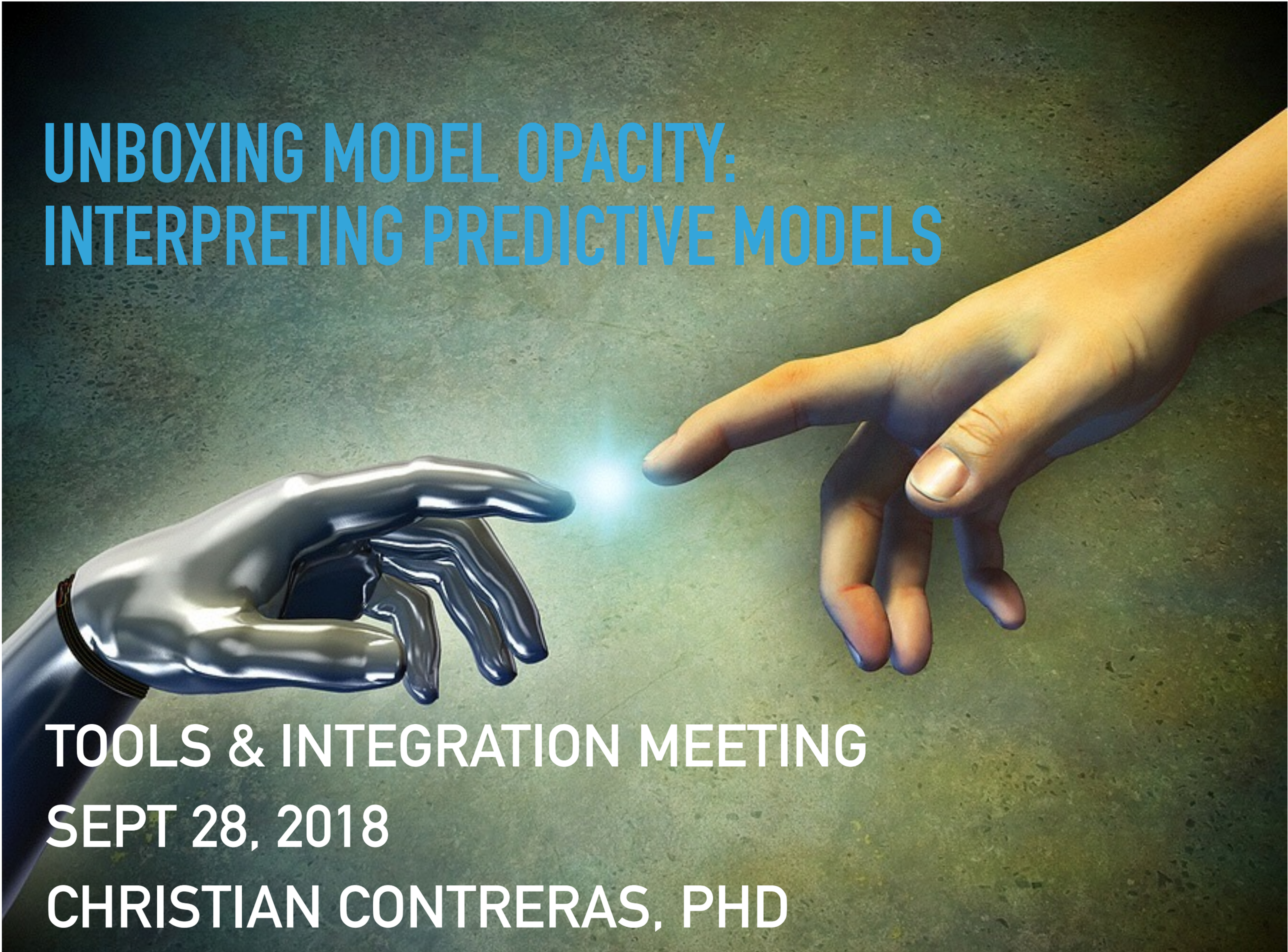
# UNBOXING MODEL OPACITY: INTERPRETING PREDICTIVE MODELS



TOOLS & INTEGRATION MEETING

SEPT 28, 2018

CHRISTIAN CONTRERAS, PHD

# GOAL OF FEATURE ATTRIBUTION METHOD

- Help better understand the model behaviour, which features mostly contribute to the output and possible reason for miss-classification.
- Attribute an effect to each feature and the sum of the effects of all feature attributions approximate the output of the black-box model.
- Faithfulness of the explanation model to the black-box model  is enforced by
  - Minimising the objective function composed of the loss function, regularisation term (penalises model complexity), and local sample weighting kernel.

# WHAT IS GAME THEORY?

- Game theory is the study of mathematical models of conflict and cooperation between intelligent rational decision-makers.

**Consider the following scenario**

- A group of people are playing a game. As a result of playing this game, they receive a certain reward.
- How can they divide this reward between themselves in a way which reflects each of their contributions?
- Cooperative Game Theory approach on fair allocation of credit with the goal of fair way for a coalition to divide its payoff.

**Players? Game? Payout? What's the connection to ML prediction and interpretability?**

- Cooperative Game Theory approach can be applied to "additive feature attributions".
- Predictions can be explained by assuming that each feature is a 'player' in a game where the prediction is the payout.
  - In a ML problem, the reward is the final prediction of the complex model, and the participants in the game are features.

# IF WE ALL COLLABORATE, HOW DO WE DIVIDE THE PAYOFF?

The solution to finding the values of attribution value predates ML and in fact it has its foundations in game theory.

- The **Shapley value** - a method from *coalitional game theory* - tells us how to fairly distribute the 'payout' or credit among the features depending on their contribution towards the total reward, with desirable properties *(see backup)*
- Assigns an importance value to each feature that represents the effect on the model prediction of including that feature, according to their **average marginal contribution** of a feature over all possible coalitions. Calculated by,

$$\phi_i(N, v) = \frac{1}{N!} \sum_{S \subseteq N \setminus \{i\}} |S|!(|N| - |S| - 1)! \Big[ v(S \cup \{i\}) - v(S) \Big]$$

for each player $i$.

Weight

marginal contribution

# UNIFIED FRAMEWORK FOR INTERPRETING PREDICTION

- **Game theory results guaranteeing** a unique solution for *additive feature attribution* methods
- **SHapley Additive exPlanations (SHAP)** – explains the output of any machine learning model using Shapley values, using a unified measure feature importance context.
  - The SHAP value for a feature is the average change in model output by conditioning on that feature when introducing features one at a time over all feature orderings.
  - Assigns a value to each feature for each prediction (i.e. feature attribution); the higher the value, the larger the feature's attribution to the specific prediction.
- Adheres to desirable properties
  - **Local accuracy:** requires the output of the local explanation model to match the original model *(truthfully explaining)*
  - **Missingness:** missing features have no attributed impact to the model predictions
  - **Consistency:** if a model changes so that some input's contribution increases or stays the same regardless of the other inputs, that input's attribution should not decrease
- The classic feature attribution methods for tree ensembles are inconsistent, meaning they can assign less importance to a feature when the true effect of that feature increases *(see backup)*.
- LIME choose the parameters heuristically; using these choices, does not recover the Shapley values.
  - Consequence is that local accuracy and/or consistency are violated, no guarantee that the feature with the highest attribution is actually the most important/influential to the model.

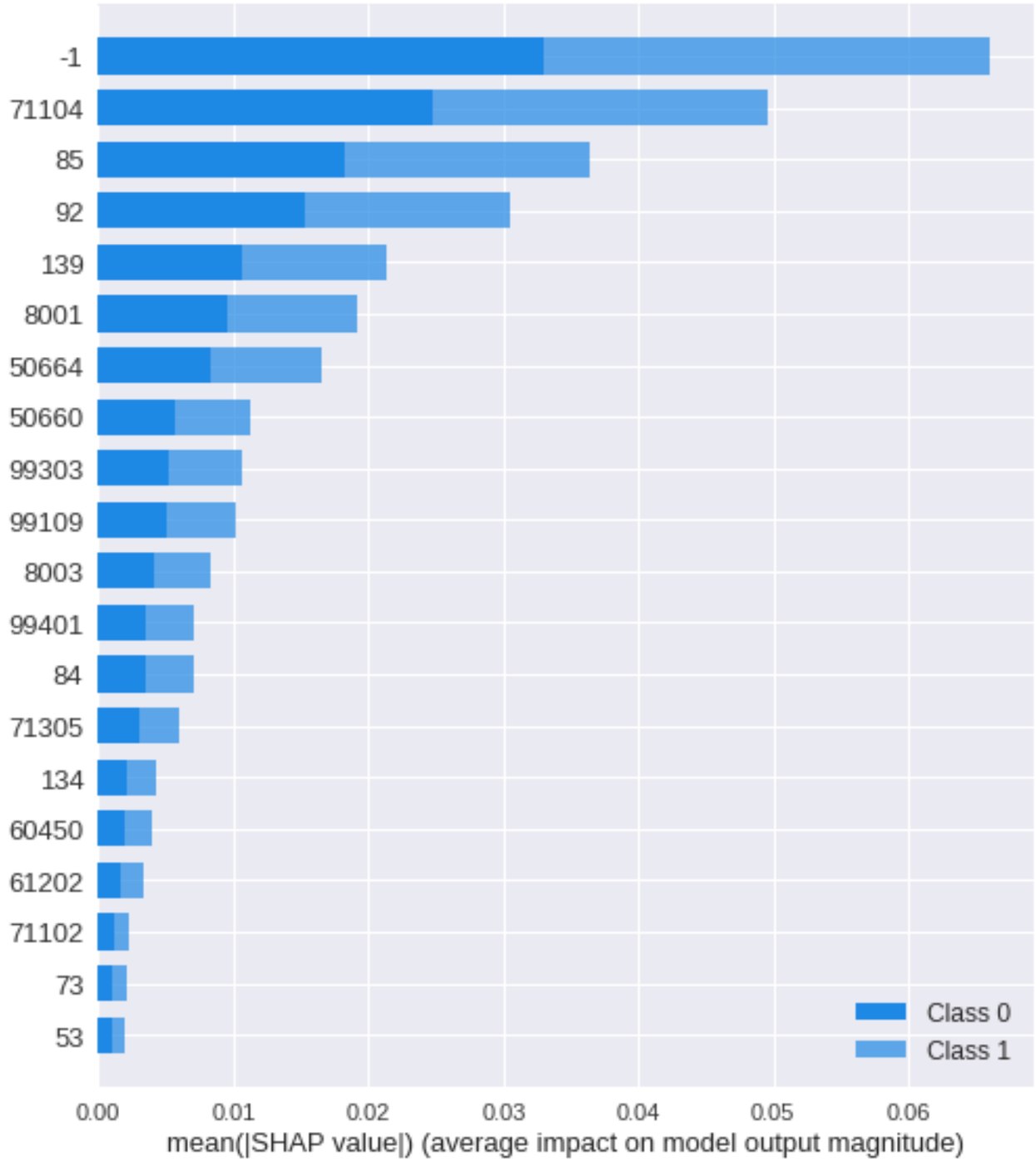
* Paper, A Unified Approach to Interpreting Model Predictions: https://arxiv.org/pdf/1705.07874.pdf

# FEATURE IMPORTANCE RANKING BASED ON SHAP VALUES



## SHAP MODEL EXPLANATION

- Class 1 corresponds to non-acdc, while class 2 represents acdc action
- Plot shows shows the feature (i.e error exit code) importance ranking based on the mean absolute sum of SHAP values
- The top 5 most contributing features to the mode's prediction are:
  - -1: ?
    - 71104: Agent Issue, Submit failed, Report to Workflow traffic controller, so an ACDC can be done.
    - 85: File-Read Error, Workflow issue, Input-file was found, but could not be read. Usually timeout when waiting for the file to open. This often happens when a file is available at only one location. The Network is overwhelmed by too many requests. Or bad worker nodes are taking the request. Secondary exit code from WMStats 8021, 8026.
    - 92: Remote file read error, Problem with one of the redirectors, Contact transfer team to investigate
    - 139: workflow issue, job runs out of virtual memory,
    - 8001: "Unable to open trivial file catalog", PluginLibraryLoadError, Site Issue, Issue with CVMFS and Singularity, Issue with CVMFS? (GGUS ticket was closed without a solution posted. Error just went away at some point), Open GGUS ticket to site with message from WMStats and CC the Transfer team on it because it is likely a transfer team issue. Open GGUS ticket with message from WMStats
    - 50664: Workflow issue, Wall Clock (wrapper initiated)



# FEATURE IMPORTANCE RANKING BASED ON PERMUTATION



## MODEL EXPLANATION

- The only exit code features given credit for model predictions are error exit code 71104, 71305, 53, 50660, 92, 61202, 139, but the last 4 only given negligible credit.
- Though this method of feature importance is not as robust as snap-value method show in the next slide (as the method assumes the features are non-correlated, and would not treat the importance measurement corruptly for instances where the features are correlated). But does provide a first order look at one exit code the model considers important.
- Exit code issue type, problem and solution
  - 71104: Agent Issue, Submit failed, Report to Workflow traffic controller, so an ACDC can be done.
  - 71305:?
  - 53: workflow issue, Workflow issue, Max. RSS.
  - 92: Remote file read error, Problem with one of the redirectors, Contact transfer team to investigate
  - 61202:?
  - 139: workflow issue, job runs out of virtual memory

| Weight | Feature |
|---|---|
| 0.0024 ± 0.0006 | 71104 |
| 0.0021 ± 0.0005 | 71305 |
| 0.0007 ± 0.0000 | 53 |
| 0.0001 ± 0.0005 | 50660 |
| 0.0001 ± 0.0005 | 92 |
| 0.0001 ± 0.0005 | 61202 |
| 0.0001 ± 0.0005 | 139 |
| 0 ± 0.0000 | 70452 |
| 0 ± 0.0000 | 70318 |
| 0 ± 0.0000 | 70 |
| 0 ± 0.0000 | 60450 |
| 0 ± 0.0000 | 11003 |
| 0 ± 0.0000 | 60307 |
| 0 ± 0.0000 | 50661 |
| 0 ± 0.0000 | 71101 |
| 0 ± 0.0000 | 50115 |
| 0 ± 0.0000 | 132 |
| 0 ± 0.0000 | 99999 |
| 0 ± 0.0000 | 71 |
| 0 ± 0.0000 | 60405 |

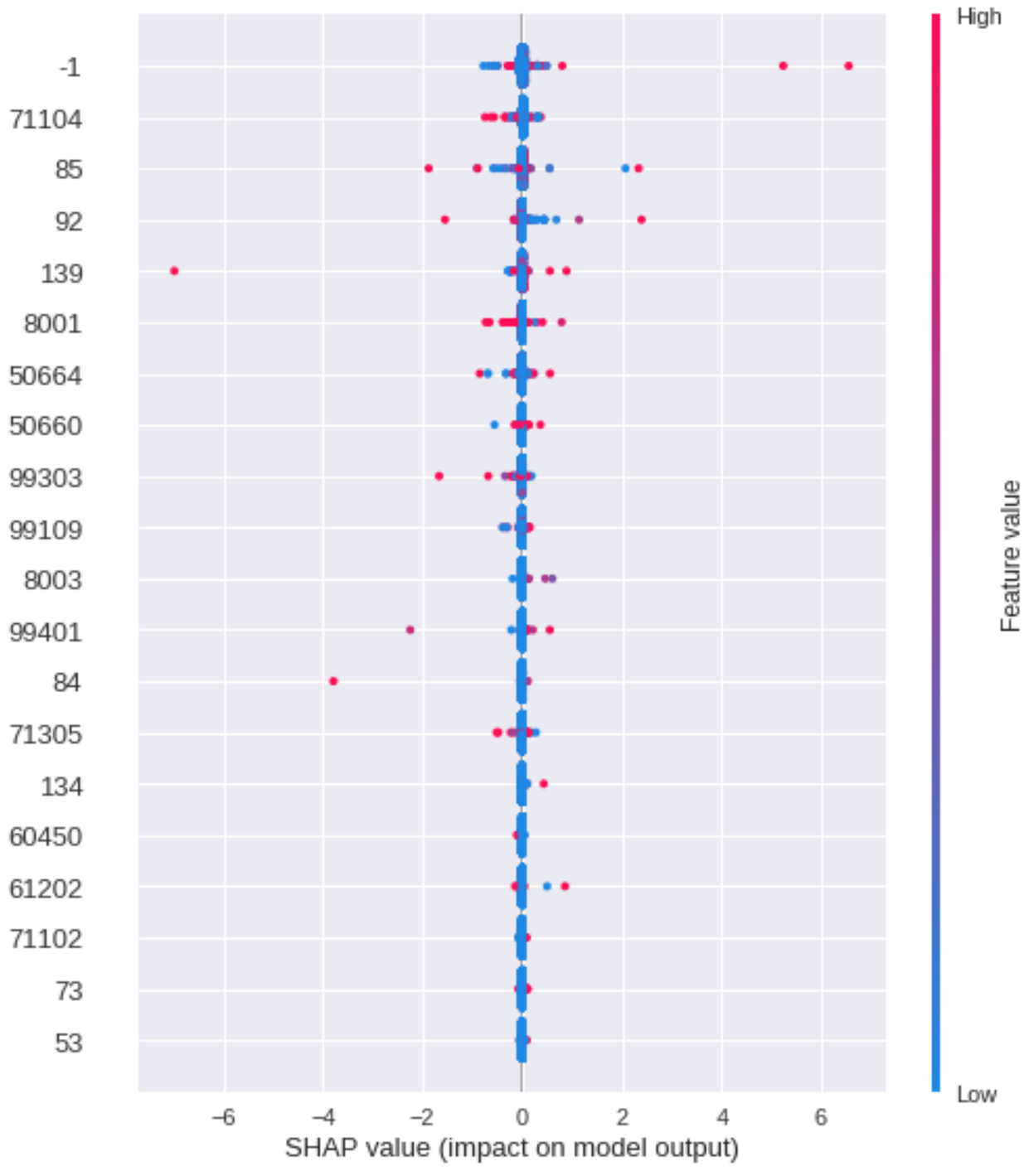
*… 33 more …*

# FEATURE IMPORTANCE RANKING BASED ON SHAP VALUES



- Plot the SHAP values of every feature for every example, shows features (exit error code number) are most important for a model, in rank order.

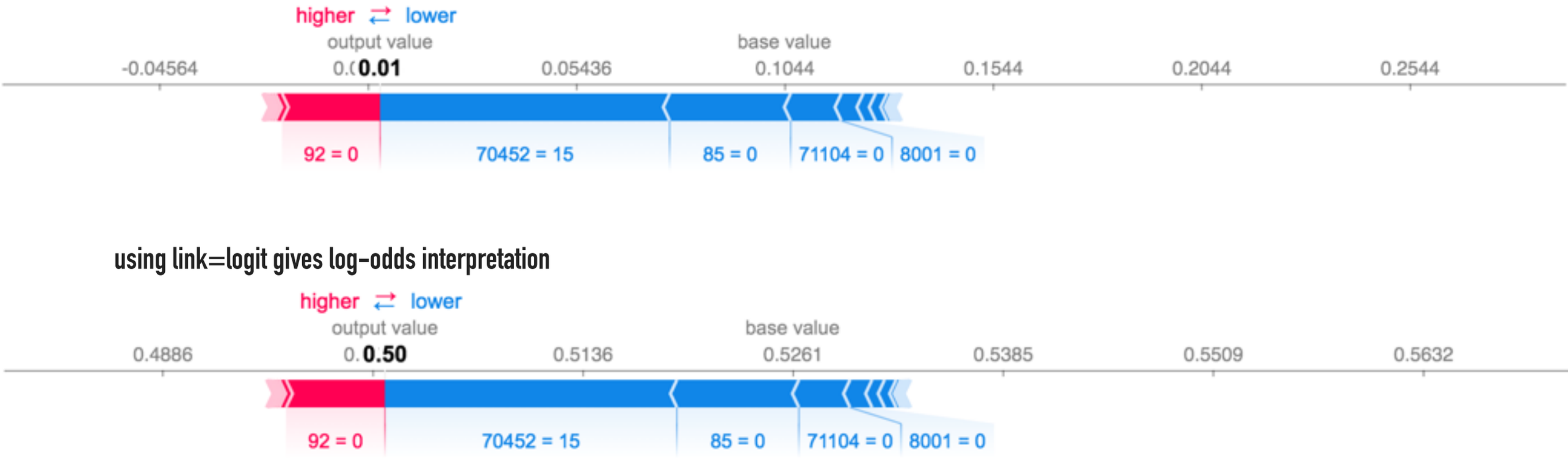
- Shap-values centred around zero indicate that particular features is given no attribution for the model's prediction.



# SHAP MODEL EXPLANATION FOR A GIVEN PREDICTION



- The model predicted acdc against it being a non-acdc scenario
- Let's see why the models thinks so,
  - The **features in blue** push the predictions towards baseline value
  - While the opposite holds true for the **features in red**.



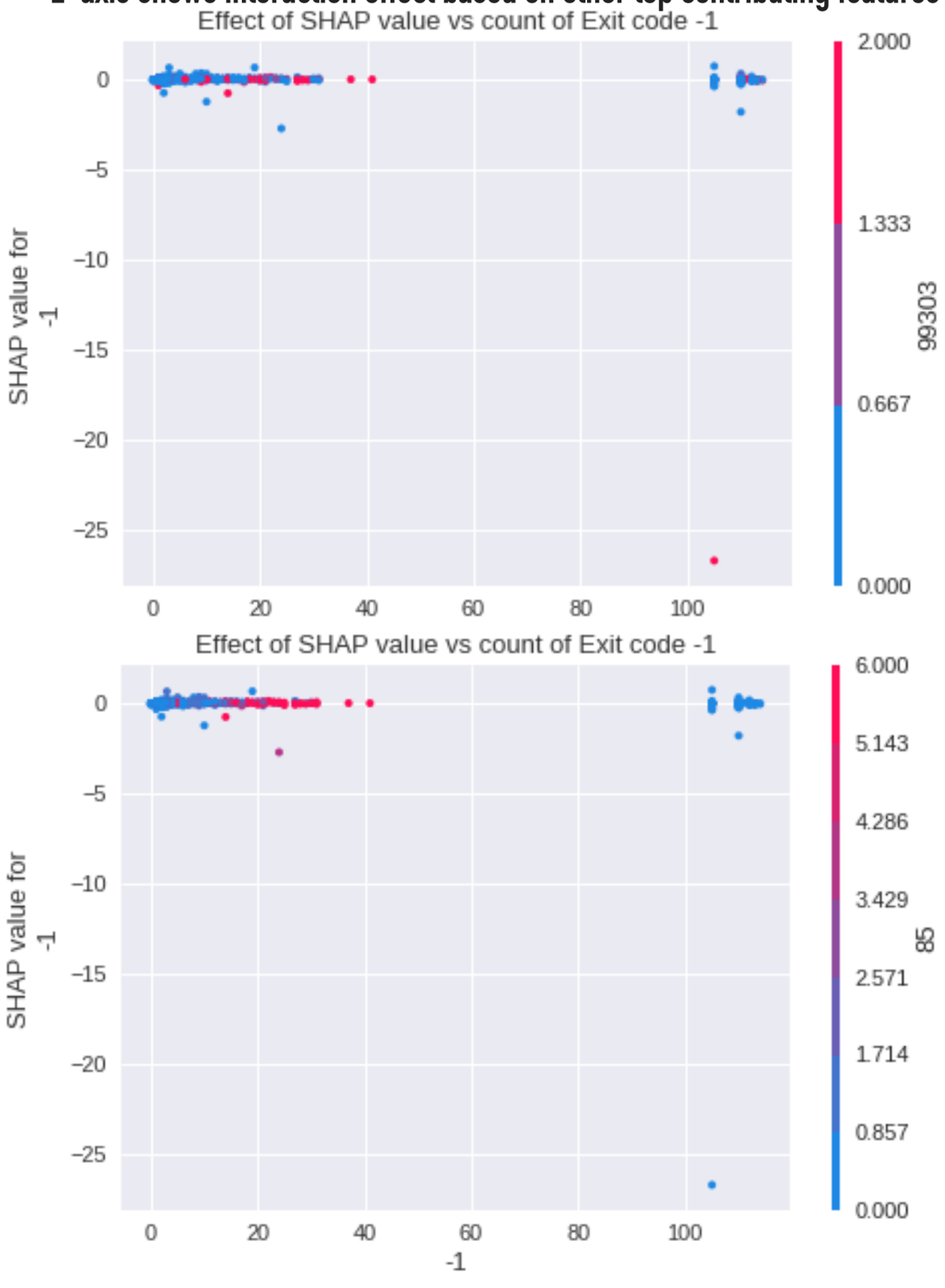

using link=logit gives log-odds interpretation

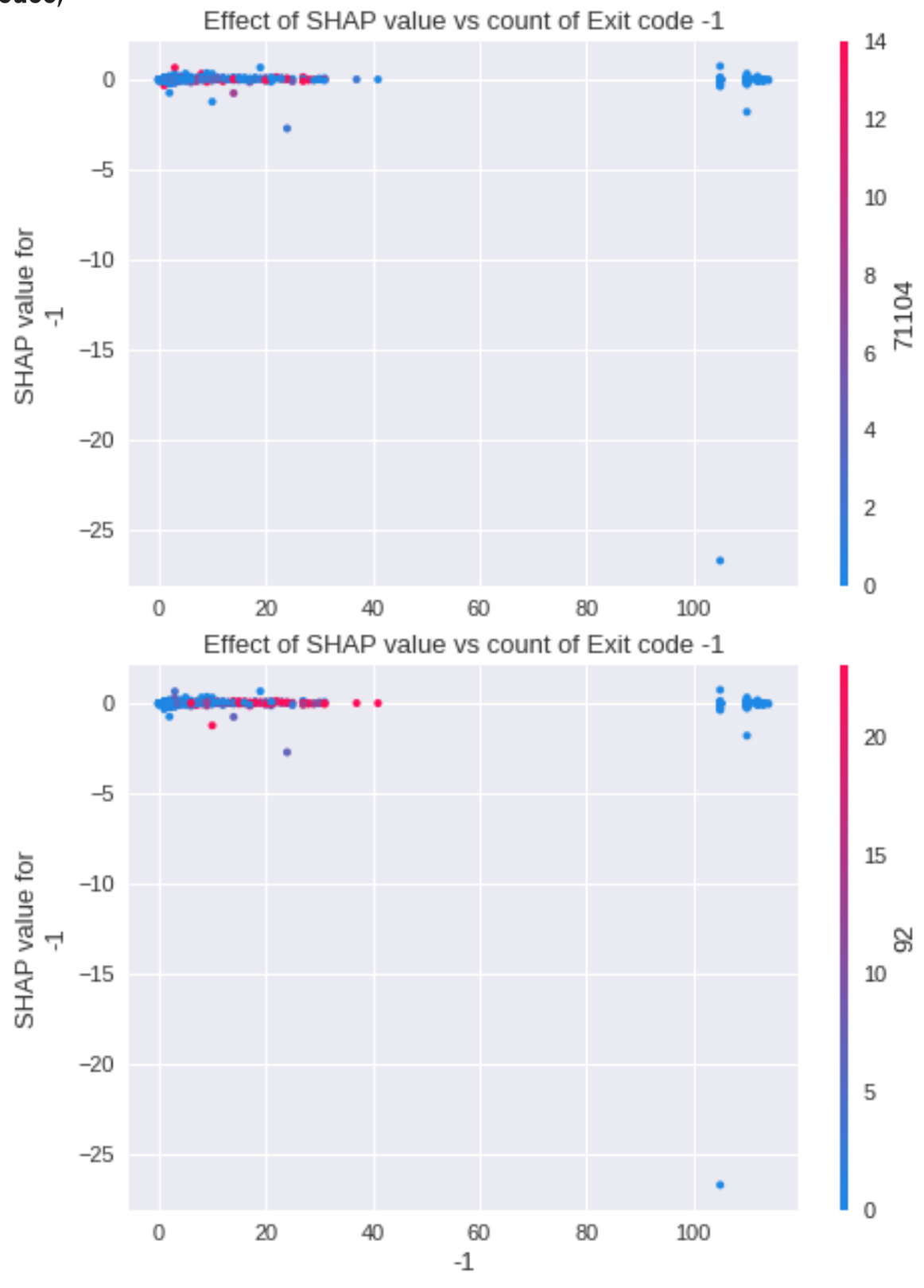
# PARTIAL DEPENDENCE PLOTS BASED ON SHAP-VALUE



**Plot shows how the shap-value changes according the the number of time exit code -1 (no exit code found json file) was triggered**

**Z-axis shows interaction effect based on other top contributing features (exit codes)**



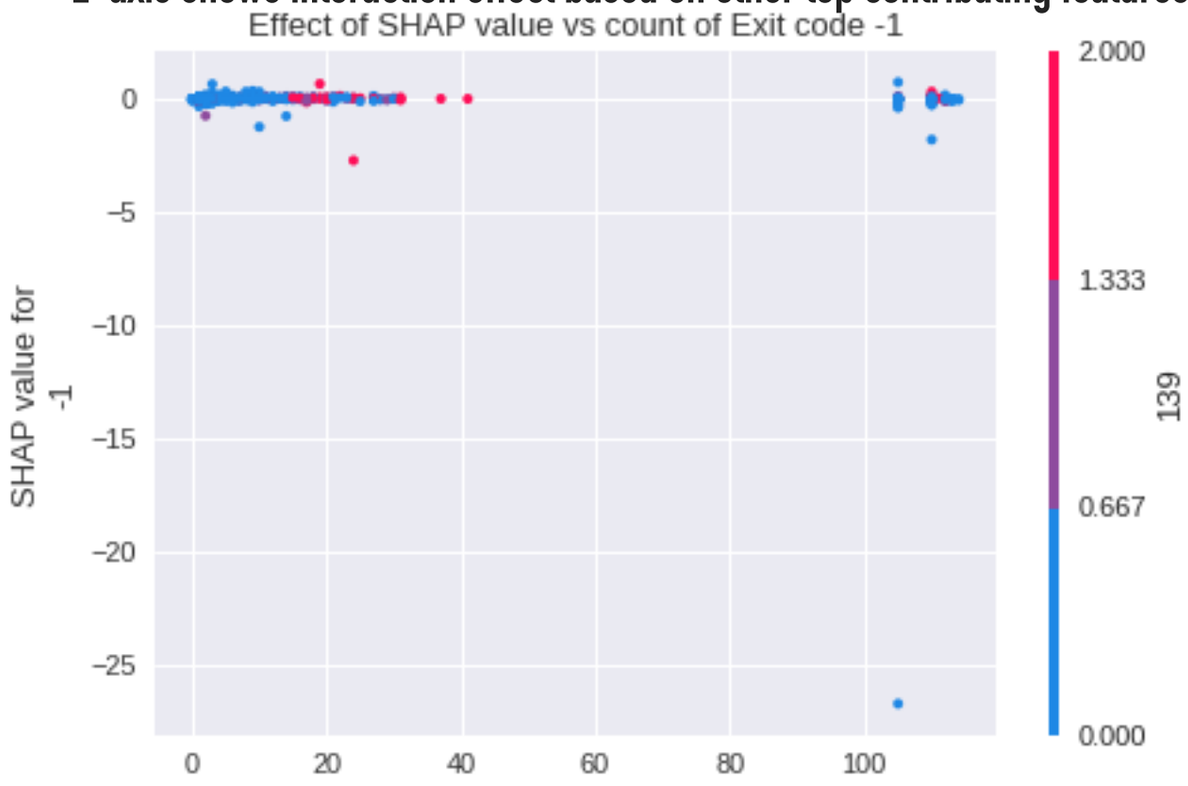



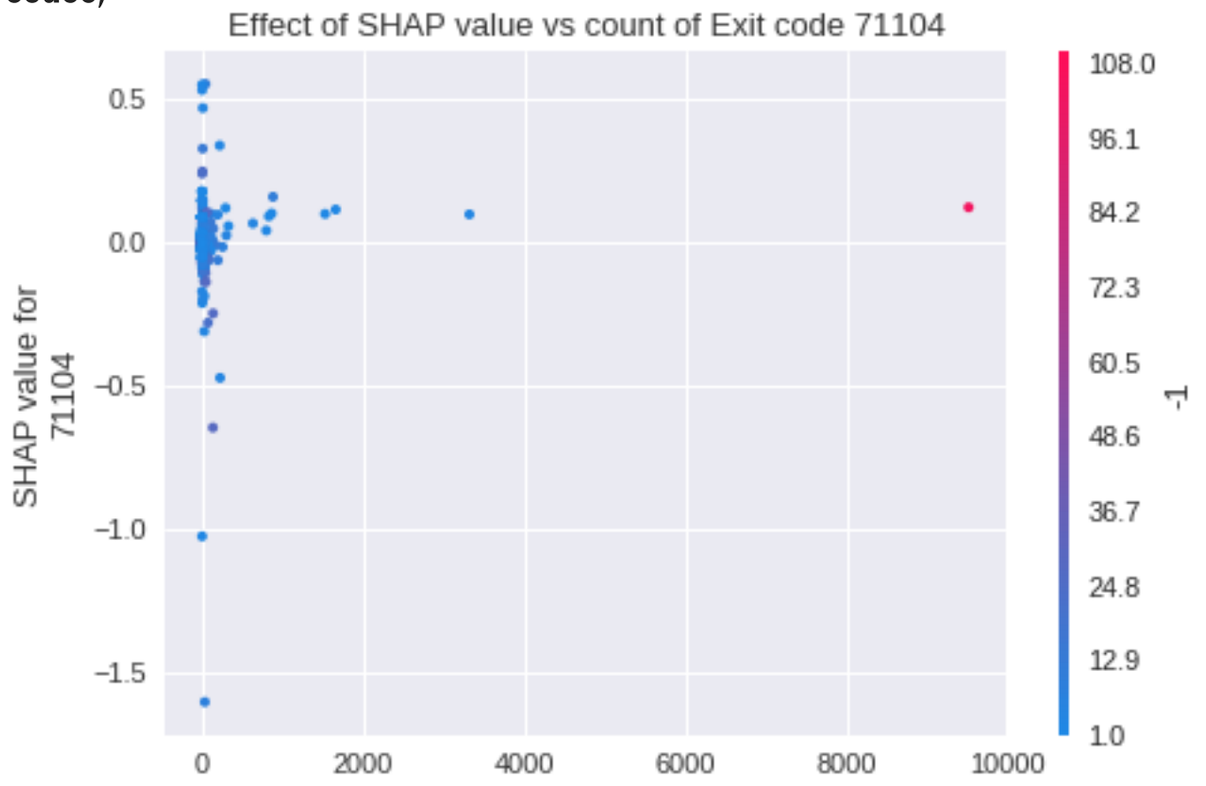
# PARTIAL DEPENDENCE PLOTS BASED ON SHAP-VALUE

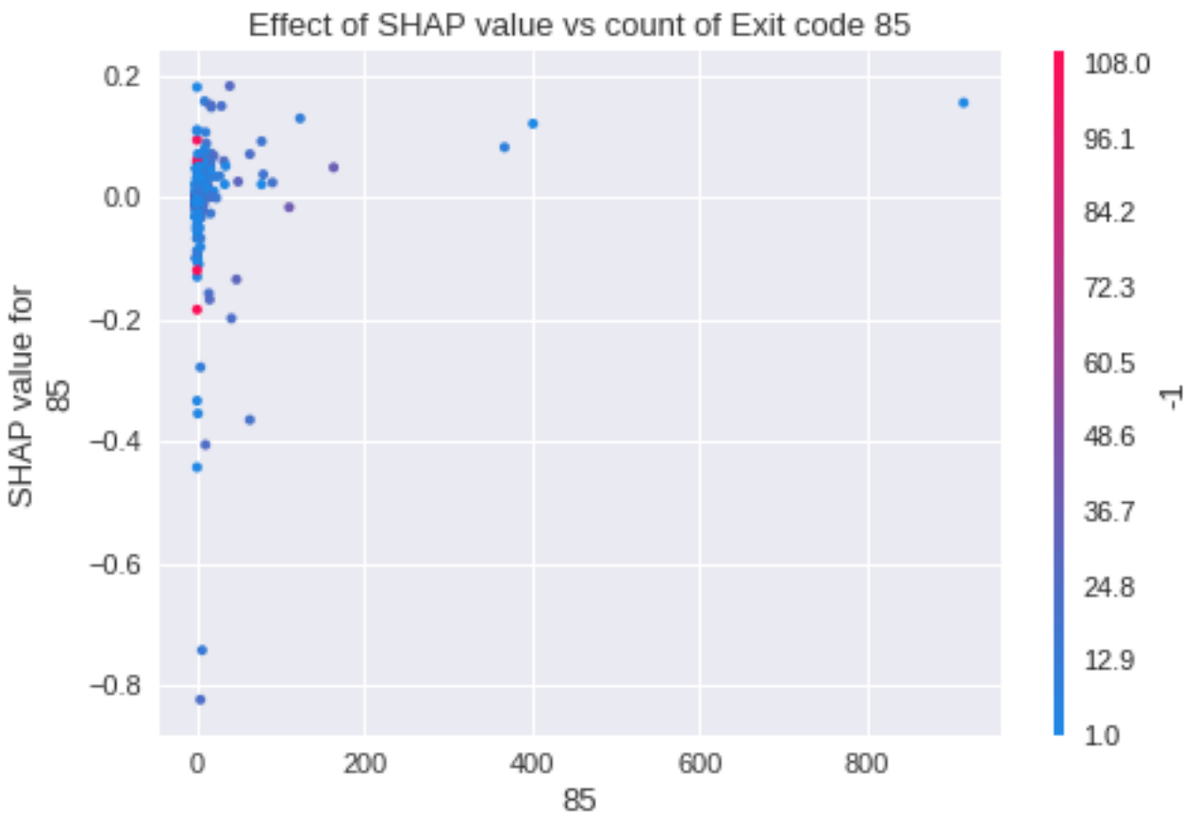


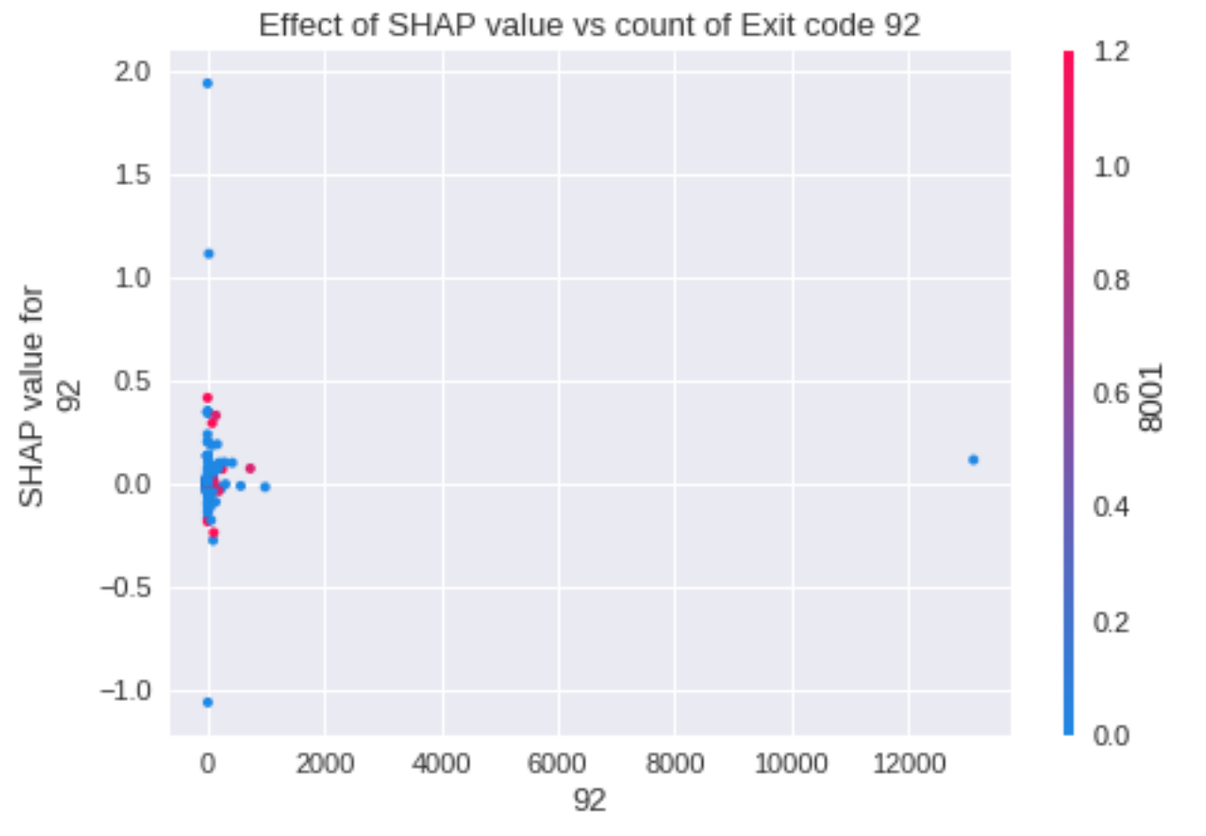
**Plot shows how the shap-value changes according the the number of time exit code -1, 71104, 85, 92 was triggered**

**Z-axis shows interaction effect based on other top contributing features (exit codes)**



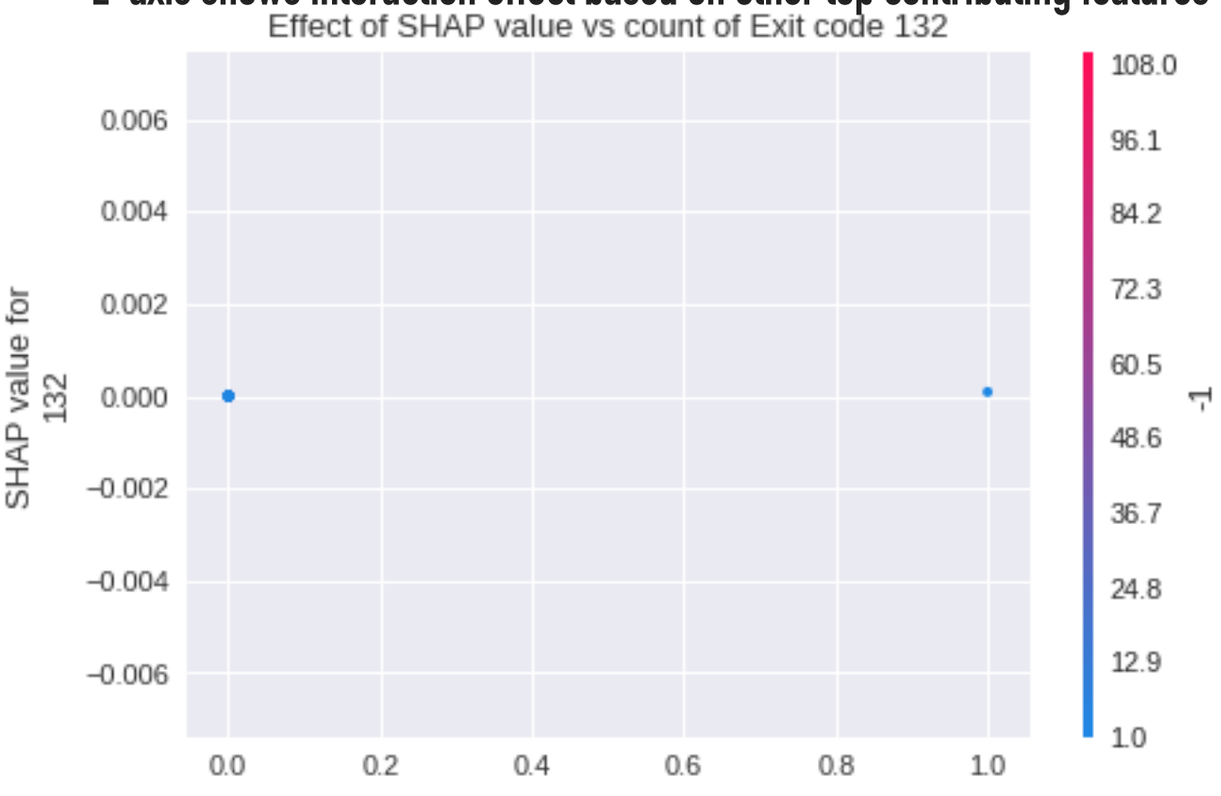



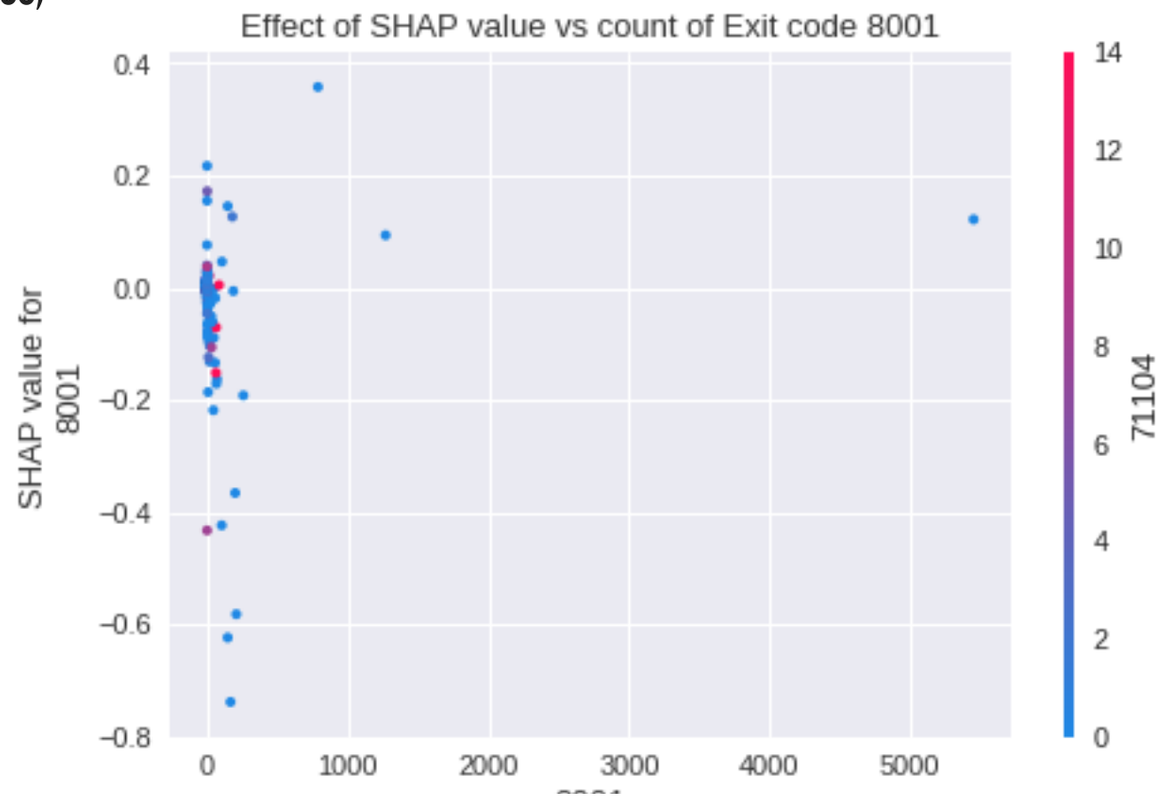
# PARTIAL DEPENDENCE PLOTS BASED ON SHAP-VALUE

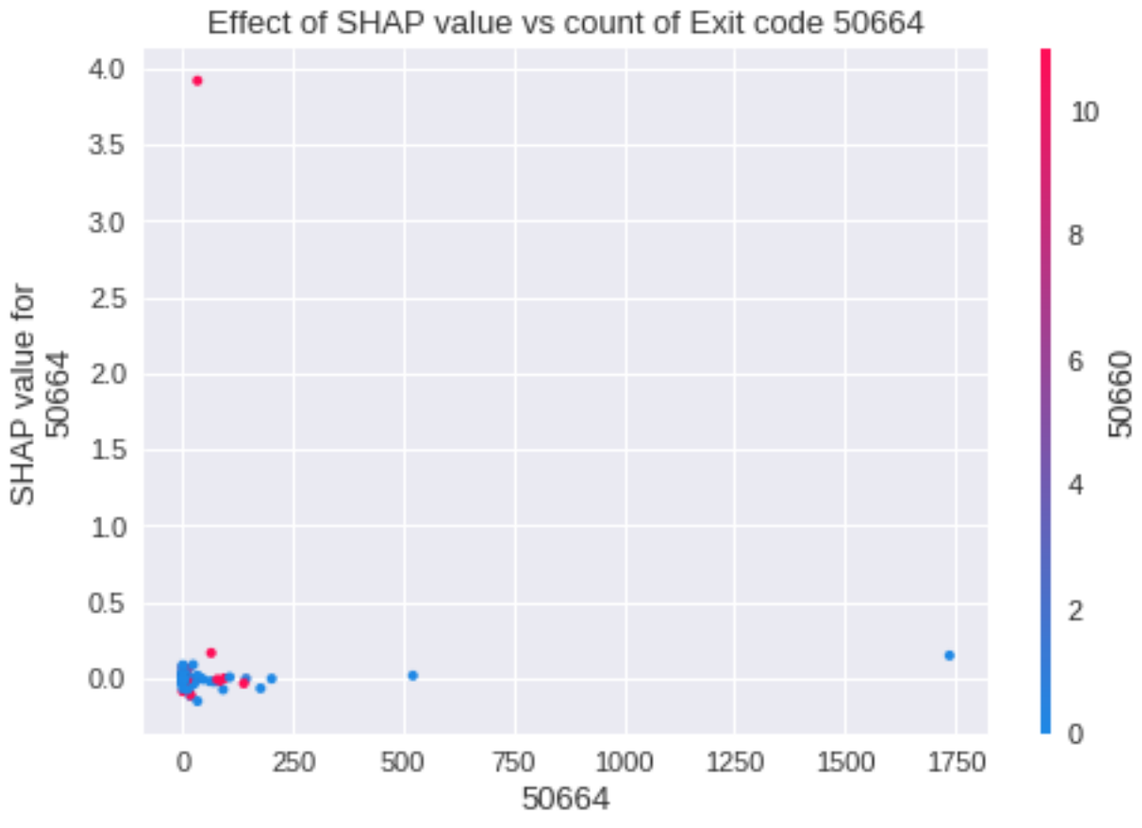


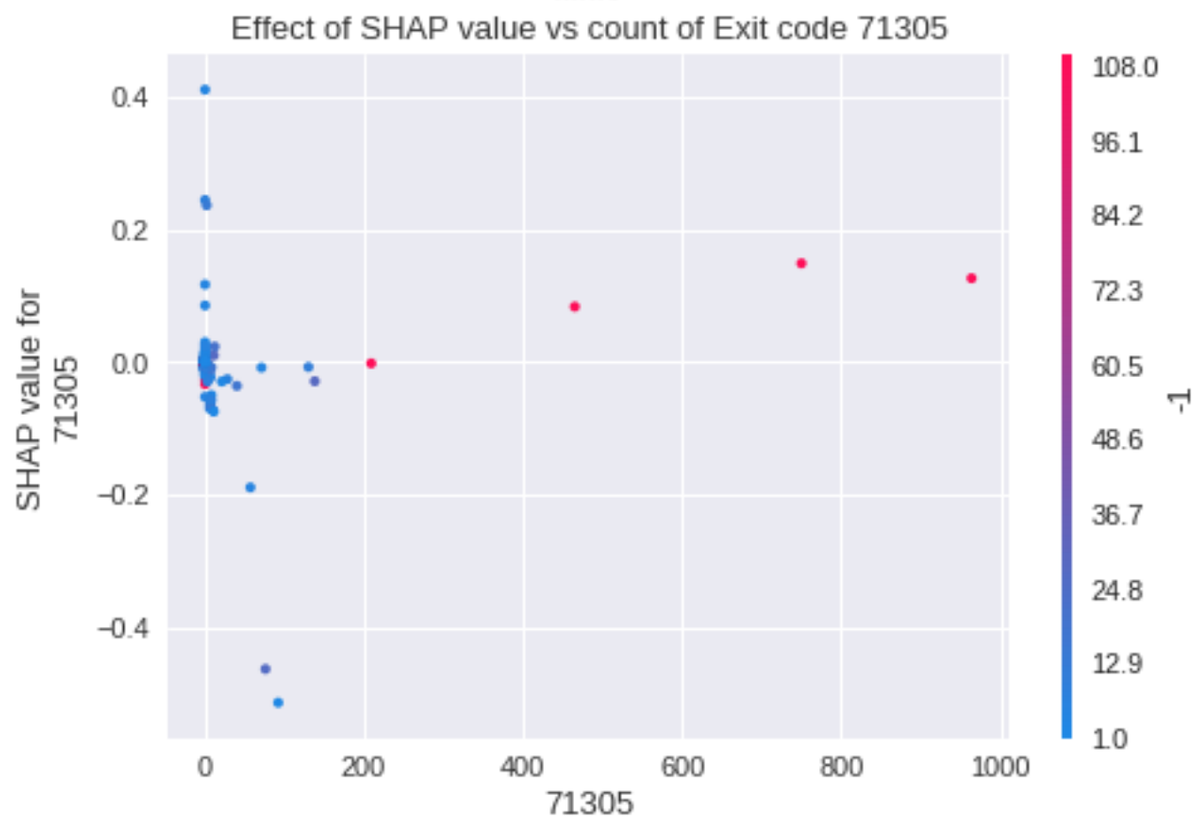
**Plot shows how the shap-value changes according the the number of time exit code 132, 8001, 50664, 71305 was triggered**

**Z-axis shows interaction effect based on other top contributing features (exit codes)**



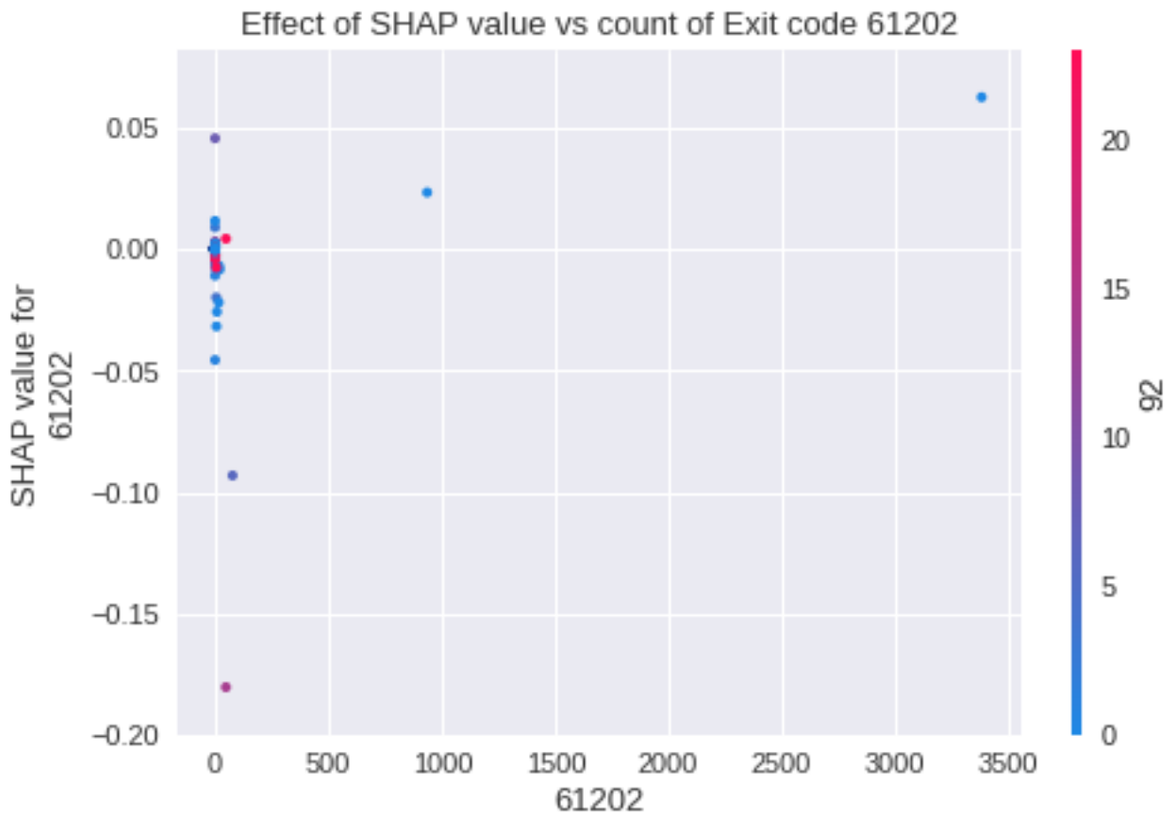



# PARTIAL DEPENDENCE PLOTS BASED ON SHAP-VALUE



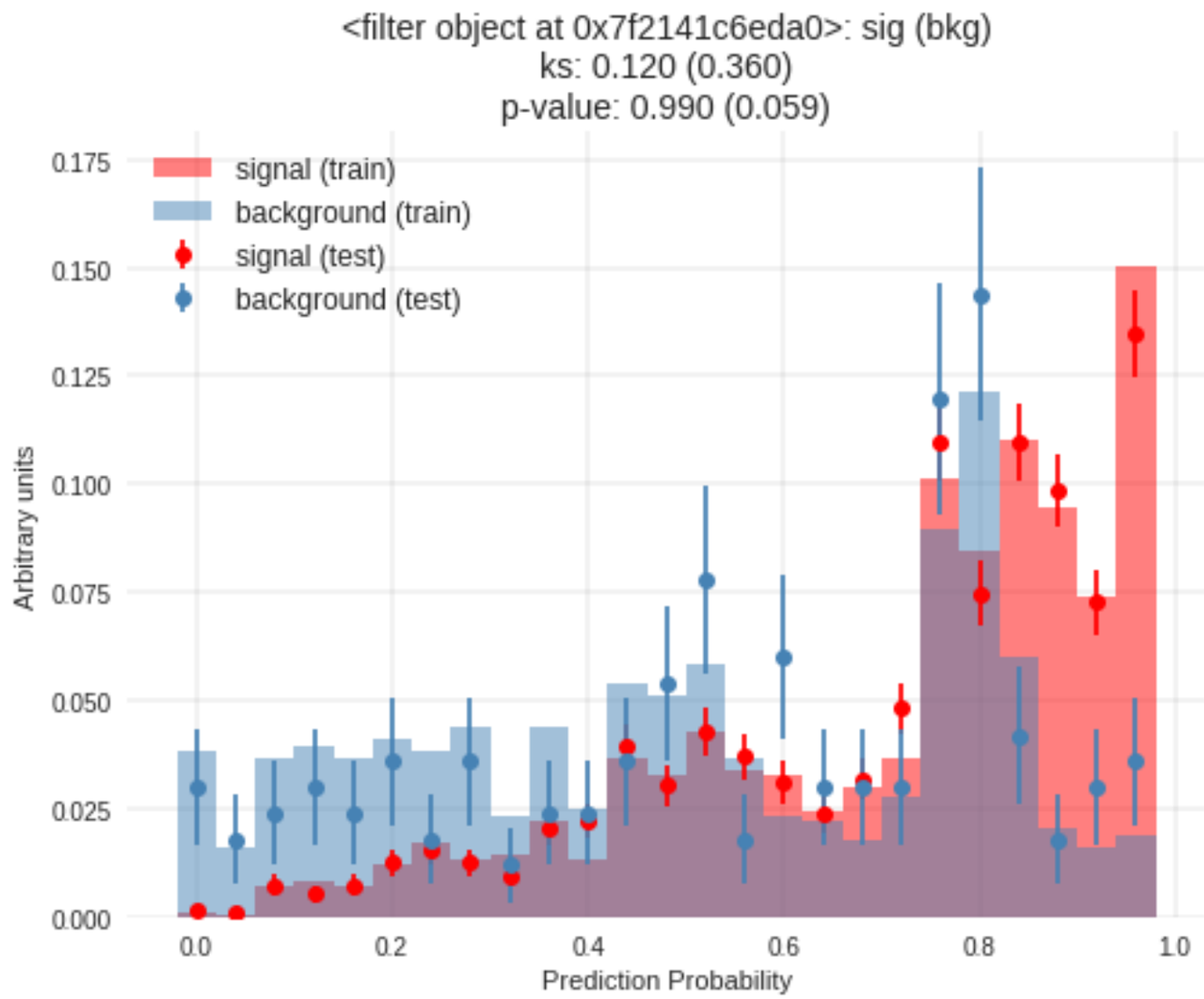
**Plot shows how the shap-value changes according the the number of time exit code 61202 was triggered**

**Z-axis shows interaction effect based on other top contributing features (exit codes)**

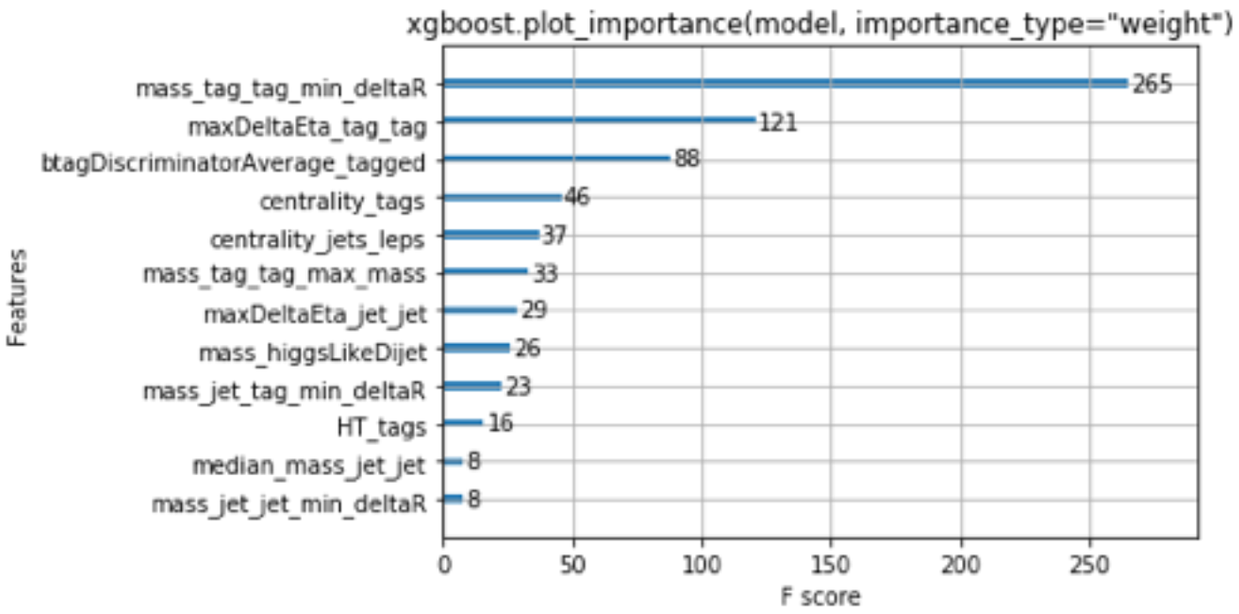


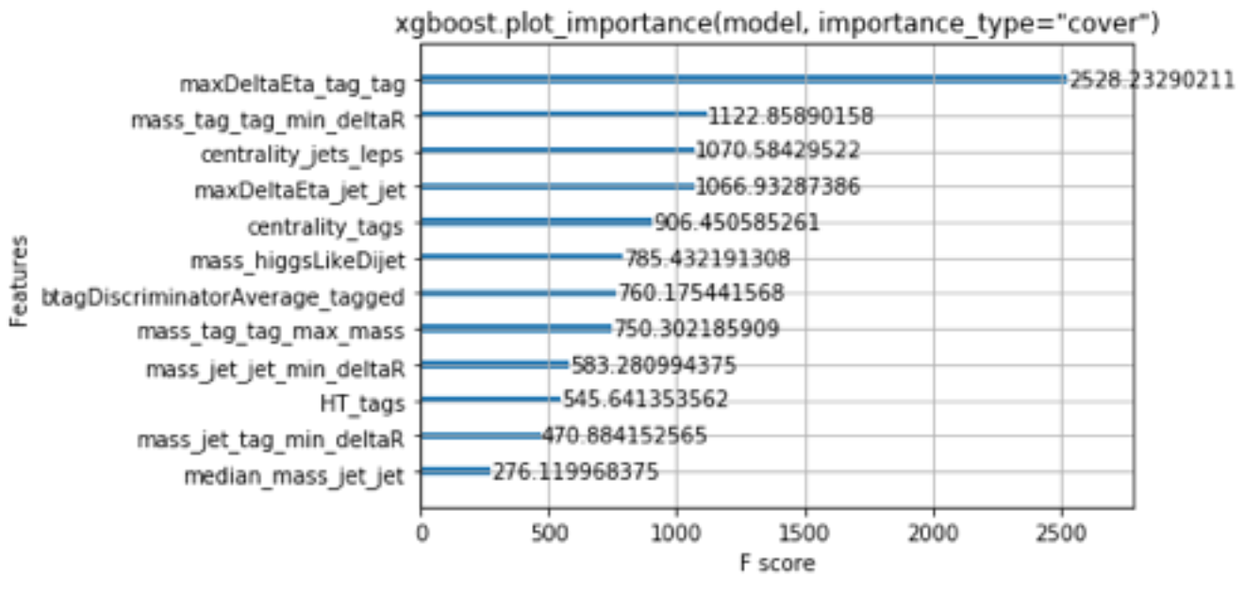

# CHECK MODEL OVER TRAINING

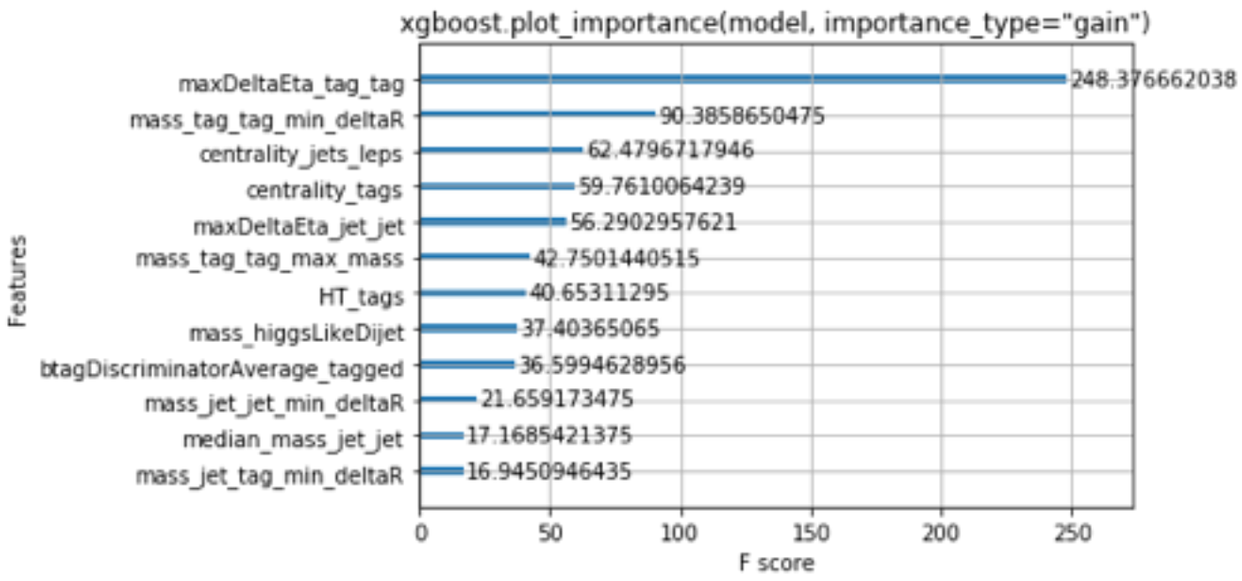




# GLOBAL FEATURE IMPORTANCE RANKING

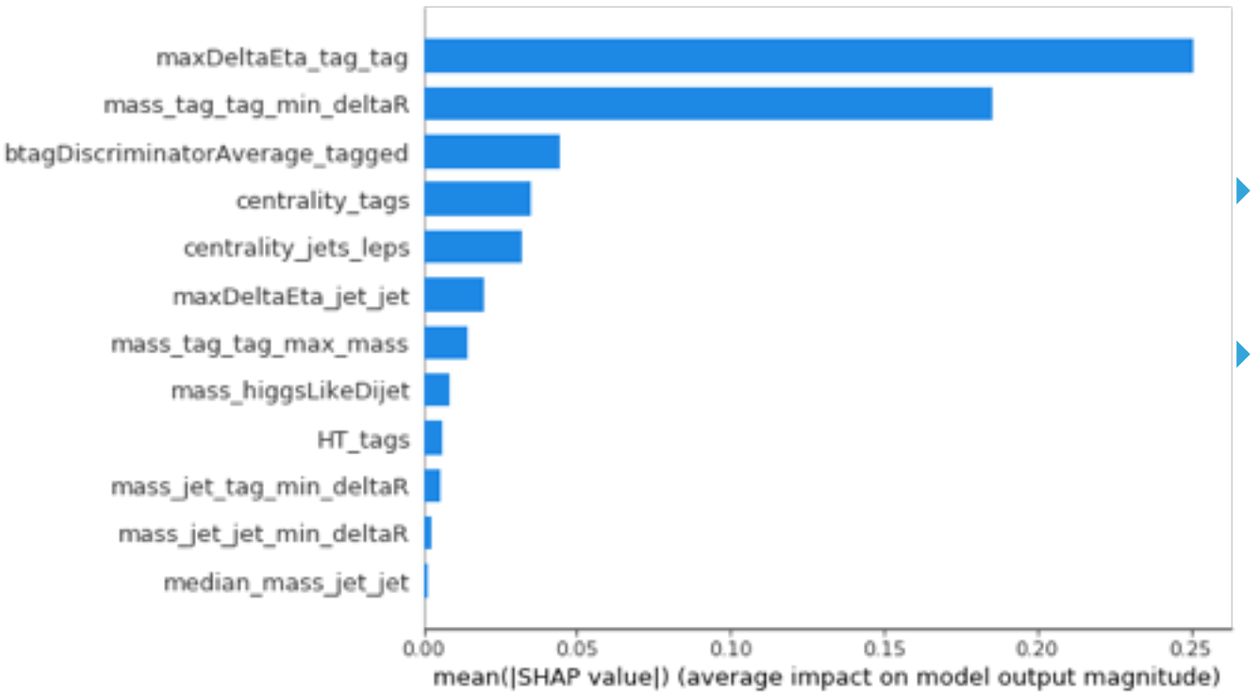



- Feature importance is often used to determine which features the models finds most valuable.
- XGboost provide out-of-the-box methods to determine the most important features

# SHAP MODEL EXPLANATION

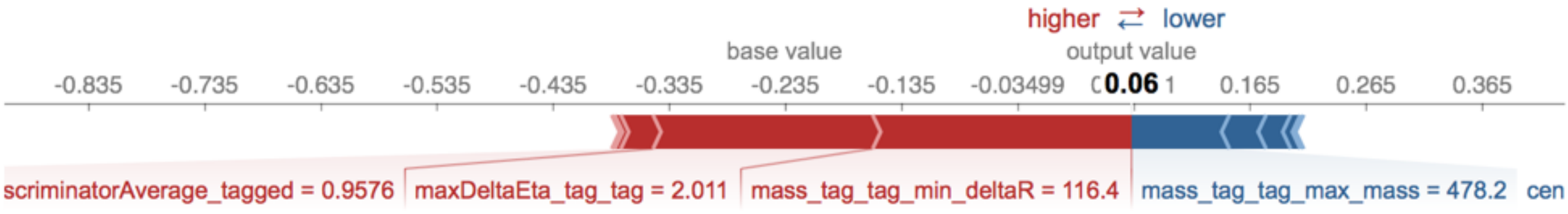



## Global interpretability

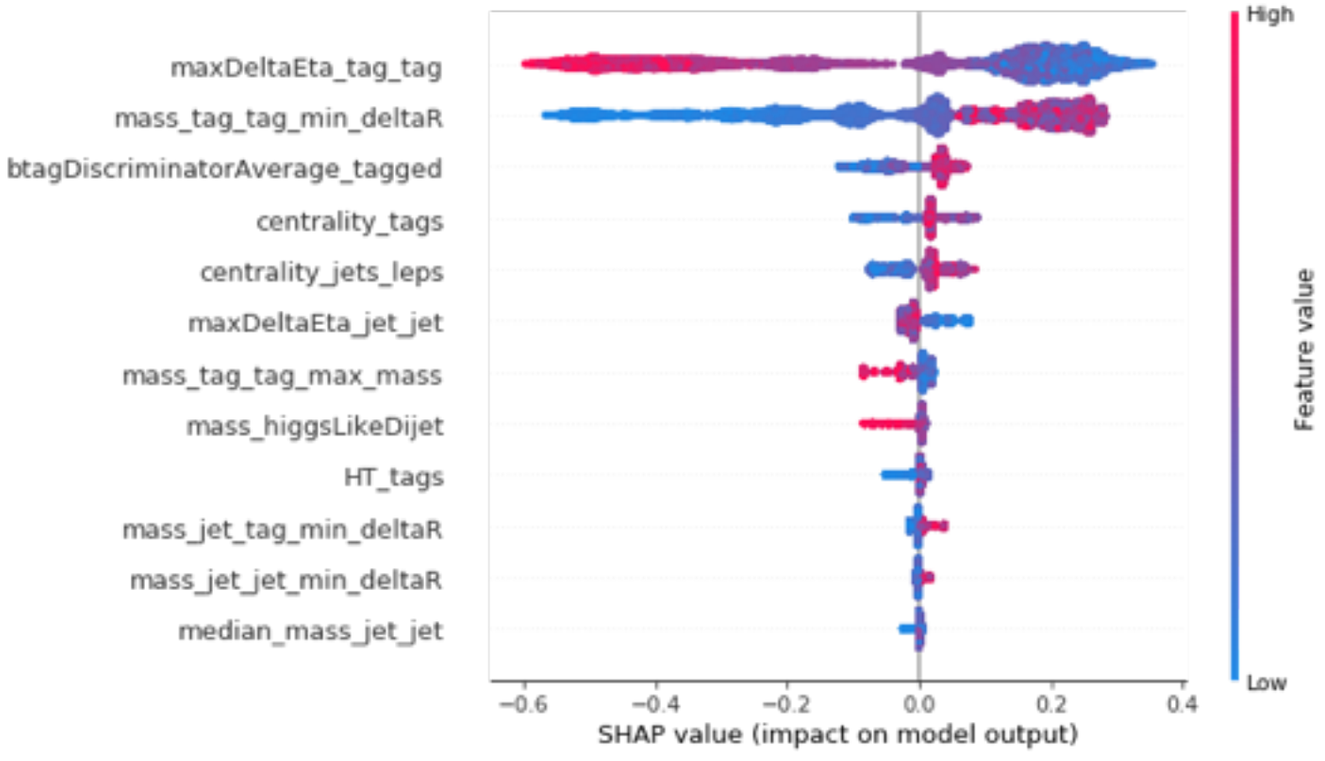
- Plot mean Shape values showing which features model finds valuable
- Features stored by the sum of SHAP value magnitudes over all samples

## Local interpretability

- The model predicted 0.06 signal against it being a background event
- Let's see why the models thinks so,
  - The **features in blue** push the predictions towards baseline value
  - While the opposite holds true for the **features in red**.



# SHAP EXAMPLE PLOTS




- Plot the SHAP values of every feature for every sample,  Shows features are most important for a model.

- SHAP dependence plot to show the effect of a single feature across the whole dataset.

- **Note:** Unlike traditional PDP, which show the average model output when changing a feature's value),these plots show interaction effects. Meaning other feature are impacting the importance of the plotted feature.