Phân tích dữ liệu từ một repo trên github

Trên 1 repo gồm có:

+Issuees : số lượng vấn đề mà có người tìm ra, trong đó có các câu hỏi thảo luận

+Pull request :

+Project: số lượng project trong repo

+fork: số lượng các bản sao được tạo ra trên từng tài khoản để thực hiện các thay đổi độc lập trên mỗi tài khoản đó

….

Lấy các dữ liệu trên, tạo thành 1 data frame

Câu hỏi :

-có bao nhiêu người commit,số lần commit của từng người?

-ai commit nhiều nhất?

-không tính other, chỉ có 4 ngôn ngữ được sử dụng, ngôn ngữ nào được sử dụng nhiều nhất,ít nhất?

-số file thêm vào, xóa đi, thay đổi nhiều nhất, ít nhất vào thời gian nào của từng file

-ngôn ngữ chủ yếu của từng người

-tổng số commit trong năm đầu, năm cuối

-năm có tổng số commit nhiều nhất

-tính phần trăm số ngôn ngữ được sử dụng

-ai commit với nhiều ngôn ngữ nhất

-tương quan giữa số lượng commit và sự thay đổi của các file

Làm sạch dữ liệu: vì id là một chuỗi số, có thể coi là số nguyên, do đó có thể lấp đầy những ô nan bằng những số nguyên. Và để không làm ảnh hưởng đến dữ liệu, em sẽ chọn những id trong khoảng id min đến id max và không trùng nhau

-Introduction:

Giới thiệu về repo: Đây là một repo một khóa học trên github có tên là React For Beginners.

Repo này gồm có: 180 commits, 2.6k forks, 2.8k stars, 38 Contributor, 31 pull requests…

Trong phần này, em sẽ phân tích dữ liệu commit của repo trên github.

-Data Cleaning & Processing:

Bộ dữ liệu được lấy trực tiếp từ hệ thống của github nên gần như sự sai lệch dữ liệu là không có. Dưới đây là thông tin dữ liệu sau khi đã được khai thác:

(df.info)

Tuy nhiên trong quá trình trích xuất, em nhận thấy rằng vẫn có những phần không tồn tại dữ liệu: Với hai thuộc tính là id\_author và login\_author chỉ có 147 non-null nhỏ hơn so với kích thước dữ liệu là 180(trên ảnh).

Do đó, để dữ liệu liên tục, em đã lấp đầy những ô dữ liệu khuyết của hai thuộc tính:

-Với id\_author: là những số nguyên nhưng những số nguyên này không phải là một dãy liên tiếp. Để mức độ ảnh hưởng của việc fillna là nhỏ nhất, em đã điền bằng những số nguyên nằm trong khoảng nhỏ nhất và lớn nhất của id hiện có và thỏa mãn với mỗi người thì id\_author là khác nhau.

Có một trường hợp có thể xảy ra: đôi lúc những lần commit có dữ liệu về id\_author và login\_author và có những lần không. Để giải quyết vấn đề này, em đã chia ra thành 2 trường hợp: nếu như đã tồn tại dữ liệu, ta thay thế nan bằng dữ liệu đó, nếu không thay bằng dữ liệu mới.

-Với login\_author: login\_author và name là ánh xạ 1-1. Do đó em điền những ô nan của login\_author bằng dữ liệu tương ứng của name.

Về dữ liệu ngoại biên: Vì đây là dữ liệu được lấy trực tiếp, ít sai sót, để tránh mất những trường hợp đặc biệt, em giữ nguyên bộ dữ liệu .

(dẫn chứng)

\*Data Analysis:

-Một tập các đuôi file được sử dụng trong commit:

(dẫn chứng)

Từ đây ta có thể kết luận: có 4 ngôn ngữ được sử dụng: js-JavaScript, html-HTML,

css-CSS, styl-Stylus, số còn lại là những file khác…

\*Phân tích về phần dữ liệu có nan:

Ở trên ta đã dự đoán rằng dữ liệu id, login author bị mất không phụ thuộc hoàn toàn vào tài khoản, mà còn phụ thuộc vào lần commit. Thật vậy: khi so sánh số lần commit trong dữ liệu nan và trong dữ liệu gốc có khác nhau: ‘Dom Gaulton’ với 4 lần commit trong dữ liệu gốc và chỉ có 3 lần commit xuất hiện trong dữ liệu nan, chứng tỏ một điều rằng có một lần commit của ‘Dom Gaulton’ dữ liệu id, login author không bị mất.

(dẫn chứng)

Một số điều em rút ra được sau khi phân tích dữ liệu nan:

(Dữ liệu nan: là một phần trong dữ liệu gốc trong đó có chứa giá trị nan)

-Có 10 người trong dữ liệu nan

-Người commit nhiều nhất là ‘Charlotte Spencer’ với 14 lần

-Thời gian đầu, cuối của những commit chứa dữ liệu nan là từ:

‘2015-11-04 20:02:24’ đến ‘2019-08-08 09:51:05’

-Số dòng bị thay đổi của các file(tổng của add và delete):

(dẫn chứng)

Nhiều nhất là file CSS với 1093, file HTML không bị thay đổi dẫn đến không có các dòng thêm vào và xóa đi đối với file HTML

-Các dòng thêm vào , xóa đi nhiều nhất vẫn là file CSS, trật tự thứ tự các file không có sự thay đổi

-Số người là committer là 9, nhiều nhất là ‘GitHub’ với 15 lần

-Về message: có nhiều message nhưng chủ yếu là thông điệp ‘fix: typo in security rules for solution’ chiếm 33.3% số message

\*Phân tích về dữ liệu sau khi đã làm sạch hoàn toàn:

-Có 44 người commit, người commit nhiều nhất là ‘Wes Bos’ với 82 lần, có 30 người commit chỉ 1 lần.

-Ngôn ngữ có số dòng bị thay đổi nhiều nhất là JavaScript với 21047 dòng(không tính other)

(dẫn chứng)

-Số dòng của các loại file thêm vào , xóa đi, thay đổi nhiều nhất vào các mốc thời gian:

(dẫn chứng)

‘Wes Bos’ và ‘’wesbos chia đều 2 vị trí người tác động đến các các file nhiều nhất

(mỗi người 6 loại)

-Số lượng các comment:khi commit gần như không có một comment nào(176 commit), chỉ có 4 lần commit là có một comment

(dẫn chứng)

-Số committer là 35, committer có số commit nhiều nhất là ‘GitHub’ với 65 lần

-Có 12 người có commiter là ‘GitHub’, chiếm 27.3% tổng số người có trong bộ dữ liệu

-Người có committer GitHub trong mỗi lần commit nhiều nhất là ‘Wes Bos’ với 41 lần

-Ngôn ngữ commit nhiều nhất của ‘Wes Bos’ là JavaScript với 11862 dòng thay đổi chiếm 43.7% so với các ngôn ngữ, chiếm 3.3% so với tất cả(tính cả other), nhưng loại file bị thay đổi nhiều nhất của ‘Wes Bos’ là ‘other’ với 334559 dòng thay đổi, chiếm 92.5%

-