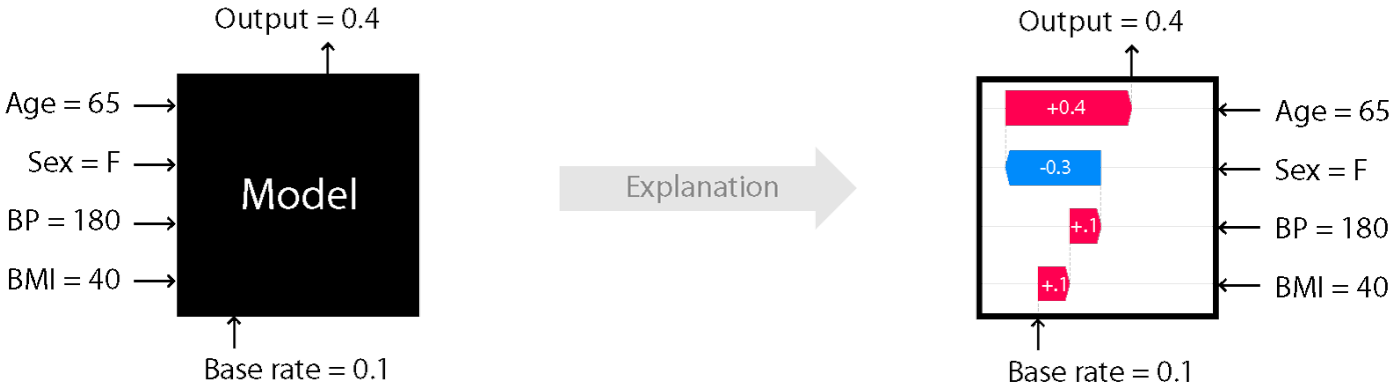
Explaining Black Box Machine Learning Models with SHAP

NLP-meeting, Brown Bag session 17.03.20

Nikolai Zhukov



Source of picture: https://github.com/slundberg/shap/blob/master/docs/index.rst

Explaining Black Box Machine Learning Models with SHAP

SHAP: **SH**apley **A**dditive ex**P**lanations. (Lundberg and Lee 2017, "A unified Approach to Interpreting Model Predictions")

Is a game theoretic approach to explain the output of any machine learning model.

Goal: explain model's predictions.

Why to use it?

- solid theory due to Shapley properties
- fast implementation (C++) for tree-based models
- global model interpretation
- debugging/exploration/monitoring of models

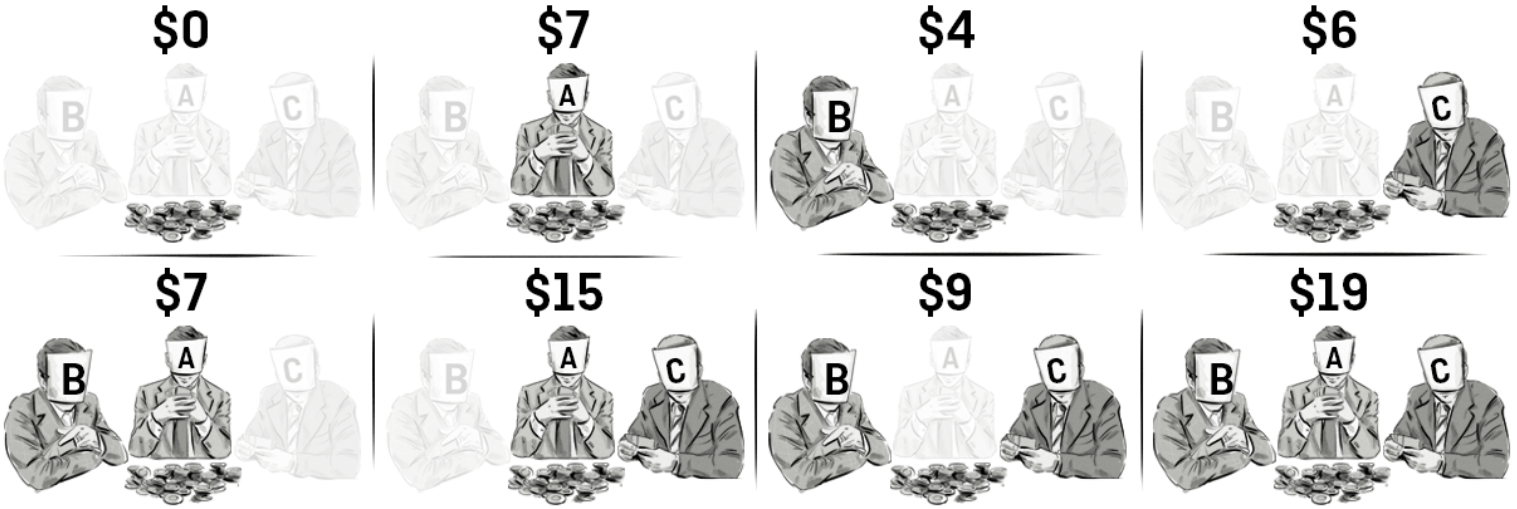
Implemented in Python (XGBoost, LightGBM, CatBoost, Pyspark and most tree-based scikit-learn models are supported in TreeExplainer).
Implementation of **D**eep **L**earning **I**mportant **F**ea**T**ures

General idea of SHAP(part with classical Shapley Values).

Shapley Values: a prediction can be explained by assuming that each feature value of the instance is a **"player"** in a **game** where the prediction is **payout**. The Shapley Value is a method from cooperative game theory, created by Loyd Shapley (1953) – tell us how to fairly distribute the "payout" among the features.

Shapley Value is average marginal contribution of a feature value across all possible coalitions.

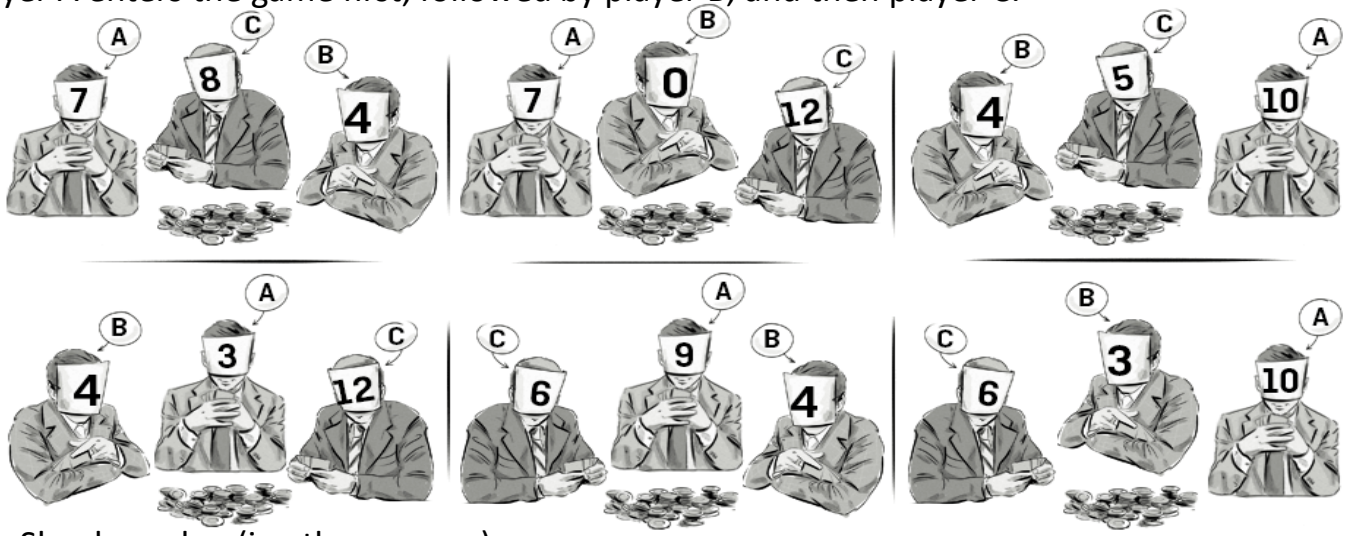
1. identifying each player's contribution when they play individually, when 2 play together, and when all 3 play together.



Source of picture: https://clearcode.cc/blog/game-theory-attribution/

General idea of SHAP(part with classical Shapley Values)

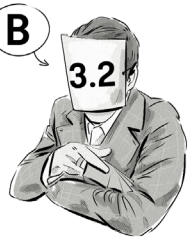
2. consider all possible orders and calculate their marginal value – e.g. what value does each player add when player A enters the game first, followed by player B, and then player C.



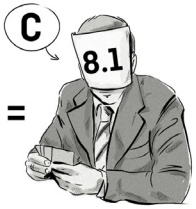
3. Calculate Shapley value (i.e. the average) for each player.

(7+7+10+3+9+10) / 6 =

(4+0+4+4+4+3) / 6 =



(8+12+5+12+6+6) / 6 =



Source of picture: https://clearcode.cc/blog/game-theory-attribution/

General idea of SHAP (predict apartment prices).

Estimation Shapley value for the feature **cat_banned**:

0. Take coalition with **cat_banned**.
Estimate prediction.

Park_nearby	Size_50	cat_banned
1	1	1

-> € 310,000

1. Remove **cat_banned** from coalition. Estimate prediction.

Park_nearby	Size_50
1	1

-> € 320,000

2. Calculate contribution of **cat_banned** (c_j).

c_j (cat_banned) =
€ 310,000 - € 320,000 = € -10,000

3. For all possible coalitions compute the predicted apartment price with and without the feature value **cat_banned** and take the difference to get the marginal contribution.

4. Estimate average of marginal contributions (=Shapley Value)

SHAP for “risk factors of cervical cancer” data

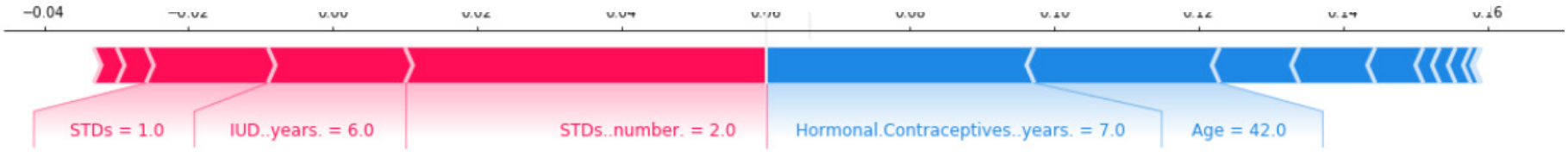
Task: prediction cervical cancer (classification).

Model: forest of randomized trees.

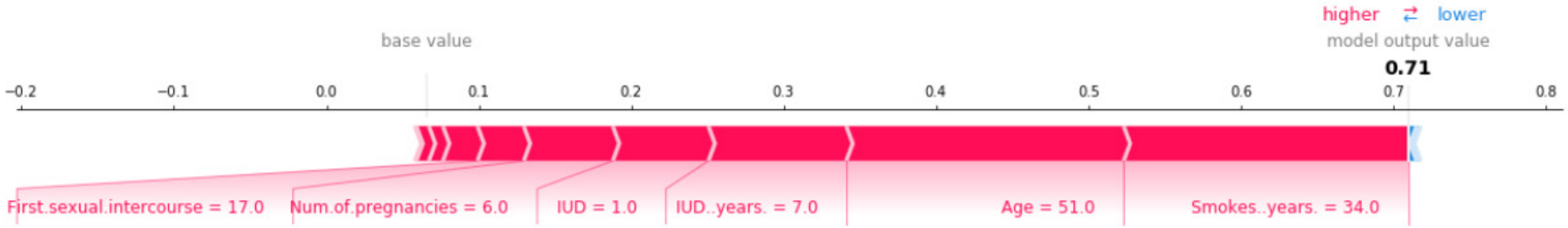
	Age	Number.of.sexual.partners	First.sexual.intercourse	Num.of.pregnancies	Smokes
0	18	4	15	1	0
1	15	1	14	1	0
2	34	1	15	1	0
3	52	5	16	4	1
4	46	3	21	4	0
...	...	...	...	...	...

Explain individual predictions

```
i = 18
shap.force_plot(explainer.expected_value[1],
shap_values[1][i,:], X.iloc[i,:], matplotlib = True)
```

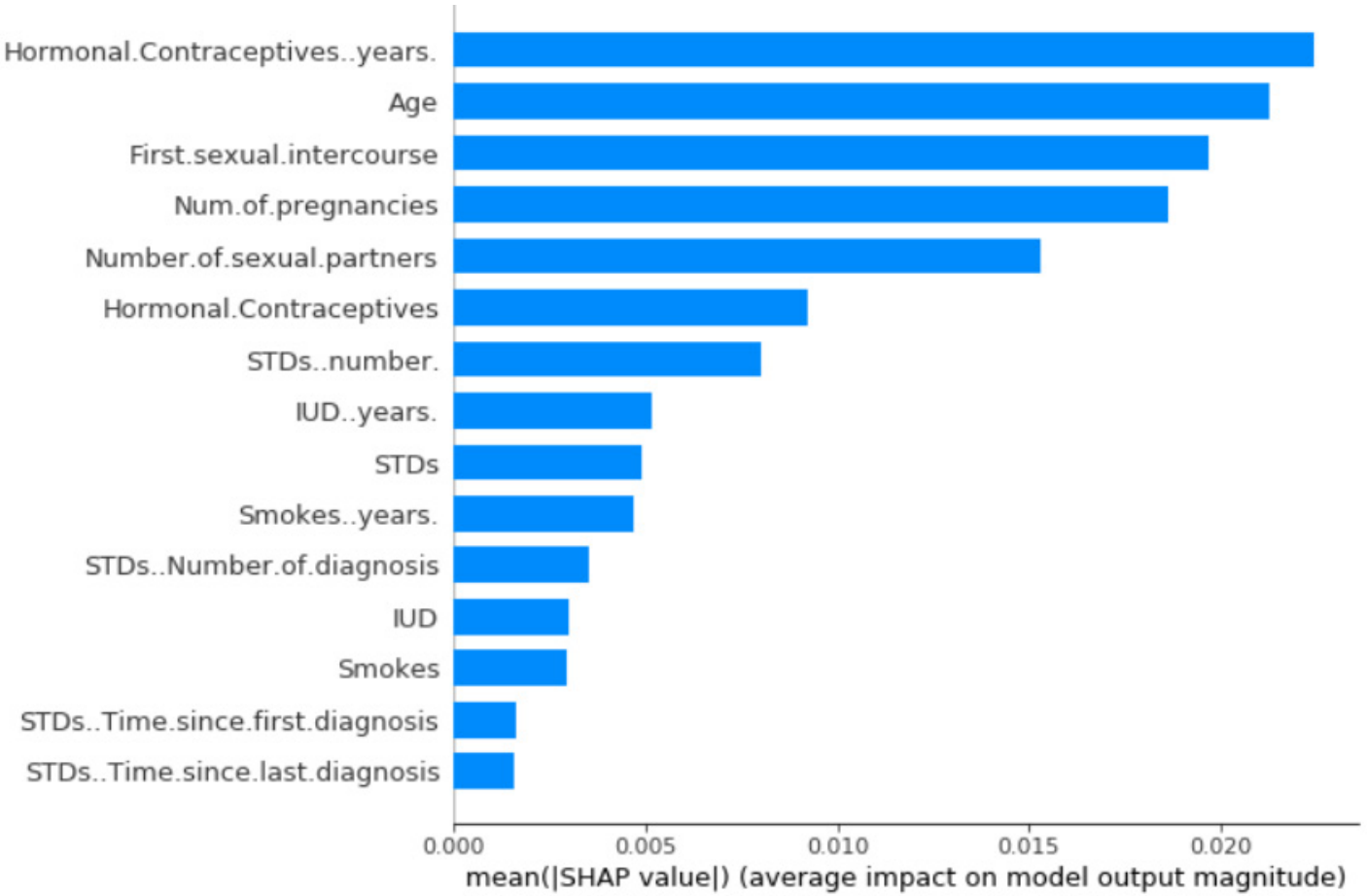



```
i = 6
```



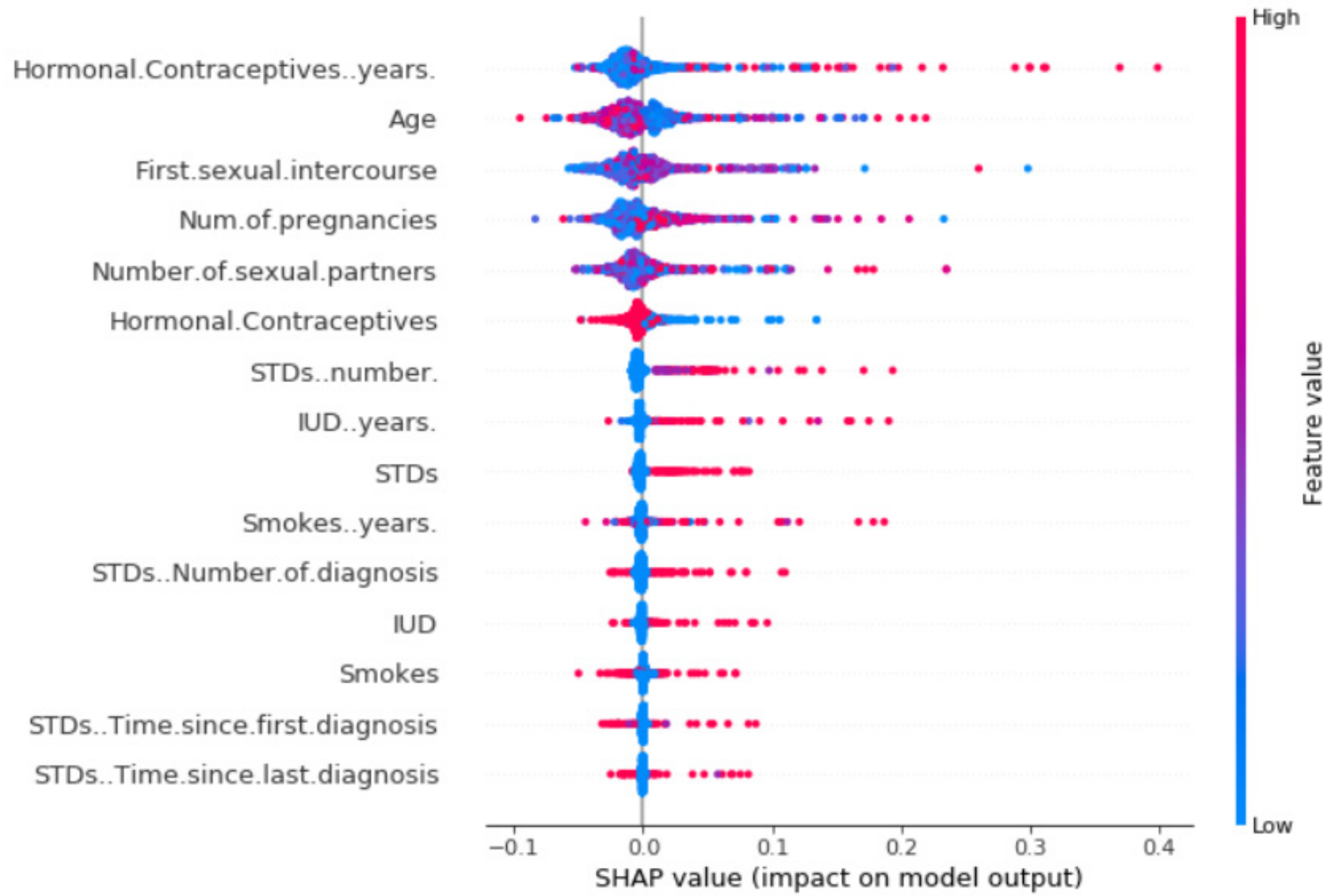
Feature importance

```
shap.summary_plot(shap_values[1], X, plot_type = "bar")
```



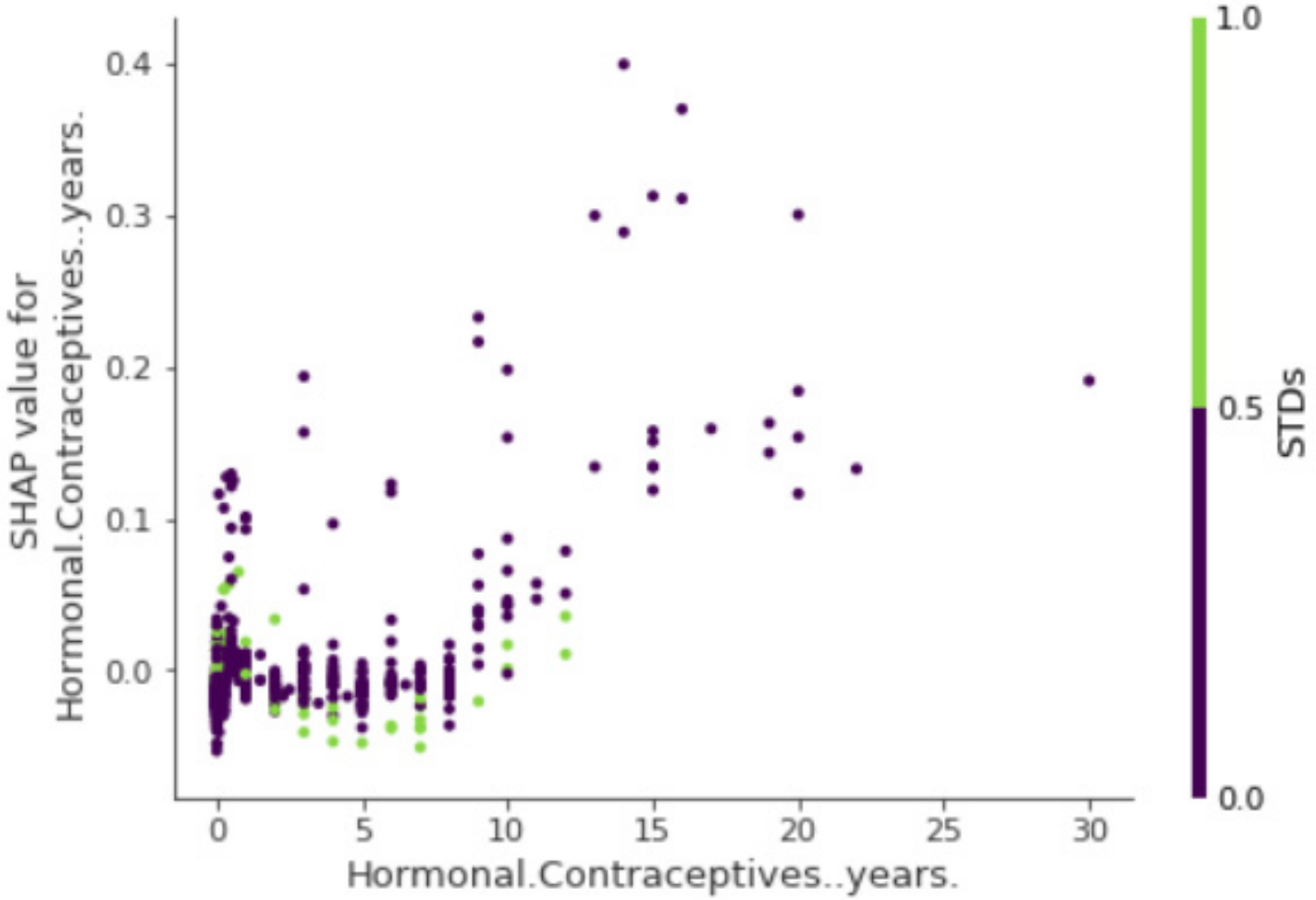
SHAP. Summary plot

```
shap.summary_plot(shap_values[1], X)
```



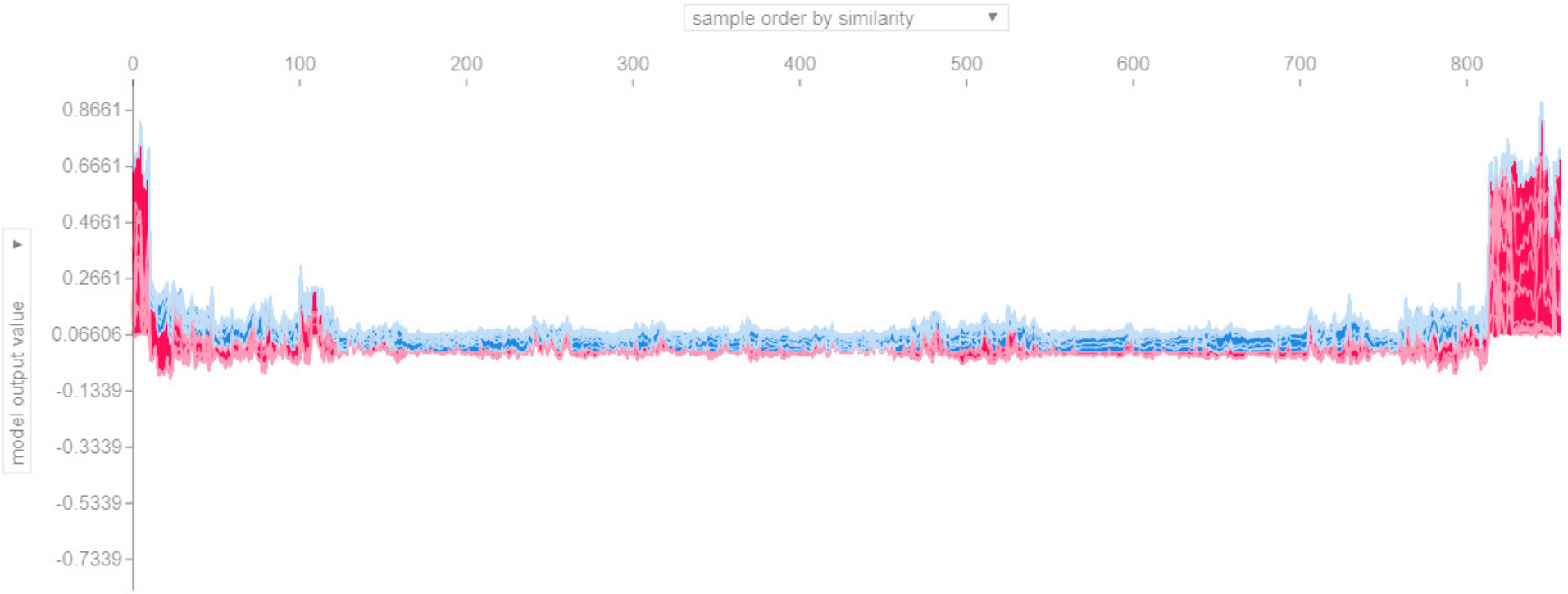
SHAP. Interaction Values

```
shap.dependence_plot("Hormonal.Contraceptives..years.",
shap_values[1], X, cmap=cmap2)
```



SHAP. Clustering

```
shap.force_plot(explainer.expected_value[1], shap_values[1], X)
```



SHAP resources:

https://github.com/slundberg/shap/blob/master/docs/index.rst

Alternative “classic Shapley Values” in R :

https://cran.r-project.org/web/packages/iml/vignettes/intro.html