PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Instituto de Informática

Unidade Praça da Liberdade

Curso: Engenharia de Software

Prof.: Laerte Xavier

Disciplina: Laboratório de Experimentação de Software

Alunos:

Laercio Nazareno Neto

Yan Max Rodrigues Sette Pinheiro

Laboratório 2 - Uma Análise Comparativa de Repositórios Python

Hipóteses

Na análise realizada sobre os 1000 repositórios mais populares no GitHub na linguagem Python, temos as seguintes hipóteses:

1) Os repositórios Python do Guido van Rossum possuem características similares;
2) Os top-1000 repositórios Python mais populares possuem características similares;
3) Os repositórios populares Python são de boa qualidade;
4) A popularidade influencia nas características de repositórios Python.

Metodologia

Através de um script em python realizando consultas via API ao GraphQL, com dados de repositórios do GitHub, foi gerado um arquivo csv com os dados.

Posteriormente, outro script realizou a leitura deste arquivo csv e baixou do GitHub cada repositório. Ao baixar, o mesmo script acessava os diretórios para obter a quantidade de linhas de código em cada um deles. Essa nova informação era então armazenada no arquivo csv e os repositórios excluídos da máquina de execução.

Definimos um limite de 10 minutos para o script baixar um arquivo inteiro, caso não obtivesse sucesso o número de linhas de código era armazenado como 0 (zero). Utilizamos de tal valor de tempo por conta da baixa capacidade das máquinas que estavam rodando o software, se cada um dos repositórios, pensando pelo pior cenário, gastasse mais do que esse tempo, poderia prejudicar as máquinas ou não dar tempo de executar tudo até a data de entrega.

Foram 15 repositórios que não conseguimos baixar do GitHub por conta deste limite de tempo. São eles:

1) shadowsocks/shadowsocks
2) 0voice/interview_internal_reference
3) kon9chunkit/GitHub-Chinese-Top-Charts
4) geekcomputers/Python
5) wangzheng0822/algo
6) openai/baselines
7) eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee/eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
8) cyrus-and/gdb-dashboard
9) sshuttle/sshuttle
10) pudo/dataset
11) arielf/weight-loss
12) raspberrypi/documentation
13) googleapis/google-cloud-python
14) ymcui/Chinese-BERT-wwm
15) OpenDroneMap/ODM

Por fim esse arquivo foi convertido para uma planilha do Excel, em que fórmulas, como as de média e mediana, foram aplicadas nas colunas para exibir os resultados.*

Definimos “Qualidade” como sendo o resultado da subtração da média da quantidade de releases pela média do tempo de atualização em dias, essa última métrica sendo o tempo em dias que o repositório não é atualizado. Então, pontos de qualidade = quantidade de releases - tempo de atualização em dias. Considerando uma qualidade boa os valores acima de 5 pontos.

*Nos resultados finais, consideramos um total de 985 repositórios analisados, excluindo com 15 informados anteriormente.

Resultados Obtidos

1) Nos repositórios do Guido, a mediana da quantidade de linhas de código é de 2858 linhas, enquanto a média é 42717,166 linhas. A mediana de atualização é de 46,5 dias enquanto a média é 75 dias. A mediana de idade é de 2,5 anos e a média é 2 anos. Por fim, a média de total de releases é 1,583 releases e a mediana é 0 release.
2) Nos top 1000 repositórios de Python, a mediana da quantidade de linhas de código é de 7646 linhas, enquanto a média é 37519,677 linhas. A mediana de atualização é de 0 dias enquanto a média é 0,548 dias. A mediana de idade é de 4 anos e a média é 4,43 anos. Por fim, a média de total de releases é 14,779 releases e a mediana é 1 release.
3) Como definimos, a qualidade dos repositórios do Guido foi -73,417 pontos. Enquanto a qualidade dos top 1000 repositórios de Python foi 14,724 pontos.

4) Os resultados para a qualidade dos 250 repositórios mais populares de Python e para os 250 repositórios menos populares de Python entre os 1000 primeiros foram, respectivamente, 18,608 pontos e 8,708 pontos.

Discussão

1) Analisando os resultados, chegamos à conclusão de que a única característica similar nos repositórios do Guido é a idade dos repositórios, visto que a diferença entre a mediana e a média da idade é próxima de 0 (zero). Então a hipótese de que os repositórios Python do Guido van Rossum possuem características similares é falsa.
2) Analisando os resultados, chegamos à conclusão de que as únicas características similares nos top 1000 repositórios em Python é a idade dos repositórios e o tempo desde a última atualização, visto que a diferença entre a mediana e a média dessas métricas é próxima de 0 (zero). Então a hipótese de que os top-1000 repositórios Python mais populares possuem características similares é falsa.
3) Analisando os resultados, chegamos à conclusão de que os repositórios do Guido não são de boa qualidade, já que o resultado da fórmula que definimos foi abaixo de 5 pontos. Já os top-1000 repositórios mais populares em Python possuem uma boa qualidade, validando nossa hipótese.
4) Analisando os resultados, chegamos à conclusão de que a popularidade influencia sim nas características de repositórios Python. Visto que, pelo resultado da fórmula que definimos, os 250 primeiros tiveram 9,9 pontos a mais do que os 250 últimos. Validando a nossa hipótese.