Consortium for Software Engineering Research

2021 Spring Meeting

Software Defect Prediction for Imbalanced Class with Applying Feed Forward Neural Network and Other Techniques

Susmita Haldar

Ph.D. student - University of Western Ontario

Professor - Fanshawe College, London Ontario

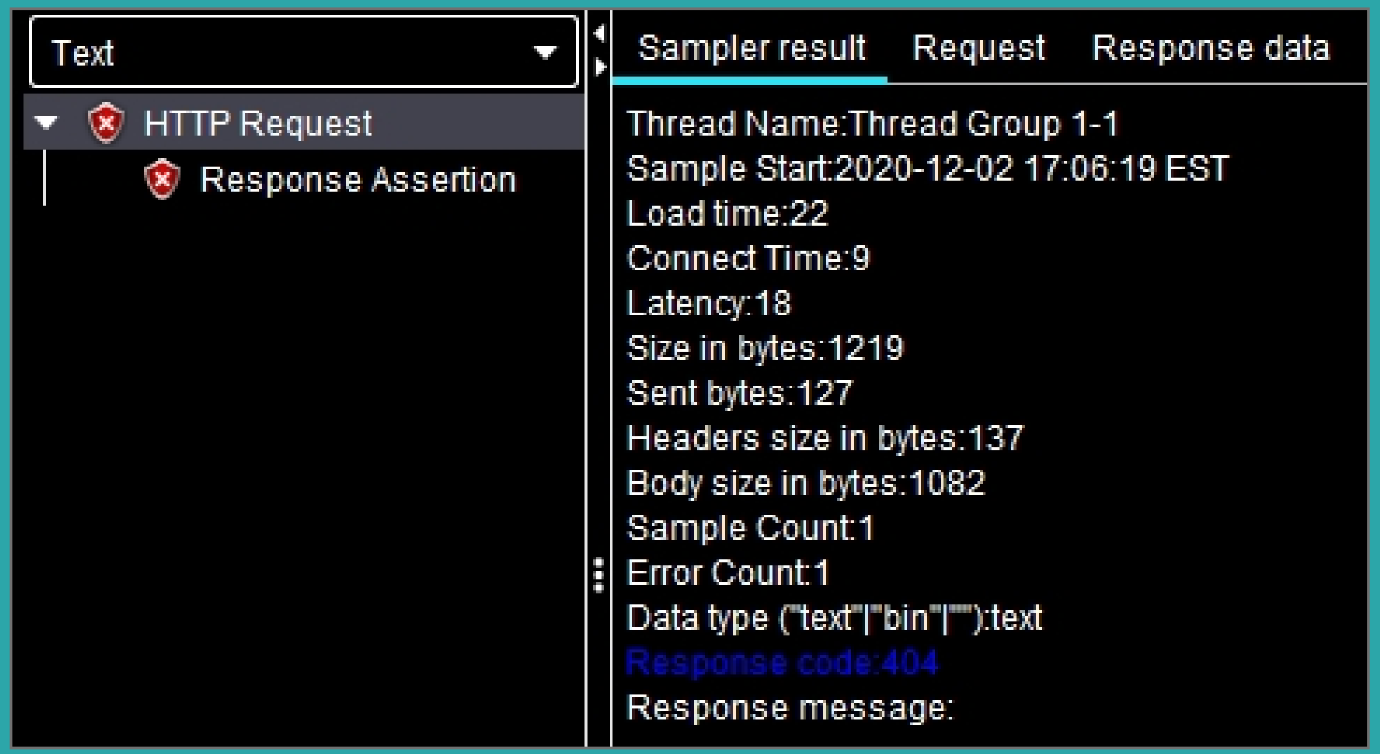
Supervisor: Dr. Luiz Capretz

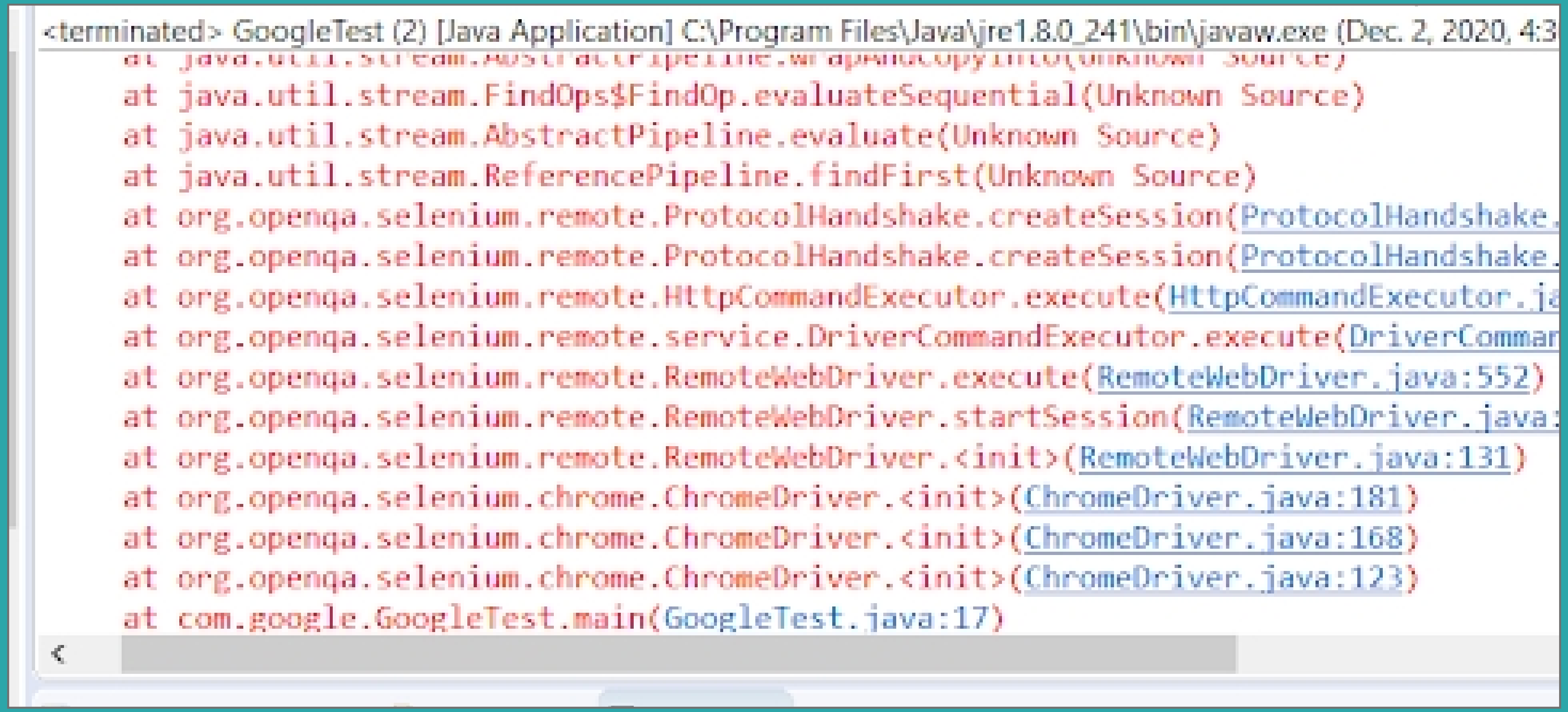
Date of presentation - May 14th, 2021

Agenda

- Defect prediction overview
- Problem description and Literature Survey
- Imbalanced dataset challenges
- Methodology
- Explainable AI using LIME

Defects overview





Decrease in ↓

Customer satisfaction
Market share, test coverage

↑ Increase in

Operating cost

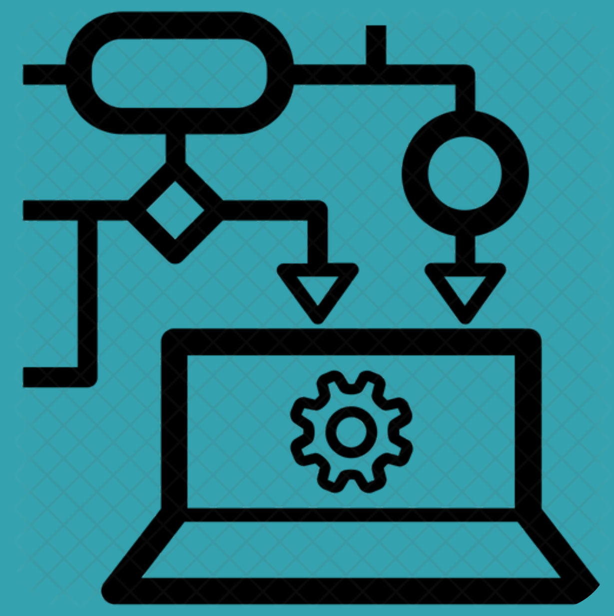
Delay in schedule

What is defect prediction?

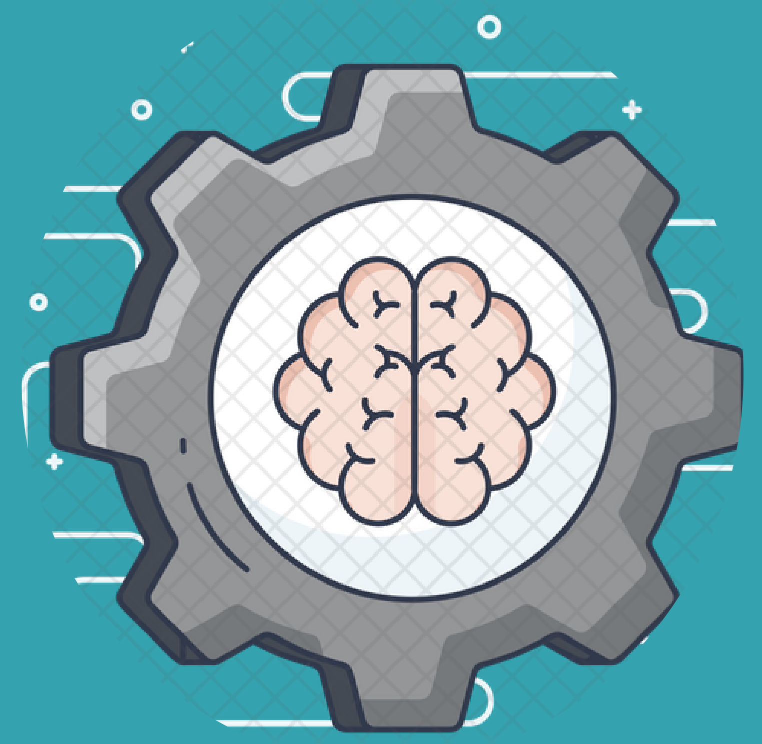
List of software modules, and past performance statistics

Source code measures (past history)					
Module No	Feature1 (LOC)	Feature2 (Difficulty)	...	FeatureN (Cycolmatic complexity)	Defect?
1	100	200		40	Yes
2	500	4		30	Yes
..	..	..	..	..	No
M	5	3		5	No

Machine learning algorithm

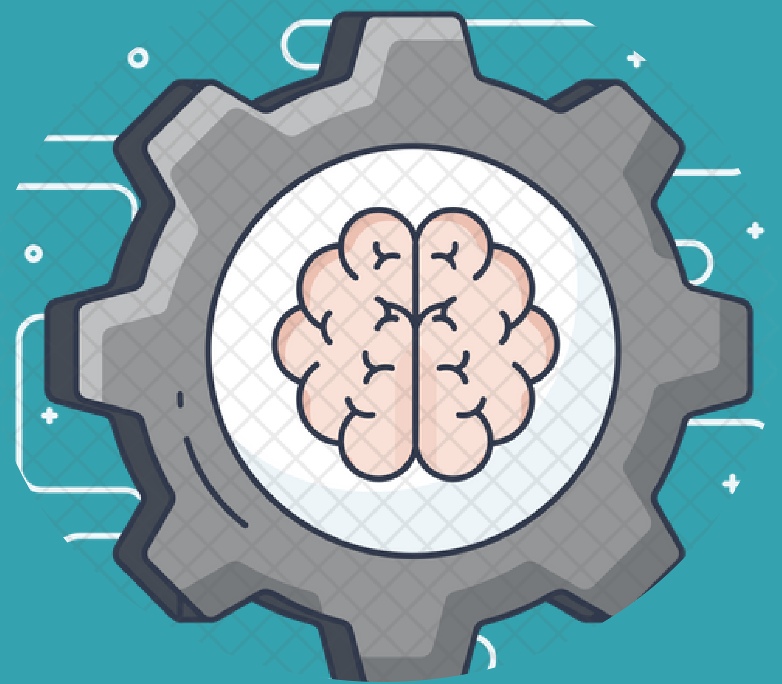


Defect Prediction model

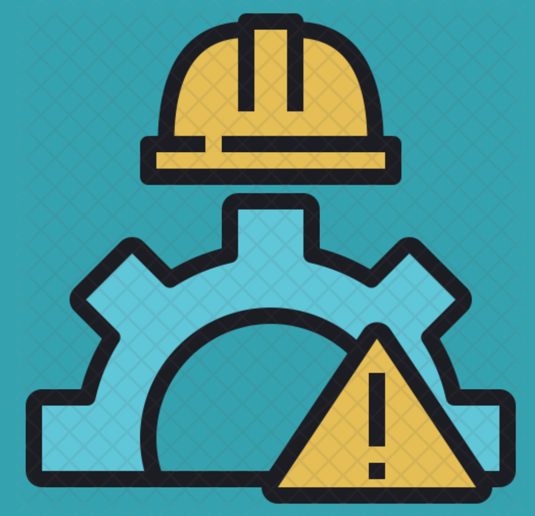


New software modules

Defect Prediction model



Prediction of faulty or non-faulty module



Use cases and challenges of defect prediction

Use Cases

- Software test planning
- Automated testing need identification
- Continuous testing
- Software reliability assessment
- Software maintenance

Challenges

- Imbalanced datasets
- Limited documentation
- Lacking explanation of algorithm
- Applying generic model
- Availability of publicly available datasets

Literature Review

Paper	Techniques Applied
1. Aleem et al.	Weka tool Binning data Missing values filled by means of attributes Comparative performance analysis.
2. Shepperd et. al.	Focused on data cleaning strategy due to having noisy data.
3. T Menzies et. al.	Defect prediction analysis based on identifying probability of failure and probability of false alarms.

Dataset description

NASA Metrics Data Program defect data sets
http://promise.site.uottawa.ca/SERepository

File	No. of observations	No. of features	Prog. Language	Defect ratio
cm1	498	22	C	9.80%
kc2	522	22	C++	28%
pc1	1109	22	C	7.30%
kc1	2109	22	C++	26%
jm1	10885	22	C	19.30%

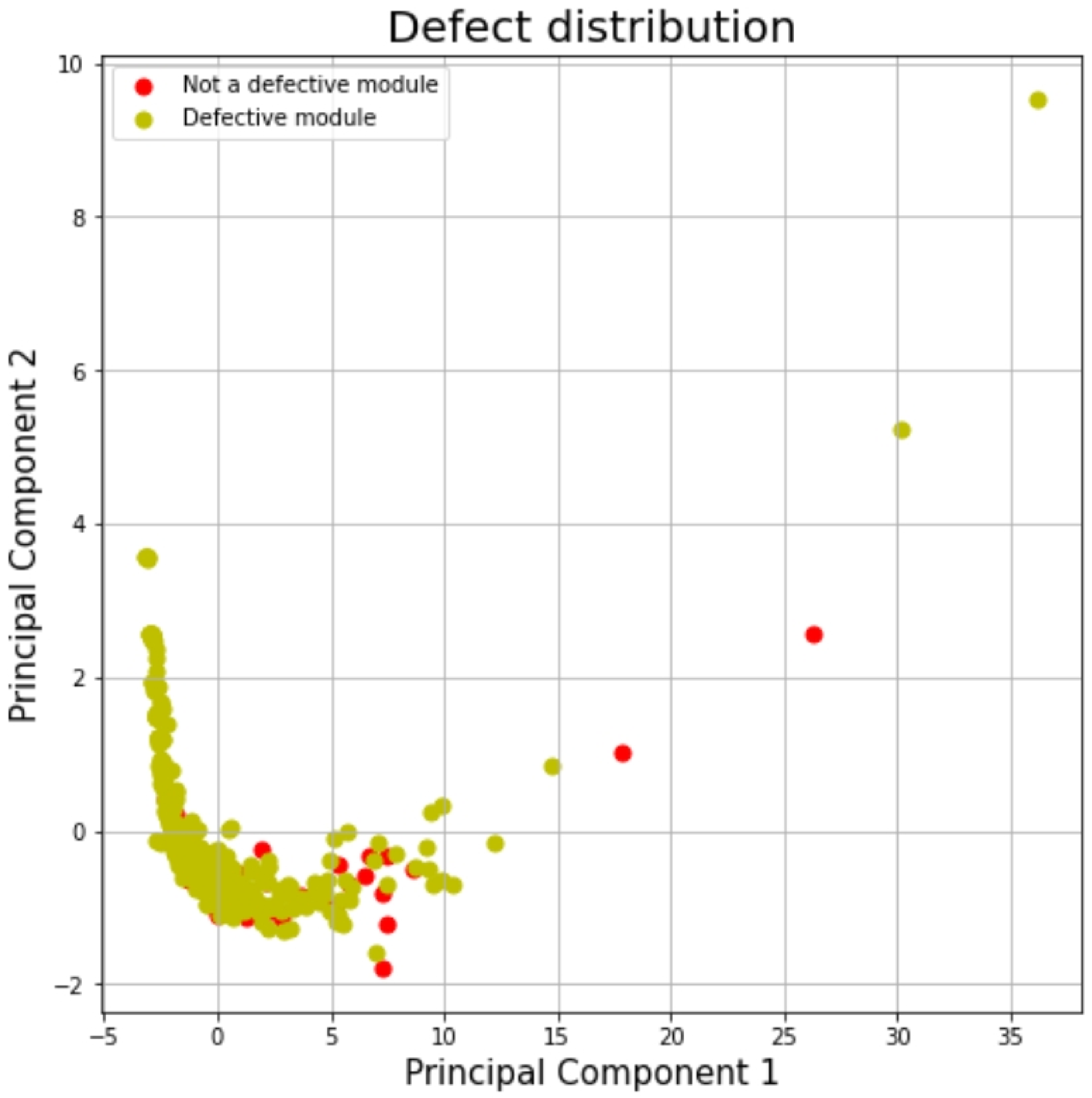
Figure 1: Dataset description

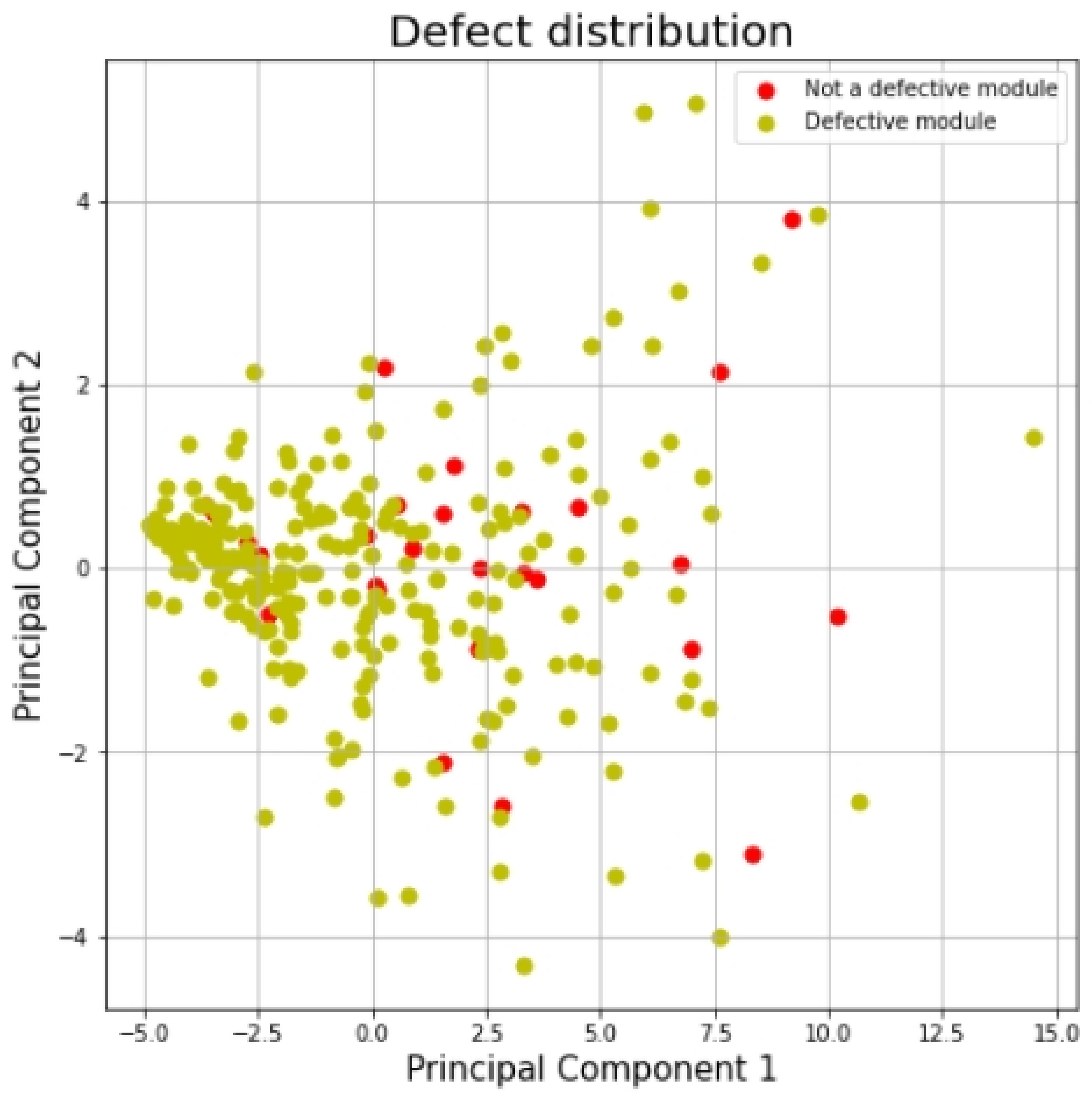
Dataset description

Feature Sequence	Feature name	Description	Type
0	loc	McCabe's line count of code	float64
1	v(g)	McCabe's cyclomatic complexity	float65
2	ev(g)	McCabe's essential coplexity	float66
3	iv(g)	McCabe's design complexity	float67
4	n	Halstead total operators + operands	float68
5	v	Halstead volume	float69
6	l	Halstead program length	float70
7	d	Halstead program difficulty	float71
8	i	Halstead intelligence	float72
9	e	Halstead effort	float73
10	b	Number of delivered bugs (from Halstead metrics)	float74
11	t	Halstead's time estimator	float75
12	loCode	Halstead's line count	float76
13	loComment	Halstead's count of lines of comments	float77
14	loBlank	Halstead's count of blank lines	float78
15	locCodeAndComment	Line of code and comment	float79
16	uniq_Op	Unique operators	float80
17	uniq_Opnd	Unique operands	float81
18	total_Op	Total operators	float82
19	total_Opnd	Total operands	float83
20	branchCount	Total branches in program	float84
21	defects	{False, true}: module has no/has reported defects	Category

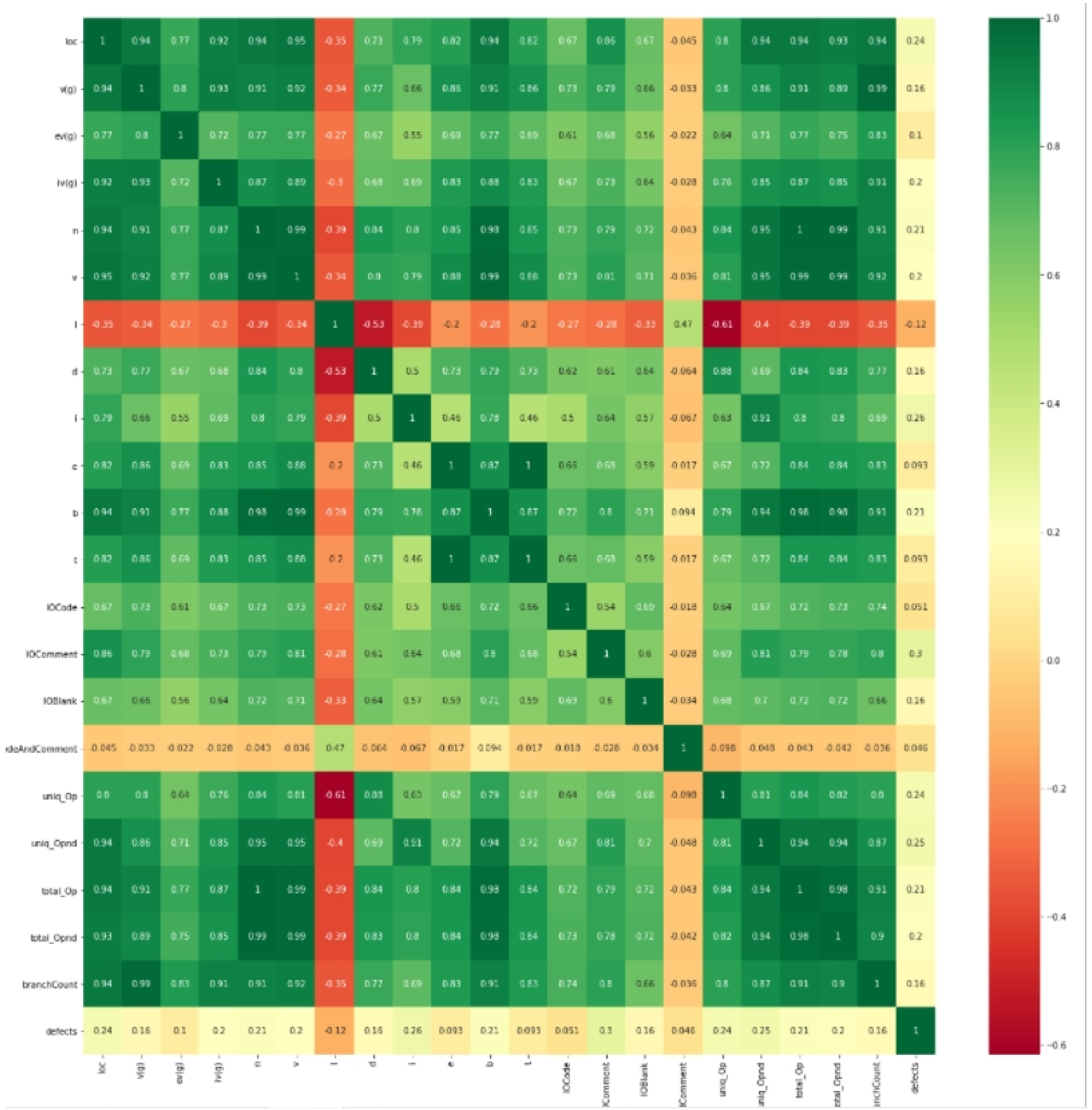
Figure 2: Description of the dataset attributes

Feature Engineering



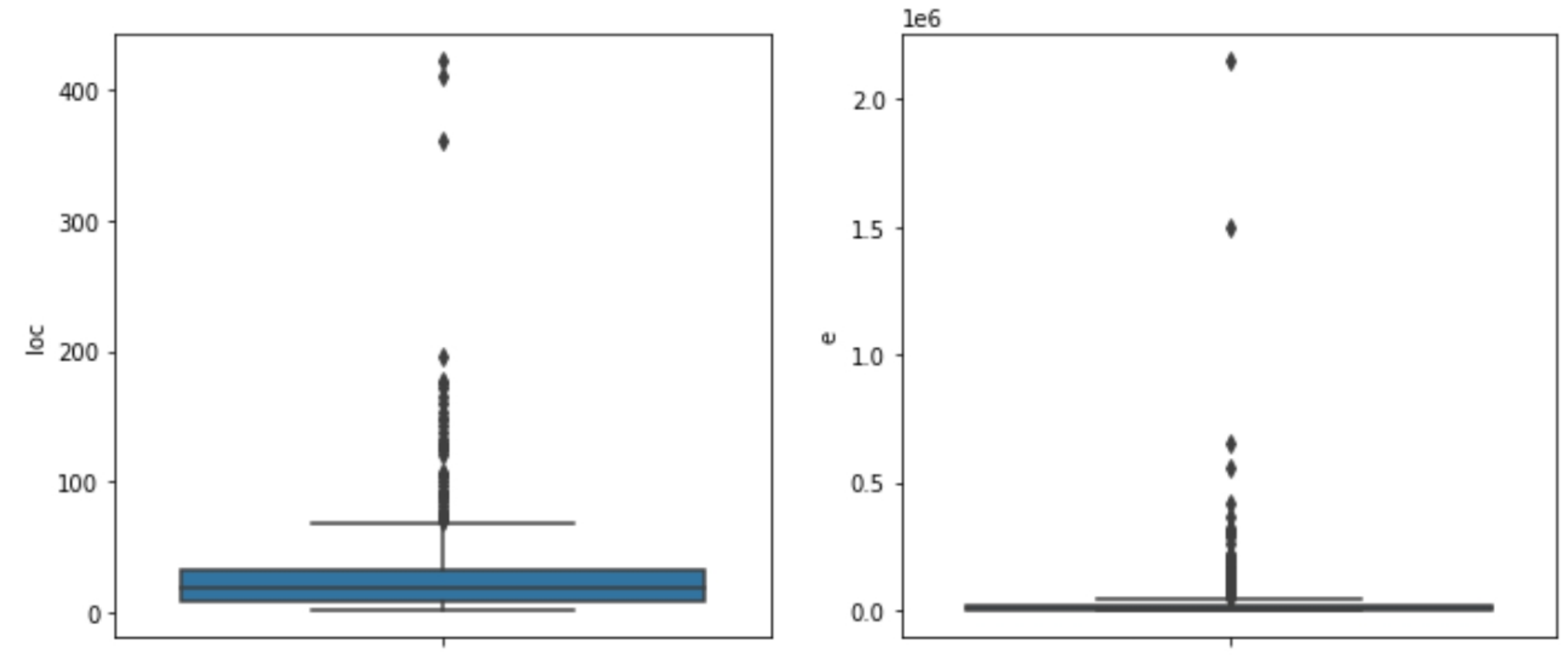


Feature Engineering

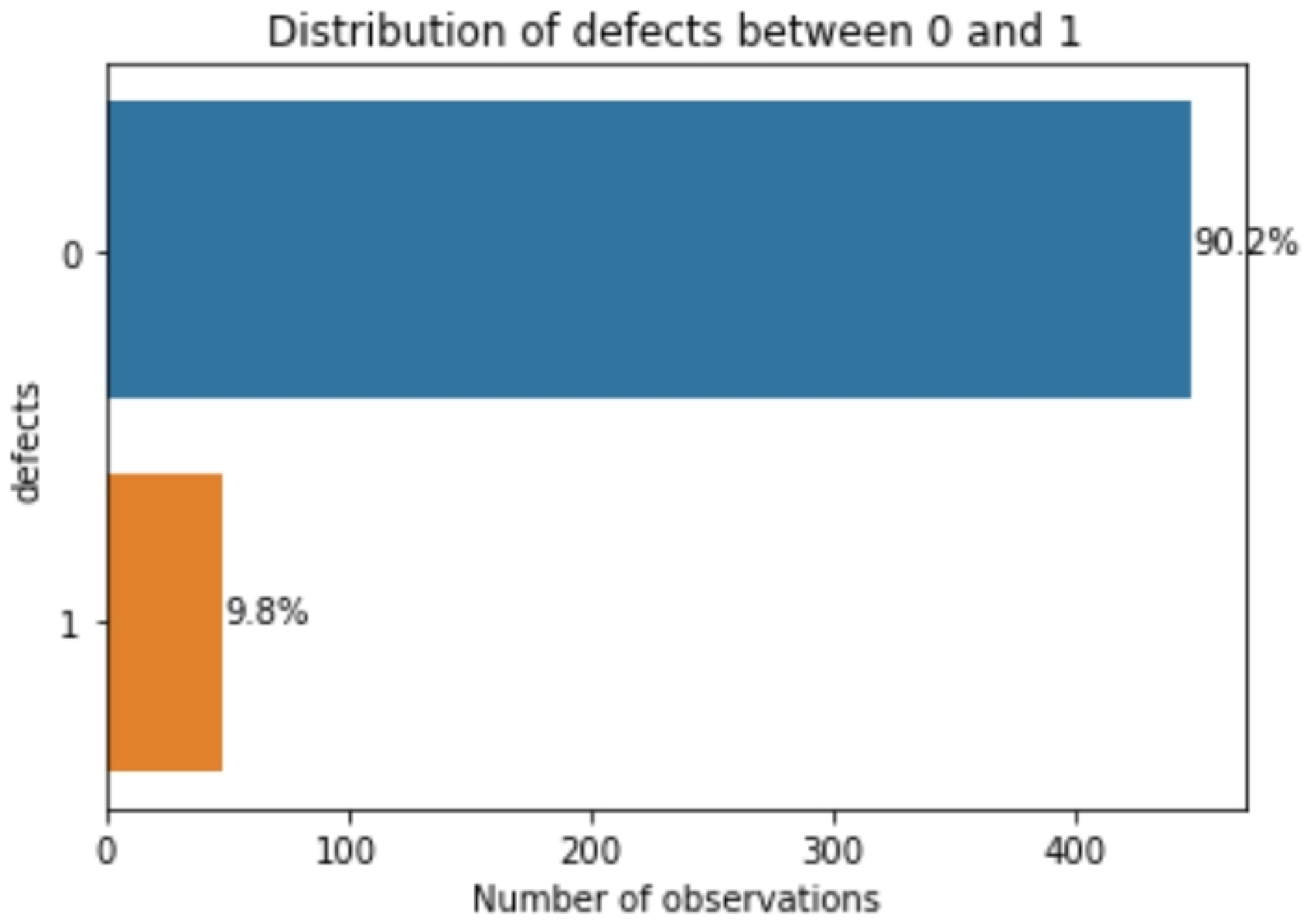


Feature Engineering

Criterion	Data Quality Category	Explanation
1	Cases with missing values	Instances that contain one or more missing observations were removed
2	Cases with conflicting features values	Instances that have 2 or more metric values that violate some referential integrity constraint were removed. The conditions are: 1) Total line of code (LOC) is less than Commented LOC since commented LOC should be a subset of LOC. 2) N is not equal to total number of operators and operands. 3) The program cyclomatic complexity is greater than total operators plus 1 since this situation is not possible.
3	Outlier removals	Outliers rely on any value that lies within the range of Q1-1.5*15 or outside the range of (Q3 + 1.5 * IQR)) where Q1 and Q3 corresponds to first and 3rd quartile.
4	Removal of duplicates	The duplicated values were removed.



Class Imbalance and SMOTE



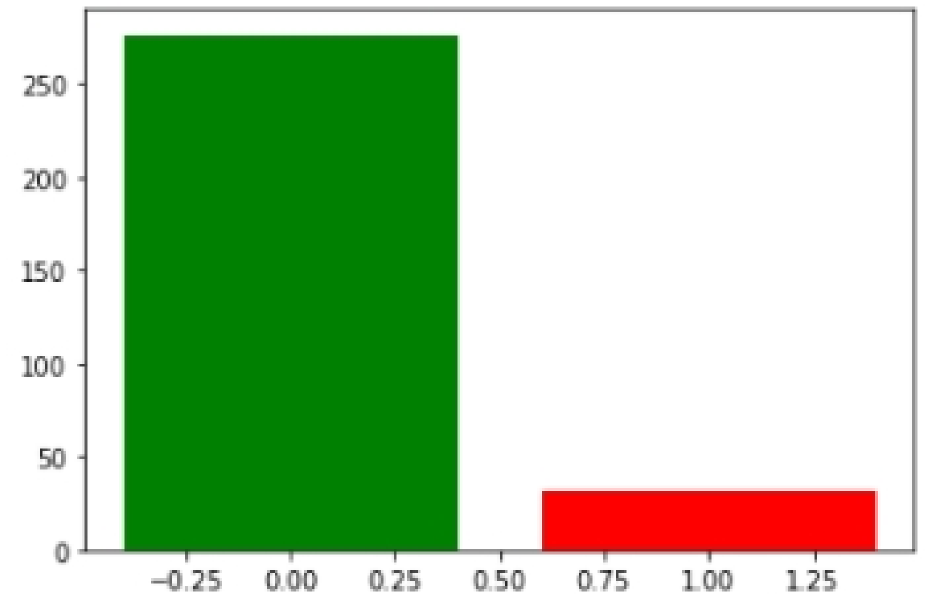
<Figure size 432x288 with 0 Axes>

SMOTE - Synthetic minority oversampling technique [Chawla et. al]

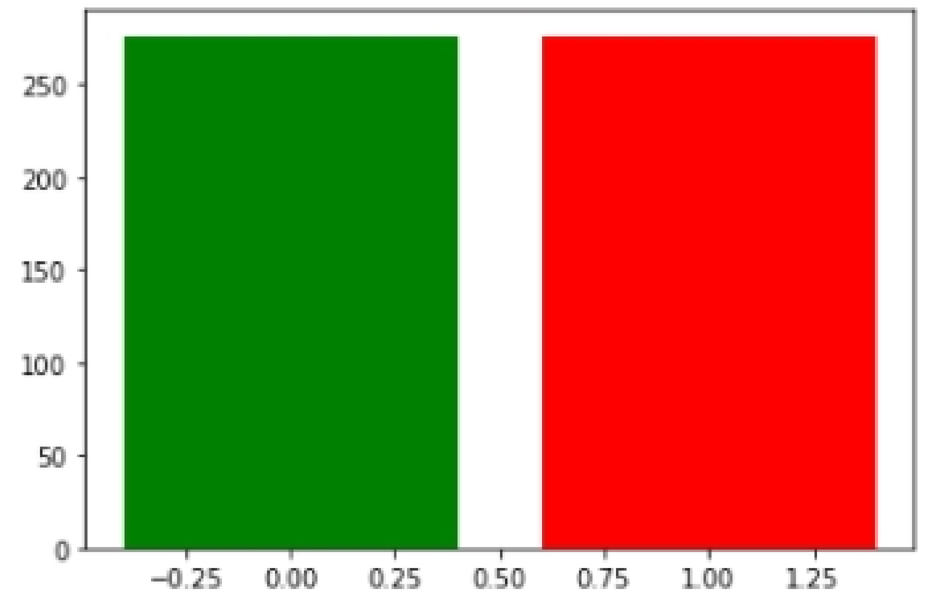
Combination of
.Oversampling of minority class
. Random underdamping of the majority class

Balancing data with SMOTE

```
Before applying SMOTE, counts of label '1' was: 32
Before applying SMOTE, counts of label '0' was: 276
```




```
After applying SMOTE, counts of label '1': 276
after applying SMOTE, counts of label '0': 276
```



Naïve Bayes algorithm

t is a classification technique based on Bayes' Theorem with an assumption of independence among predictors:

P(c|x) = (P(x|c) * P(c)) / P(x)

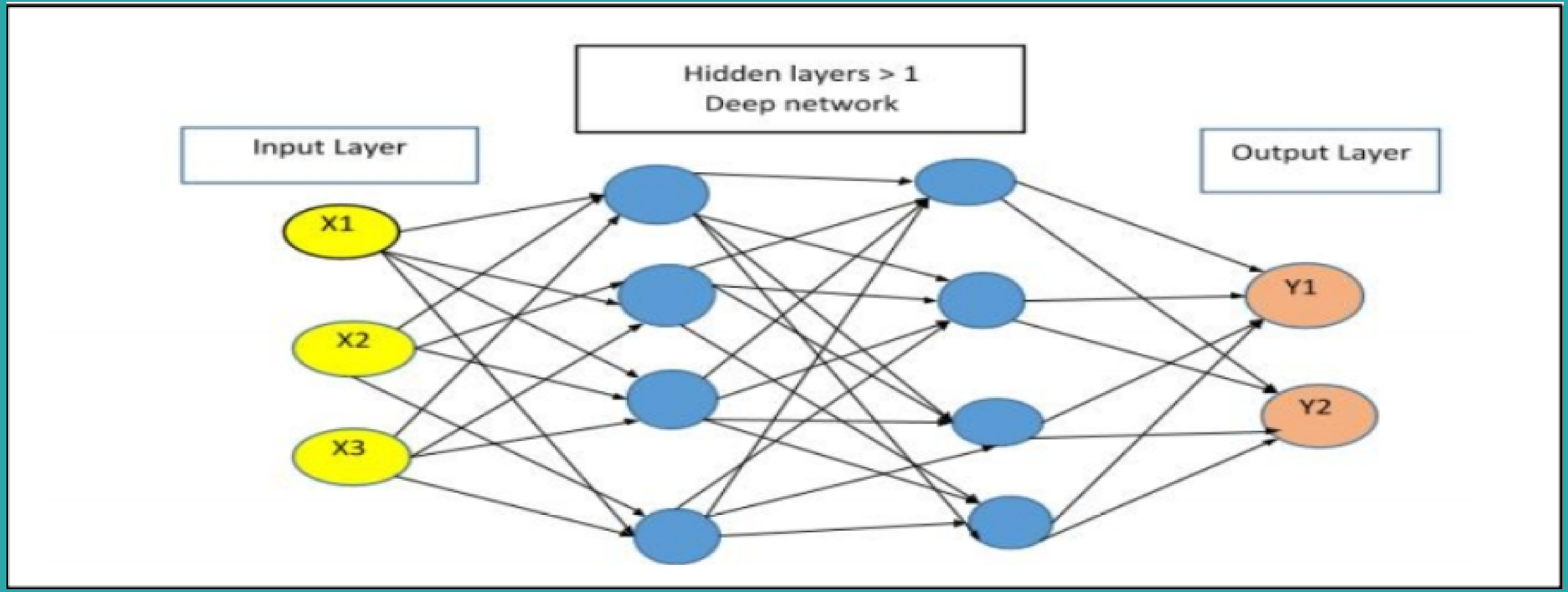
Where P(c|x) stands for Posterior Probability
(P(x|c) -> Likelihood
P(c) -> Class prior probability
P(X) -> Predictor prior probability

SVM and Adaboost classifier

SVM: Support vector machine is a supervised machine learning algorithm which defines a hyperplane which can split the data in the most optimal way such that there is a wine margin among the hyperplane and the observations.

Adaboost Classifier: AdaBoost classifier is an estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset

Feed Forward Neural



A general version of the neural network is called Feed Forward Neural Network is basically a collection of neurons. Each of these neurons or layers are vertically concatenated.

Evaluation strategy

- Accuracy
- Precision
- Recall
- FI Score
- AUC score

Confusion Matrix

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

Precision = True Positive / (True Positive + False Positive)

Recall = True Positive / (True Positive + False Negative)

F1 score = 2 *((Precision*Recall)/(Precision+Recall))

ROC curve comparison

```
[[285   0]
 [ 15   0]]
```

	precision	recall	f1-score	support
0	0.95	1.00	0.97	285
1	0.00	0.00	0.00	15
accuracy			0.95	300
macro avg	0.47	0.50	0.49	300
weighted avg	0.90	0.95	0.93	300

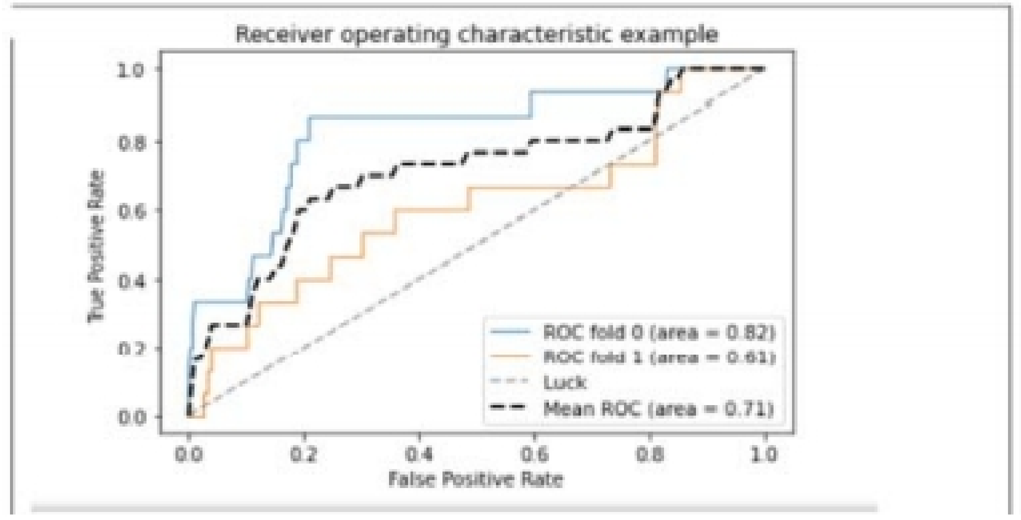


Figure 7: SVM model before applying smote.

```
[[283   2]
 [ 12   3]]
```

	precision	recall	f1-score	support
0	0.96	0.99	0.98	285
1	0.60	0.20	0.30	15
accuracy			0.95	300
macro avg	0.78	0.60	0.64	300
weighted avg	0.94	0.95	0.94	300

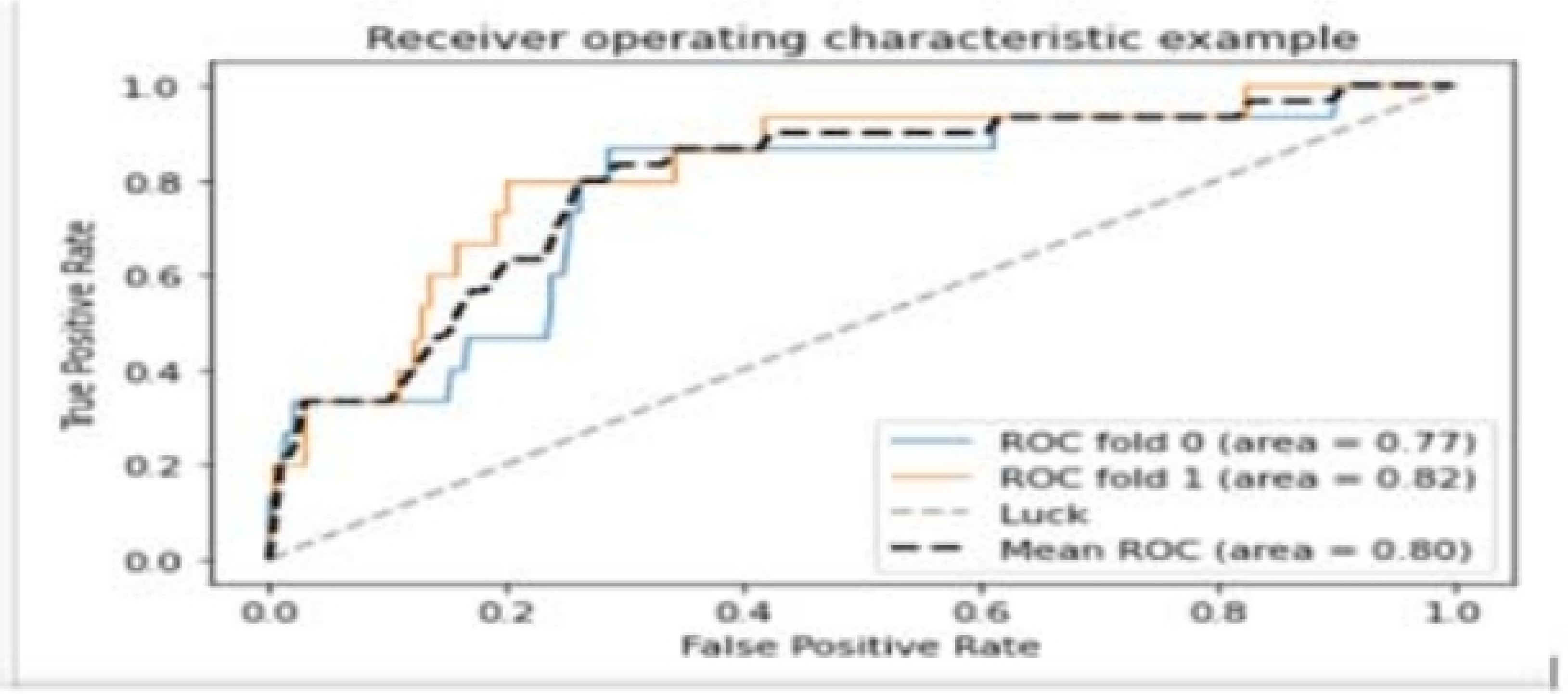


Figure 8 shows the result of applying SMOTE on the same file

Result comparison-SVM model

	Imbalanced data					Applying SMOTE				
	Accuracy	Mean AUC	Precision	Recall	FI Score	Accuracy	Mean AUC	Precision	Recall	FI Score
pc1	95	71	90	95	93	95	**80**	**94**	95	**94**
kc1	78	45	61	78	68	69	**50**	**64**	69	66
kc2	79	51	62	79	69	79	**57**	**73**	79	72
jm1	81	**61**	74	81	77	77	60	**75**	77	76
cm1	91	61	83	91	87	91	61	83	91	87

Result comparison (continued)

Final result:

Naïve Bayes algorithm

	Imbalanced data					Applying SMOTE				
	Accuracy	Mean AUC	Precision	Recall	FI Score	Accuracy	Mean AUC	Precision	Recall	FI Score
pc1	96	70	92	96	96	96	**71**	92	96	94
kc1	81	59	65	81	72	81	59	65	81	72
kc2	75	68	56	75	64	75	65	56	75	64
jm1	84	**58**	71	84	77	84	59	71	84	77
cm1	92	65	85	92	88	83	64	87	83	85

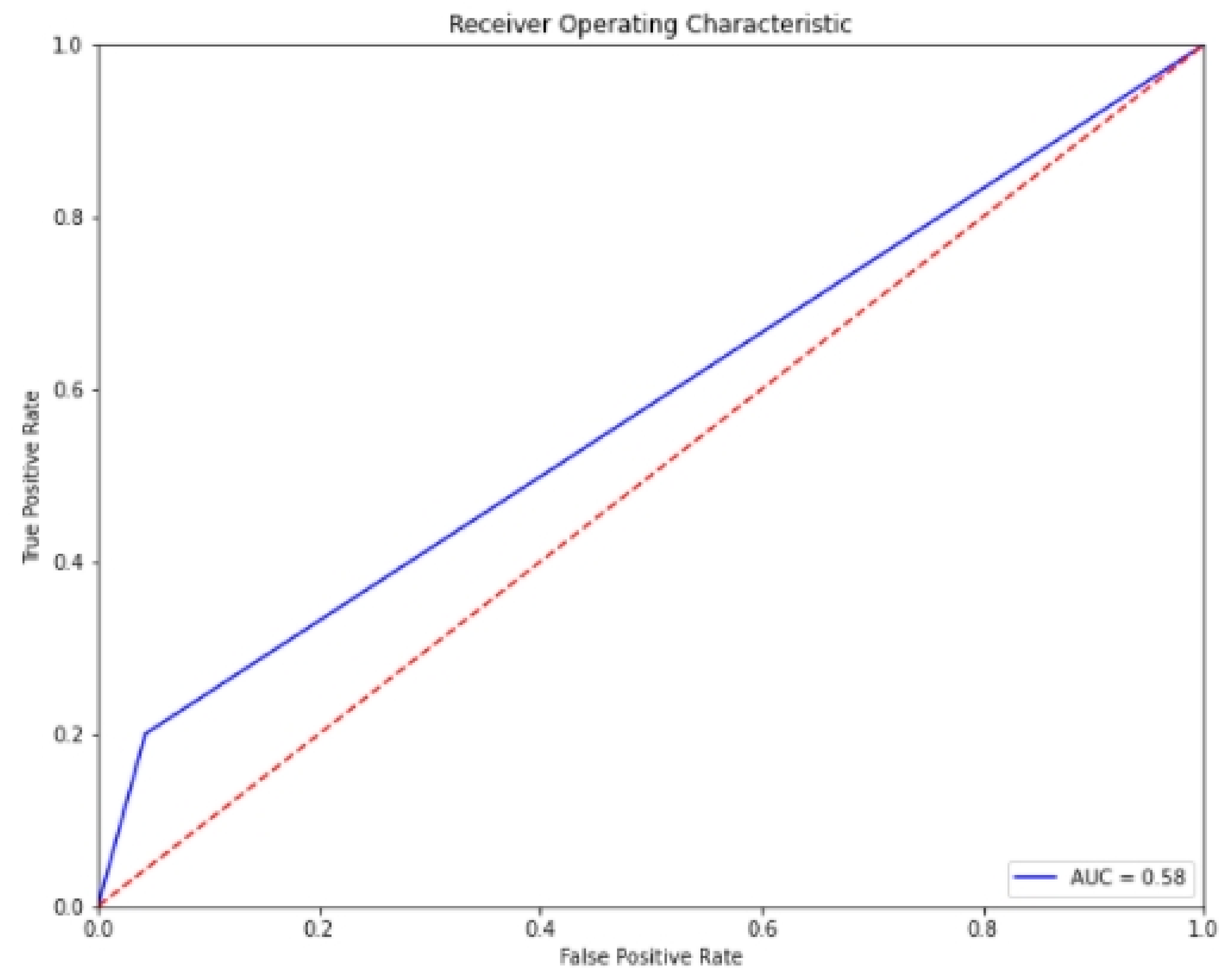
Logistic Regression

	Imbalanced data					Applying SMOTE				
	Accuracy	Mean AUC	Precision	Recall	FI Score	Accuracy	Mean AUC	Precision	Recall	FI Score
pc1	96	89	92	96	94	96	87	92	96	94
kc1	81	60	65	81	72	70	60	**75**	70	72
kc2	75	70	56	75	64	75	64	**71**	75	71
jm1	84	**63**	71	84	77	74	63	**78**	74	58
cm1	92	65	85	92	88	91	71	88	91	89

Adaboost classifer

	Imbalanced data					Applying SMOTE				
	Accuracy	Mean AUC	Precision	Recall	FI Score	Accuracy	Mean AUC	Precision	Recall	FI Score
pc1	96	89	92	96	94	53	**66**	**94**	53	**66**
kc1	81	58	65	81	72	23	**51**	**71**	23	15
kc2	75	59	72	75	72	70	63	73	70	71
jm1	84	**59**	71	84	84	16	46	**2**	16	4
cm1	91	51	84	91	88	62	58	86	62	71

Result from Feed Forward Neural Network

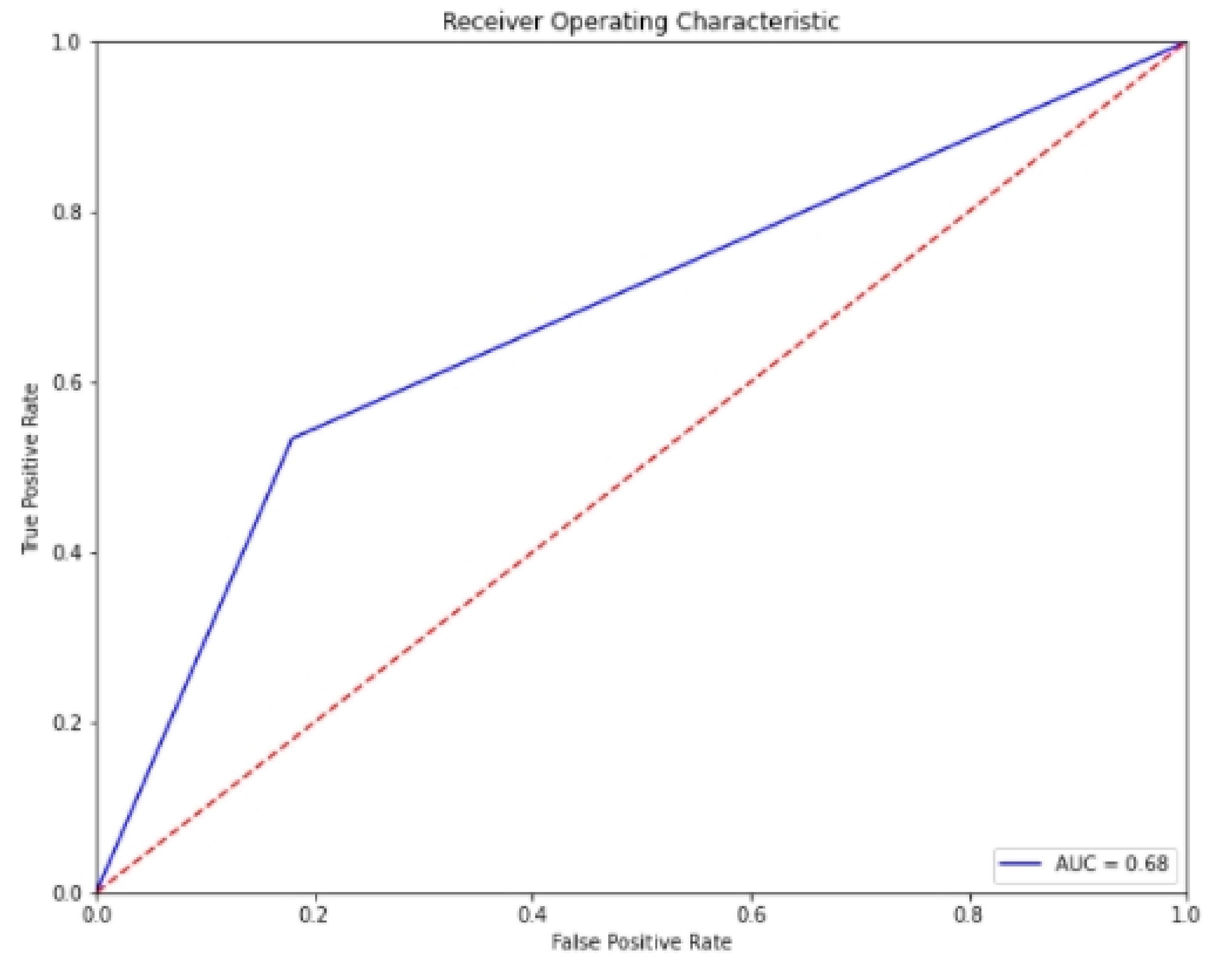



```
[[112   5]
 [ 12   3]]
              precision    recall  f1-score   support

           0       0.90      0.96      0.93       117
           1       0.38      0.20      0.26        15

    accuracy                           0.87       132
   macro avg       0.64      0.58      0.60       132
weighted avg       0.84      0.87      0.85       132

0.8712121212121212
```




```
[[96 21]
 [ 7  8]]
              precision    recall  f1-score   support

           0       0.93      0.82      0.87       117
           1       0.28      0.53      0.36        15

    accuracy                           0.79       132
   macro avg       0.60      0.68      0.62       132
weighted avg       0.86      0.79      0.81       132

0.7878787878787878
```

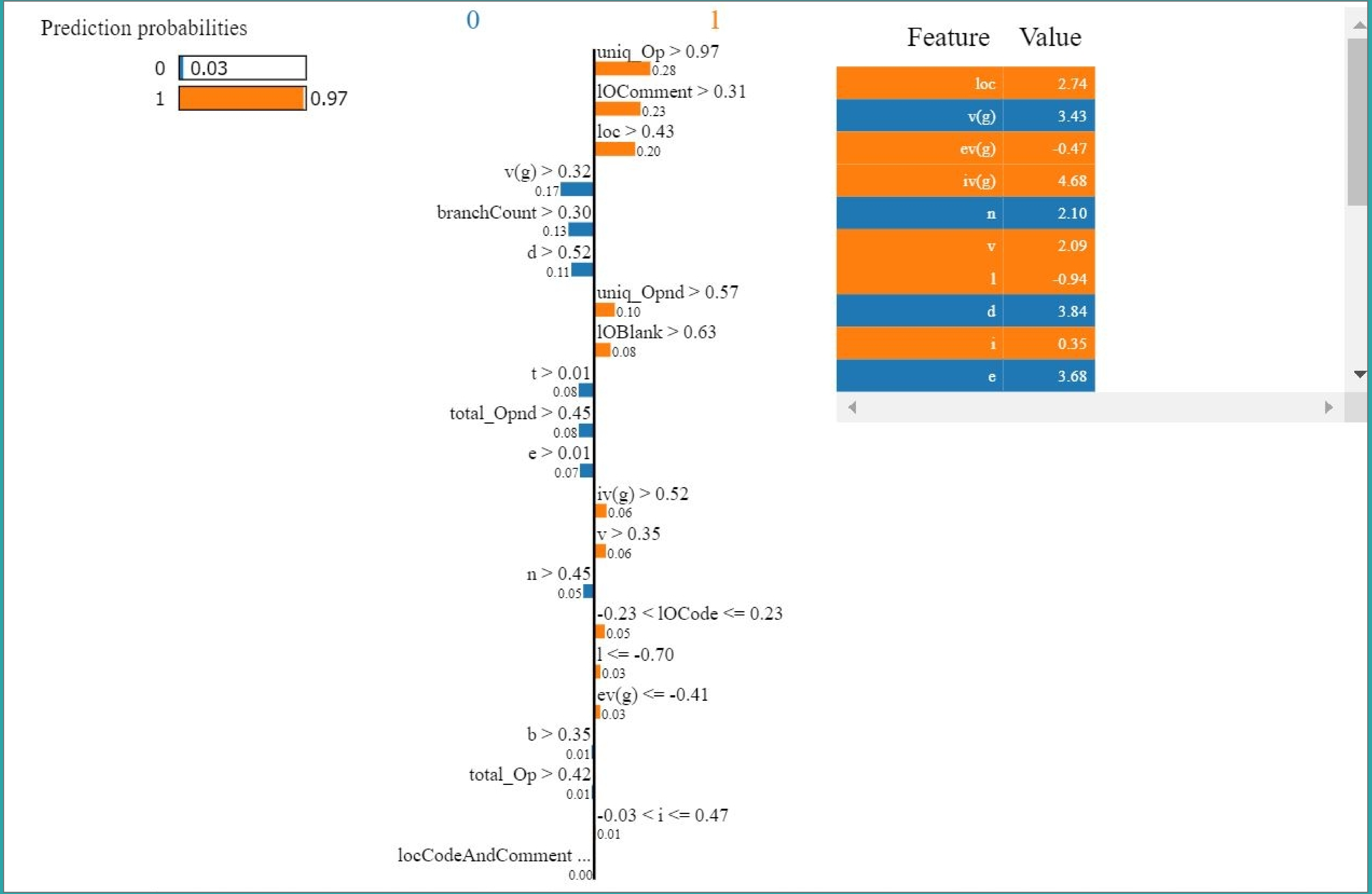
Explainable AI and LIME

Need for :
. Transparency
. Auditability

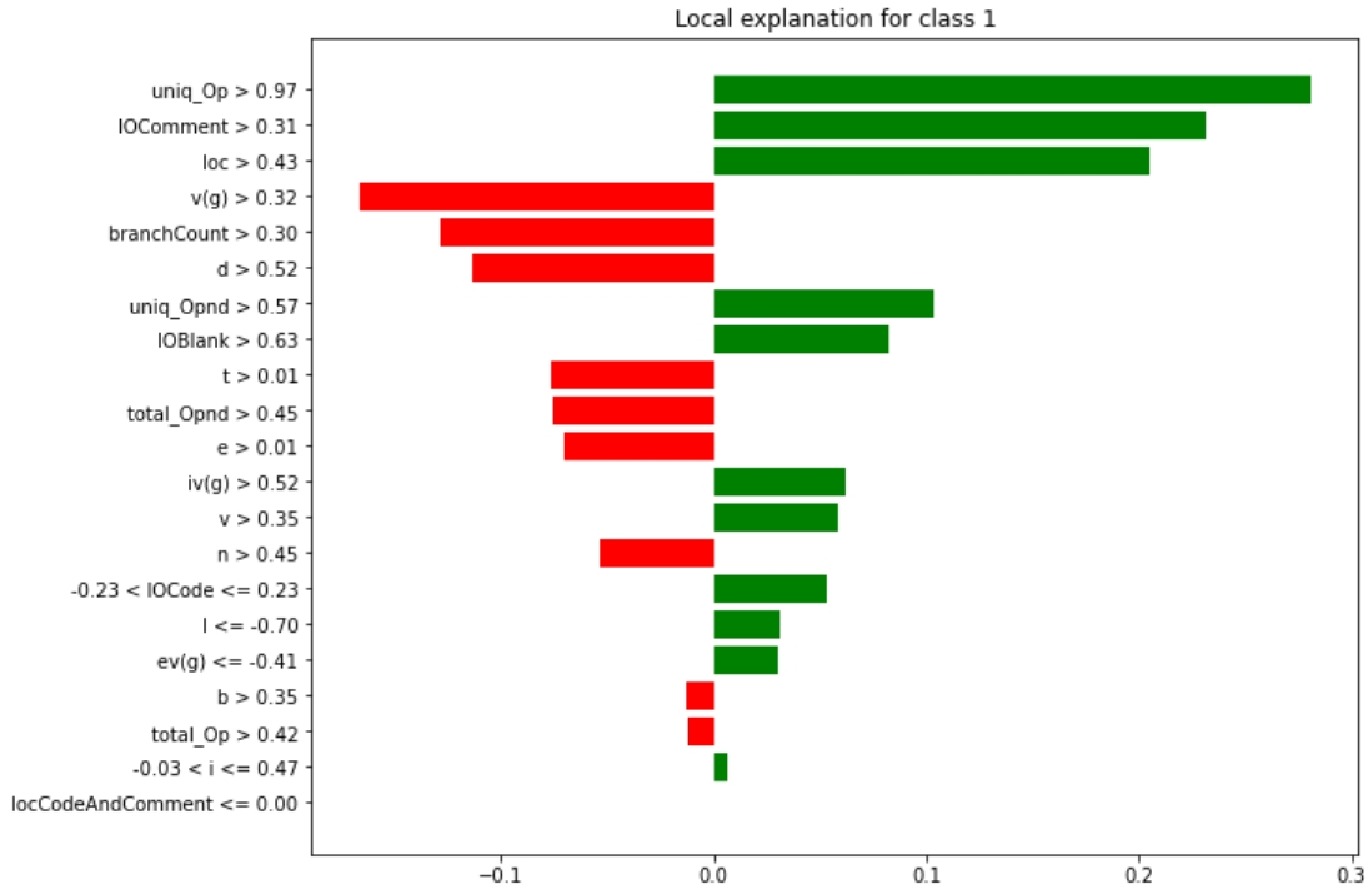
Local Interpretable Model-Agnostic Explanations (LIME) which unpack and understand the inner correlations and innerworkings of what would normally be considered a "black box" model.
AI Explanation technique that explains the predictions of any classifier in an
interpretable and faithful manner, by learning an interpretable model locally around the prediction.

Explain ability of the predicted value

Feature	Value
loc	133
v(g)	30
ev(g)	1
iv(g)	26
n	605
v	4185.91
l	0.01
d	77.46
i	54.04
e	324260.81
b	1.4
t	18014.49
lOCode	3
lOComment	91
lOBlank	55
locCodeAndComment	0
uniq_Op	50
uniq_Opnd	71
total_Op	385
total_Opnd	220
branchCount	34
defects	TRUE



Lime



Local explanation for class 1
uniq_Op > 0.97
lOComment > 0.31
loc > 0.43
v(g) > 0.32
branchCount > 0.30
d > 0.52
uniq_Opnd > 0.57
lOBlank > 0.63
t > 0.01
total_Opnd > 0.45
e > 0.01
iv(g) > 0.52
v > 0.35
n > 0.45
-0.23 < lOCode <= 0.23
l <= -0.70
ev(g) <= -0.41
b > 0.35
total_Op > 0.42
-0.03 < i <= 0.47
locCodeAndComment <= 0.00
−0.1
0.0
0.1
0.2
0.3

Conclusion and future work

Integrate defect prediction algorithm with continuous testing

Apply the same technique in other datasets

Performance tuning

References

1. Saiqa Aleem, Luiz Fernando Capretz,Faheem Ahmed, Benchmarking machine learning techniques for defect prediction

2.Martin Shepperd Qinbao Song, Zhongbin Sun Carolyn Mair, Data Quality: Some Comments on the NASA Software Defect Data Sets

3.Nitesh V. Chawla, Kevin W. Bowyer, W. Philip Kegelmeyer, SMOTE: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research, June 2002

4. T. Menzies, J. Distefanor A., Orrego and R. Chapman, "Assessing Predictors of Software Defects", 2004, Proceedings, workshop on Predictive Software Models, Chicago.

Thank you and questions?