Indian Statistical Institute, Kolkata at PR-SOCO 2016 : A Simple Linear Regression Based Approach

Kripabandhu Ghosh[1,2]
Swapan Kumar Parui[1]

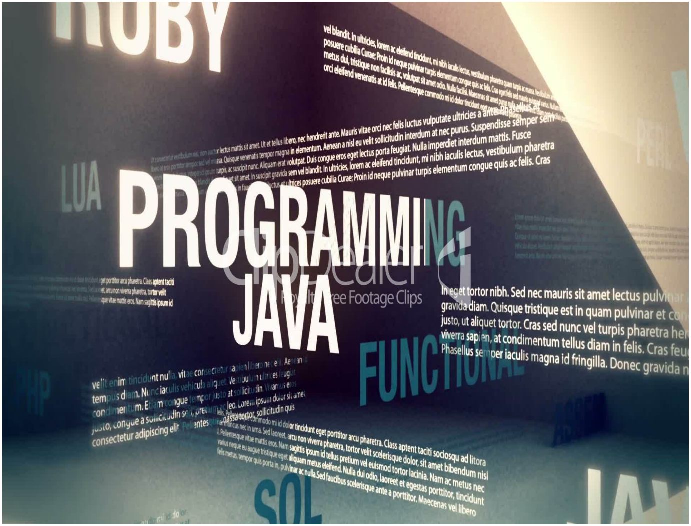
[1]**Indian Statistical Institute, Kolkata, India**

[2]**Indian Institute of Technology, Kanpur, India**

Objective

To predict the BIG5 personality traits of a person from her Java program code

Programming



Programming and personality

Programming and personality

Programming and personality

DELETE A SEMICOLON FROM CODE
AND ALL PROGRAMMERS LOSES THEIR MINDS
quickmeme.com

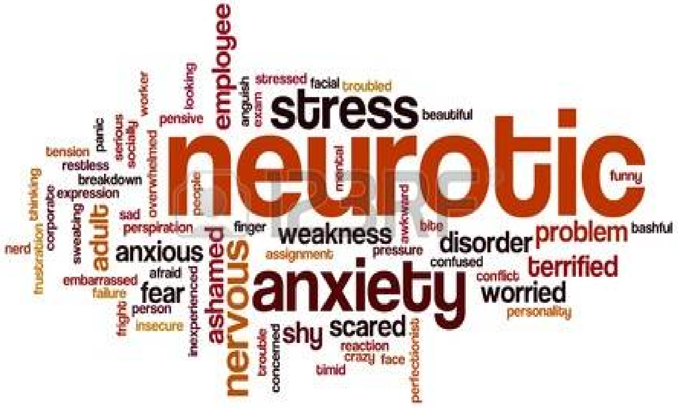
Outline

1. BIG5 personality
2. Features
3. Methodology
4. Results
5. Analysis

BIG5 personality

BIG5 personality traits

BIG5 personality : Neuroticism



BIG5 personality : Neuroticism

BIG5 personality : Neuroticism

Motivation

Neurotics exhibit low emotional stability and so is likely to be less methodical in writing a code.

BIG5 personality : Extroversion

expressive
leads
intuition
joyful
self-assured
urban
loquacious
quote
happy
stress
attitude
gossip
irrational
vitality
blabbing
rambling
elated
yapping
excited
shouting
business
outspoken
optimistic
gab
friendly
confident
chatty
hip
forward
outgoing
chatterbox
individual
interpersonal
communicate
joy
personality
psychology
blabbermouth
extroverted
babbling
positive
bold
happiness
ramble
people
extrovert
daring
interactions
concept
gossiper
introvert
chat
success
talkative
laughter
behavior
cheerful
talk
vibrant
successful
chitchat
fearless

BIG5 personality : Extroversion

BIG5 personality : Extroversion

Motivation

Extroverts are likely to express themselves and possibly provide meaningful comments in their code.

Outline

1. BIG5 personality
2. Features
3. Methodology
4. Results
5. Analysis

Features

FEATURES

Features

Determining factors

- Readibility
- Efficiency

Features : Multi-line comments (MLC)

- The number of genuine comment words in multi-line comments, i.e., between /* and */ found in the program code.
- We have not considered the cases where lines of code were commented.
- Eliminate code lines – E.g., using **[a-zA-Z][a-zA-Z]*[ ]*(** matching **System.out.println("Even");** used in a Java code.
- This feature value was normalized by dividing it by the total number of words in the program file
- Indicator of code readability and meticulousness of the coder.

Features : Multi-line comments (MLC)

Feature	Positive example	Negative example

MLC	/**
	* Make the hash table logically empty.
	*/
		/*System.out.println(“Even”);
		printQ(qEven);
		System.out.println(“Odd”);
		printQ(qOdd);*/

SLC	// Create a new double-sized, empty table	//String[] ss = linea.readLine().split(“ ”);
NES	for (int i=1; i<=casos; i++)	for (int i = 1; i< = casos; i++)
IS	import java.io.FileNotFoundException	import java.io.*

Features : Single-line comments (SLC)

- This is the number of genuine single-line comment words in single line comments, i.e., comments following “//”.
- We have not considered the cases where lines of code were commented.
- Eliminate code lines – same as MLC.
- This feature value was normalized by dividing it by the total number of words in the program file.
- Indicator of code readability and meticulousness of the coder.

Features : Single-line comments (SLC)

Feature	Positive example	Negative example

MLC	/**
	* Make the hash table logically empty.
	*/
		/*System.out.println(“Even”);
		printQ(qEven);
		System.out.println(“Odd”);
		printQ(qOdd);*/

SLC	// Create a new double-sized, empty table	//String[] ss = linea.readLine().split(“ ”);
NES	for (int i=1; i<=casos; i++)	for (int i = 1; i< = casos; i++)
IS	import java.io.FileNotFoundException	import java.io.*

Features : Non-existent spaces (NES)

- This is the number of lines containing non-existent spaces
- *i=1; i<=casos;* as opposed to *i = 1; i< = casos;*
- Regular expression **[a-z][a-z]* [a-z][a-z]*[=<>+]** (e.g., int i=1)
- This feature value was normalized by dividing it by the total number of lines in the program file.
- Indicator of code readability and meticulousness of the coder.

Features : Non-existent spaces (NES)

Feature	Positive example	Negative example

MLC	/**
	* Make the hash table logically empty.
	*/
		/*System.out.println(“Even”);
		printQ(qEven);
		System.out.println(“Odd”);
		printQ(qOdd);*/

SLC	// Create a new double-sized, empty table	//String[] ss = linea.readLine().split(“ ”);
NES	for (int i=1; i<=casos; i++)	for (int i = 1; i< = casos; i++)
IS	import java.io.FileNotFoundException	import java.io.*

Features : Import Specific (IS)

- This is the number of instances where the programmer exported the specific libraries only
- E.g., cases of import **java.io.FileNotFoundException** as opposed to **import java.io.***
- This feature value was normalized by dividing it by the total number of lines in the program file.
- Indicator of code efficiency as well as experience, prudence of the coder

Features : Import Specific (IS)

Feature	Positive example	Negative example

MLC	/**
	* Make the hash table logically empty.
	*/
		/*System.out.println(“Even”);
		printQ(qEven);
		System.out.println(“Odd”);
		printQ(qOdd);*/

SLC	// Create a new double-sized, empty table	//String[] ss = linea.readLine().split(“ ”);
NES	for (int i=1; i<=casos; i++)	for (int i = 1; i< = casos; i++)
IS	import java.io.FileNotFoundException	import java.io.*

Outline

1. BIG5 personality
2. Features
3. Methodology
4. Results
5. Analysis

Methodology

METHODOLOGY

Method : Multiple Linear Regression

- Four features – explanatory variables
- Each of the five BIG Five traits is the dependent variable.

Method : Multiple Linear Regression

For a program code p, given as follows:

$$\begin{aligned} score_{BIG5}(p) =& \alpha + \beta_1 MLC(p) + \beta_2 SLC(p) \\ &+ \beta_3 NES(p) + \beta_4 IS(p) \end{aligned} \tag{1}$$

We calculate the values of α and β_i, $i = 1, 2, 3, 4$ from the training data using the linear regression implementation in R.[a]

[a] https://www.r-bloggers.com/r-tutorial-series-multiple-linear-regression/

Outline

1. BIG5 personality
2. Features
3. Methodology
4. Results
5. Analysis

Results

RESULTS

Results

1. **Run1.txt**: The values of the dependent variables were generated on the test data using the regression equation (1) learned from the training data.
2. **Run2.txt**: For this run, for each BIG Five trait, we calculated the values of the dependent variables given by the linear regression equation (1) on the training set. We then calculated the error and removed the files in the training set with the three highest error values. We then trained the linear regression on the new training set and calculated the coefficients. Finally, values of the dependent variables were calculated on the test data. The purpose of this run is to remove some outliers from the training set.

Results : RMSE

Method	NEUROTICISM	EXTROVERSION	OPENNESS	AGREEABLENESS	CONSCIENTIOUSNESS
Run1.txt	10.22	**8.60**	7.16	9.60	9.99
Run2.txt	10.04	10.17	7.36	9.55	10.16
Baseline (bow)	10.29	9.06	7.74	9.00	8.47
Baseline (mean)	10.26	9.06	7.57	9.04	8.54
Reported best	9.78	8.60	6.95	8.79	8.38

Table : Root Mean Squared Error (RMSE). The best result produced by our submitted runs when compared to all the submitted runs is shown in bold.

Results : PC

Method	NEUROTICISM	EXTROVERSION	OPENNESS	AGREEABLENESS	CONSCIENTIOUSNESS
Run1.txt	**0.36**	0.35	0.33	0.09	-0.20
Run2.txt	0.27	0.04	0.27	0.11	-0.13
Baseline (bow)	0.06	0.12	-0.17	0.20	0.17
Baseline (mean)	0.00	0.00	0.00	0.00	0.00
Reported best	0.36	0.47	0.62	0.38	0.33

Table : Pearson Product-Moment Correlation (PC). The best result produced by our submitted runs when compared to all the submitted runs is shown in bold.

Outline

1. BIG5 personality
2. Features
3. Methodology
4. Results
5. Analysis

Analysis

ANALYSIS

Analysis

BIG5 Trait	α (Intercept)	β_1 (MLC)	β_2 (SLC)	β_3 (NES)	β_4 (IS)
Neuroticism	55.30	10.82	-331.58	-57.15	-282.14
Extroversion	39.58	50.49	261.44	67.38	163.28
Openness	46.63	46.07	98.92	28.20	49.48
Agreeableness	42.521	-1.103	78.905	90.909	196.740
Conscientiousness	-1.708	-1.708	225.988	-67.633	135.353

Table : The regression coefficients for Run1

Analysis : Insights

- The negative value of high magnitude of β_2 indicates that a person who frequently provides Single Line Comments (SLC) in her code is likely to exhibit low Neuroticism.
- The negative value of high magnitude of β_4 indicates that a person who tends to import libraries selectively, is likely to have low Neuroticism
- The positive values of β_1, β_2 and β_4 indicates that a person who tends to provide genuine comments (both Multi Line and Single Line) and import specific libraries in her code is likely to have high Extrovertion.
- The observations for *Openness* are almost identical to those for *Extroversion*.

Thank you!

Backspace
Scree
Home
Page Up
Thank You
Page Down
"
'
Shift
Insert