Witold Drzewakowski
XAI - hw 4, mimuw, 09.11.2023

Task A

In this task I explain the prediction of XGBoost (and a decision tree in the last part) of dataset `pc1` from openml, using SHAP values as implemented in shap package and dalex.

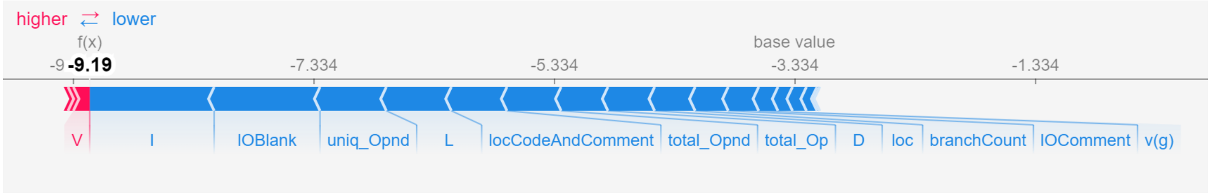
Two examples are selected:

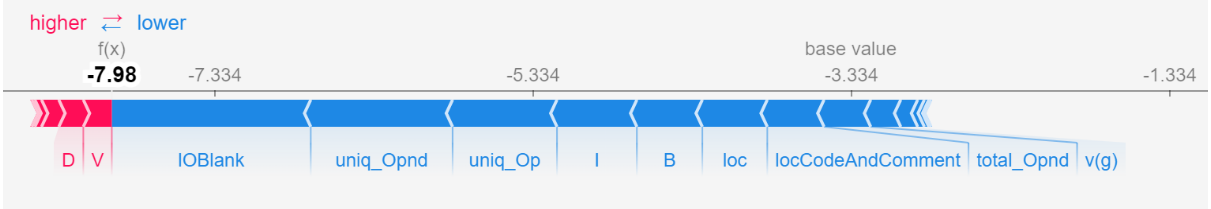
```
['loc', 'v(g)', 'ev(g)', 'iv(G)', 'N', 'V', 'L', 'D', 'I', 'E', 'B', 'T',
'lOCode', 'lOComment', 'locCodeAndComment', 'lOBlank', 'uniq_Op', 'uniq_Opnd',
'total_Op', 'total_Opnd', 'branchCount']
[-0.17965684  0.14622415 -0.30461524 -0.05804766 -0.04113199 -0.124238
  -0.65785186  1.44415841 -0.54233399 -0.05619656 -0.13163209 -0.0561963
  -0.16184783 -0.33849269 -0.27260169 -0.46398689  0.20572007 -0.35883074
  -0.08232116  0.01270306  0.18712682]
[-0.3919152  -0.16970835 -0.30461524  0.08642    -0.44068116 -0.37483328
   0.36514007 -0.59330894 -0.36281855 -0.16144188 -0.37070117 -0.1614421
  -0.38506885 -0.33849269 -0.27260169 -0.21769343 -0.64593743 -0.42401123
  -0.434575   -0.44506423 -0.1560938 ]
```

which are classified by XGBoost as

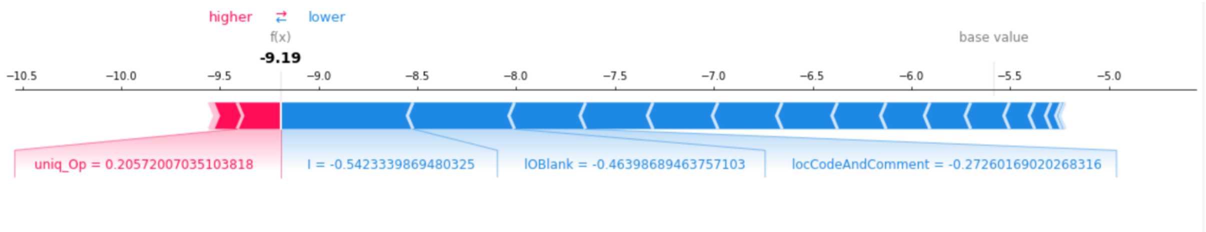
```
[9.9989825e-01 1.0177162e-04]
[9.9965680e-01 3.4319365e-04]
```

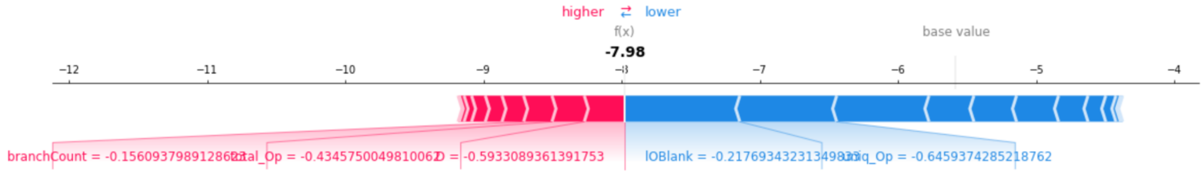
I calculate explanations on those examples using shap package and dalex, which are visualized below. For the explanations from shap package:





For the explanations from dalex:





I also print and manually inspect the pairs of most important features (indices) for 10 examples from two packages

shap package	dalex
[[15 8] [17 15] [14 13] [8 0] [15 14] [20 15] [8 15] [8 15] [0 15] [8 17]]	[[15 8] [16 15] [15 13] [0 18] [8 15] [8 15] [14 16] [5 18] [15 0] [8 17]]

We can see that, as discussed in the lecture, the importance of different features can obviously vary depending on an example.

By splitting the feature indices into those that have positive and negative contribution, we can see easily see that there is a lot of features such that the sign of their contribution flips from example to example:

```
ind, positive, negative
0 [ 3  5 16] [ 0  1  2  4  6  7  8  9 10 12 13 14 15 17 18 19 20]
1 [ 3  4  5  7  9 20] [ 0  1  2  6  8 10 12 13 14 15 16 17 18 19]
2 [ 0  1  2  3  5  7 12 15 16 17 20] [ 4  6  8  9 10 13 14 18 19]
3 [ 0  1  3  4  6  7  9 12 13 18 19 20] [ 2  5  8 10 14 15 16 17]
4 [ 1  3  5  8  9 15 17] [ 0  2  4  6  7 10 12 13 14 16 18 19 20]
```

For example the feature 16 - `uniq_Op` has this property (between 0th and 1st example).

It is also clear that the explanations provided by the two packages are different. Even though both methods calculate shap values, there might be differences in the approximation algorithm and data handling that lead to different results in some cases. Both packages can also differently approach the marginalization vs conditioning question.

In the last part of the exercise, I train a decision tree and explain its predictions using shap. I again print the most important features for the 10 first examples from test set:

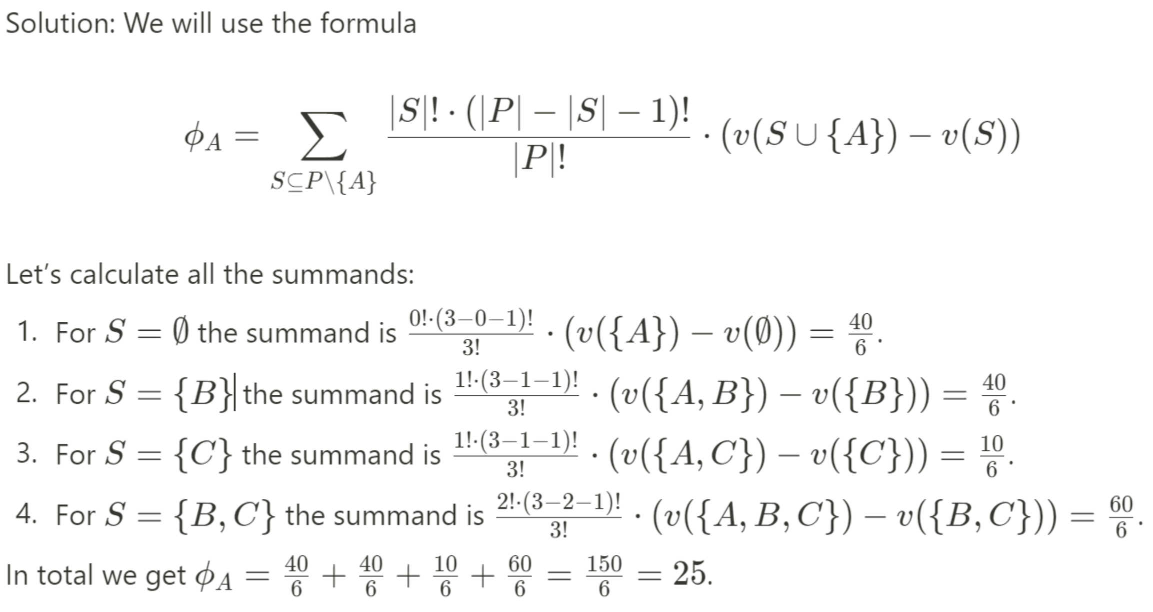
Decision Tree	XGBoost
`[ 8 15]`	`[[15  8]`
`[8 0]`	` [17 15]`
`[ 8 13]`	` [14 13]`
`[15  0]`	` [ 8  0]`
`[15  8]`	` [15 14]`
`[ 2 15]`	` [20 15]`
`[15  0]`	` [ 8 15]`
`[ 0 15]`	` [ 8 15]`
`[ 0 15]`	` [ 0 15]`
`[8 0]`	` [ 8 17]]`

Clearly they are different (examples 2st, 3rd, etc). Shap values depend on the predictions which vary since the models are different.

Task B

Calculate Shapley values for player A given the following value function

```
v() = 0
v(A) = 20
v(B) = 20
v(C) = 60
v(A,B) = 60
v(A,C) = 70
v(B,C) = 70
v(A,B,C) = 100
```



Solution: We will use the formula

$$\phi_A = \sum_{S \subseteq P \setminus \{A\}} \frac{|S|! \cdot (|P| - |S| - 1)!}{|P|!} \cdot (v(S \cup \{A\}) - v(S))$$

Let's calculate all the summands:

1. For $S = \emptyset$ the summand is $\frac{0! \cdot (3-0-1)!}{3!} \cdot (v(\{A\}) - v(\emptyset)) = \frac{40}{6}$.
2. For $S = \{B\}$ the summand is $\frac{1! \cdot (3-1-1)!}{3!} \cdot (v(\{A, B\}) - v(\{B\})) = \frac{40}{6}$.
3. For $S = \{C\}$ the summand is $\frac{1! \cdot (3-1-1)!}{3!} \cdot (v(\{A, C\}) - v(\{C\})) = \frac{10}{6}$.
4. For $S = \{B, C\}$ the summand is $\frac{2! \cdot (3-2-1)!}{3!} \cdot (v(\{A, B, C\}) - v(\{B, C\})) = \frac{60}{6}$.

In total we get $\phi_A = \frac{40}{6} + \frac{40}{6} + \frac{10}{6} + \frac{60}{6} = \frac{150}{6} = 25.$