

Beyond the Black Box

Interpreting ML models with SHAP

Avik Basu
(He/Him)

About Me

- Based in Sunnyvale, CA
- Staff Data Scientist at Intuit
- Editor at PyOpenSci
- Engineering + Data Science
- Love PS5 + Steam Deck 🎮
- Football + Tennis 🎾
- Driving is therapy! 🚗



Agenda

1. Why does explainability matter?
2. SHAP theory and intuition
3. Case Study 1: Decision Trees
4. Case Study 2: Deep Neural Networks
5. Limitations

Explainable AI

Why does it matter?

- ML models are not just used to classify cats vs dogs
- Used in very crucial industries / use-cases
 - *Resume screening for a job application*
 - *Medical image diagnosis*
 - *Credit risk for analysis for loan application*
- Ethical concerns

Shapley Values

Concept

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! (|N| - |S| - 1)!}{|N|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]$$

L. S. Shapley and others, "A value for n-person games," 1953

Shapley Values

Concept

Example

- 3 friends start a business and bring in \$20k profit
- Alice alone: \$5k profit
- Bob alone: \$3k profit
- Charlie alone: \$4k profit

How can they fairly split the \$20k profit?

Shapley Values

Concept

- Originates from game theory
- Fairly distribute rewards among “players” in a cooperative “game”
- In machine learning terms
 - Player == Feature
 - Game == ML Task
 - ML task can be classification, regression, etc.

SHAP

SHapley Additive exPlanations

- Specific implementation of Shapley values
- Explain ML predictions
- Model Agnostic
 - Decision Trees
 - Neural Networks
 - Any kind of model really
- Linear model of feature contributions

S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," CoRR, 2017

SHAP

Mathematical intuition

$$f(x) = E[f(X)] + \sum_{i=1}^M \phi_i(x)$$

- $f(x)$ is the prediction of a single instance
- $E[f(X)]$ is mean prediction of the dataset or background data
- $\phi_i(x)$ is the SHAP value for feature i for input x

How do we calculate SHAP?

The actual calculation is
 $O(2^M)$

For 3 features, we need to test all 8 combinations!

SHAP calculation is exponential

Example

Question

How much does each feature contribute to the loan application?

- Age
- Income
- Credit Score

Approach

Try all combination of features.

1. {} - no features
2. {Age}
3. {Income}
4. {Credit}
5. {Age, Income}
6. {Age, Credit}
7. {Income, Credit}
8. {Age, Income, Credit}

SHAP calculates how much each feature helps on average across all combinations!

SHAP

Approximation methods

Kernel SHAP

- Applicable to any model type
- Slow for large datasets

Tree SHAP

- Optimized for tree-based models
- Exploits the hierarchical structure

Deep SHAP

- Estimation for deep neural networks
- Leverage backpropagation

Now let's see how all this plays
out!

Case Study I: Decision Trees

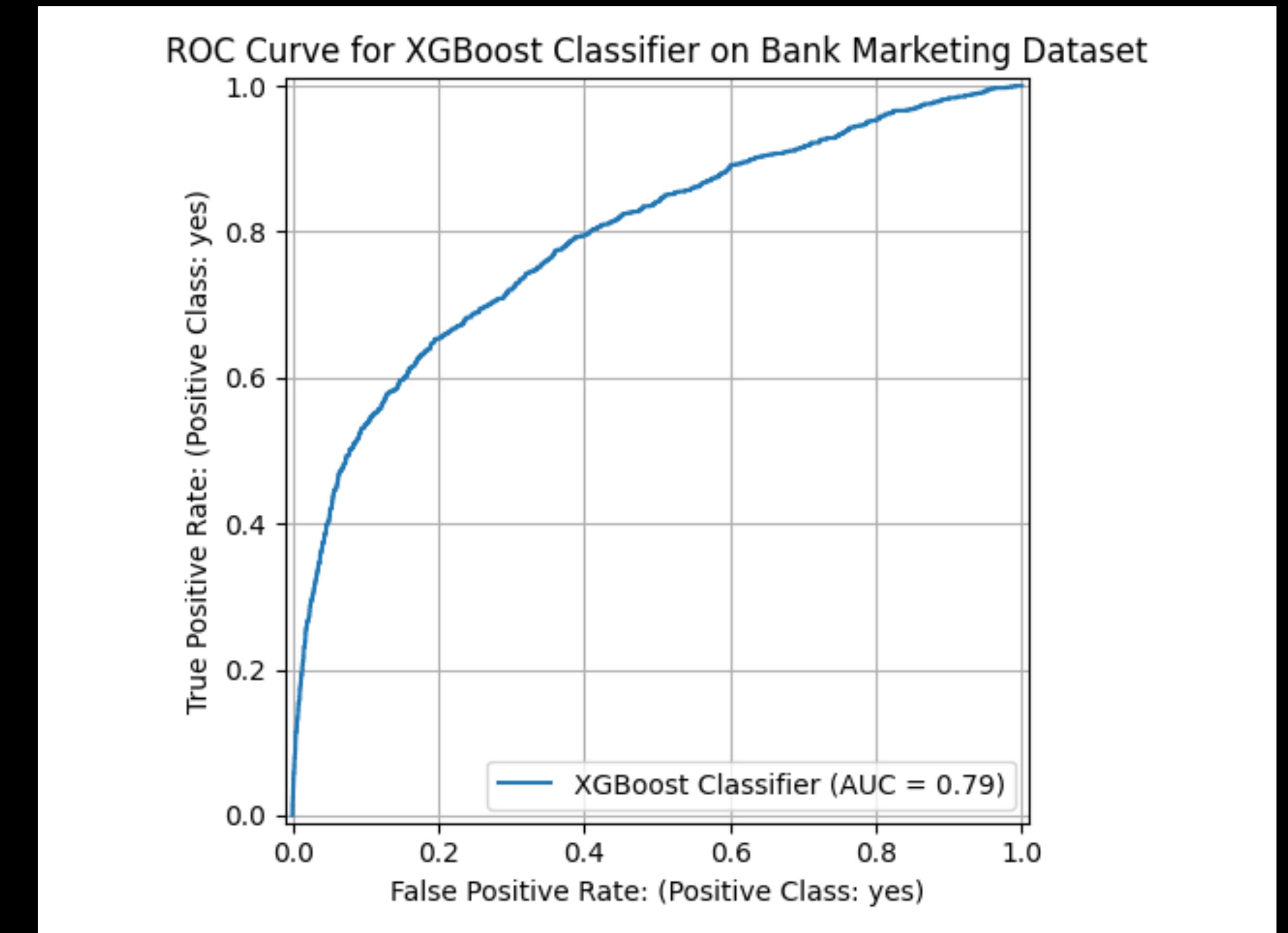
Business Problem

A Portuguese bank wants to predict which customers will subscribe to a term deposit based on direct marketing campaigns, i.e. phone calls

SHAP for Decision Trees

Bank Marketing Dataset

- Tabular data
- Features include
 - Customer demographics
 - Financial details
 - Previous interaction history
- Classes: 2 (yes, no)
- Number of total samples: 45,211
- Model used: XGBoost Classifier



SHAP for Decision Trees

Calculation

```
import shap
```

```
# xgb is the trained XGBoost classifier  
# x_test is the test dataset of shape (9043, 15)
```

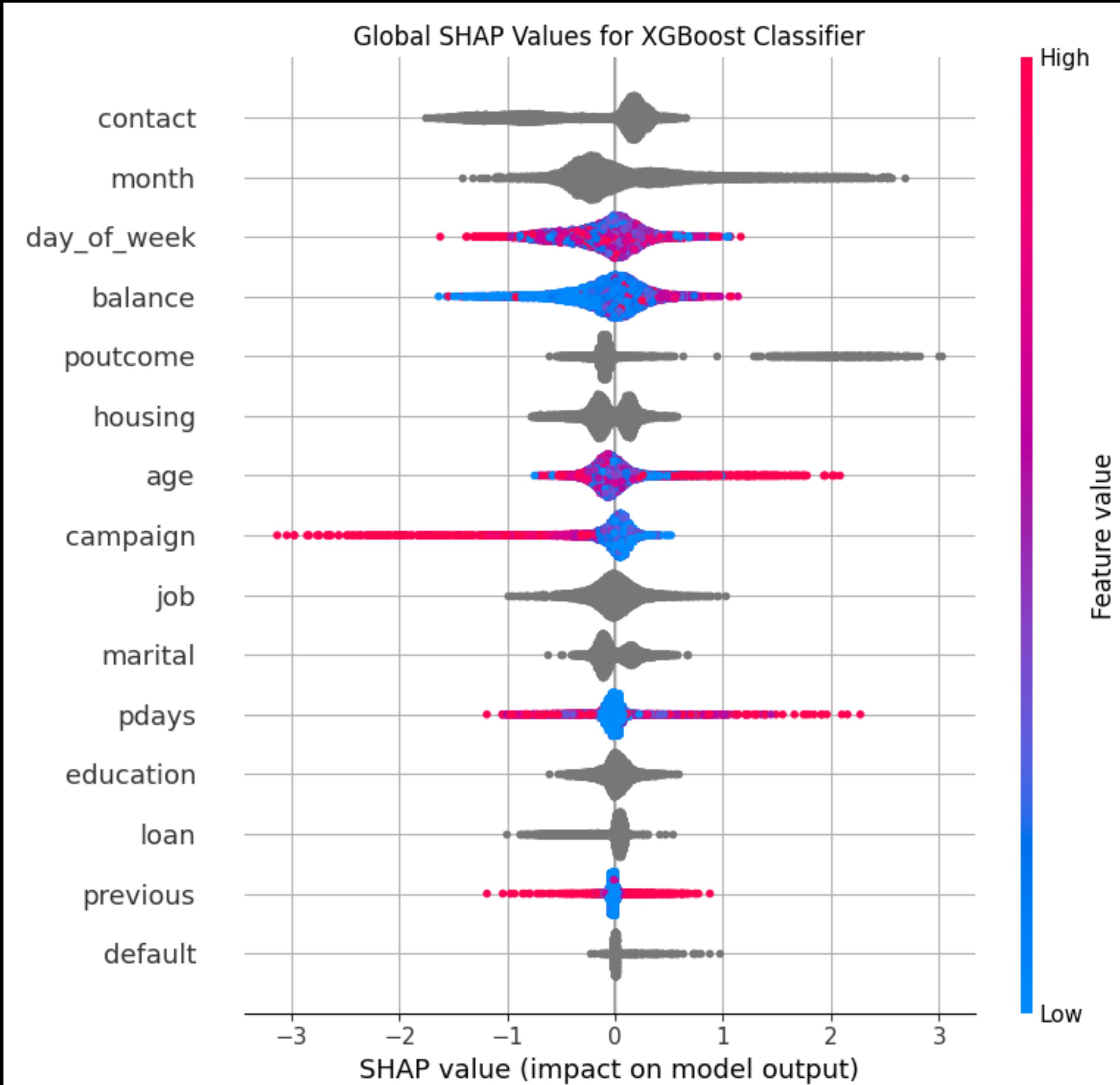
```
# Setup TreeSHAP explainer  
explainer = shap.TreeExplainer(xgb)  
shap_values = explainer.shap_values(x_test)
```

```
# shap_values shape: (9043, 15)
```

SHAP for Decision Trees

Global explanations with beeswarm plot

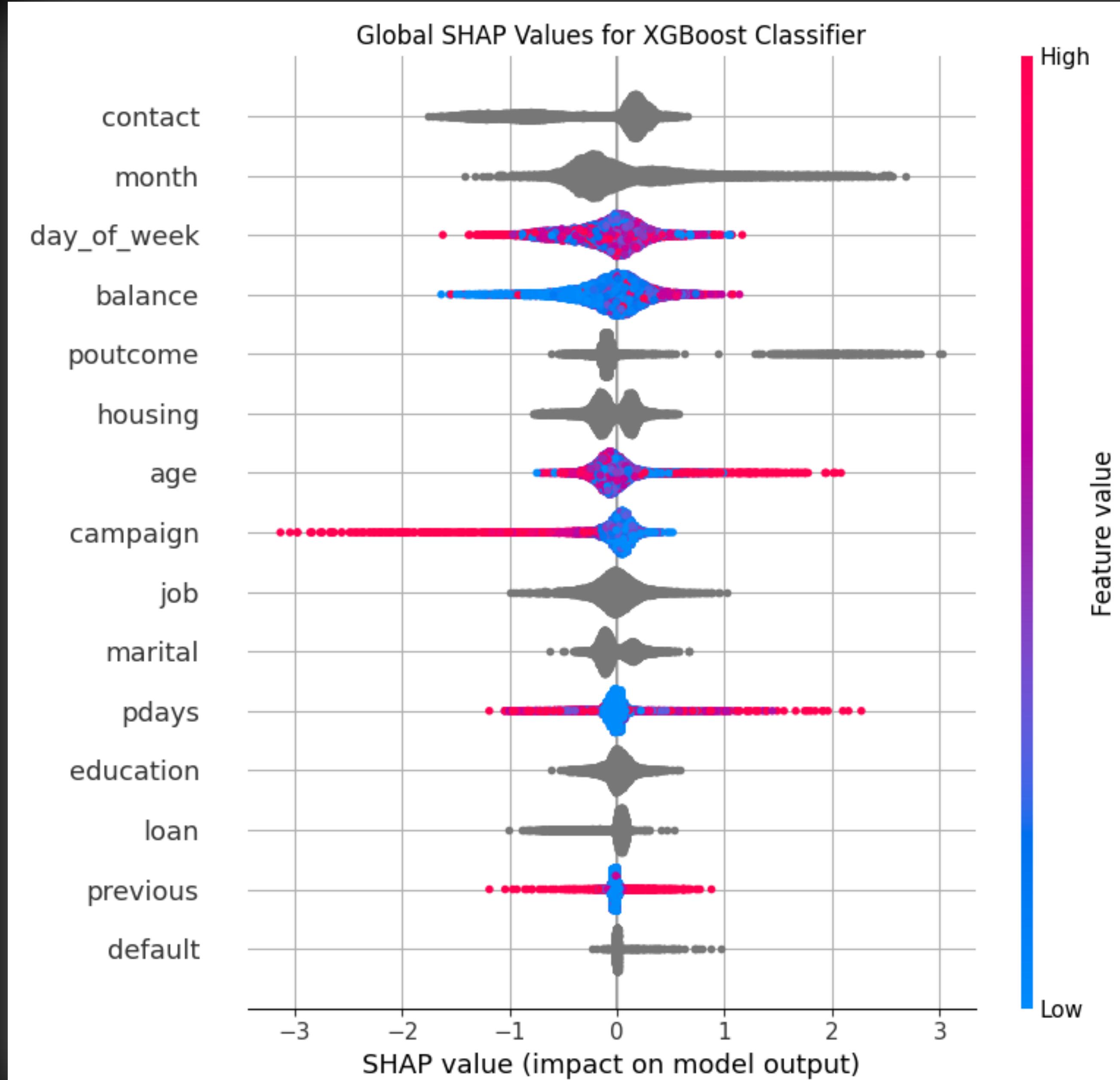
- High-level summary of the model's behavior
- Useful for understanding the overall impact of each feature
- Features are ordered by importance



SHAP for Decision Trees

Global explanations with beeswarm plot

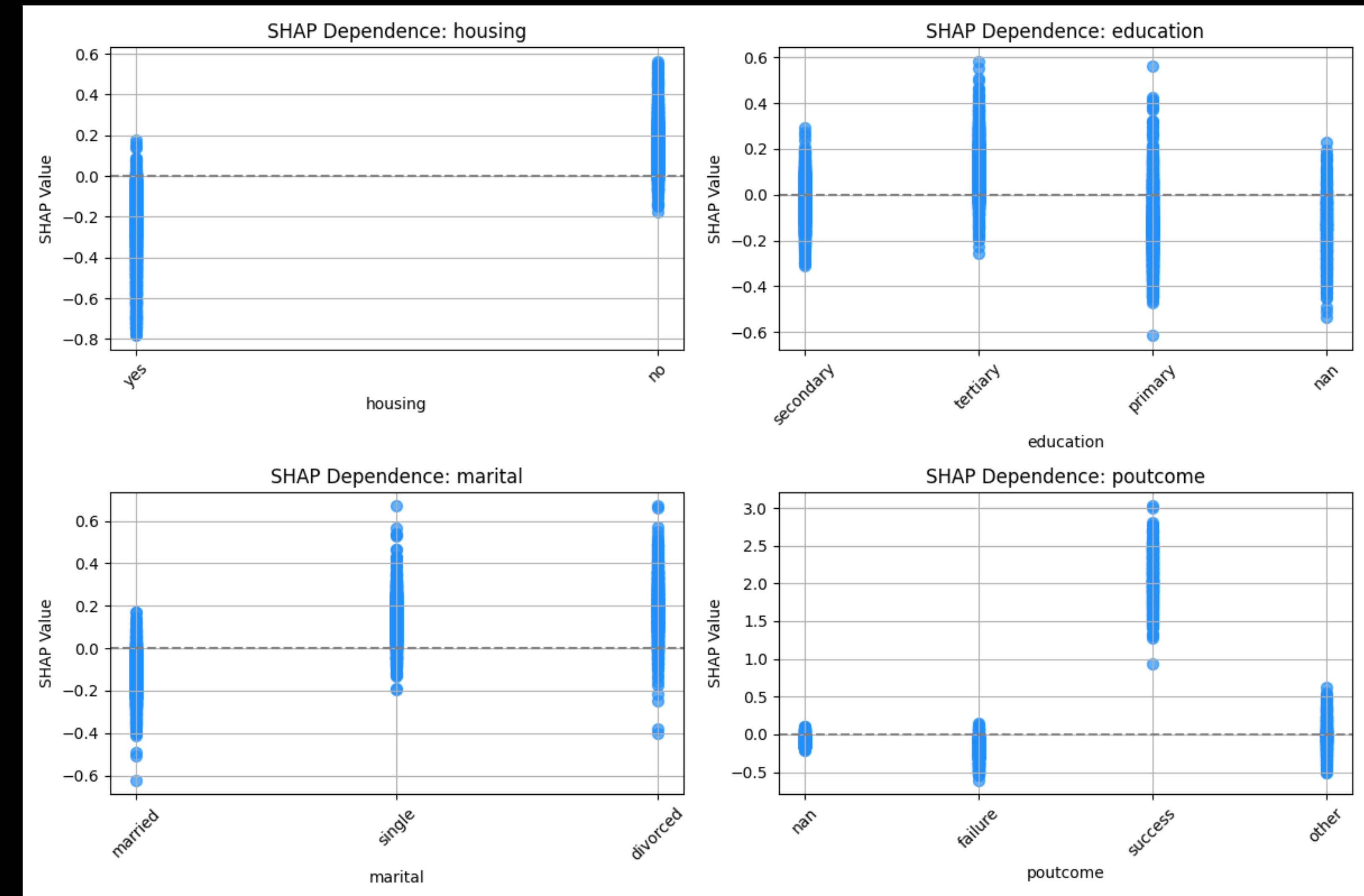
- SHAP values represent the impact each feature has on the model's predictions
- Positive SHAP values indicate that the feature pushes the model prediction higher, i.e. towards the "yes" class
- The distance from the center (zero) indicates magnitude
- Categorical variables are displayed in gray



SHAP for Decision Trees

Dependency plots

- Deeper analysis for a single feature and its effect on model predictions
- Especially useful for categorical features



Case Study II: Neural Networks

SHAP for Convolutional Neural Networks

Human Activity Recognition dataset

- Time series sensor data collected from a smartphone
- A smartphone is attached to the waist of the individual
- Data collected from 30 individuals within the 19-48 years age group

Six Activities

1. *Walking* 
2. *Walking Upstairs* 
3. *Walking Downstairs* 
4. *Sitting* 
5. *Standing* 
6. *Laying* 

SHAP for Convolutional Neural Networks

Problem formulation

Dataset

- Multivariate time series
- 560+ features (all numeric)
 - Raw accelerometer readings (X, Y, Z axes)
 - Raw gyroscope readings (X, Y, Z axes)
 - Engineered features in time and frequency domain (mean, std, etc.)
- 6 classes
- Objective: Multiclass classification

Model

- 1D CNN using PyTorch
- Normalize the data [0, 1]
- Convert into window sequences

```
class CNNClassifier(nn.Module):
    def __init__(self, input_dim: int, num_classes: int = 6):
        super(CNNClassifier, self).__init__()
        self.layer1 = nn.Sequential(
            nn.Conv1d(input_dim, 32, kernel_size=3, stride=2, padding=1),
            nn.Tanh(),
        )
        self.layer2 = nn.Sequential(
            nn.Conv1d(32, 16, kernel_size=3, stride=2, padding=1),
            nn.Tanh(),
        )
        self.flatten = nn.Flatten()
        self.fc = nn.Linear(256, num_classes)

    def forward(self, x: torch.Tensor) -> torch.Tensor:
        x = self.layer1(x)
        x = self.layer2(x)
        x = x.view(x.size(0), -1)
        x = self.fc(x)
        return x
```

SHAP for CNN

Model Performance

Class	F1-score	Support
Walking	0.95	496
Walking Upstairs	0.93	471
Walking Downstairs	0.90	420
Sitting	0.85	467
Standing	0.87	501
Laying	0.94	528

SHAP for CNN

SHAP calculation

```
import shap
import numpy as np

# Create balanced background dataset (20 samples per class)
background_data = create_balanced_background(
    x_train_seq, y_train_seq, n_per_class=20
)

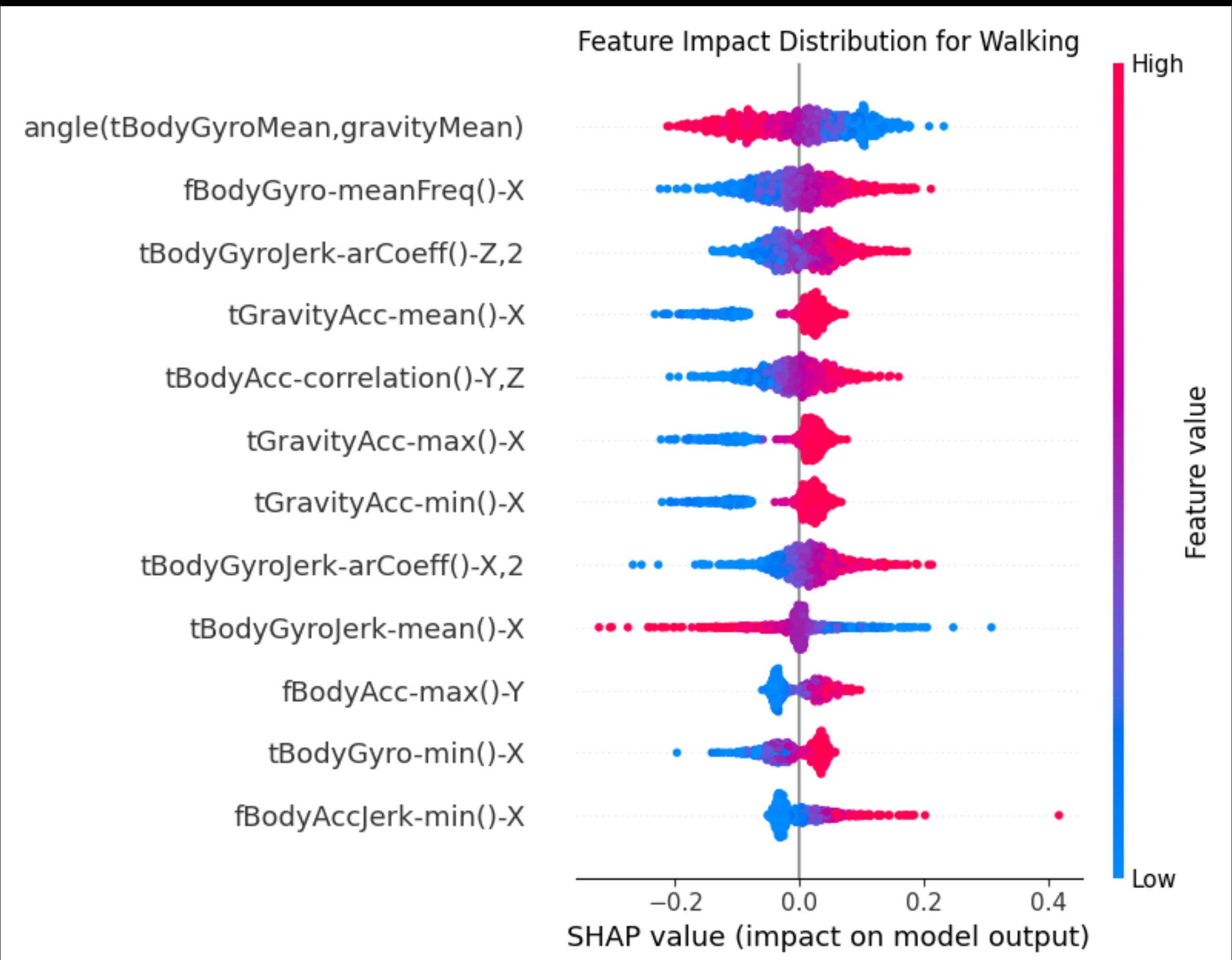
# Setup DeepSHAP explainer
explainer = shap.DeepExplainer(model, background_data)
shap_values: np.array = explainer.shap_values(x_test_seq[:1000])
# shap_values shape: (1000, 561, 64, 6)

# Extract feature importance and visualize
feature_importances = shap_values[:, :, -1, :] # Last time step
shap.plots.beeswarm(feature_importances[:, :, 0]) # Walking class
```

SHAP for CNN

Global explanations

- In a multiclass setting, we have a set of shap values for every class in a **one-vs-all** fashion
- **Positive** SHAP values pushes the feature towards the positive class (Walking)
- **Negative** SHAP values reduces the model's prediction towards all the negative class (all other classes)



SHAP for CNN

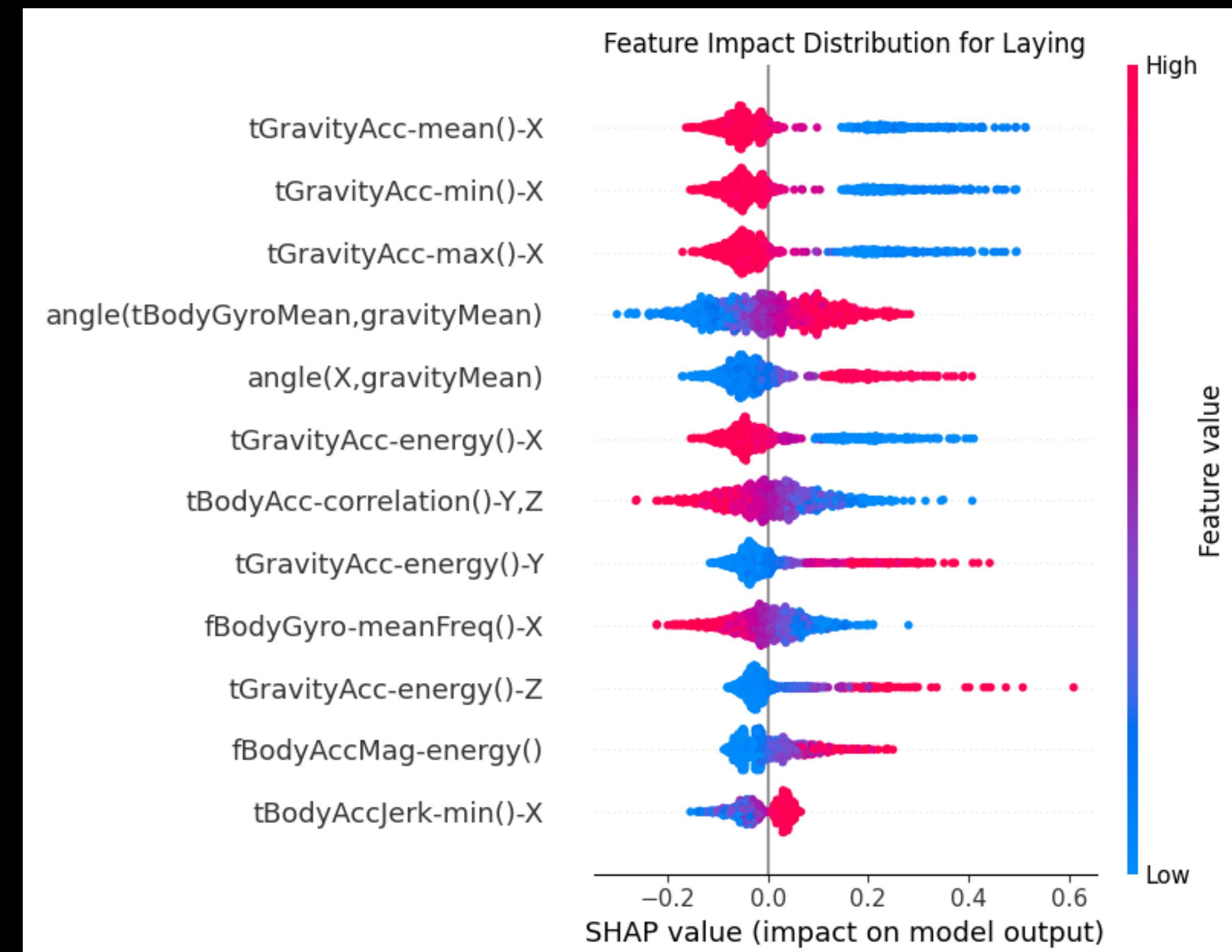
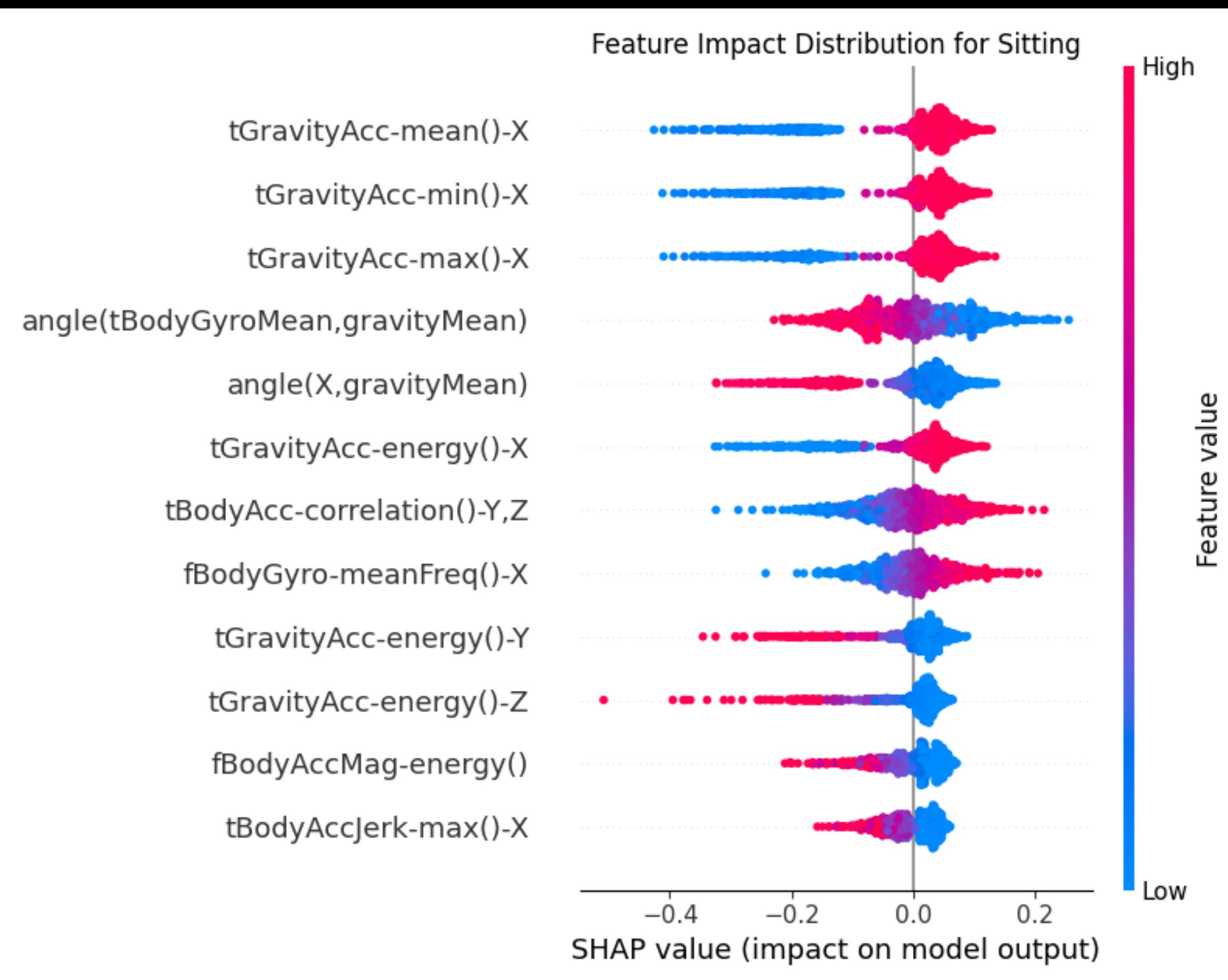
Global explanations: Walking

- When *Walking*, the angle between the mean gyroscope signal and the mean gravity signal is smaller
- Low values of gravitational acceleration along the X axis pushes the prediction away from the *Walking* class



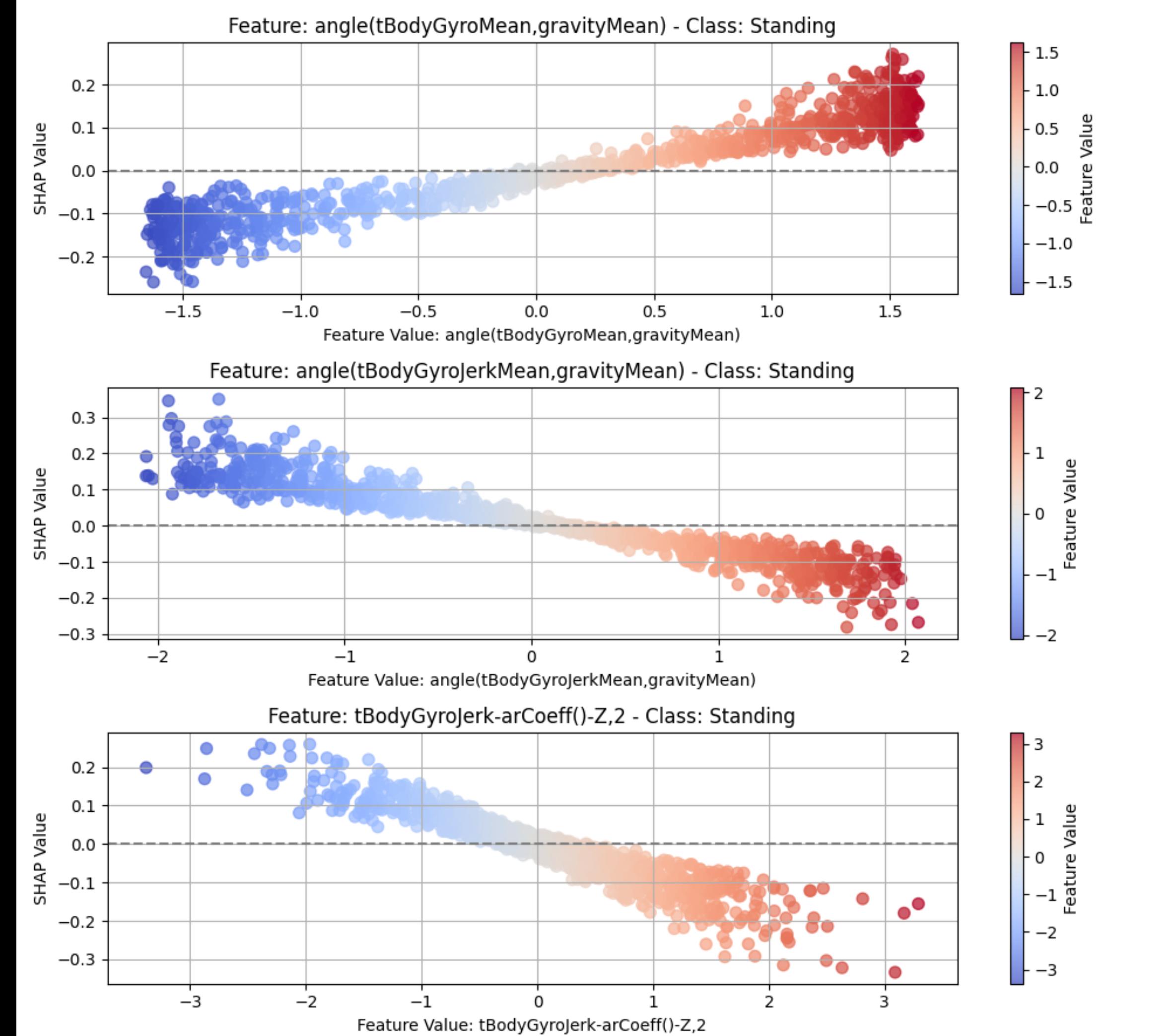
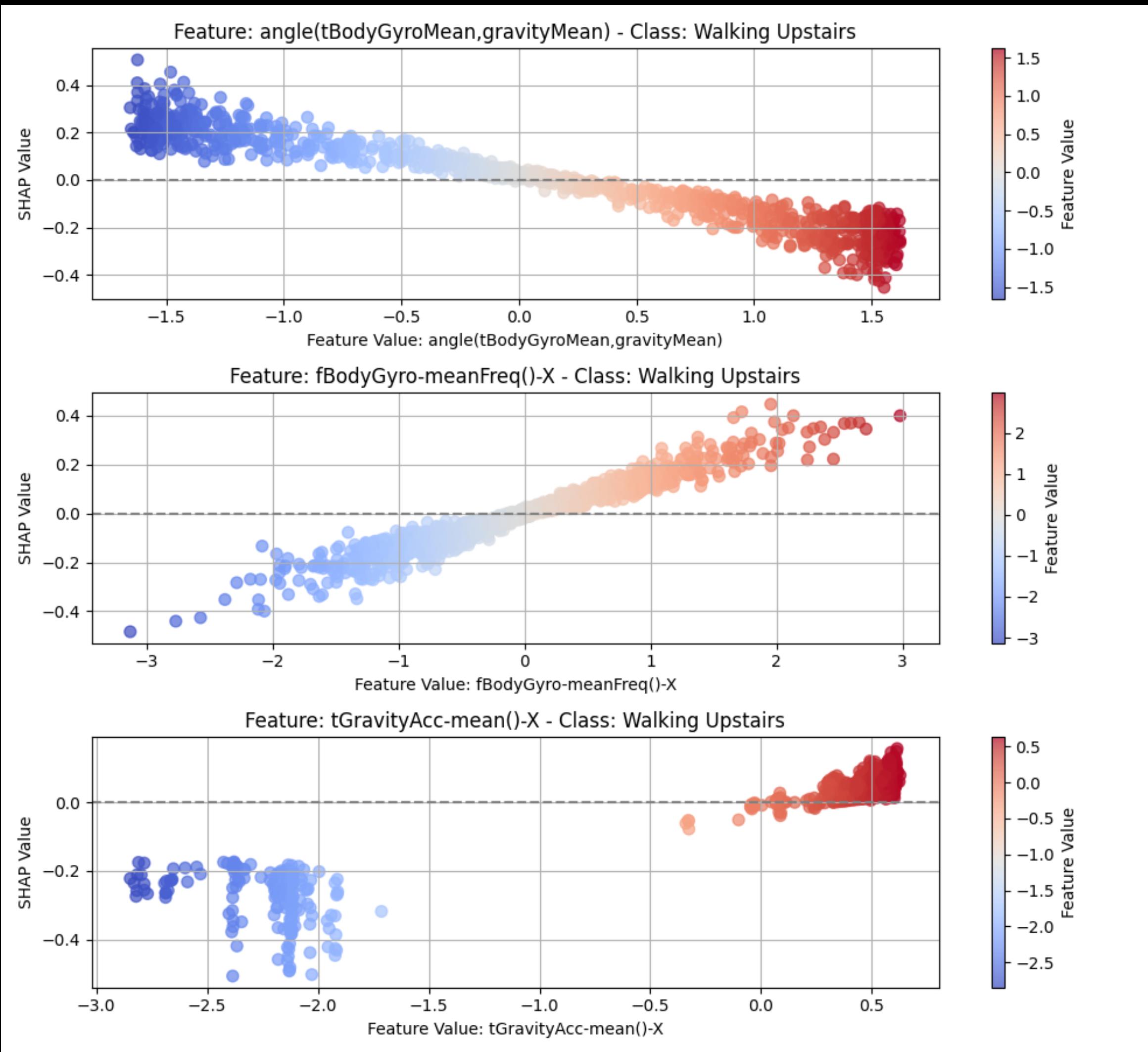
SHAP for CNN

Global explanations: Sitting vs Laying



SHAP for CNN

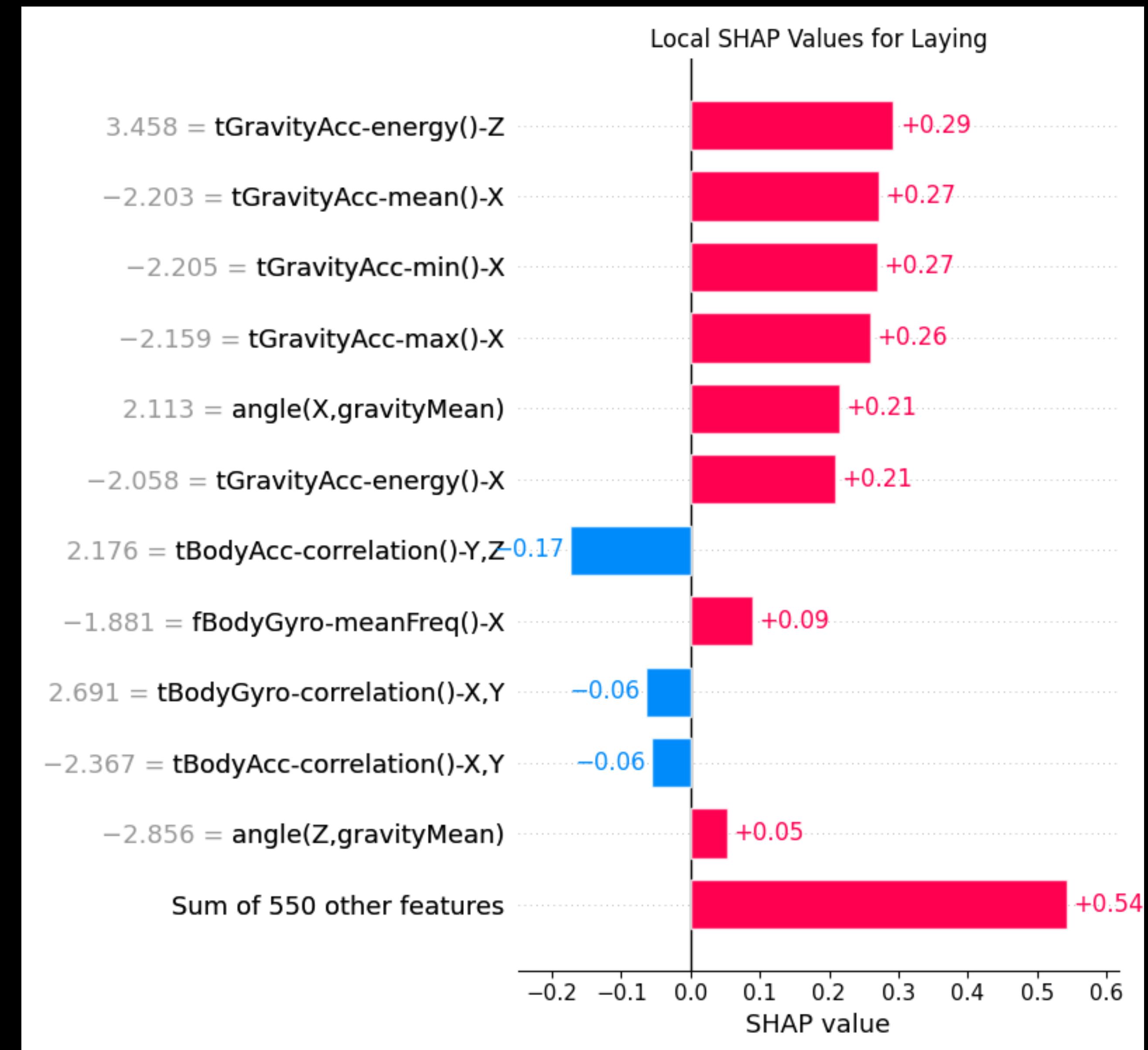
Dependency plots



SHAP for CNN

Local explanations

- Helps understand factors influencing individual predictions
- Positive SHAP values push the prediction towards the “*Laying*” class
- Negative SHAP values push the prediction away from the “*Laying*” class



Things to look out for!

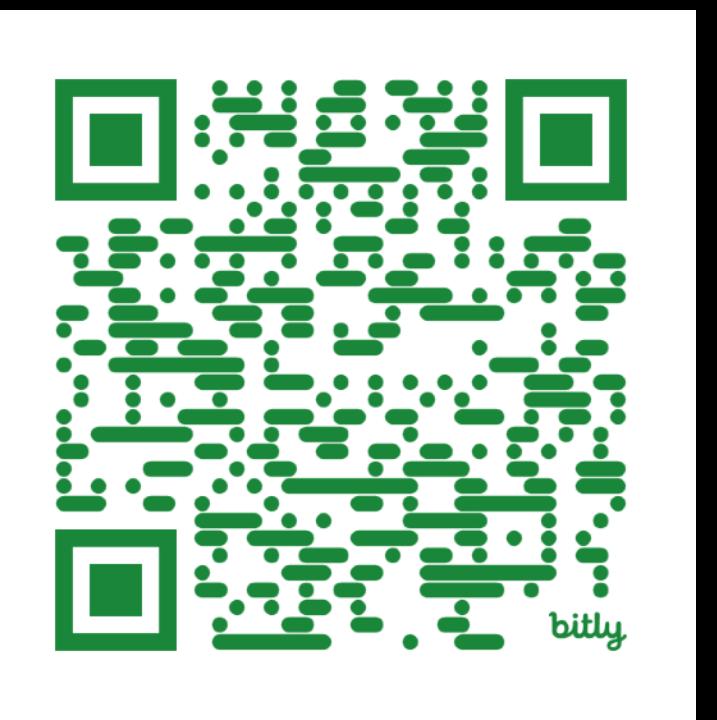
Limitations

- Explains the Model, not Reality!
- For large datasets, the calculation is very resource-intensive
- SHAP values can be misleading when features are highly correlated
- SHAP values show feature contributions, NOT causal effects
- Results depend on the choice of background dataset

Thank You! 🙏



Github with code and
slides



Connect on LinkedIn