



Group 4, Presentation 2

By: Bill Yu, Kevin Amado, Leo Wei



Project Objective


We want to **predict whether an issue on a specific GitHub repository will be fixed** by analysing the issue title, description and discussion comments.



Data Collection - Selected Project

github.com/PowerShell/PowerShell

README.md

 **PowerShell**

Welcome to the PowerShell GitHub Community! PowerShell Core is a cross-platform (Windows, Linux, and macOS) automation and configuration tool/framework that works well with your existing tools and is optimized for dealing with structured data (e.g. JSON, CSV, XML, etc.), REST APIs, and object models. It includes a command-line shell, an associated scripting language and a framework for processing cmdlets.

Windows PowerShell vs. PowerShell Core


Although this repository started as a fork of the Windows PowerShell code base, changes made in this repository do not make their way back to Windows PowerShell 5.1 automatically. This also means that [issues tracked here](#) are only for PowerShell Core 6 and higher. Windows PowerShell specific issues should be reported with the [Feedback Hub app](#), by choosing "Apps > PowerShell" in category.

New to PowerShell?








If you are new to PowerShell and would like to learn more, we recommend reviewing the [getting started](#) documentation.


Data Collection - Project Steps

main 1 branch 0 tags Go to file Add file <> Code

 **kamadorueda** step-1: minor refactors

4523589 2 days ago 12 commits

 step-0-raw-data	step-1: add structured data	2 days ago
 step-1-structured-data	step-1: minor refactors	2 days ago
 step-2-labeling	step-2: exclude pull requests	2 days ago
 step-3-visualization	step-3: visualization	2 days ago
 README.md	step-0: collect issues via api	8 days ago
 format-json.sh	step-0: collect issues via api	8 days ago
 presentation-1.pdf	step-0: collect issues via api	8 days ago

README.md 

Project

Selected repository: <https://github.com/PowerShell/PowerShell>.

Data Collection - Get issues

Executable File | 11 lines (10 sloc) | 241 Bytes

```
1  #!/bin/sh -eux
2
3  page=0
4  while true; do
5      page=$((page + 1))
6      sleep 2
7      gh api \
8          -H "Accept: application/vnd.github+json" \
9          "/repos/PowerShell/PowerShell/issues?state=all&per_page=100&
10         > "issues/${page}.json"
11  done
```

```
[
{
  "active_lock_reason": null,
  "assignee": null,
  "assignees": [],
  "author_association": "NONE",
  "body": "### Prerequisites\n\n- [X] Write a descriptive title.\n- [X] Make sure you are a\n  "closed_at": null,
  "comments": 0,
  "comments_url": "https://api.github.com/repos/PowerShell/PowerShell/issues/18550/comments",
  "created_at": "2022-11-13T22:43:25Z",
  "events_url": "https://api.github.com/repos/PowerShell/PowerShell/issues/18550/events",
  "html_url": "https://github.com/PowerShell/PowerShell/issues/18550",
  "id": 1447123128,
  "labels": [
    {
      "color": "CAD1A6",
      "default": false,
      "description": "The issue is new and needs to be triaged by a work group.",
      "id": 2674564170,
      "name": "Needs-Triage",
      "node_id": "MDU6TG6fZWwyNjc0NTY0MTcw",
      "url": "https://api.github.com/repos/PowerShell/PowerShell/labels/Needs-Triage"
    }
  ]
}
```

Data Collection - Get comments

```
1  #!/bin/sh -eux
2
3  cat issue-numbers.lst | while read -r issue_number; do
4      page=0
5      while true; do
6          page=$((page + 1))
7          sleep 1
8          gh api \
9              -H "Accept: application/vnd.github+json" \
10             "/repos/PowerShell/PowerShell/issues/${issue_number}/comments?per_page=100&page=${page}" \
11             > "comments/issue-${issue_number}-page-${page}.json"
12
13             if jq -er 'length == 0' < "comments/issue-${issue_number}-page-${page}.json"; then
14                 rm "comments/issue-${issue_number}-page-${page}.json"
15                 break
16             fi
17         done
18     done
```

```
1  [
2      {
3      "author_association": "MEMBER",
4      "body": "@palladia Got master reset, want to rebase `dev/debug` and open a new PR?\n",
5      "created_at": "2016-01-19T18:18:51Z",
6      "html_url": "https://github.com/PowerShell/PowerShell/pull/10#issuecomment-172939510",
7      "id": 172939510,
8      "issue_url": "https://api.github.com/repos/PowerShell/PowerShell/issues/10",
9      "node_id": "MDEyOklzc3VlQ29tbWVudDE3MjkzOTUxMA==",
10     "performed_via_github_app": null,
11     "reactions": {
12         "+1": 0,
13         "-1": 0,
14         "confused": 0,
15         "eyes": 0,
16         "heart": 0,
17         "hooray": 0,
18         "laugh": 0,
19         "rocket": 0,
20         "total_count": 0,
21         "url": "https://api.github.com/repos/PowerShell/PowerShell/issues/comments/172939510",
22     },
23     "updated_at": "2016-01-19T18:18:51Z",
24     "url": "https://api.github.com/repos/PowerShell/PowerShell/issues/comments/172939510"
```



Quality of Preprocessing

```
5 def main() -> None:
6     for issues_path in sorted(glob.glob("step-0-raw-data/i
7         with open(issues_path, mode="r", encoding="utf-8")
8             issues = json.load(issues_file)
9
10    for issue in issues:
11        if "pull_request" in issue:
12            continue
13
14        print(issue["number"])
15        issue = {
16            "_url": f"https://github.com/PowerShell/Po
17            "number": issue["number"],
18            "state": issue["state"],
19            "title": issue["title"],
20            "body": issue["body"],
21            "created_at": issue["created_at"],
22            "closed_at": issue["closed_at"],
23            "updated_at": issue["updated_at"],
24            "author": issue["user"]["login"],
25            "comments": [],
26        }
27
28    for comments_path in sorted(
29        glob.glob(
30            f"step-0-raw-data/comments/issue-{issu
31    )
32    ,
```

```
1  {
2      "_url": "https://github.com/PowerShell/PowerShell/issues/10000",
3      "author": "VIKITALA",
4      "body": "I am trying to get column data and assign it to a variable which I want to l
5      "closed_at": "2019-06-25T07:30:37Z",
6      "comments": [
7          {
8              "author": "VIKITALA",
9              "author_association": "NONE",
10             "body": "I have managed to get only data in the column without column headers. Bel
11             "created_at": "2019-06-25T07:33:30Z",
12             "updated_at": "2019-06-25T07:33:30Z"
13         }
14     ],
15     "created_at": "2019-06-25T07:10:30Z",
16     "number": 10000,
17     "state": "closed",
18     "title": "How can I get only data without column headers when i use Invoke-Sqlcmd",
19     "updated_at": "2019-06-28T12:53:33Z"
20 }
```

Quality of Labeling

Excel

2022-fall-ensf-612-project-labeling - Saved

Search (Alt + Q)

File

Home

Insert

Draw

Page Layout

Formulas

Data

Review

View

Automate

Help

Calibri

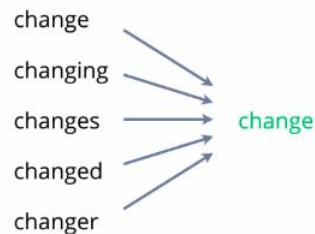
11

B

Quality of Features

- Text extracted from the Github pages
- Text Preprocessing:
 - Hyperlinks and Mentions
 - Stopwords
 - Spelling correction
 - Stemming/Lemmatization
- Extracting features with Bag of words

Stemming vs Lemmatization





ML Model & Training

- TF-IDF (Term Frequency - Inverse Document Frequency)
 - Term frequency: summarizes how often a given word appears within a document
 - Inverse document frequency: down scales word that appear a lot across documents
- Naive Bayes Classifier
- SVM