Intro

Instructions

In our study, we aim to **investigate (i) how easy to interpret the visual explanations of bug predictions are, and (ii) how good automatic visual explanations generated with LIME and SHAP are.**

The survey will take 15-20 min to complete. It consists mainly of three parts:

1) Demographic: 3 questions are formulated related to your role, experience in software measurement, and working experience
2) Interpretability: One visual local-explanation for each instance obtained with Random Forest (presence and no presence of bugs) will be shown. Then you should analyze such explanations, select one and provide your own interpretation.
3) Explainability: you will be asked questions (formulated in two different scales) that are used to measure the ability of **LIME or SHAP** to provide explanations about bug predictions obtained with **Random Forest** (RF) algorithm.

Your survey responses will remain confidential and your name will not be given to any external parties. Your information will only be used for aggregate research purposes.

The research team
CITIUS-USC

Consent

By agreeing to participate in this study, you:
Confirm that you are of legal age.
Confirm that you have read and understood the preceding information.
You know that your participation is voluntary and anonymous.
You understand that you are free to leave the study at any time, without the need to

explain the reasons for your abandonment and without any consequences for you.
You agree to participate in the above study.
You consent that the information collected in this research may be shared, with the assurance of its anonymity, with other teams through collaborative research networks or repositories, for non-profit research purposes.

- ○ Agree
- ○ Do not agree

Personal experience

Which of the following best describes your current role?

- ○ Product development
- ○ Software architecture
- ○ Strategic management
- ○ Product planning
- ○ Quality assurance/software testing
- ○ End-user perspective
- ○ Operational management
- ○ Maintenance/evolution perspective
- ○ Legal perspective
- ○ External businness perspective
- ○ Software engineering researcher
- ○ [ ] Other (please indicate)

What is your current experience working with software measurement?

- ☐ No experience
- ☐ I have taught software metrics to my students, but using only toy examples
- ☐ I have learned software metrics, but using only toy examples
- ☐ I have supervised several academic projects with the usage of metrics as main asset
- ☐ I have been involved in various projects using software metrics to make decisions

How many years of working experience do you have?

- ○ 0-4 years
- ○ 5-9 years

- 10-14 years
- 15-19 years
- 20-24 years
- 25 years and over

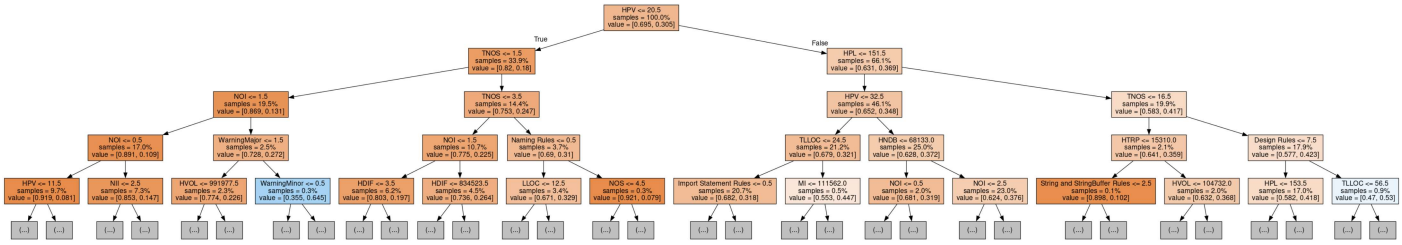
Case Description

Bug prediction using the BugHunter dataset

We use the public BugHunter dataset, which is a bug dataset that contains before/after fix states of source code elements at file, class, and method levels. The dataset includes 15 selected projects on GitHub, written in Java.

Using the **Random Forest (RF)** algorithm, we trained a model to predict future bugs (failures) based on **source code metrics at the method level**.
An example decision tree output by the RF model trained is shown below. As you can see the 4-level decision tree is not actually readable because of the large number of features. To address this issue, we have used two **explainability techniques** that can help to understand why an instance is classified as buggy or non-buggy.

In the next section, three examples of visual explanations applied to our prediction results are presented for your analysis. You should select just one of them.



Q2 LIME_pred

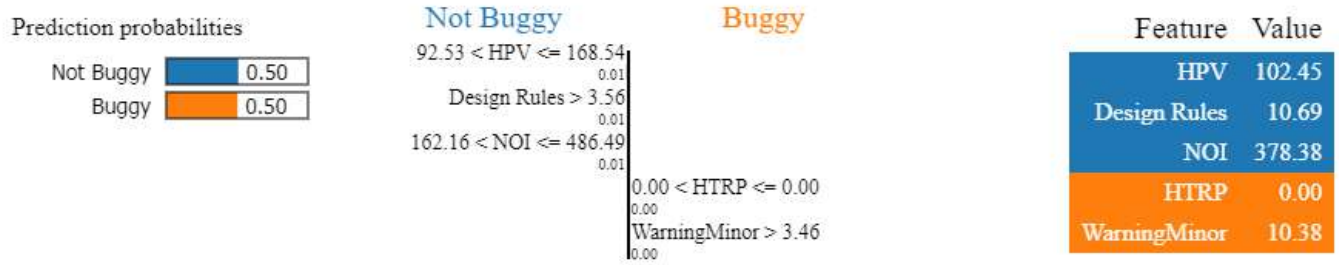
Examples of visual explanation for bug prediction results derived from the Random Forest algorithm

0: non-buggy

1: buggy

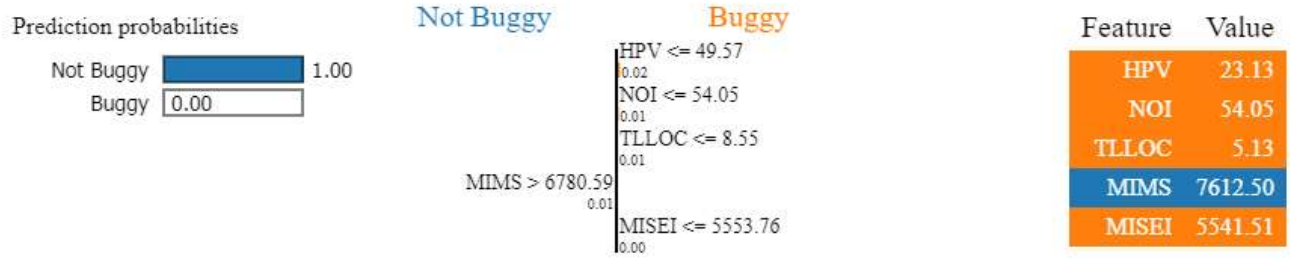
LIME EXPLANATION 1

```
Intercept 0.6319996305979336
Prediction_local [0.61778668]
Right: 0.503
```



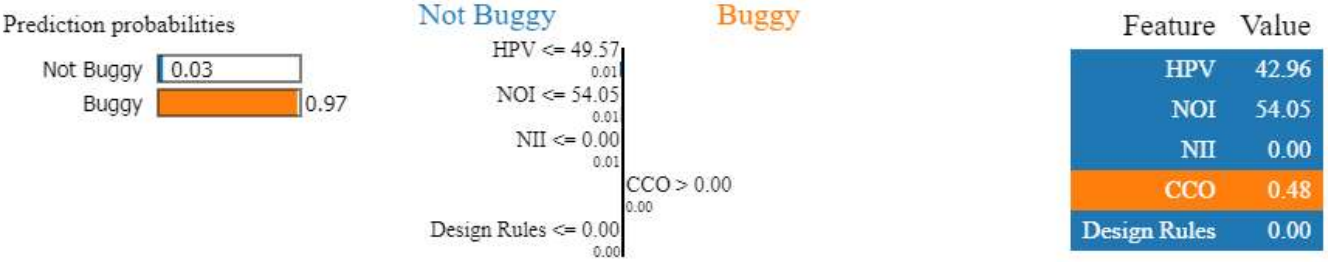
LIME EXPLANATION 2

```
Intercept 0.6215735416162742
Prediction_local [0.65190753]
Right: 1.0
```



LIME EXPLANATION 3

```
Intercept 0.381350631979124
Prediction_local [0.35403957]
Right: 0.966
```



SOURCE CODE METRICS: If you need a description of any metric, please click here

CLOC	Comment Lines of Code	HPL	Halstead Program Length
LOC	Lines of Code	HPV	Halstead Program Vocabulary
LLOC	Logical Lines of Code	HTRP	Halstead Time Required to Program
CD	Comment Density	HVOL	Halstead Volume
DLOC	Documentation Lines of Code	MIMS	Maintainability Index (Microsoft version)
NL	Nesting Level	MI	Maintainability Index (Original version)
NLE	Nesting Level Else-If	MISEI	Maintainability Index (SEI version)
NII	Number of Incoming Invocations	MISM	Maintainability Index (Source Meter version)
NOI	Number of Outgoing Invocations	NUMPAR	Number of Parameters
NOS	Number of Statements	TCD	Total Comment Density

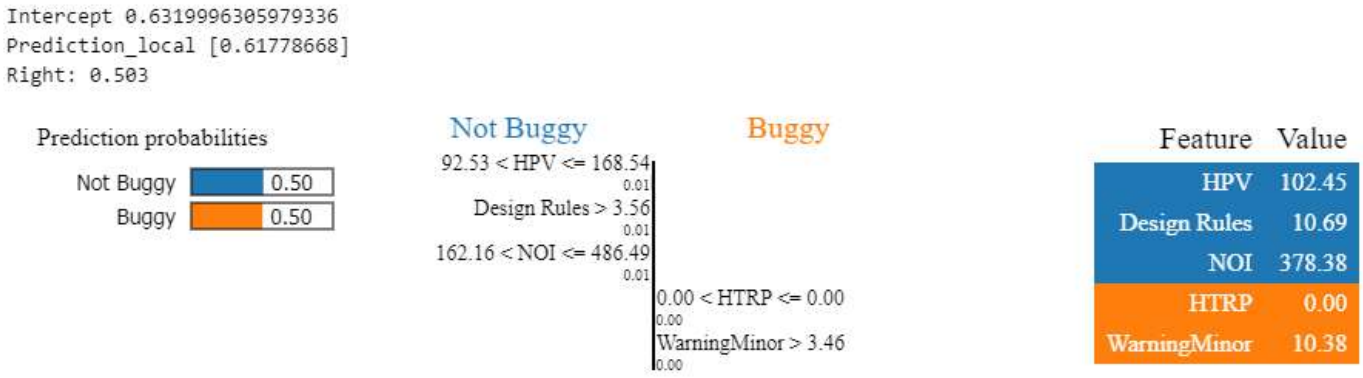
HCPL	Halstead Calculated Program Length	TCLOC	Total Comment Lines of Code
HDIF	Halstead Difficulty	TLOC	Total Lines of Code
HEFF	Halstead Effort	TLLOC	Total Logical Lines of Code
HNDB	Halstead Number of Delivered Bugs	TNOS	Total Number of Statements

Select the explanation which you may interpret and (probably) which can help you with making some decision

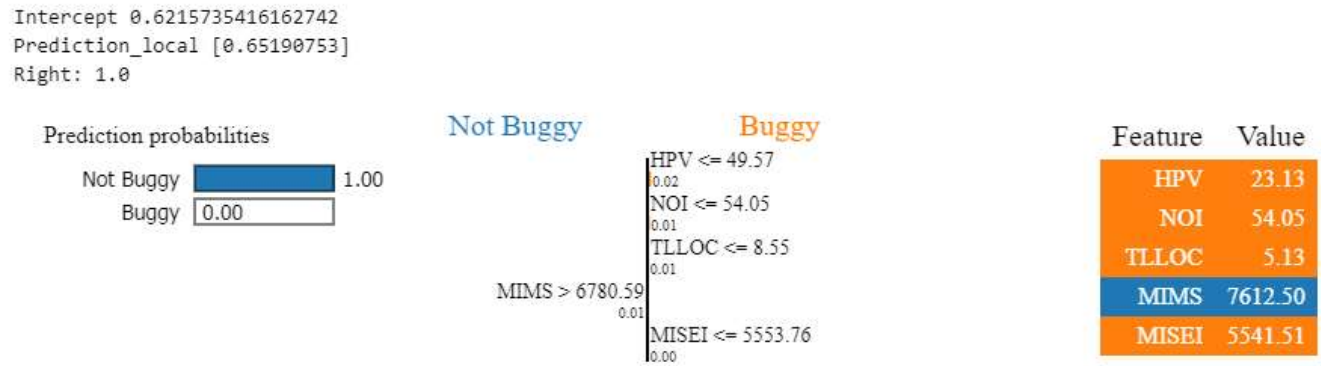
- LIME EXPLANATION 1
- LIME EXPLANATION 2
- LIME EXPLANATION 3

Please, write your own interpretation of the selected visual explanation for predicting presence or non-presence of bugs

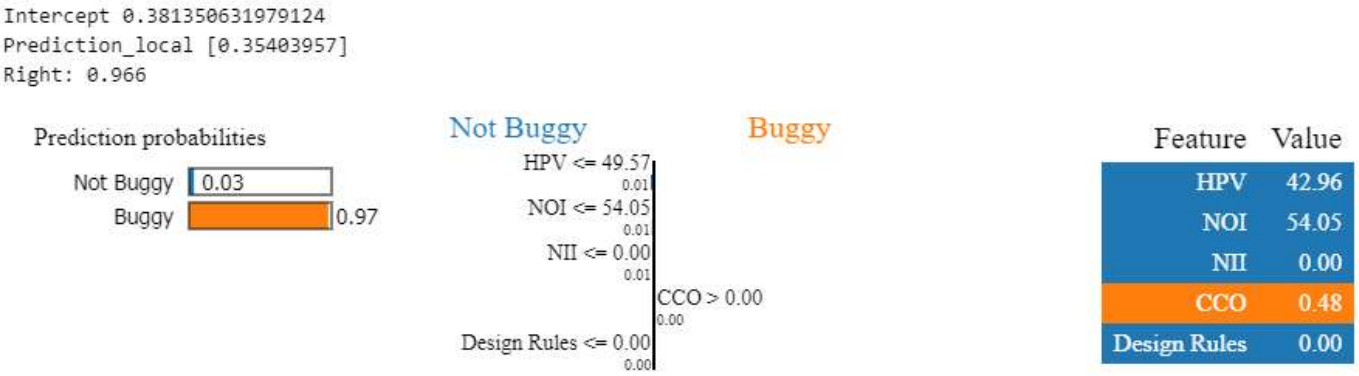
LIME EXPLANATION 1



LIME EXPLANATION 2



LIME EXPLANATION 3



Considering the selected visual explanation, for each item listed below, please indicate your extent of agreement:

	Strongly disagree	disagree	Neither agree nor disagree	Agree	Strongly agree
I understand why the model has classified a software code element (java method) as buggy/non-buggy.	○	○	○	○	○
I think the explanation provided for the reasons why java methods have been classified as buggy/non-buggy.	○	○	○	○	○
I think the visual presentation increased my trust in the prediction model.	○	○	○	○	○
I am satisfied with the visual explanation provided.	○	○	○	○	○

For each item listed below, response with yes or no

	Yes	No
The explanation helps me **understand** the result provided by the RF model.	○	○

	Yes	No
The explanation generated is **satisfying.**	○	○
The explanation is **sufficiently detailed.**	○	○
The explanation is **sufficiently complete.**	○	○
The explanation is **actionable**, that is, it helps me know how to use the results derived from the RF algorithm.	○	○
The explanation lets me know how **accurate or reliable** are the results derived from the RF model is.	○	○
The explanation lets me know how **trustworthy** are the results derived from the RF model is.	○	○

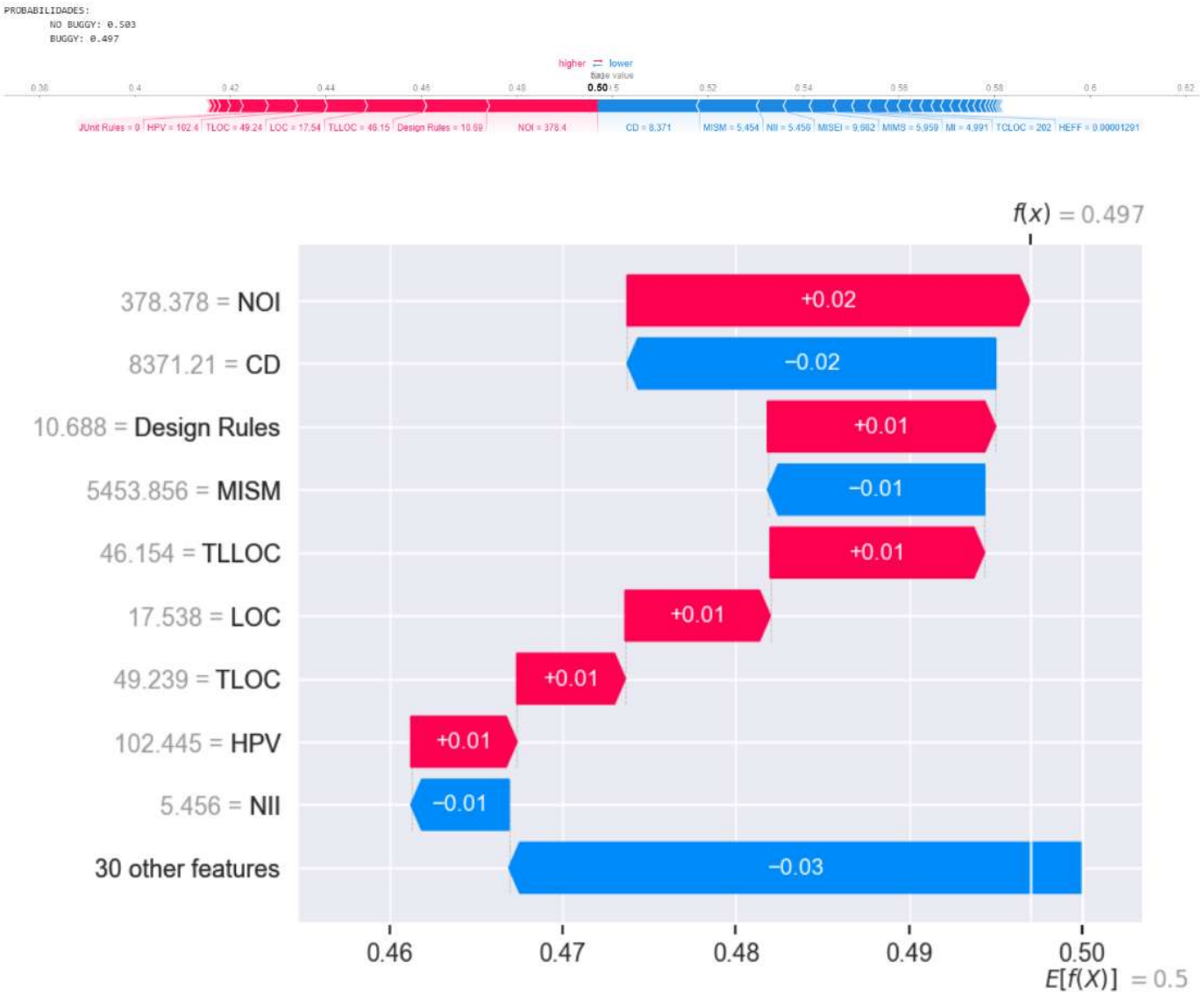
What are your thoughts on the explanations generated with LIME? How do you think it could be improved?

Q4 SHAP_pred

Examples of visual explanation for bug prediction results derived from the Random Forest algorithm

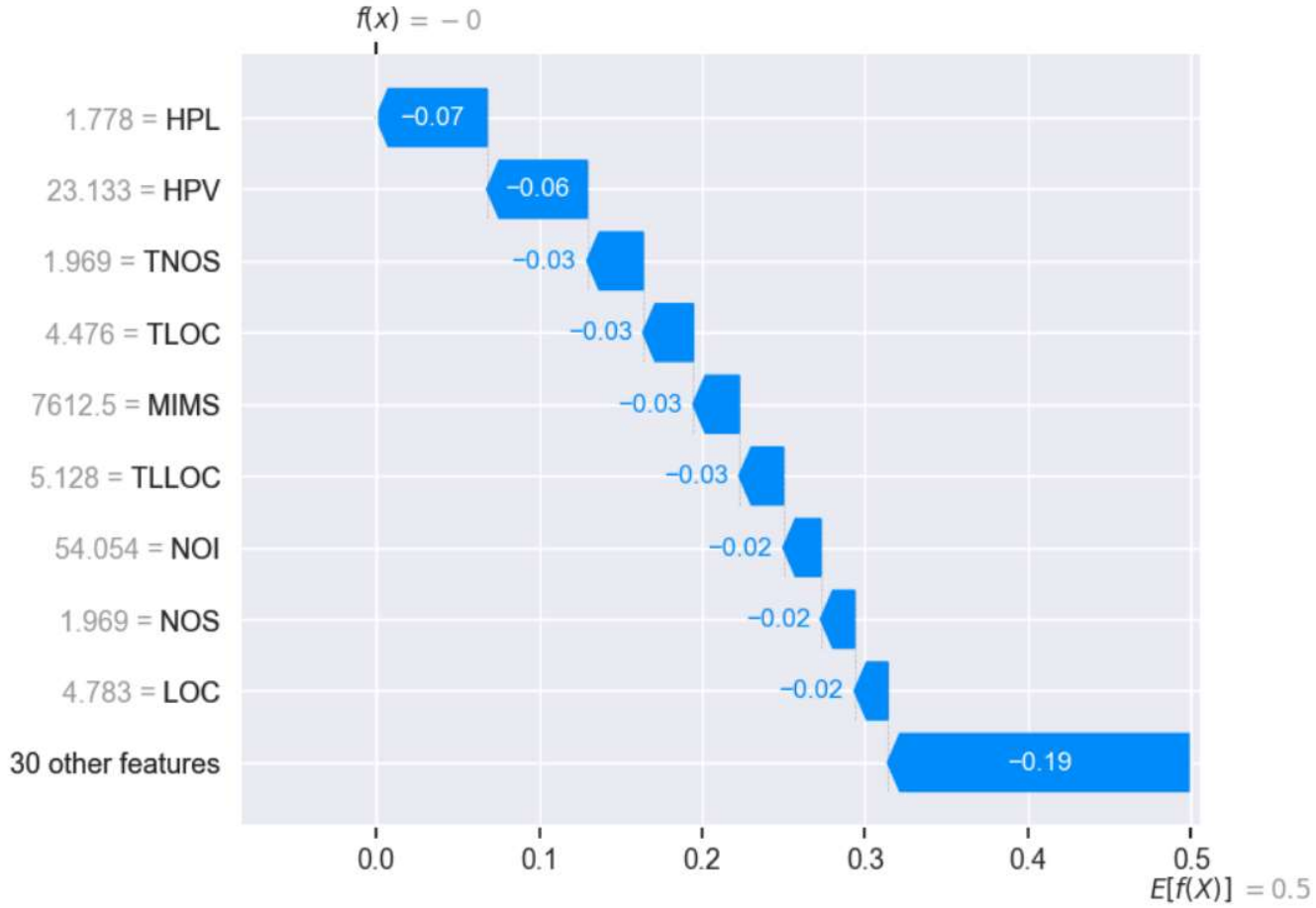
SHAP LOCAL EXPLANATION 1

Probabilities: No buggy 0.503; buggy 0.497



SHAP LOCAL EXPLANATION 2

Probabilities: No buggy 1; buggy 0



SHAP LOCAL EXPLANATION 3

Probabilities: No buggy 0.034; buggy 0.966

SOURCE CODE METRICS: If you need a description of any metric, please click here

CLOC	Comment Lines of Code	HPL	Halstead Program Length
LOC	Lines of Code	HPV	Halstead Program Vocabulary
LLOC	Logical Lines of Code	HTRP	Halstead Time Required to Program
CD	Comment Density	HVOL	Halstead Volume
DLOC	Documentation Lines of Code	MIMS	Maintainability Index (Microsoft version)
NL	Nesting Level	MI	Maintainability Index (Original version)
NLE	Nesting Level Else-If	MISEI	Maintainability Index (SEI version)
NII	Number of Incoming Invocations	MISM	Maintainability Index (Source Meter version)
NOI	Number of Outgoing Invocations	NUMPAR	Number of Parameters
NOS	Number of Statements	TCD	Total Comment Density
HCPL	Halstead Calculated Program Length	TCLOC	Total Comment Lines of Code
HDIF	Halstead Difficulty	TLOC	Total Lines of Code
HEFF	Halstead Effort	TLLOC	Total Logical Lines of Code
HNDB	Halstead Number of Delivered Bugs	TNOS	Total Number of Statements

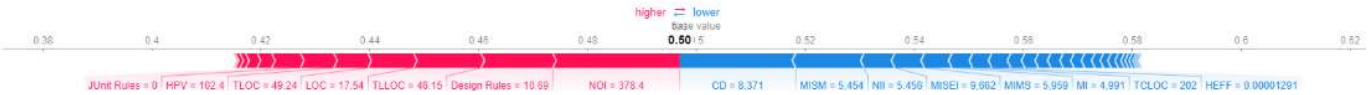
Select one of the 3 explanations which you may interpret and (probably) which can help you with making some decision

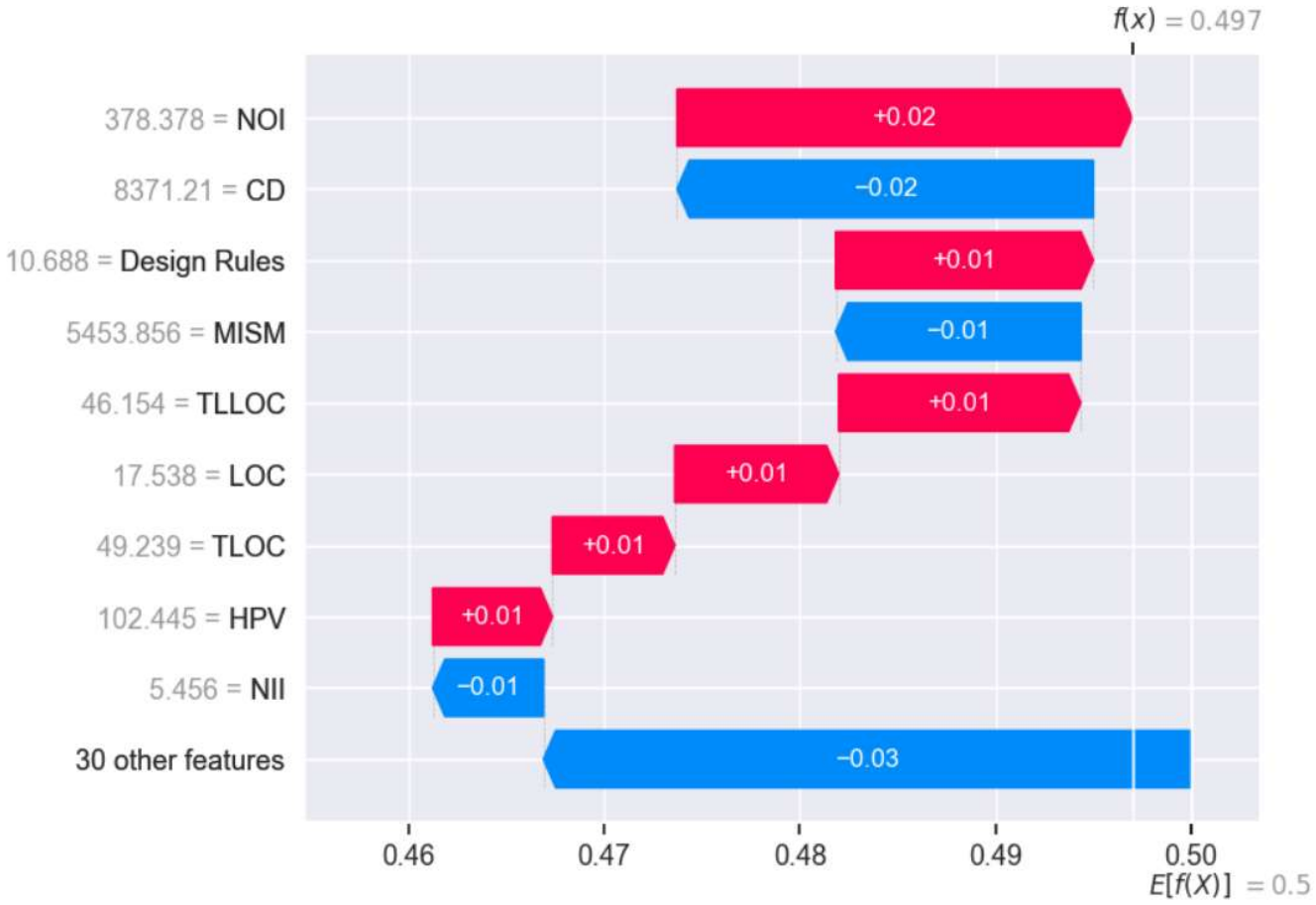
- ○ SHAP EXPLANATION 2
- ○ SHAP EXPLANATION 1
- ○ SHAP EXPLANATION 3

Please, write your own interpretation of the **selected visual explanation** provided for predicting presence or non presence of bugs

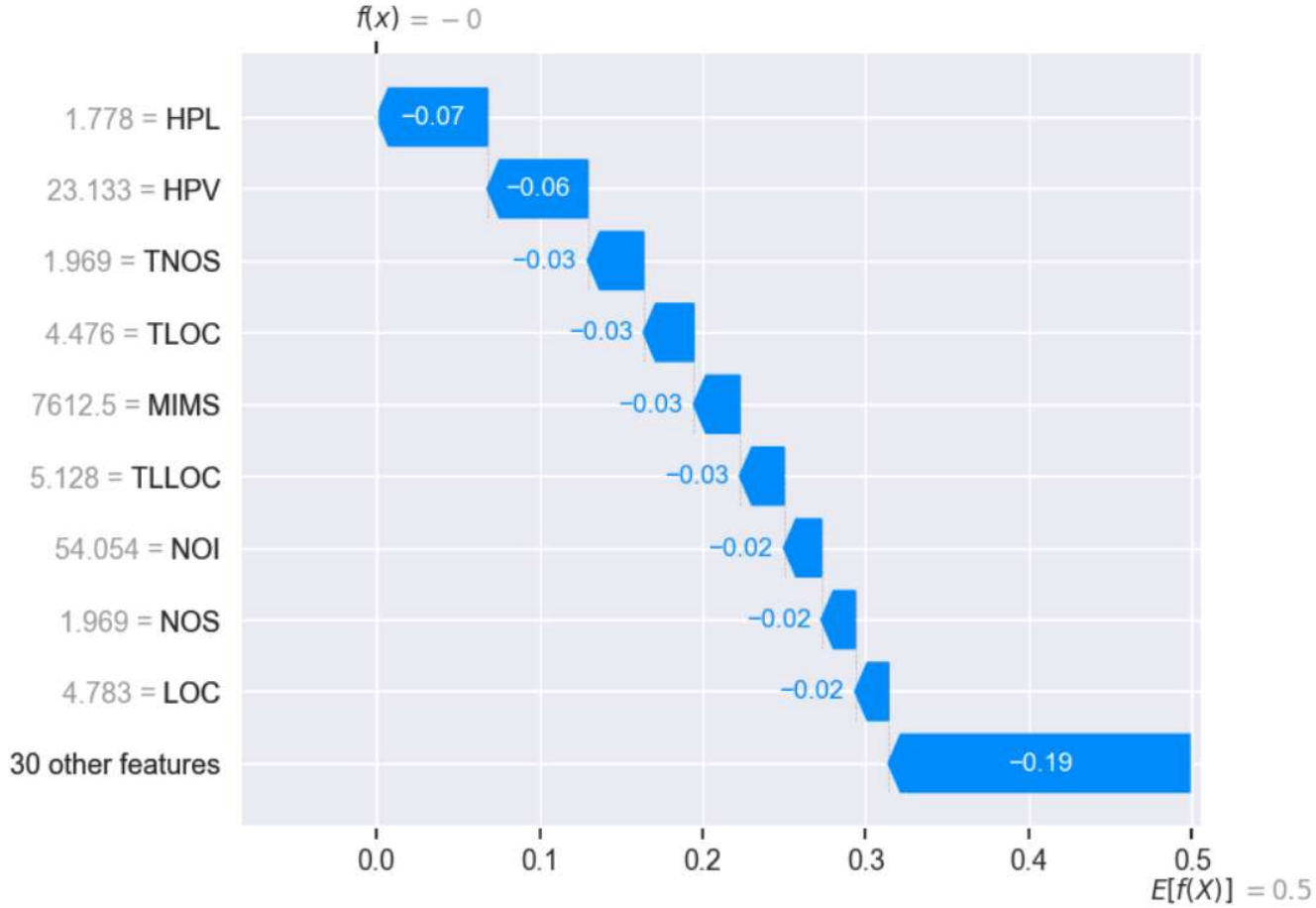
SHAP LOCAL EXPLANATION 1

```
PROBABILIDADES:
        NO BUGGY: 0.503
        BUGGY: 0.497
```

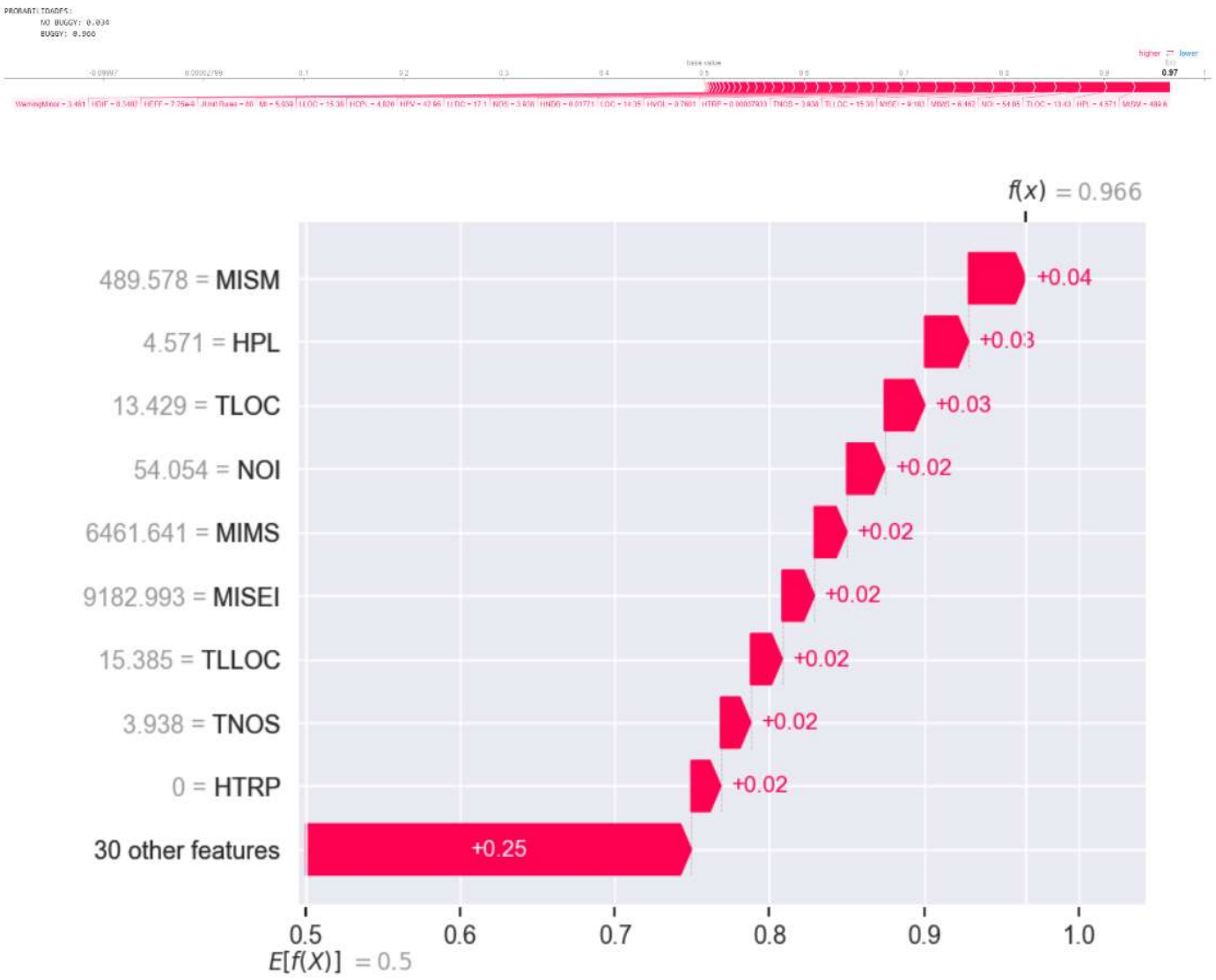




SHAP LOCAL EXPLANATION 2



SHAP LOCAL EXPLANATION 3



Considering the selected visual explanation generated with SHAP, for each item listed below, please indicate your extent of agreement:

	Strongly disagree	disagree	Neither agree nor disagree	Agree	Strongly agree
I understand why the model has classified a software code element (java method) as buggy/non-buggy.	○	○	○	○	○
I think the explanation provided for the reasons why java methods have been classified as buggy/non-buggy.	○	○	○	○	○
I think the visual presentation increased my trust in the prediction model	○	○	○	○	○

	Strongly disagree	disagree	Neither agree nor disagree	Agree	Strongly agree
I am satisfied with the explanation provided.	○	○	○	○	○

For each item listed below, response with yes or no

	Yes	No
The explanation helps me **understand** the result provided by the RF model.	○	○
The explanation generated is **satisfying.**	○	○
The explanation is **sufficiently detailed.**	○	○
The explanation is **sufficiently complete.**	○	○
The explanation is **actionable**, that is, it helps me know how to use the results derived from the RF algorithm.	○	○
The explanation lets me know how **accurate or reliable** are the results derived from the RF model is.	○	○
The explanation lets me know how **trustworthy** are the results derived from the RF model is.	○	○

What are your thoughts on the explanations generated with SHAP? How do you think could be improved?