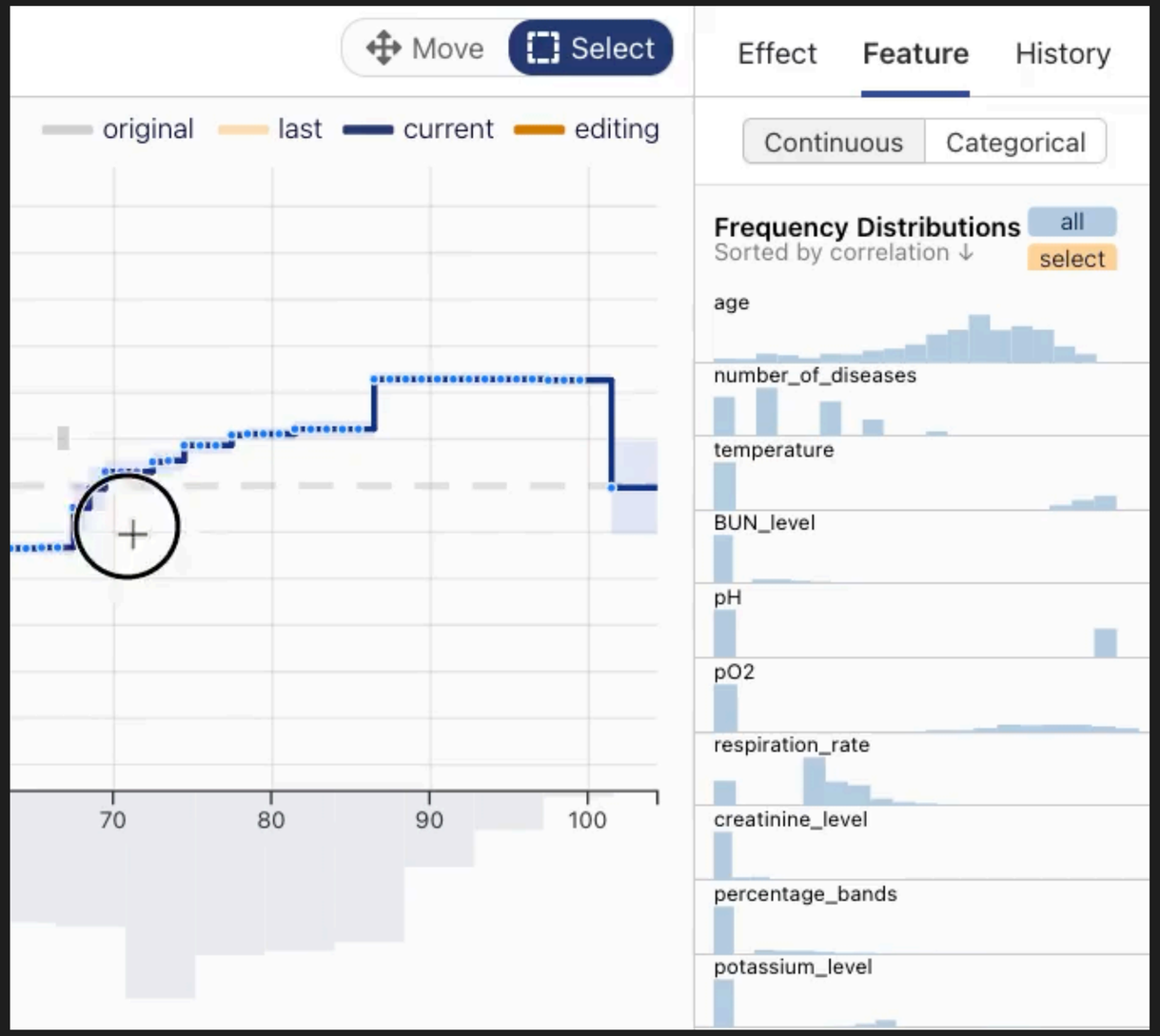
How would it affect the accuracy?

How would it affect my protected group?

## Feature Correlations

Other proxy features?

Features need to edit next?



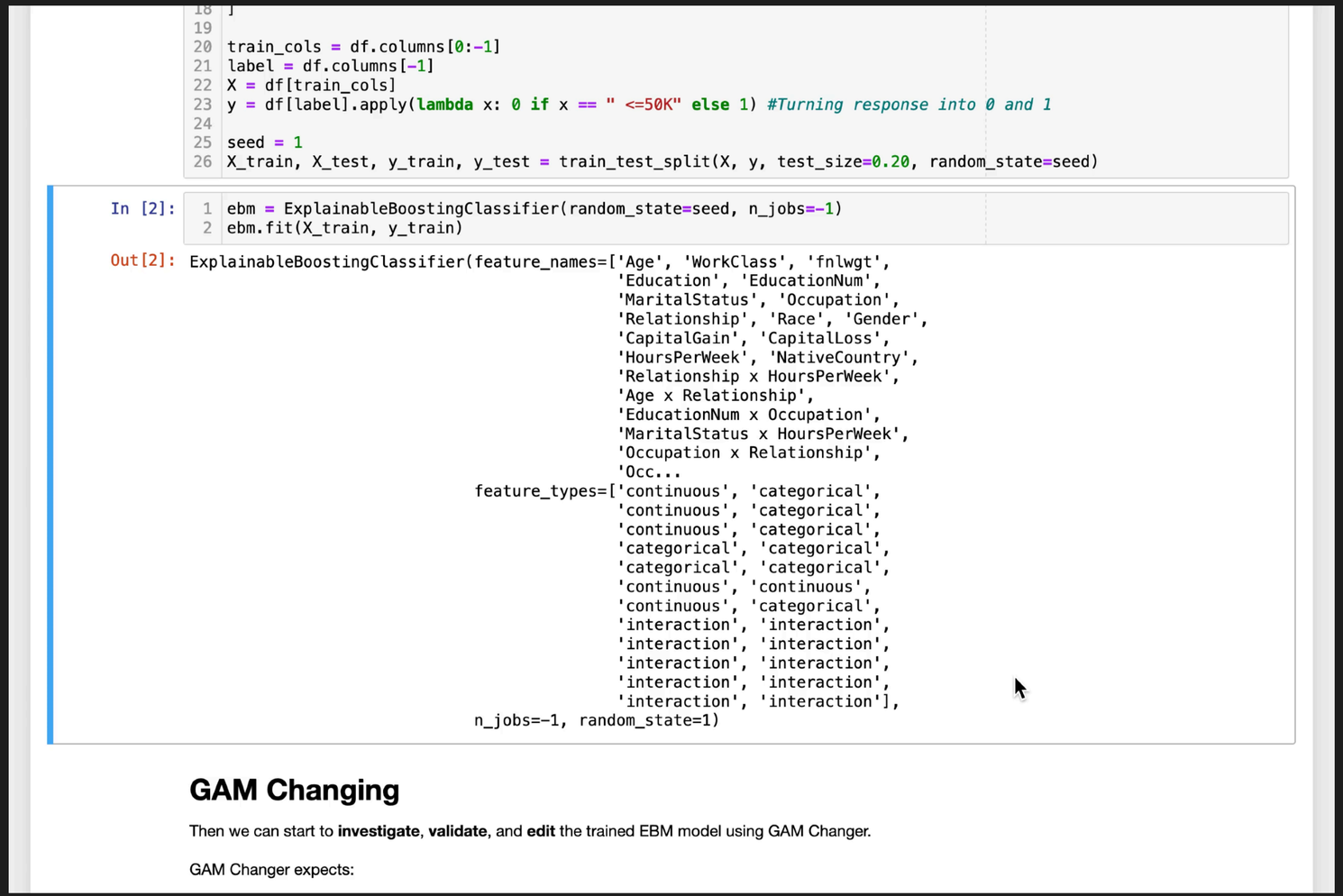
# P3 Reversibility and Documentation

- Values and knowledge **change** over time
- **Disposable** experiment
- **Collaborative** editing

```
git commit -m "Change Age's effect for people older than 85"
```

# P4 Low Barrier to Entry

- **No need to** install/write code
- Fit into the **current workflow**
- **Private and secure**
- **Democratize** responsible ML



```
]

train_cols = df.columns[0:-1]
label = df.columns[-1]
X = df[train_cols]
y = df[label].apply(lambda x: 0 if x == " <=50K" else 1) #Turning response into 0 and 1

seed = 1
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=seed)
```

```
In [2]:
ebm = ExplainableBoostingClassifier(random_state=seed, n_jobs=-1)
ebm.fit(X_train, y_train)
```

```
Out[2]: ExplainableBoostingClassifier(feature_names=['Age', 'WorkClass', 'fnlwgt',
                                                    'Education', 'EducationNum',
                                                    'MaritalStatus', 'Occupation',
                                                    'Relationship', 'Race', 'Gender',
                                                    'CapitalGain', 'CapitalLoss',
                                                    'HoursPerWeek', 'NativeCountry',
                                                    'Relationship x HoursPerWeek',
                                                    'Age x Relationship',
                                                    'EducationNum x Occupation',
                                                    'MaritalStatus x HoursPerWeek',
                                                    'Occupation x Relationship',
                                                    'Occ...
                                    feature_types=['continuous', 'categorical',
                                                   'continuous', 'categorical',
                                                   'continuous', 'categorical',
                                                   'categorical', 'categorical',
                                                   'categorical', 'categorical',
                                                   'continuous', 'continuous',
                                                   'continuous', 'categorical',
                                                   'interaction', 'interaction',
                                                   'interaction', 'interaction',
                                                   'interaction', 'interaction',
                                                   'interaction', 'interaction',
                                                   'interaction', 'interaction'],
                                    n_jobs=-1, random_state=1)
```

## GAM Changing

Then we can start to **investigate**, **validate**, and **edit** the trained EBM model using GAM Changer.

GAM Changer expects:

# GAM CHANGER

bit.ly/**demo-msr**

Align ML Model Behaviors with Human Knowledge

## GAM Changer: Editing Generalized Additive Models with Interactive Visualization


Zijie J. Wang[1] Alex Kale[2] Harsha Nori[3] Peter Stella[4] Mark Nunnally[4] Duen Horng Chau[1]
Mihaela Vorvoreanu[3] Jennifer Wortman Vaughan[3] Rich Caruana[3]
[1]Georgia Tech [2]University of Washington [3]Microsoft Research [4]NYU Langone Health

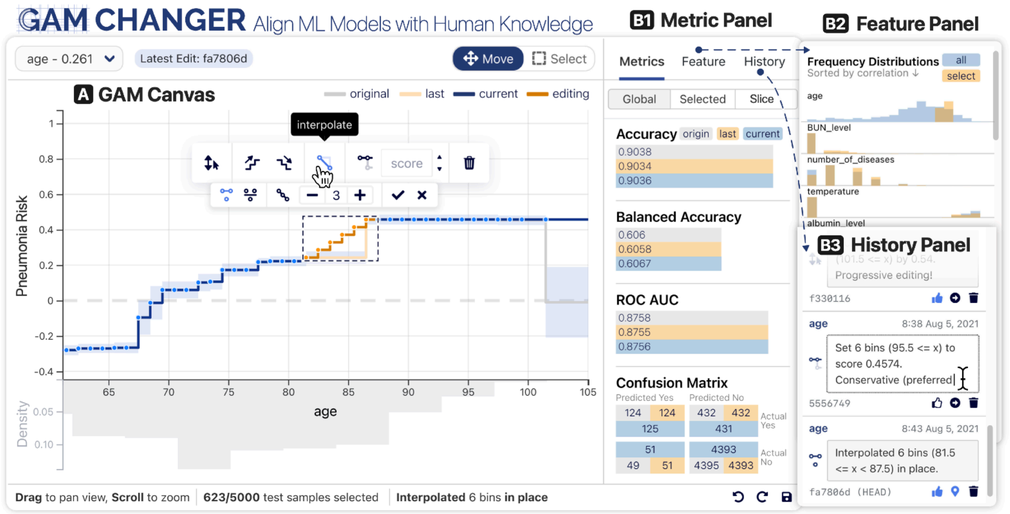




**Figure 1: GAM Changer empowers domain experts and data scientists to easily and responsibly align model behaviors with their domain knowledge and values, via direct manipulation of GAM model weights. For example, A the *GAM Canvas* enables doctors to interpolate the predicted risk of dying from pneumonia to match their domain knowledge of a gradual risk increase from age 81 to age 87. GAM Changer promotes accountable editing and elucidates potential tradeoffs induced by the edits. B1 The *Metric Panel* provides real time feedback on model performance. B2 The *Feature Panel* helps users identify characteristics of affected samples and promotes awareness of fairness issues. To enable reversible transparent model edits, B3 the *History Panel* allows the doctor to compare and revert changes, as well as document their motivations and editing contexts.**


### ABSTRACT

Recent strides in interpretable machine learning (ML) research reveal that models exploit undesirable patterns in the data to make predictions, which potentially causes harms in deployment. However,


### 1 INTRODUCTION

It is crucial to understand how machine learning (ML) models used in high-stakes settings (e.g., healthcare, finance, and criminal justice) make predictions (Fig. 2A). Recently, researchers have made

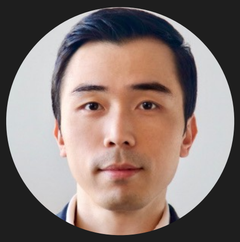

**Jay Wang**
@jay4w Georgia Tech

Alex Kale
Uni of Washington

Harsha Nori
Microsoft

Peter Stella
NYU Langone Health

Mark Nunnally
NYU Langone Health



Polo Chau
Georgia Tech

Mickey Vorvoreanu
Microsoft Research

Jenn Wortman Vaughan
Microsoft Research

Rich Caruana
Microsoft Research

# StickyLand

Break the Linear Presentation of Computational Notebooks

bit.ly/stickyland

pip install stickyland

Thanks!

## StickyLand: Breaking the Linear Presentation of Computational Notebooks


Zijie J. Wang
Georgia Tech
jayw@gatech.edu

Katie Dai
Georgia Tech
kdai7@gatech.edu

W. Keith Edwards
Georgia Tech
keith@cc.gatech.edu

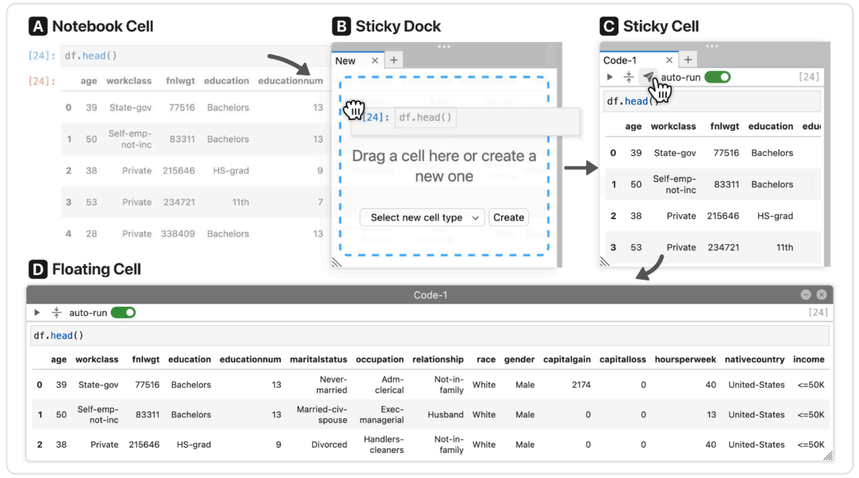




**Figure 1: The StickyLand user interface persists on top of a computational notebook, creating an always-on workspace for data scientists to display text and execute code. (A) A user can drag an existing *Notebook Cell* to (B) the *Sticky Dock* that mounts on the edge of the notebook. (C) It creates a *Sticky Cell* that stays at the same location even when the user scrolls the notebook. (D) The user can further convert this cell to a *Floating Cell* that floats on top of the notebook with customizable position and size.**


### ABSTRACT

How can we better organize code in computational notebooks? Notebooks have become a popular tool among data scientists, as they seamlessly weave text and code together, supporting users to rapidly iterate and document code experiments. However, it is often challenging to organize code in notebooks, partially because always shown on the screen, users can quickly access their notes, instantly observe experiment results, and easily build interactive dashboards that support complex visual analytics. Case studies highlight how our tool can enhance notebook users's productivity and identify opportunities for future notebook designs. StickyLand is available at https://github.com/anonchi/stickyland.


Jay Wang @Jay4W

Katie Dai

Keith Edwards

Georgia Tech