

# *“Automatically Assessing Code Understandability”* **Reanalyzed:** Combined Metrics Matter

**Asher Trockman**, Keenen Cates, Mark Mozina,  
Tuan Nguyen, Christian Kästner, Bogdan Vasilescu

# Automatically Assessing Code Understandability: How far are we?

Simone Scalabrino, Gabriele Bavota, Christopher Vendome,  
Mario Linares-Vásquez, Denys Poshyvanyk, Rocco Olivetto

- Motivation: Understandability...
  1. is crucial for maintenance
  2. could predict defects
- Understandability metric: extremely useful

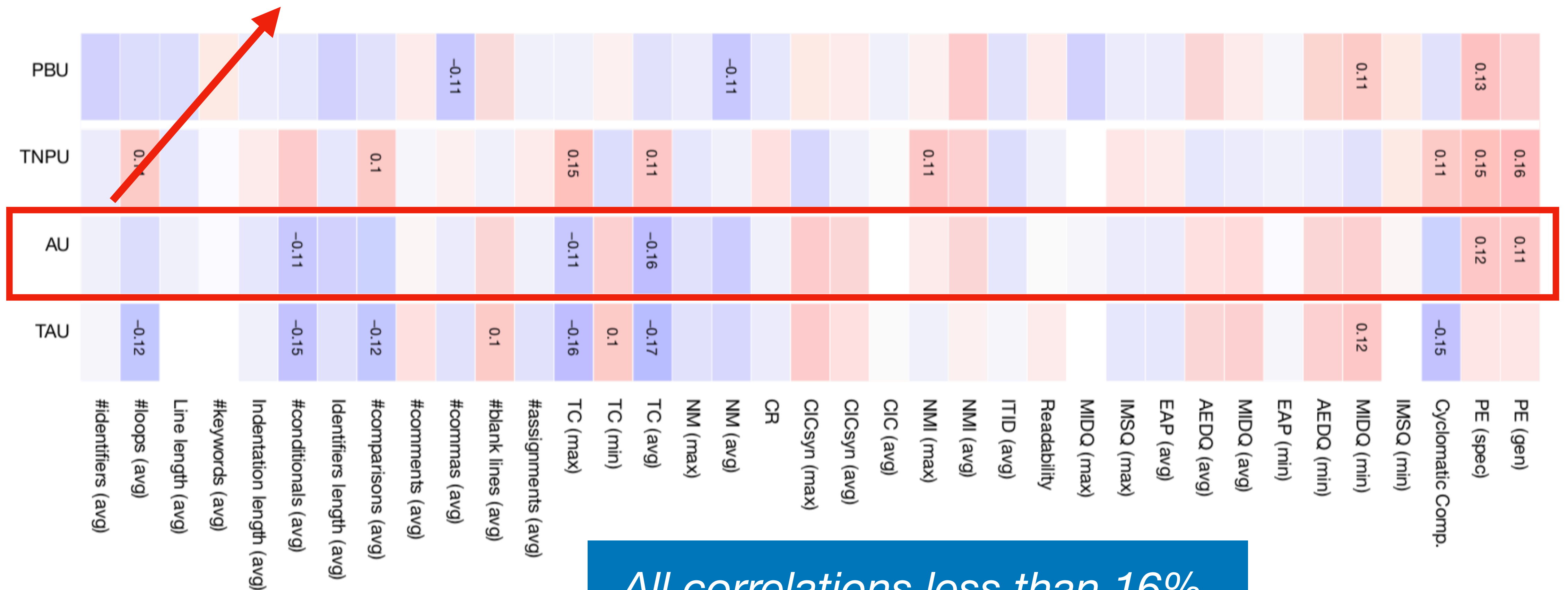
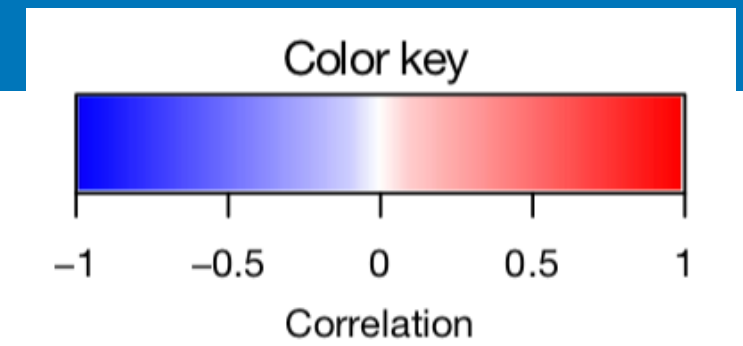
# Automatically Assessing Code Understandability: How far are we?

Simone Scalabrino, Gabriele Bavota, Christopher Vendome,  
Mario Linares-Vásquez, Denys Poshyvanyk, Rocco Olivetto

- 46 developers quizzed on 8 Java snippets
- Recorded 121 code-related metrics for the snippets
- $n = 324$  observations,  $p = 121$  features

# Original study: Individual correlations only

## Understandability vs. **121** Metrics



*All correlations less than 16%.*

from “Automatically Assessing Code Understandability”, Scalabrino et al. (2017)

# Our reanalysis: Combined metrics → Logistic models

- Improvement: **multiple regression** models
  - (Understandability ~ Combination of metrics +  $\epsilon$ )
- Public data set: ***Thank you, Scalabrino et al.!***
- Caveat: High dimensionality (121 metrics)
  - Solution: Automatic variable selection
  - e.g., **forward stepwise selection** and **LASSO**



# What explains understandability?

## 1. Developer Experience

If a developer has 5 or more years of programming experience, their **odds of understanding increase by 200%** on average.

### 1. Forward-Stepwise-Selected Understandability Classifier

Variable	Coefficient
(Intercept)	-0.398
Indentation.length..dft.	0.241
Literals..Visual.Y.	-0.275
JavaProfessional	1.228
X.parameters	-0.307
NMI..avg.	0.442**
MIDQ..max.	-0.392*
X.periods..avg.	-0.325*
Professional	1.014*
Line.length..max.	-0.488**
TC..avg.	-0.793***
N	324
Log Likelihood	185.00
AIC	394.00
BIC	439.37
$R_m^2$	0.2870
$R_c^2$	0.4106

\*\*\*p < .001; \*\*p < .01; \*p < .05

# What explains understandability?

## 2. Maximum Line Length

Increasing the maximum line length by one character **decreases the odds of understanding by 2%.**

Takeaway: keep lines short.

### 1. Forward-Stepwise-Selected Understandability Classifier

Variable	Coefficient
(Intercept)	-0.398
Indentation.length..dft.	0.241
Literals..Visual.Y.	-0.275
JavaProfessional	1.228
X.parameters	-0.307
NMI..avg.	0.442**
MIDQ..max.	-0.392*
X.periods..avg.	-0.325*
Professional	1.014*
Line.length..max.	-0.488**
TC..avg.	-0.793***
N	324
Log Likelihood	185.00
AIC	394.00
BIC	439.37
$R_m^2$	0.2870
$R_c^2$	0.4106

\*\*\* p < .001; \*\* p < .01; \* p < .05

# What explains understandability?

## 3. Narrow Meaning Identifiers<sup>1</sup>

Increasing NMI, a measure of descriptiveness of variable names, by one unit **increases the odds of understanding by 80%**.

Takeaway: use specific variable names.

### 1. Forward-Stepwise-Selected Understandability Classifier

Variable	Coefficient
(Intercept)	-0.398
Indentation.length..dft.	0.241
Literals..Visual.Y.	-0.275
JavaProfessional	1.228
X.parameters	-0.307
NMI..avg.	0.442**
MIDQ..max.	-0.392*
X.periods..avg.	-0.325*
Professional	1.014*
Line.length..max.	-0.488**
TC..avg.	-0.793***
N	324
Log Likelihood	185.00
AIC	394.00
BIC	439.37
$R_m^2$	0.2870
$R_c^2$	0.4106

\*\*\* p < .001; \*\* p < .01; \* p < .05



# What explains understandability?

By combining metrics on developer experience, code readability, and more...

**Pseudo-R<sup>2</sup> = 41%**

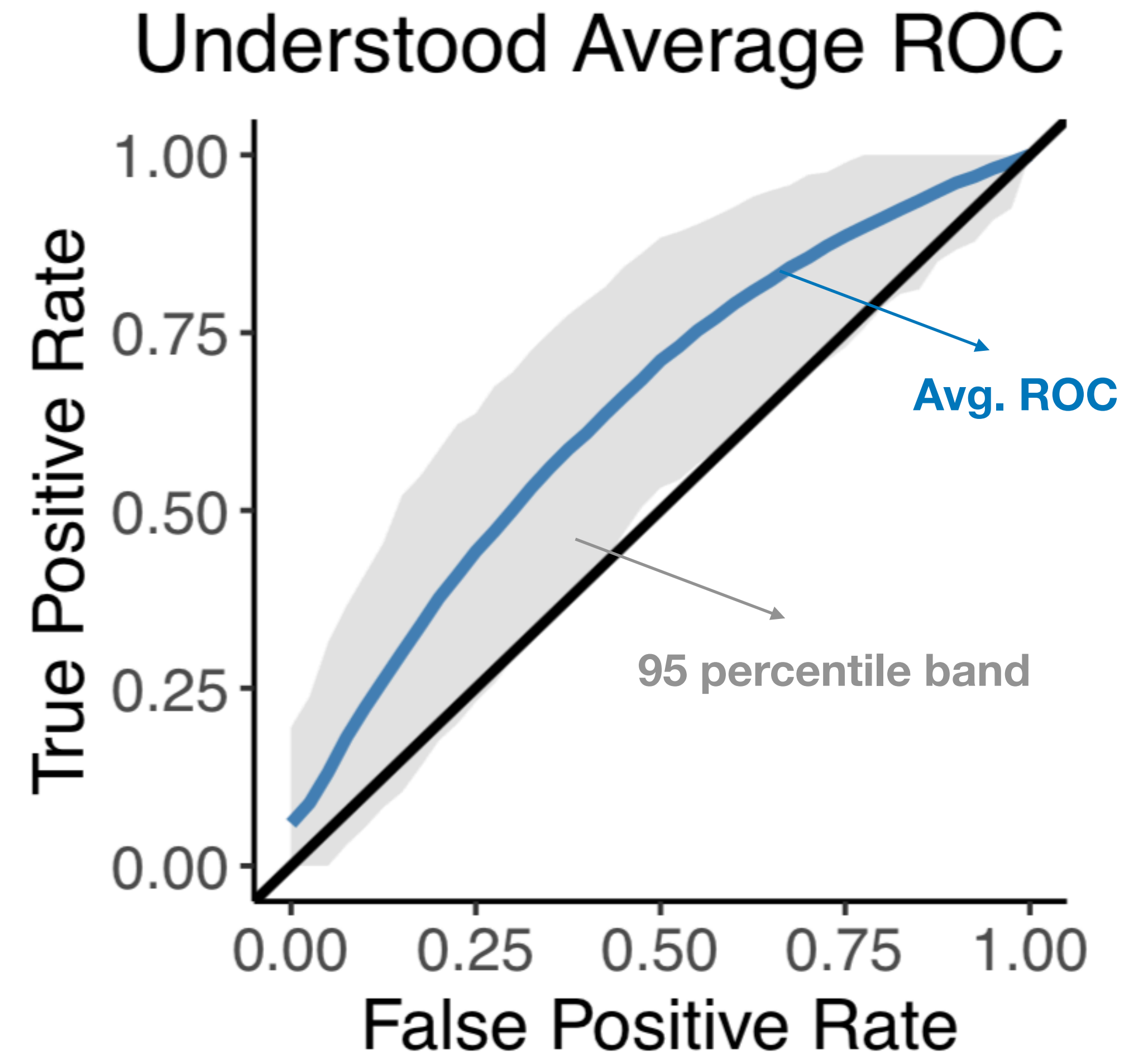
## 1. Forward-Stepwise-Selected Understandability Classifier

Variable	Coefficient
(Intercept)	-0.398
Indentation.length..dft.	0.241
Literals..Visual.Y.	-0.275
JavaProfessional	1.228
X.parameters	-0.307
NMI..avg.	0.442**
MIDQ..max.	-0.392*
X.periods..avg.	-0.325*
Professional	1.014*
Line.length..max.	-0.488**
TC..avg.	-0.793***
N	324
Log Likelihood	185.00
AIC	394.00
BIC	439.37
R <sub>m</sub> <sup>2</sup>	0.2870
R <sub>c</sub> <sup>2</sup>	0.4106

\*\*\* p < .001; \*\* p < .01; \* p < .05

# Can we *predict* understandability?

- Binary classifier (Logistic)
- Understood or not
- **Random cross validation**
- **Avg. AUC: 0.64**
  - i.e., ranks an easy-to-understand snippet above a hard-to-understand one 64% of the time



## Original Study

*Correlations with individual metrics...*

**Can we measure understandability?**

**NO**

(Not with existing individual metrics.)

## Our Reanalysis

*Linear models with combined metrics...*

**Can we measure understandability?**

**YES**

(With more data.)

# Creating a Metric of Code Understandability:

**Now**

***Future Work***

**46** developers



**1000** developers

Small dataset  
Simple models  
~64% accuracy

Big data  
Advanced models  
Useful in real world