

**Professor - Fanshawe College, London Ontario** 

- 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** 
    - **Supervisor: Dr. Luiz Capretz**
    - Date of presentation May 14th, 2021







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



### Defects overview

Text	•	Sampler result Request Response data
▼ ⊗ HTTP Request ⊗ Response Assertion		Thread Name:Thread Group 1-1 Sample Start:2020-12-02 17:06:19 EST Load time:22 Connect Time:9 Latency:18 Size in bytes:1219 Sent bytes:127 Headers size in bytes:137 Body size in bytes:1082 Sample Count:1 Error Count:1 Data type ("text" "bin" ""):text Response code:404 Response message:



**Decrease in** 



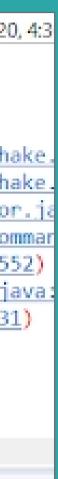
**Customer satisfaction** Market share, test coverage



erminated > GoogleTest (2) [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (Dec. 2, 2 acjava.ucti.scream.woscraccripetine.wrapenucopytrico(onknown_source)	020
at java.util.stream.FindOps\$FindOp.evaluateSequential(Unknown Source)	
at java.util.stream.AbstractPipeline.evaluate(Unknown Source)	
at java.util.stream.ReferencePipeline.findFirst(Unknown Source)	
at org.openqa.selenium.remote.ProtocolHandshake.createSession(ProtocolHand	lsha
at org.openqa.selenium.remote.ProtocolHandshake.createSession(ProtocolHand	lsha
at org.openqa.selenium.remote.HttpCommandExecutor.execute( <u>HttpCommandExecu</u>	
at org.openqa.selenium.remote.service.DriverCommandExecutor.execute(Driver	
at org.openqa.selenium.remote.RemoteWebDriver.execute( <u>RemoteWebDriver.java</u>	
at org.openqa.selenium.remote.RemoteWebDriver.startSession( <u>RemoteWebDriver</u>	
at org.openqa.selenium.remote.RemoteWebDriver. <init>(<u>RemoteWebDriver.java:</u></init>	131
at org.openqa.selenium.chrome.ChromeDriver. <init>(<u>ChromeDriver.java:181</u>) at org.openqa.selenium.chrome.ChromeDriver.<init>(<u>ChromeDriver.java:168</u>)</init></init>	
at org.openga.selenium.chrome.ChromeDriver. <init>(ChromeDriver.java:100)</init>	
at com.google.GoogleTest.main(GoogleTest.java:17)	



**Operating cost Delay in schedule** 



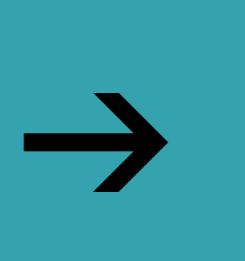


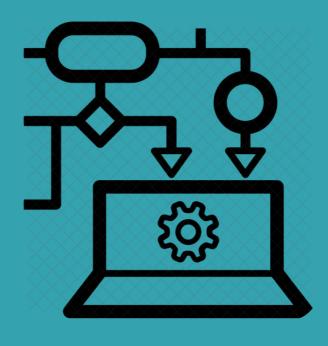




### List of software modules, and past performance statistics

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

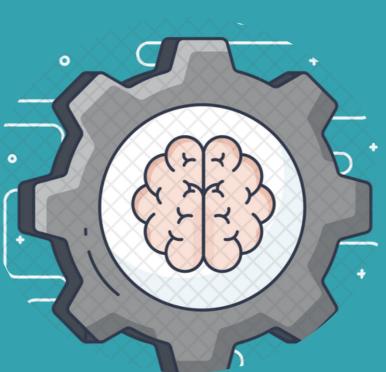




New software modules

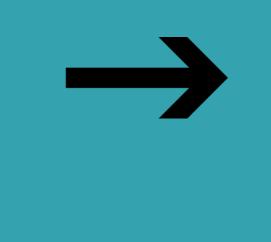


**Defect Prediction** model



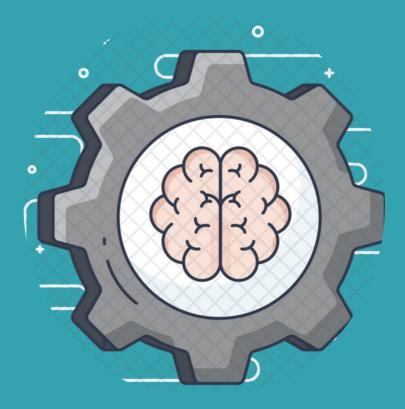
# What is defect prediction?

Machine learning algorithm





### **Defect Prediction** model



**Prediction of faulty or** non-faulty module







### **Use Cases**

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

### Use cases and challenges of defect prediction

### Challenges

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





### Literature Review

<u> </u>	
Paper	Techniques Applied
1. Aleem et al.	Weka tool Binning data Missing values filled Comparative perform
2. Shepperd et. al.	Focused on data clea
3. T Menzies et. al.	Defect prediction and failure and probability

- l by means of attributes mance analysis.
- eaning strategy due to having noisy data.

alysis based on identifying probability of ty 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	С	9.80%
kc2	522	22	C++	28%
pc1	1109	22	С	7.30%
kc1	2109	22	C++	26%
jm1	10885	22	С	19.30%
Figure 1:	Dataset descripti	on		



# Dataset description

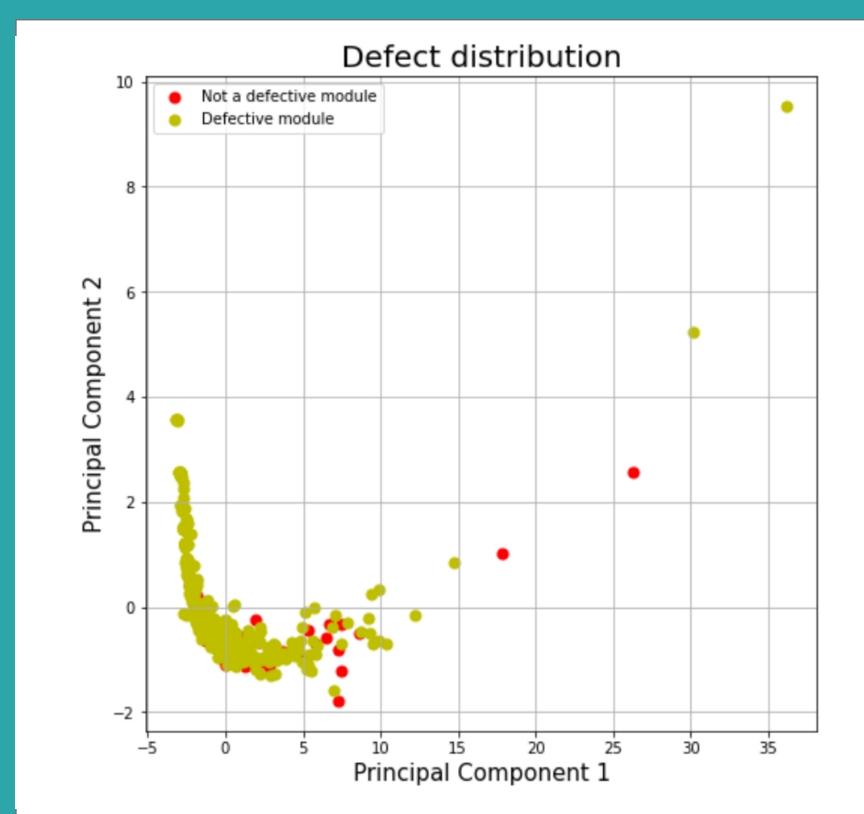


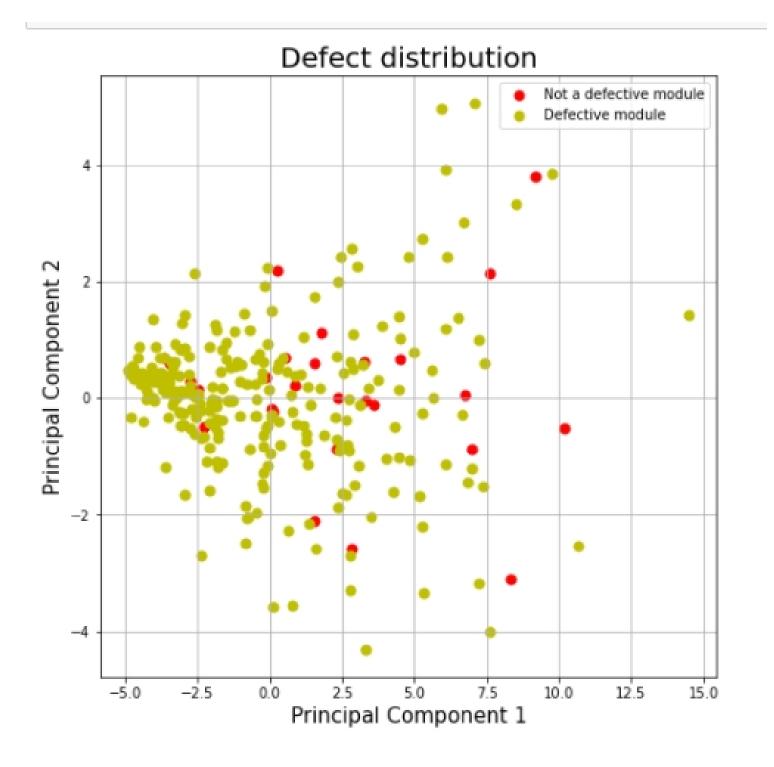
Feature	Feature name	Description	Turne	
Sequence	1	Description	Type	
0	loc	McCabe's line count of code	float64	
1	v(g)	McCabe's cyclomatic complexity	float65	
2	ev(g)	McCabe's essential coplexity	float66	
3	iv(g)	McCabe's design complexity	float67	
4	n	Halstead total operators + operands	float68	
5	v	Halstead volume	float69	
6	I. I.	Halstead program length	float70	
7	d	Halstead program difficulty	float71	
8	i	Halstead intelligence	float72	
9	e	Halstead effort	float73	
10	L.	Number of delivered bugs (from Halstead		
10	b	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	Categor	

Figure 2: Description of the dataset attributes



# Feature Engineering









loc -	1	0.94	0.77	0.92	0.94	0.95			0.79	0.82	0.94	0.62			0.67	-0.045		0.94	0.94	0.93	0.94	0.24
v(g)	0.94					0.92			0.66	0.85		0.86		0.79	0.66	-0.033		0.86	0.91	0.89	0.99	0.16
ev(g) -	0.77								0.55				0.61		0.56	-0.022	0.64				0.83	0.1
iv(g) -	0.92		0.72	1	0.87	0.89						0.83			0.64	-0.028		0.85	0.87		0.91	0.2
n-	0.94	0.91		0.87	1	0.99	4.39	0.84			0.98	0.65		0.79	0.72	-0.043	0.84	0.95		0.99	0.91	0.21
			0.77	0.89	0.99				0.79	0.83	0.99	0.68						0.95	-	0.99		
v		0.92												0.81		-0.036	0.81		0.99		0.92	0.2
1-	40.35				-0.39			-0.53	-0.39	-0.2		-0.2				0.47	-0.61	-0.4	-0.39	-0.39		-0.12
d -	0.73				0.84		4.53	1	0.5			0.73	0.62	0.61	0.64	-0.064	0.88	0.69	0.84			0.16
1-	0.79	0.66	0.55			0.79	4.39	0.5	1	0.46		0.46	0.5	0.64	0.57	-0.067	0.63	0.91	0.8			0.26
e ·	0.82	0.86		0.83	0.85	0.68	-0.2		0.46		0.87		0.65		0.59	-0.017		0.72	0.84	0.84	0.83	0.093
b	0.94	0.91	0.77	0.88	0.98	0.99		0.79	0.78	0.87	١	0.87		0.8		0.094	0.79	0.94	0.98	0.98	0.91	0.21
t	0.82	0.86		0.83	0.85	0.88	-0.2		0.46		0.87		0.65		0.59	0.017			0.84	0.84	0.83	0.093
IOCode -	0.67		0.61	0.67				0.62	0.5	0.65		0.66	1	0.54		-0.018	0.64					0.051
IOComment -	0.86	0.79						0.61	0.64		0.8		0.54	1	0.6	-0.028		0.81	0.79			0.3
IOBlank -	0.67	0.65	0.56	0.64				0.64	0.57	0.59		0.59		0.6	1	-0.034					0.66	0.16
eAndComment -	-0.045	-0.033	0.022	-0.028	-0.043	-0.036	0.47	-0.064	-0.067	0.017	0.094	0.017	0.013	-0.028	-0.034	1	-0.096	-0.045	0.043	-0.042	-0.036	0.046
				0.76	0.84	0.61		0.03			0.79						1			0.02		0.24
uniq_Op ·	0.0		0.64						0.63							-0.090		0.61	0.84	0.02	0.0	
uniq_Opnd ·	0.94				0.95	0.95	-0.4		0.91		0.94					-0.048			0.94	0.94	0.87	0.25
total_Op -	0.94	0.91		0.87		0.99	40.39	0.84		0.84	0.98	0.84				-0.043	0.84	0.94		0.98	0.91	0.21
total_Opnd ·	0.93				0.99	0.99	4.39			0.84	0.98	0.84		0.78		-0.042	0.82	0.94	0.98			0.2
branchCount -	0.94	0.99	0.83	0.91	0.91	0.92	-0.35	0.77	0 69	0.83	0.91	0.83	0.74	0.B	0.66	-0.036	0.8	0.87	0.91	0.9	1	0.16
defects -	0.24	0.16	0.1	0.2	0.21	0.2	-0.12	0.16	0.26	0.093	0.21	0.093	0.051	0.3	0.16	0.046	0.24	0.25	0.21	0.2	0.16	1
	kc.	- (6)A	cv(g) -	- (6)vi	ė	×.	-	ť	-	÷	4	4	DCode -	Comment -	108lank -	omment -	- d0_pru	- pud -	tatal Op -	tal Oprid -	nchCount -	defects -
														ů,	-	ŝ	-	Ť,	-	rta.	and	

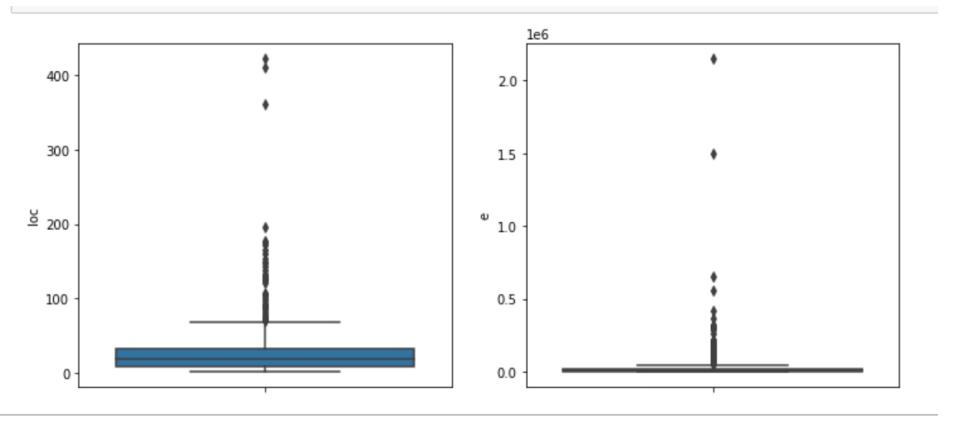
- 0.4 - 0.2 - 0.0 -0.2



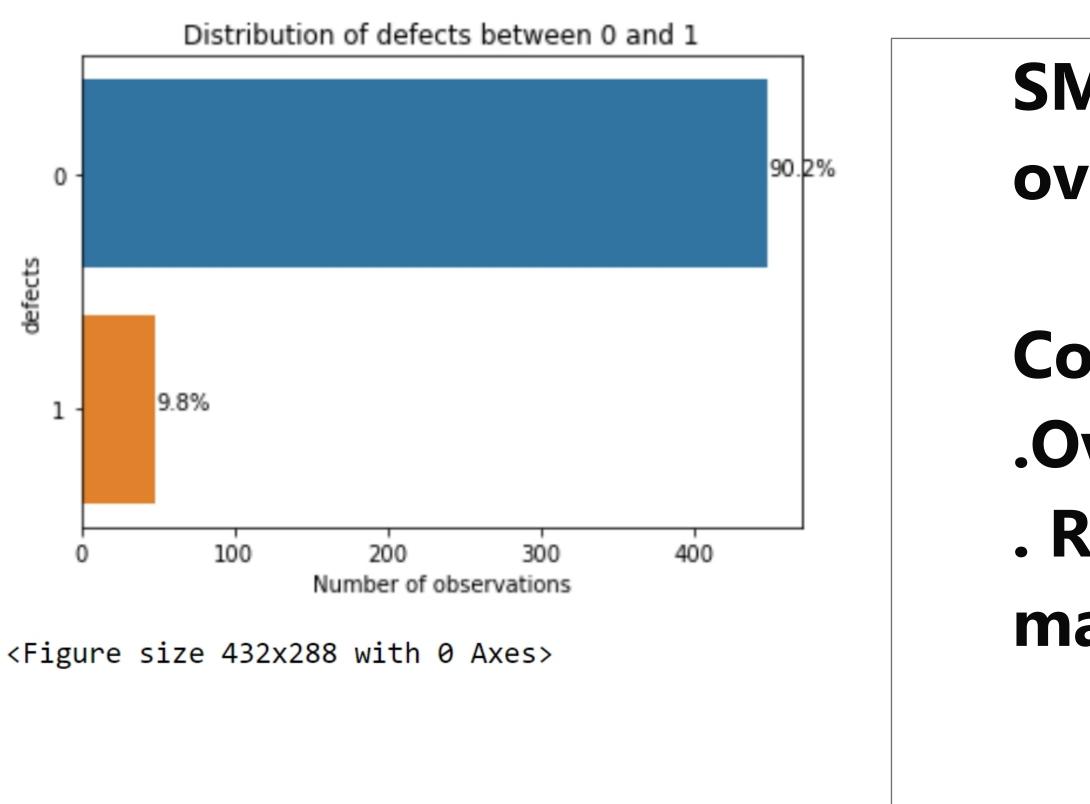


# Feature Engineering

Criterion	Data Quality	Explanation
Criterion	Data Quality	Explanation
	Category	
1	Cases with	Instances that contain one or more
	missing values	missing observations were
		removed
2	Cases with	Instances that have 2 or more
	conflicting	metric values that violate some
	features values	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	The duplicated values were
·	duplicates	removed.
I	duplicates	removed.







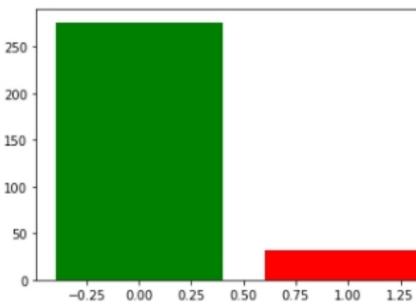
# **Class Imbalance and SMOTE**

- **SMOTE Synthetic minority** oversampling technique [Chawla et. al]
- **Combination of .Oversampling of minority class** . Random underdamping of the majority class

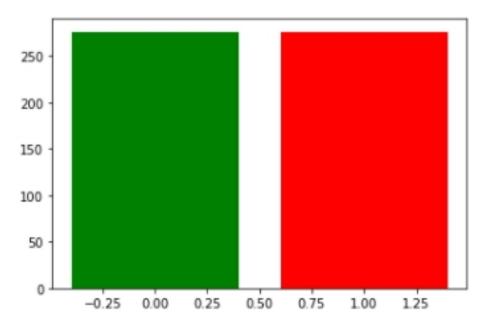




Before applying SMOTE, counts of label '1' was: 32 Before applying SMOTE, counts of label '0' was: 276



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



### Balancing data with SMOTE





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

the hyperplane and the observations.

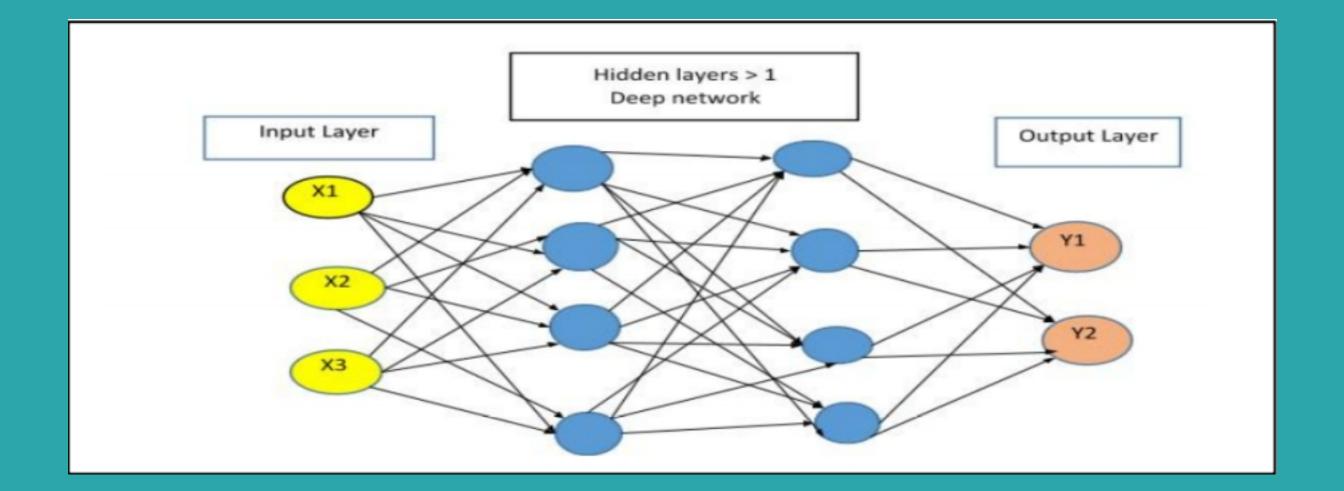
Adaboost Classifier: AdaBoost classifier is an estimator that additional copies of the classifier on the same dataset

- 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
- begins by fitting a classifier on the original dataset and then fits



### 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.





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

Actual	
Actual	



### **Confusion Matrix**

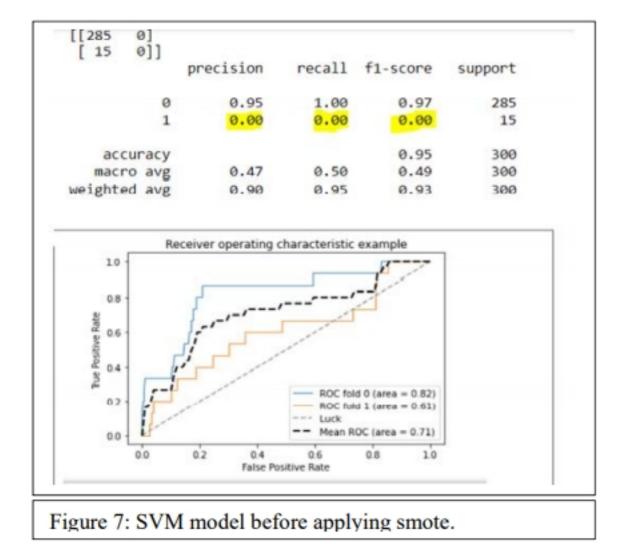
	Predicted	0	Predicted	1
0	TN		FP	
1	FN		TP	

**Precision = True Positive / (True Positive + False Positive) Recall = True Positive / (True Positive + False Negative)** F1 score = 2 \*( (Precision\*Recall)/(Precision+Recall))





### **ROC curve comparison**



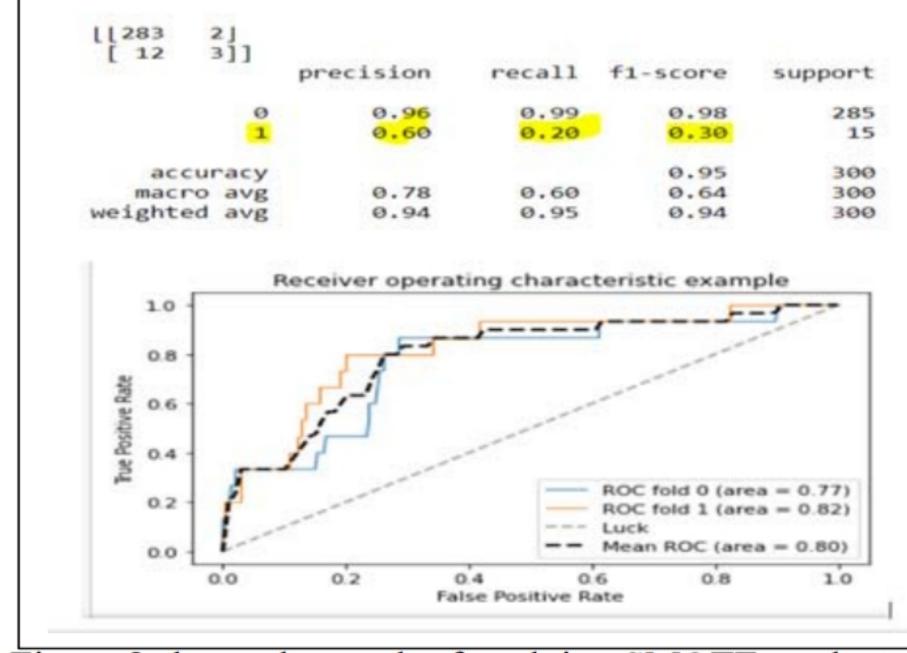


Figure 8 shows the result of applying SMOTE on the same file





# **Result comparison-SVM model**

			m <mark>balanced</mark>	data			Арр	lying SMOT	Έ	
	Accuracy	Mean	Precision	Recall	FI Score	Accuracy	Mean	Precision	Recall	FI Score
		AUC					AUC			
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	<mark>91</mark>	61	83	91	87	91	61	83	91	87





# Result comparison (continued)

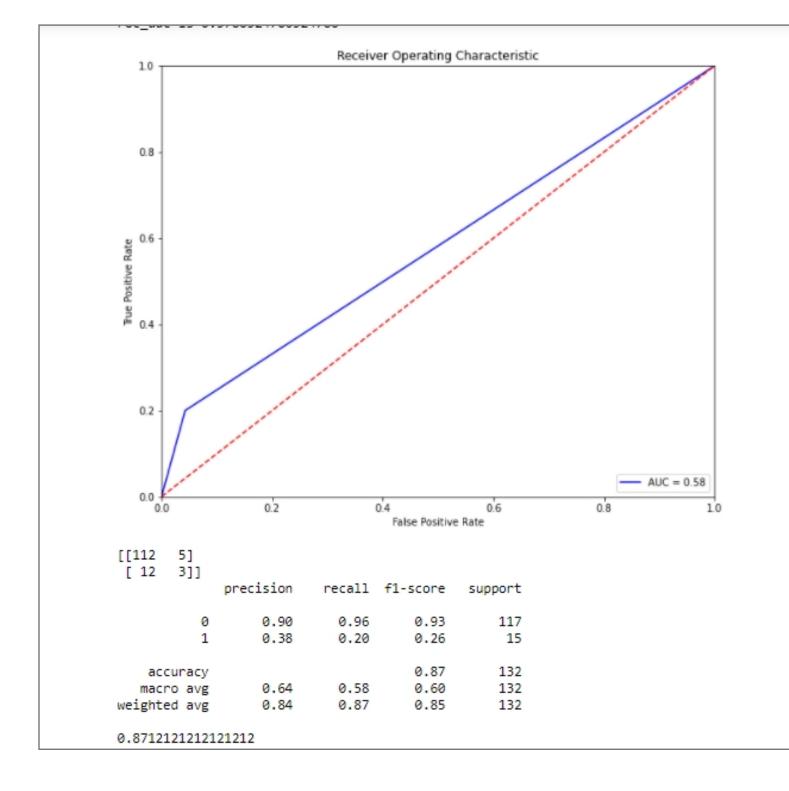
### Final result:

	ui 1050				Naïve Bay	es algorithn	n			
			mbalanced	data			App	olying SMOT	ΓE	
	Accuracy Mean Precisio		Precision	Recall	FI Score	Accuracy	Mean	Precision	Recall	FI Score
		AUC					AUC			
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 Re	egression					
Imbalanced data							App	olying SMOT	ΓE	
	Accuracy	Mean	Precision	Recall	FI Score	Accuracy	Mean	Precision	Recall	FI Score
		AUC					AUC			
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					
		Ir	mbalanced	data			App	olying SMOT	TE	
	Accuracy	Mean	Precision	Recall	FI Score	Accuracy	Mean	Precision	Recall	FI Score
		AUC					AUC			
pc1	96	89	92	96	94	53	66	94	53	66
kc1	81	58	65	81	72	23	51	71	23	15
kc2	75	<mark>5</mark> 9	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

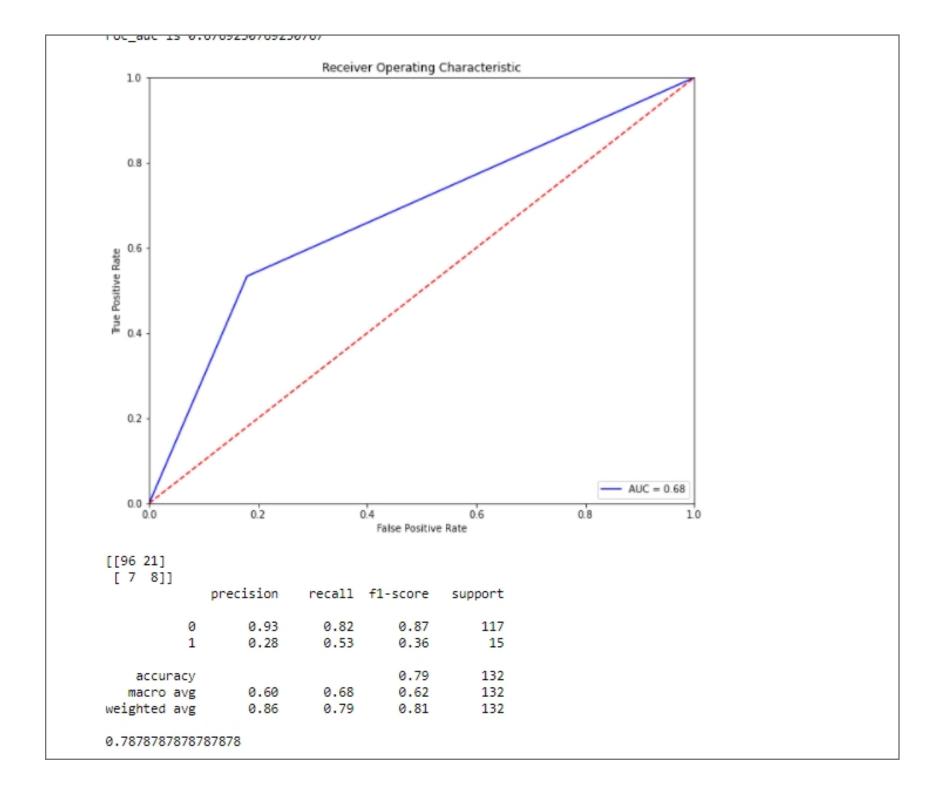








### **Result from Feed Forward Neural Network**











# Explainable Al and LIME

### **Need for :**

- . Transparency
- . Auditability

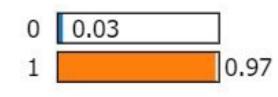
what would normally be considered a "black box" model. classifier in an model locally around the prediction.

- Local Interpretable Model-Agnostic Explanations (LIME) which unpack and understand the inner correlations and innerworkings of Al Explanation technique that explains the predictions of any
- interpretable and faithful manner, by learning an interpretable

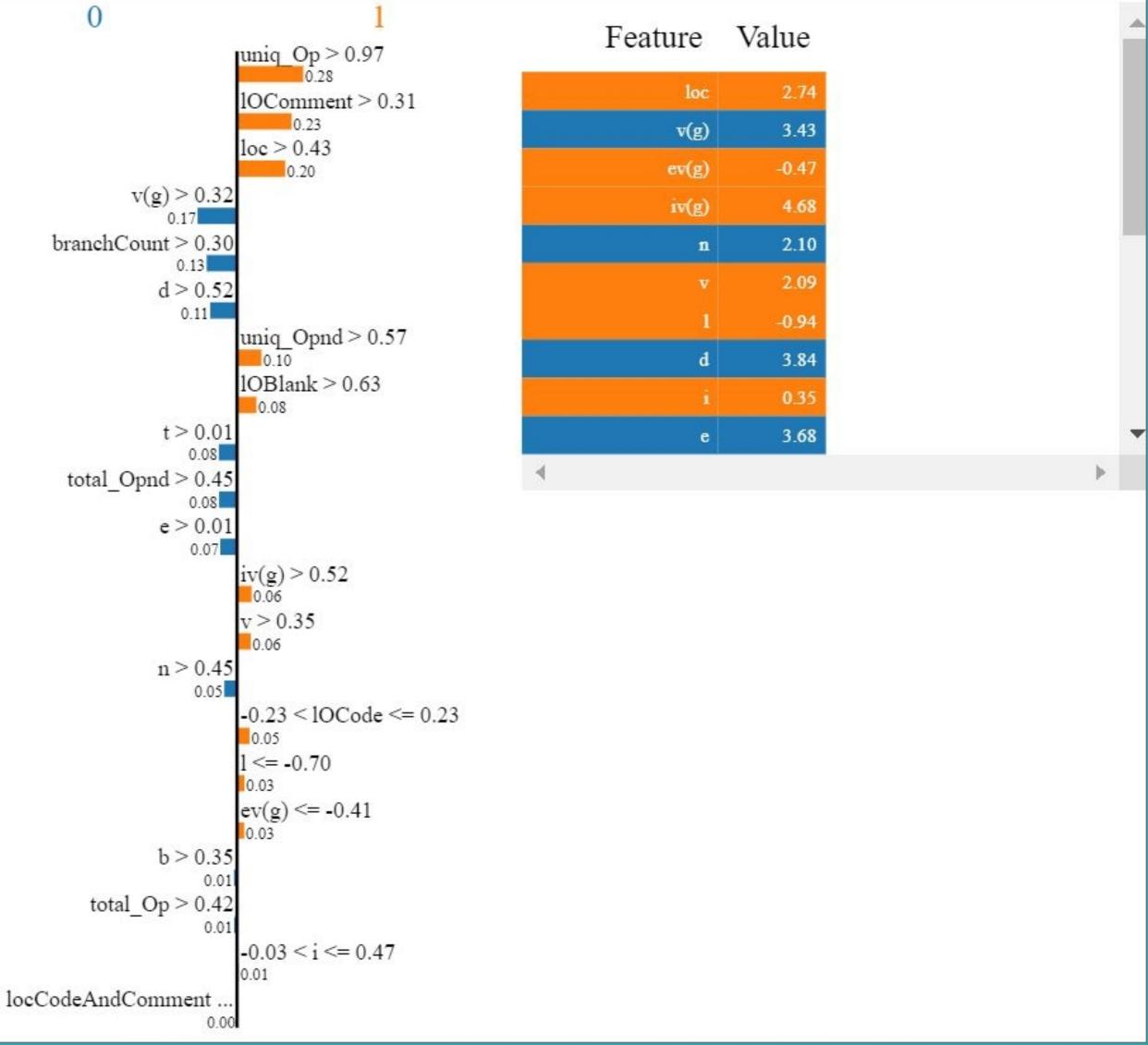


### Explain ability of the predicted value

### Prediction probabilities



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





### Lime





uniq\_Op > 0.97 -IOComment > 0.31 loc > 0.43 v(g) > 0.32 branchCount > 0.30 d > 0.52 uniq\_Opnd > 0.57 IOBlank > 0.63 t > 0.01 total\_Opnd > 0.45 e > 0.01 iv(g) > 0.52 v > 0.35 n > 0.45 --0.23 < IOCode <= 0.23 I <= -0.70 ev(g) <= -0.41 b > 0.35 total\_Op > 0.42 -0.03 < i <= 0.47 locCodeAndComment <= 0.00

### Local explanation for class 1

	-			
- 1				
0	.0	0.1	0.2	0.3





### Integrate defect prediction algorithm with continuous testing

### Apply the same technique in other datasets **Performance tuning**

### **Conclusion and future work**







techniques for defect prediction

**Comments on the NASA Software Defect Data Sets** 

over-sampling technique, Journal of Artificial Intelligence Research, June 2002

**Defects**", 2004, Proceedings, workshop on Predictive Software Models, Chicago.

1. Saiqa Aleem, Luiz Fernando Capretz, Faheem Ahmed, Benchmarking machine learning

- 2.Martin Shepperd Qinbao Song, Zhongbin Sun Carolyn Mair, Data Quality: Some
- 3.Nitesh V. Chawla, Kevin W. Bowyer, W. Philip Kegelmeyer, SMOTE: synthetic minority
- 4. T. Menzies, J. Distefanor A., Orrego and R. Chapman, "Assessing Predictors of Software



# Thank you and questions?



