INTERPRETABLE MACHINE LEARNING WITH SHAP

Brianna Frederick

General Mills

brianna.frederick56@gmail.com

AGENDA

- What is ML Model Explainability?
  - Why is it important?
- SHAP Intro
- LIME Intro
- SHAP/LIME Comparison
- Example Output
- Python Examples
- Q&A

- Python code and data available at: https://github.com/briannafrederick/interpretable_ml_shap

ABOUT ME

My Work

- General Mills (~4 yrs)
  - Data Science Manager, Enterprise Data Capabilities
  - Senior Data Scientist, Enterprise Data Capabilities
  - Data Scientist, Global Consumer Insights
- Hamline University, Course Instructor
  - MSBA Practicum
  - Forecasting and Modeling

My Life

- Daughter, Jane (21 months)
- Husband, JP
- 2 Good Pups, Robin and Baron and 1 Good Cat, Karl

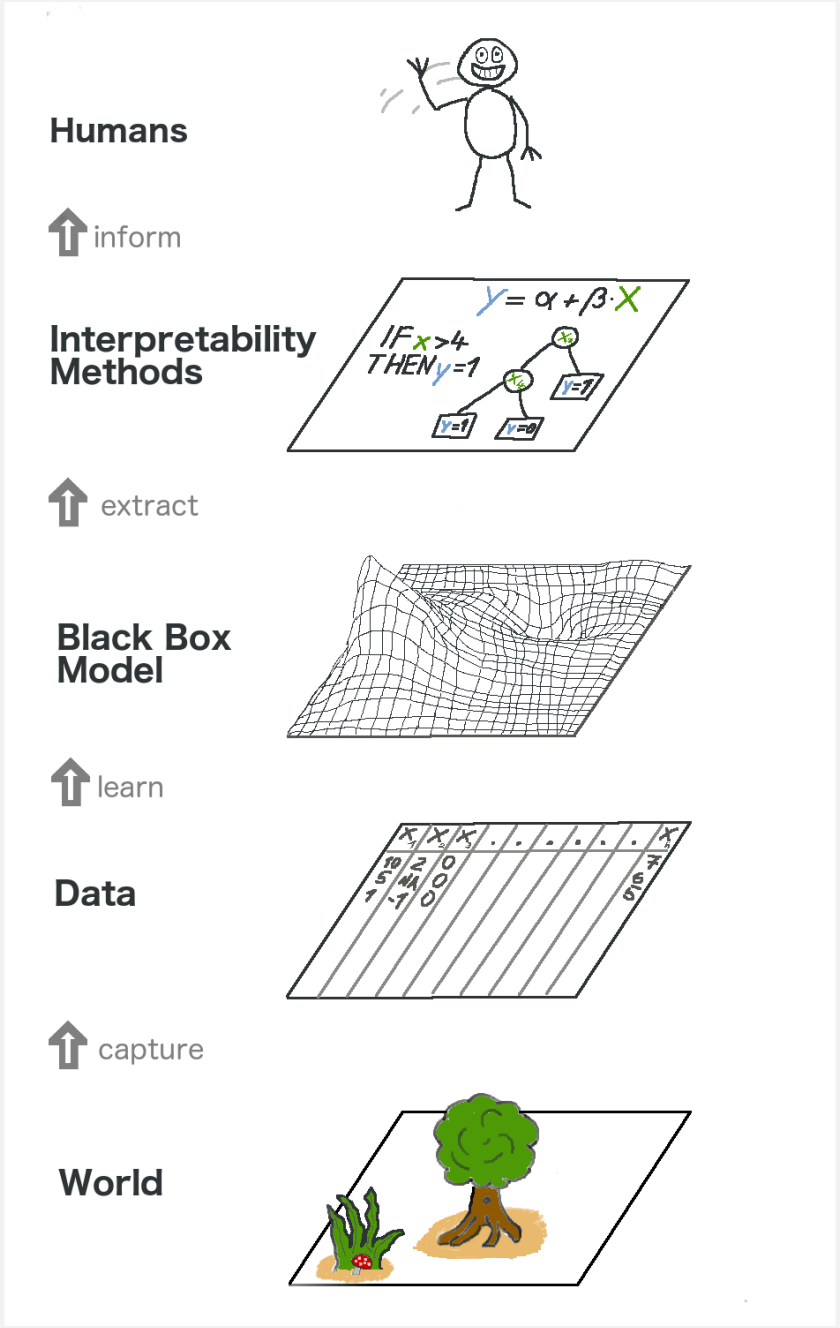
What is ML Model Explainability?

WHY IS MODEL EXPLAINABILITY IMPORTANT?

- In machine learning models, complexity often increases with accuracy
- High complexity can lead to low interpretability

- We want to allow our stakeholders to understand and use results in the proper way
  - Many applications require that we understand how individual model inputs impact the output

Black boxes can miss the most important part of the story!



Interpretable models are more:

- Informative
- Trustworthy
- Conducive to decision-making

https://towardsdatascience.com/interpretable-machine-learning-1dec0f2f3e6b

IS THIS JUST FEATURE IMPORTANCE?

- **Feature importance** is the *relative importance* of the features in a model
  - Typically uses the sum or *average improvement in model fit* when a feature is added

- **SHAP and LIME** values measure the *influence* of a feature
  - Compares *model predictions* with and without a feature

SHAP Intro

BACKGROUND

- SHAP (SHapley Additive exPlanations)
  - Shapley values – used in game theory to determine how much each player has contributed to success/failure in a cooperative game
    - Game: model, player: feature or collection of features
    - Average marginal contribution of a feature value over all possible coalitions
    - Fairly distribute both gains and losses to several players working in a coalition - in the case of machine learning, the model features are the players
- Considers the effects of features on individual datapoints, then aggregates the results
- Created by Scott Lundberg and Su-In Lee from the University of Washington[1]

EXAMPLE FROM GAME THEORY

Game: Paying for a pitcher of beer
Players: A, B, C
Payoff: Maximum quantity of beer the group can buy by pooling their money

Principles:

- What everyone pays should equal the total bill *(what everyone contributes should equal total reward)*
- If two people pay the same amount, they should receive the same amount of beer *(equal value contributed = equal reward)*
- If someone pays nothing, they should drink no beer ☹
- The group can have multiple rounds (rewards are additive)

EXAMPLE FROM GAME THEORY

A: $8

B: $7

C: $12

$4

30oz

$6

50oz

$9

70oz

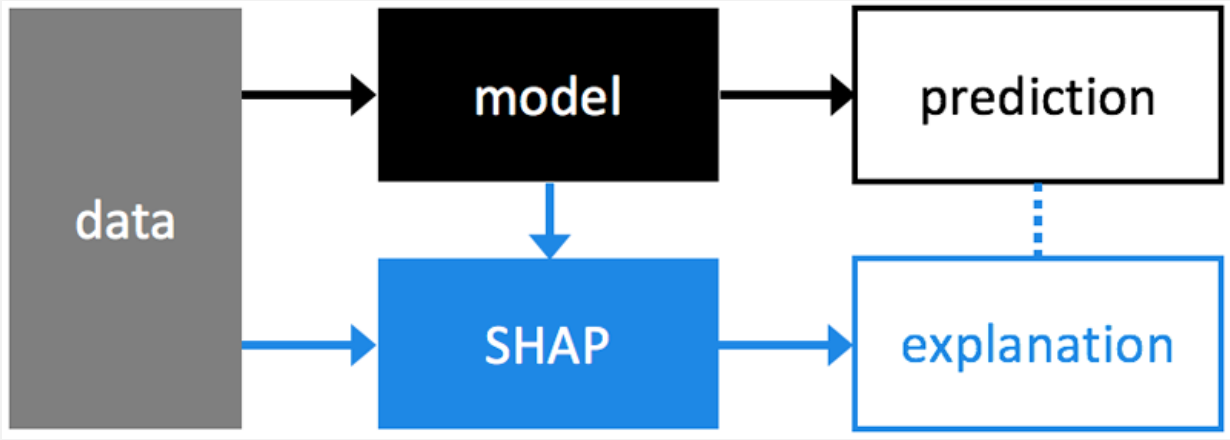
B = (30, 50, 70), P = ($4, $6, $9)

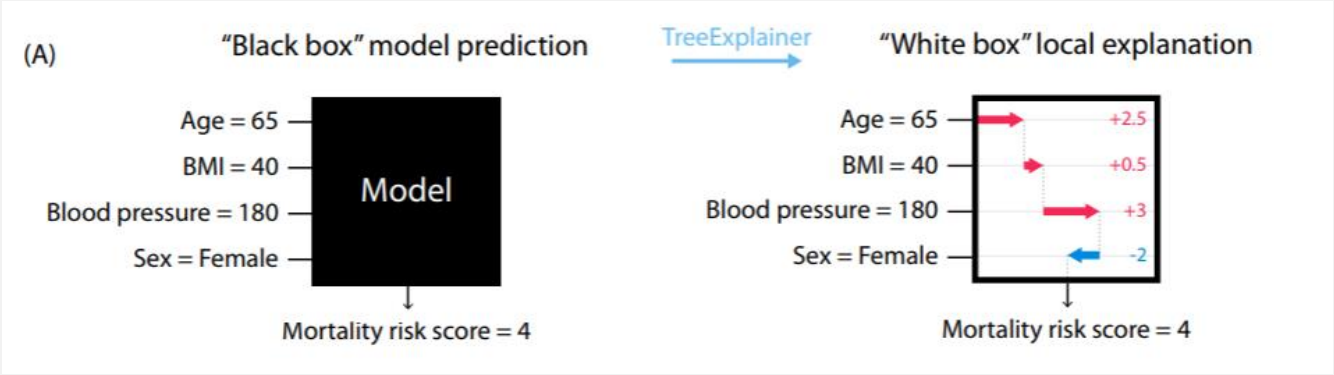
<u>A few possible outcomes*:</u>

- A and B contribute:
  - $15 → 120oz (A: 64oz, B: 56oz)
- Only B contributes:
  - $6 → 50oz (B: 50oz)
- All contribute:
  - $19 → 150oz (A: 63.2oz, B: 55.3oz, C: 31.6oz)

*There are three players, so $2^3=8$ possible outcomes in a single round

SHAP UNDER THE HOOD





https://shap.readthedocs.io/en/latest/#

SHAP ADVANTAGES/SHORTCOMINGS

Advantages

- Allows calculation of global contributions
- Shapley values consider all possible predictions for an instance using all possible combinations of inputs
  - Local accuracy and consistency are guaranteed (consistency in that sampling is not done to calculate contributions)

Shortcomings

- Sensitive to high correlations among features
  - Important to do feature selection prior to modeling
- Can be computationally expensive
- SHAP will always use all of the features in the data to explain the output (not for sparse explanations)

LIME Intro

LIME INTRO

- LIME (Local Interpretable Model-Agnostic Explanations)
  - Uses local surrogate models to approximate individual predictions
    - Intuition is that it is easier to approximate complex models locally (near the prediction that we want to explain)
    - Surrogate models generate explanations by approximating the underlying model with an interpretable one (for example, as a linear model)
- Considers the effects of features on individual data points
- Created by Marco Ribeiro, Sameer Singh, and Carlos Guestrin from the University of Washington2

LIME ADVANTAGES/SHORTCOMINGS

Advantages

- Faster and less computationally expensive than SHAP

Shortcomings

- Only allows for local approximations of contributions
  - Single observation explanations only
- Subset of SHAP without accuracy and consistency guarantee

SHAP/LIME Comparison

SHAP OR LIME

- SHAP
  - Produces globally consistent explanations
    - Estimates the effect of all features on all data points
      - To calculate exactly would be ***very*** computationally expensive
    - The values add up to the true prediction of the model
  - High accuracy and reliability

- LIME
  - Produces local explanations (given a specific sample)
    - Gives contributions at a single data point of interest
  - Faster, but less accurate
  - Doesn't work on all model types
    - Namely, *XGBoost*

LIME can be thought of as a subset of SHAP

MY RECOMMENDATION

Use SHAP unless the computational costs are too high, in which case, use LIME

SHAP Examples

DATA

- Data on secondary student achievement in mathematics at two Portuguese schools. The data attributes include student grades, demographic, social and school related features. Data was by using school reports and questionnaires.

- Classification example: Predict whether a student will pass their final math course

- Data source: https://archive.ics.uci.edu/ml/datasets/Student+Performance

*Example code will use SHAP TreeExplainer, which is ideal for trees and ensembles of trees

DATA

$$Predicted\ Value = Expected\ Value + \sum SHAP\ Values$$

- Expected Value: *Value predicted in the absence of any features*
  - For example, if we only had the instance of passing and failing we'd take the mean as our expected value (0.41 in our case)

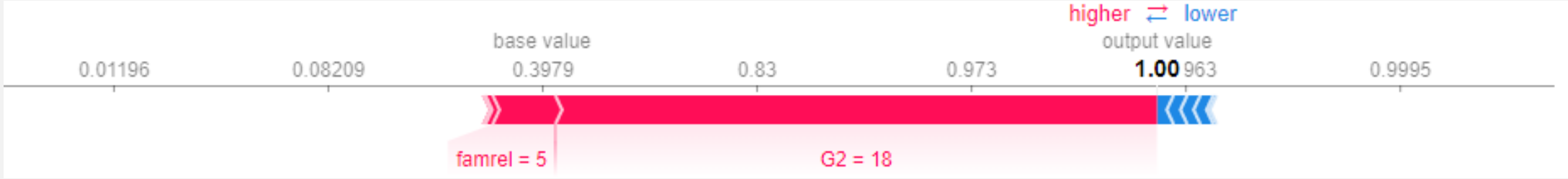
- SHAP Value: *Average marginal contribution of a feature value across all possible coalitions*

FORCE PLOT: EXPLAIN SINGLE PREDICTION

Force plots allow us to visualize individual model predictions. The "base value" is the model's average prediction over the training data set. The "output value" is the predicted value for this observation - whether or not that a student will pass their final math course.

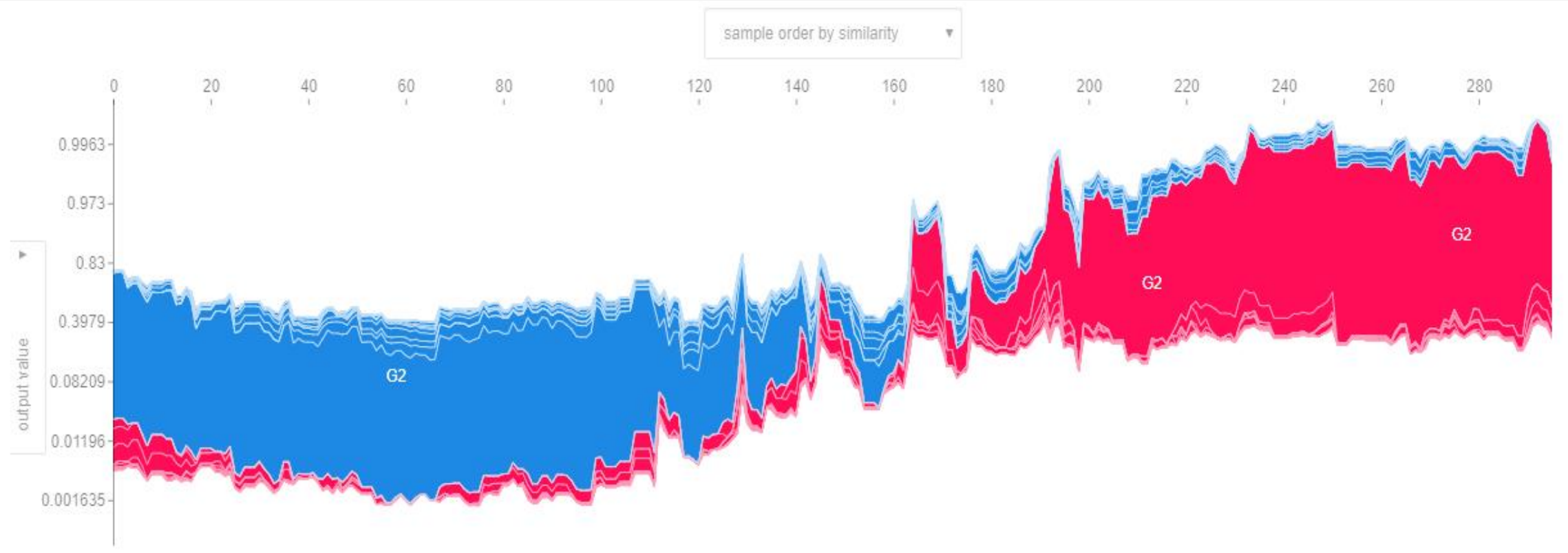
The prediction is whether a student will pass their final math course.

- Red arrows represent feature effects (SHAP values) that drive the prediction higher
- Blue arrows are effects that drive the prediction value lower.
- Each arrow's size represents the magnitude of the corresponding feature's effect



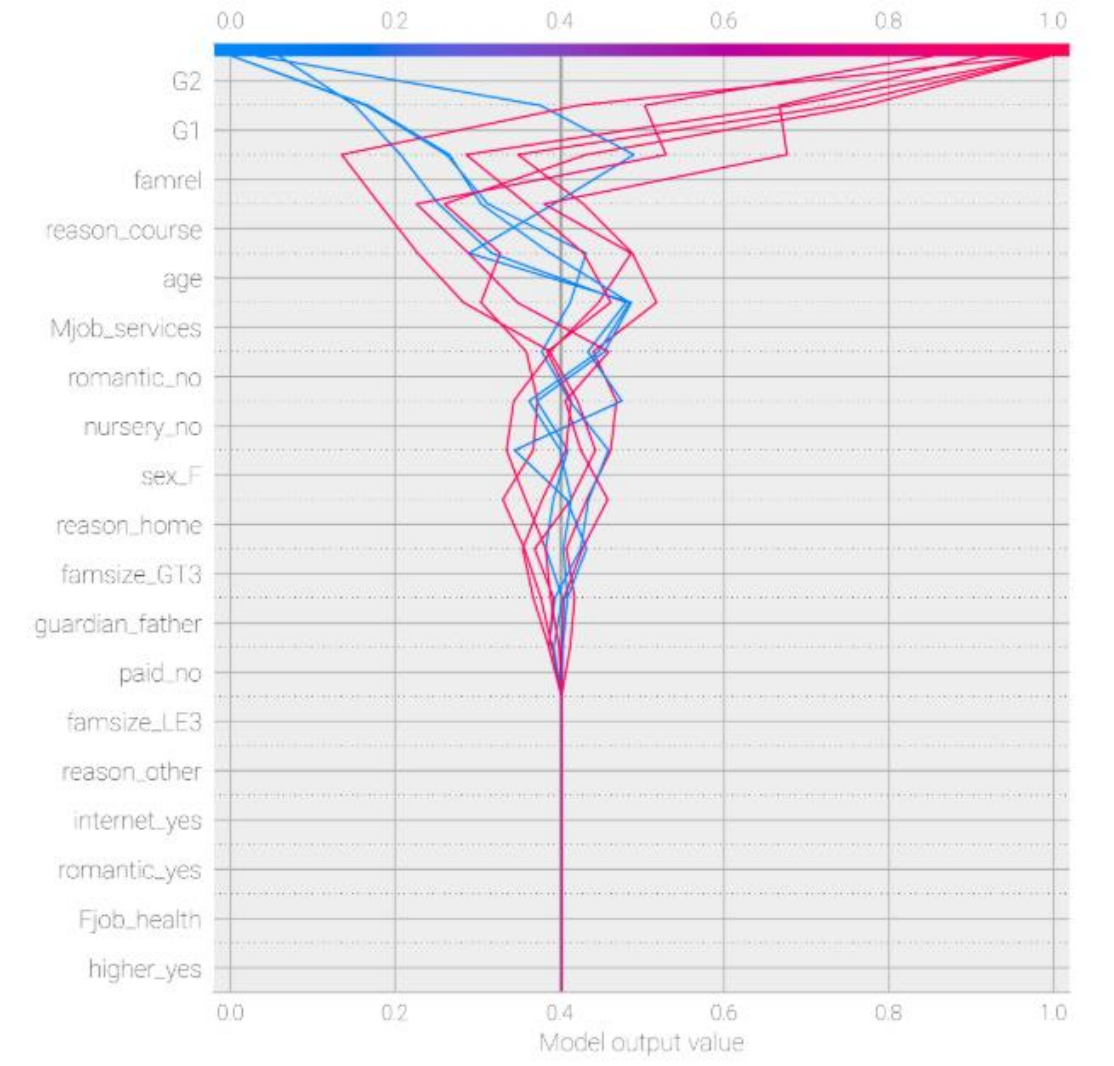
FORCE PLOT: EXPLAIN ALL PREDICTIONS

Force plots can also allow us to visualize all model predictions at once. Think of this as individual force plots rotated vertically and stacked horizontally.



DECISION PLOT: EXPLAIN PATH TO PREDICTION

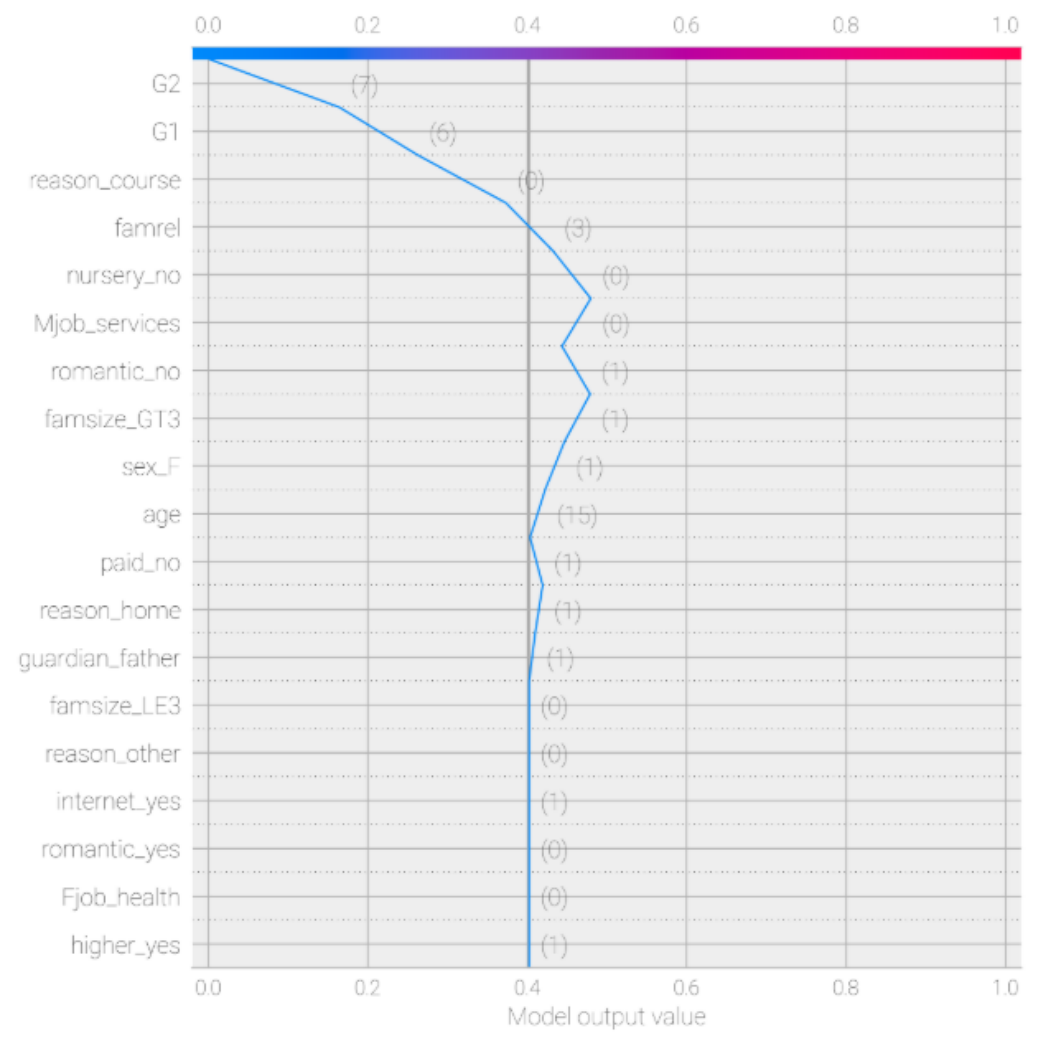
Decision plots show how complex models arrive at their predictions (i.e., how models make decisions). Below we see the decision path for the first 10 predictions.



- The x-axis represents the model's output. In this case, the units are probabilities.
- The y-axis lists the model's features. By default, the features are ordered by descending importance. The importance is calculated over the observations plotted. (This will typically differ from the importance ordering for the entire dataset.
- Each observation's prediction is represented by a colored line. At the top of the plot, each line strikes the x-axis at its corresponding observation's predicted value. This value determines the color of the line on a spectrum.
- Moving from the bottom of the plot to the top, SHAP values for each feature are added to the model's base value. This shows how each feature contributes to the overall prediction.
- At the bottom of the plot, the observations converge at expected value.

DECISION PLOT: EXPLAIN PATH TO PREDICTION

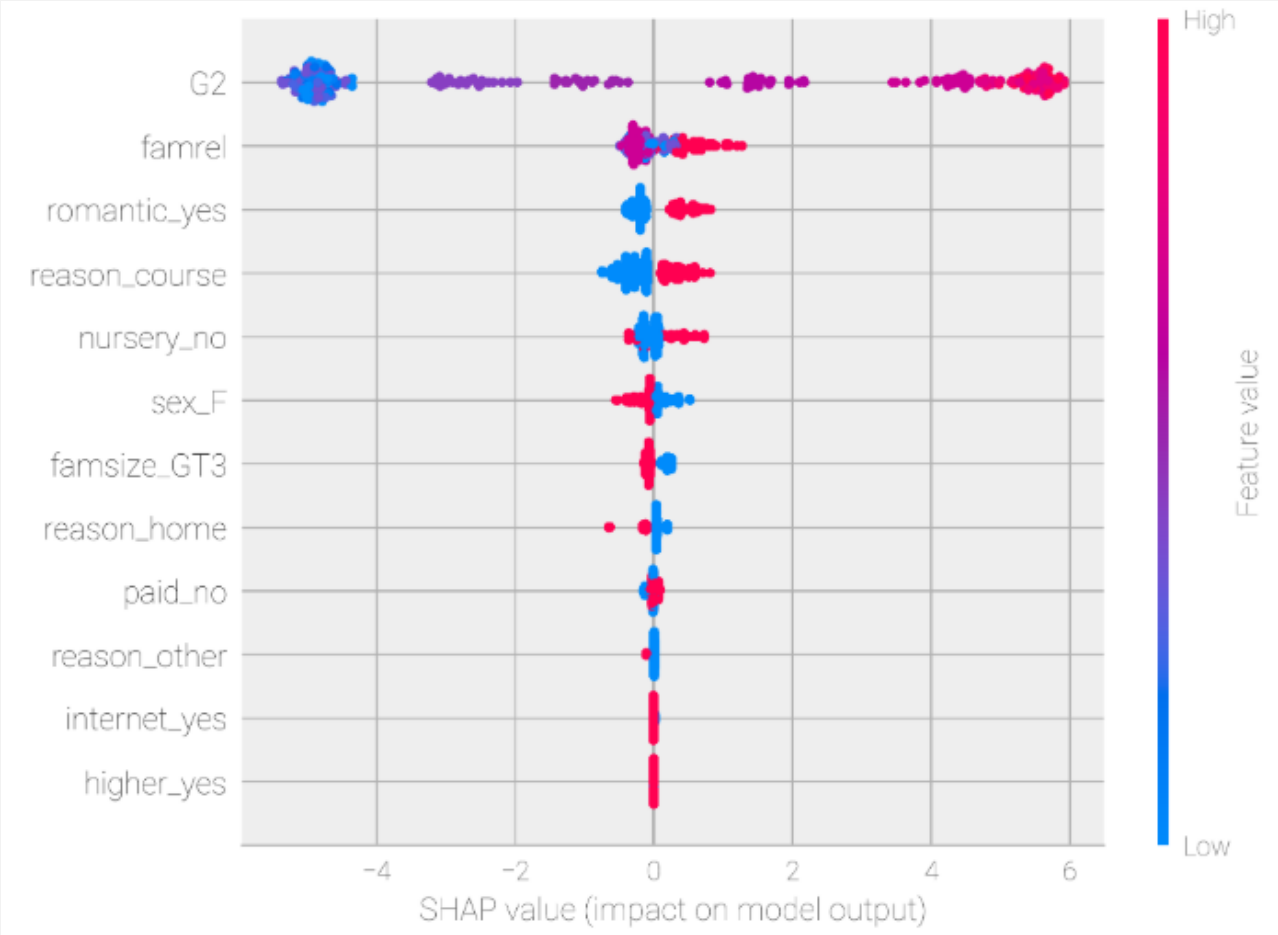
Decision plots show how complex models arrive at their predictions (i.e., how models make decisions). Below we see the decision path for a single prediction.



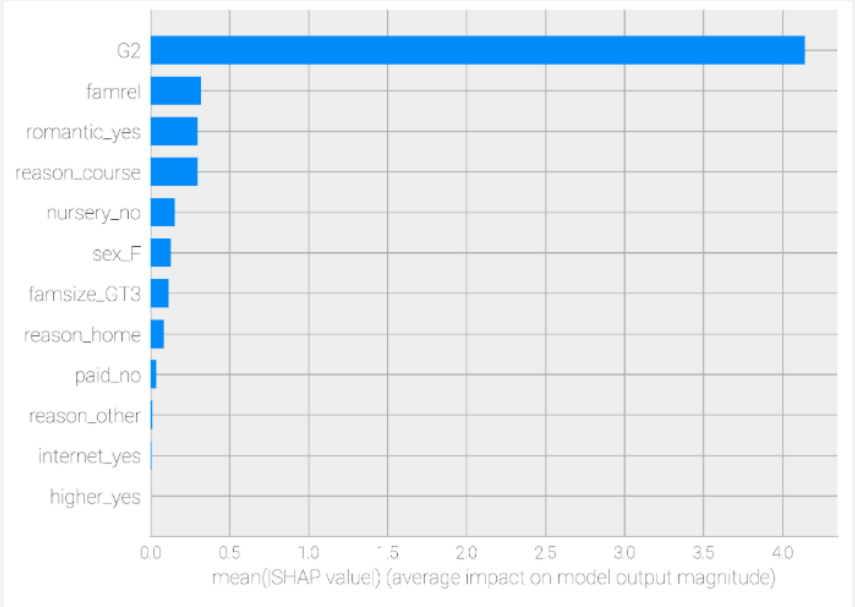
- For this observation, it is predicted that the student will fail their final math course (notice that the model output value is 0)
- You can follow the vertical path from the bottom at 0.4 (the expected value) to see how each feature positively or negatively impacts the prediction
- G2 (grade in second math course) is the most impactful feature, and in this impacts the prediction negatively in this case.

SUMMARY PLOT

A summary plot is an extension of a traditional feature importance plot. It details both feature importance AND feature effects. Each point on the plot is a Shapley value for a feature and an instance. In this case, G2 (grade in second math course) is the most important feature.

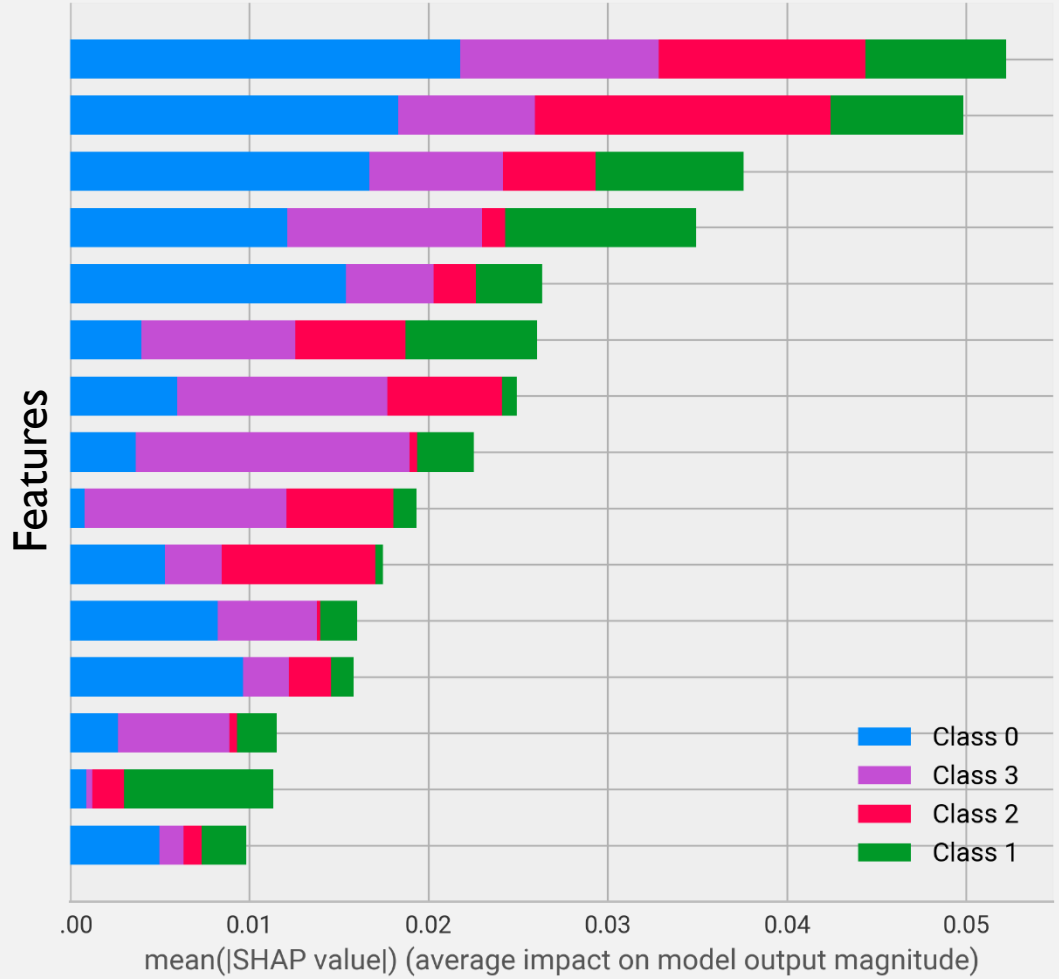


- The position on the y-axis is determined by the feature and on the x-axis by the Shapley value. The color represents the value of the feature from low to high. Overlapping points are jittered in y-axis direction, so we get a sense of the distribution of the Shapley values per feature.



SUMMARY PLOT (MULTICLASS CLASSIFICATION)

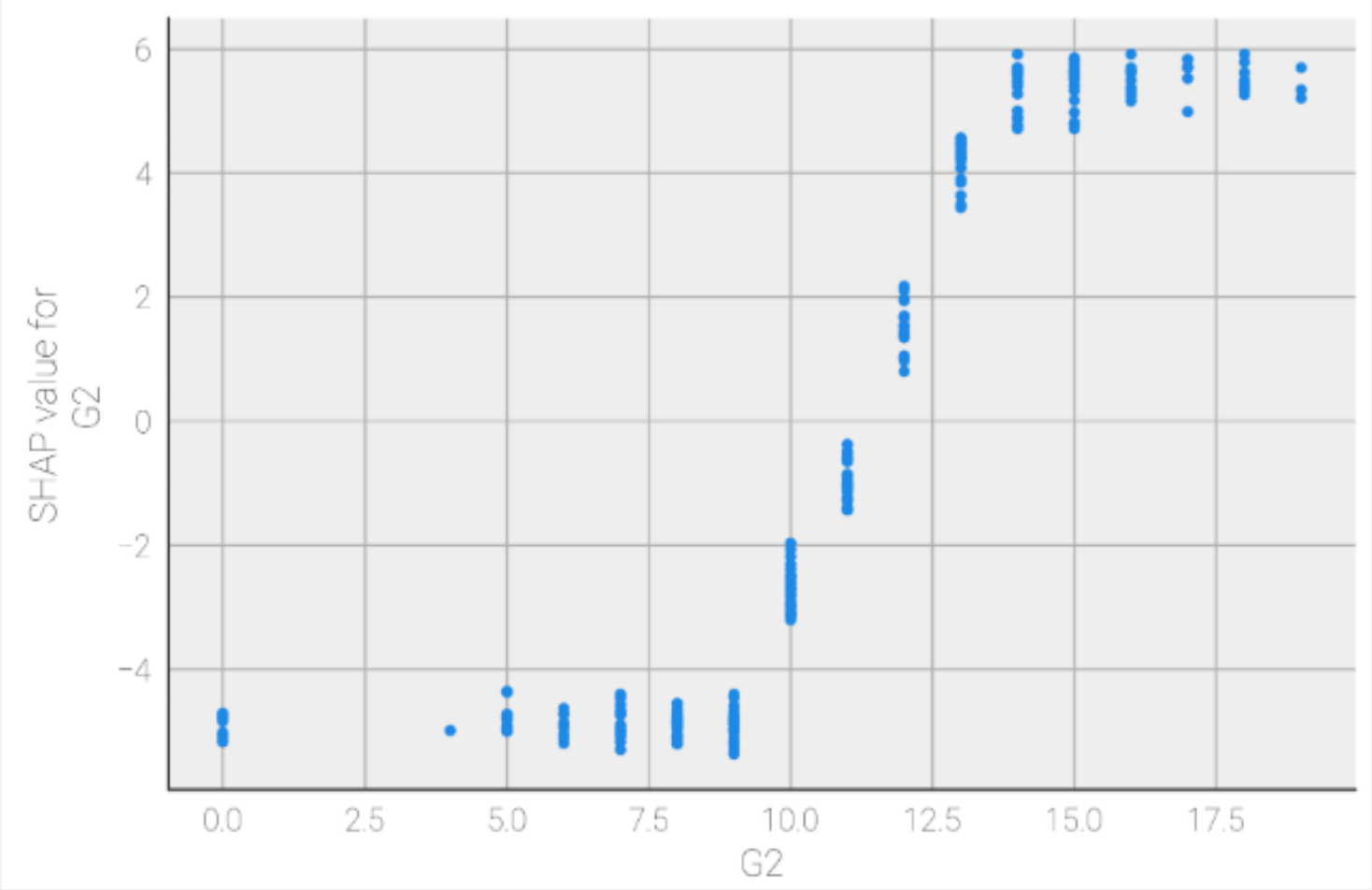
A summary plot for multiclass classification (used post-clustering in this instance) allows us to understand which features are most prevalent in the clusters



*This is just a visual example and is not using the education data

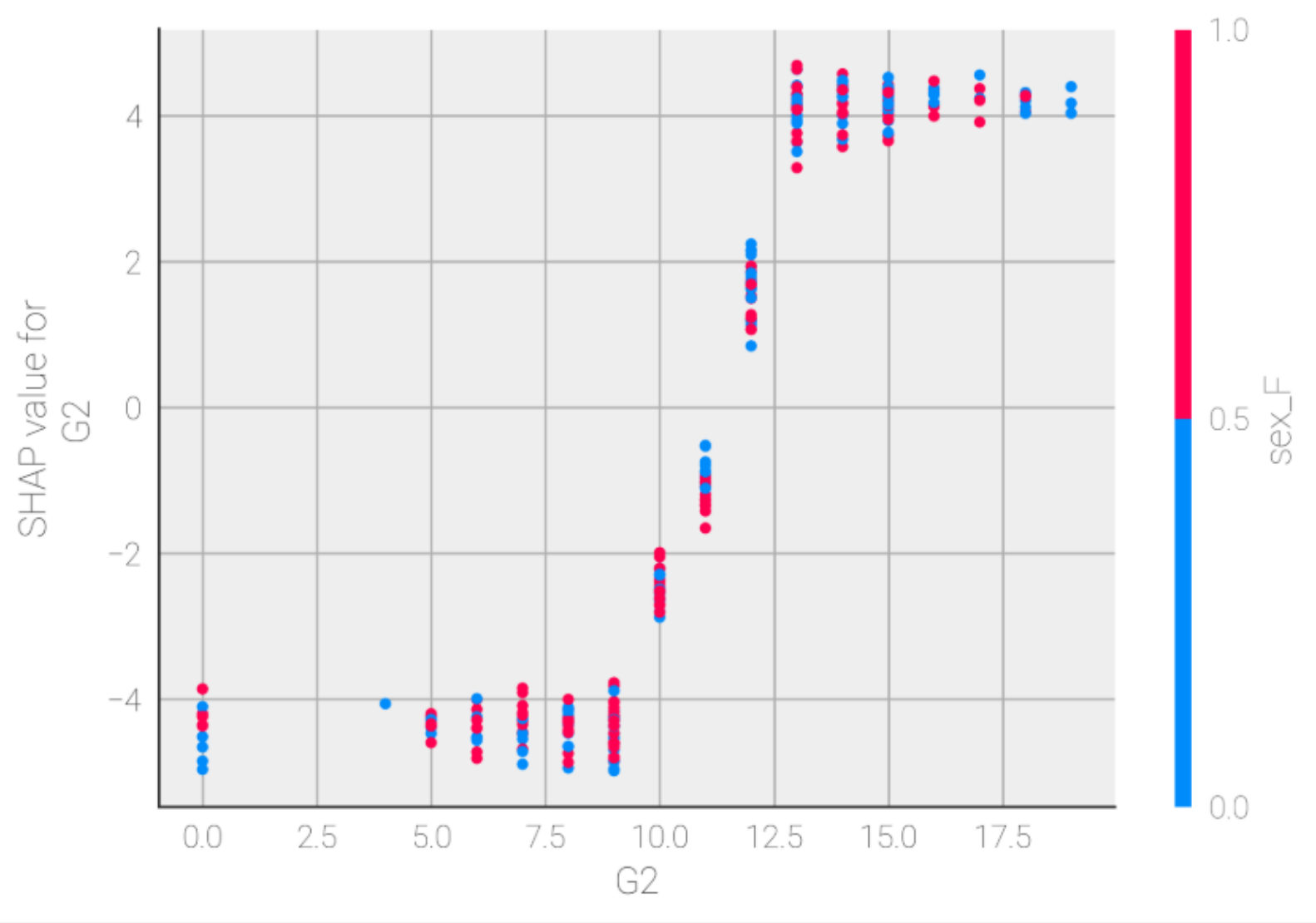
DEPENDENCE PLOT

A dependence plot is a scatter plot that shows the effect a single feature has on the predictions made by the model. In this example the log-odds of a passing grade increases noticeably for G2 = 10.



- Each dot is a single prediction (row) from the dataset.
- The x-axis is the value of the feature (from the X matrix).
- The y-axis is the SHAP value for that feature, which represents how much knowing that feature's value changes the output of the model for that sample's prediction. For this model the units are log-odds of passing.

DEPENDENCE PLOT



- The color corresponds to a second feature that may have an interaction effect with the feature we are plotting. If an interaction effect is present between this other feature and the feature we are plotting it will show up as a distinct vertical pattern of coloring.

Appendix

EXAMPLE FROM GAME THEORY: POTENTIAL OUTCOMES

B = (30, 50, 70), P = ($4, $6, $9)

Coalition	Potential Contribution	Total Contribution	Individual Contribution	Total Reward	Individual Rewards
None	$0	$0	$0	0oz	0oz
A Only	$8	$6	$6	50oz	50oz
B Only	$7	$6	$6	50oz	50oz
C Only	$12	$10	$10	80oz	80oz
A & B	$15	$15	A:$8, B:$7	120oz	A:64oz, B:56oz
A & C	$20	$19	A:$8, C:$11	150oz	A:63.2oz, C:86.8oz
B & C	$19	$19	B:$7, C:$12	150oz	B:55.3oz, C:94.7oz
All	$27	$19	A:$8, B:$7, C:$4	150oz	A:63.2oz, B:55.3oz, C:31.6oz

OTHER TYPES OF EXPLAINERS

- Kernel Explainer
  - Output of any function
- Linear Explainer
  - Linear models with independent features
- Gradient Explainer
  - Deep learning
- Deep Explainer
  - Deep learning (fast, but approximate)

HELPFUL RESOURCES

- Interpretable Machine Learning
- SHAP Paper
- SHAP Documentation
- SHAP for Python (Github)
- SHAP for R (Github)
- LIME Paper
- LIME Documentation
- LIME for Python (Github)
- LIME for R (Github)

Less common interpretability packages:

- ELI5
- Skater

Thank you

Q & A