Regression Case Study Intro

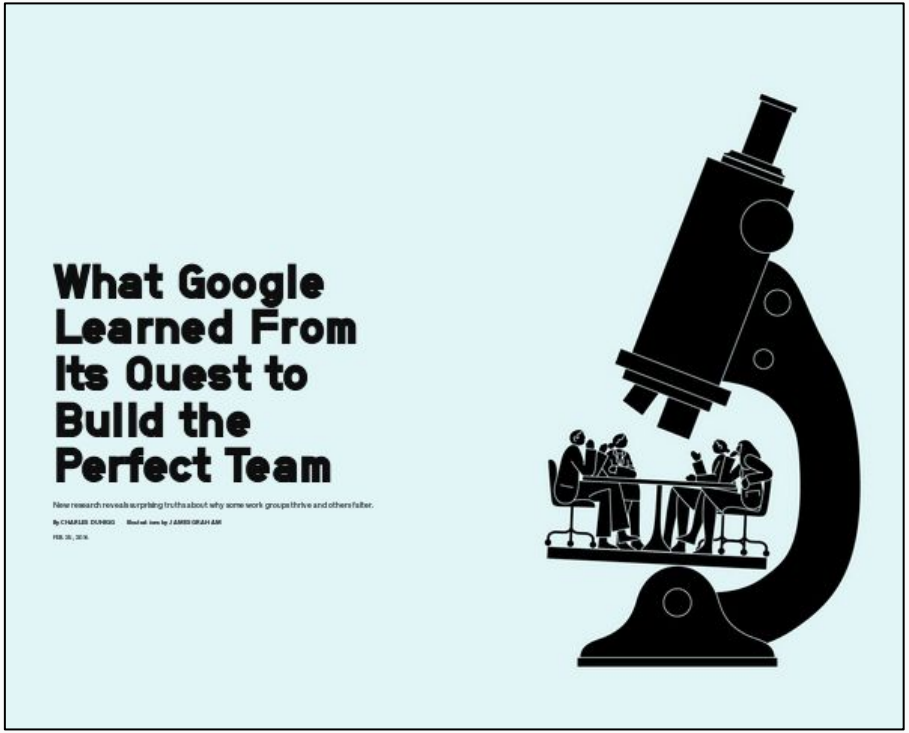
Kristie Wirth
Frank Burkholder

galvanize

Overview

- Congratulations on sticking with it (really!).
- What makes a good team
- Project workflow ideas
- Working with missing values
- Collaborating with Git
- Have fun!

What makes teams succeed



https://www.nytimes.com/2016/02/28/magazine/what-google-learned-from-its-quest-to-build-the-perfect-team.html

“Who” didn’t seem to matter

galvanize

‘We had lots of data, but there was nothing showing that a mix of specific personality types or skills or backgrounds made any difference. The “who” part of the equation didn’t seem to matter.’

Goal: Collective intelligence

galvanize

> ‘As long as everyone got a chance to talk, the team did well. But if only one person or a small group spoke all the time, the collective intelligence declined.’

Goal: Collective intelligence

galvanize

> ‘As long as everyone got a chance to talk, the team did well. But if only one person or a small group spoke all the time, the collective intelligence declined.’

Please take the time to listen to each other. You’ll learn something, your team will likely do better, and you’ll like each other more.

CRoss-Industry Standard Process for Data Mining (CRISP-DM)

- Business understanding
  - Review project objectives and planned use cases
  - Define the problem to be explored
  - Create a basic plan to achieve the objectives
- Data understanding
  - Data collection
  - Explore data - visualization, identify quality problems, find interesting patterns
- Data preparation
  - Cleaning and transforming data into something usable for modeling purposes
- Modeling
  - Testing models
  - Testing hyperparameters
- Evaluation
  - Evaluate the model's performance
  - Review the steps taken to confirm they meet the objectives
- Deployment
  - Creating something for the end user - writing a report, developing an interactive web app, making a presentation

General Case Study Tips

- Remember - this is the first time you're working through a whole project by yourselves! (Many) things will probably go wrong…
- Divide & conquer - assign people to work on different areas such as...
  - EDA
  - Data cleaning
  - Generating new features
  - Creating/testing models
- First make it work, then make it better
  - First, make your data usable in whatever way is fastest
  - Then, make a terrible model that runs
  - Have a minimal viable product before you start trying fancier ideas
- Set a code freeze time for when you stop testing models and start tying everything together - save at least an hour for this at the end of the day

Working with Null Values

- Some nulls have meaning - e.g., missing number of clicks means 0 clicks
  - Fill these with what value makes sense
- Drop all rows with nulls
  - Worst option
  - Pandas .dropna
- Fill with the column mean
  - Okay option
  - Pandas .fillna
- Create regressions to predict missing values and fill with these predictions
  - Awesome option for the overachievers!
  - See next slide

Pseudocode - Filling Nulls with Regression

```
for column in columns:

    Get indices of rows in which the given column is not null (.index)

    Store the other columns of information for non-null column data (.iloc)

    Store the non-null column data (.iloc)

    Instantiate model to predict values

    Get indices in which the given column is null (.index)

    Store the other columns of information for the null column data (.iloc)

    Predict the values of the given column where it is blank

    Fill in the values that are blank using these predictions (.iloc)
```

Collaborating on Github

Goal: Team members will work on their own parts of the project and then combine all these parts into one repo on Github. At the end of the project, all team members will have an up-to-date fork of the case-study repo.

Case study high-level workflow suggestion

galvanize

1) One team member should fork the case study. This will be called the upstream repo.
2) All other team members should fork the upstream repo.
3) Everyone clones their own forked repos down to their local machines.
4) On your local machine, create and checkout a branch to work on. No one works on the master branch, even the upstream owner!
5) Do your work.
6) Push your branch to your fork.
7) Issue a pull request to merge your fork with the upstream repo. The owner of the upstream repo will accept your pull request and merge it into the upstream master branch, then delete your branch.

In this process, everything will eventually be merged into the master branch in the upstream repo. This will be the "production" code that everyone will have a copy of in the end.

1. One person creates upstream

Frank-W-B / **regression-case-study** Private
forked from zipfian/regression-case-study

Watch 16 | Star 0 | Fork 223

Code | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

Case study for regression (week 4).

Edit

Add topics

16 commits | 1 branch | 0 releases | 6 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

This branch is 2 commits ahead of zipfian:master. Pull request Compare

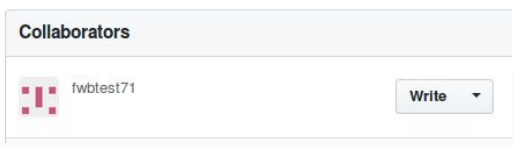
Frank-W-B Add fb solution files — Latest commit 7f39063 9 hours ago

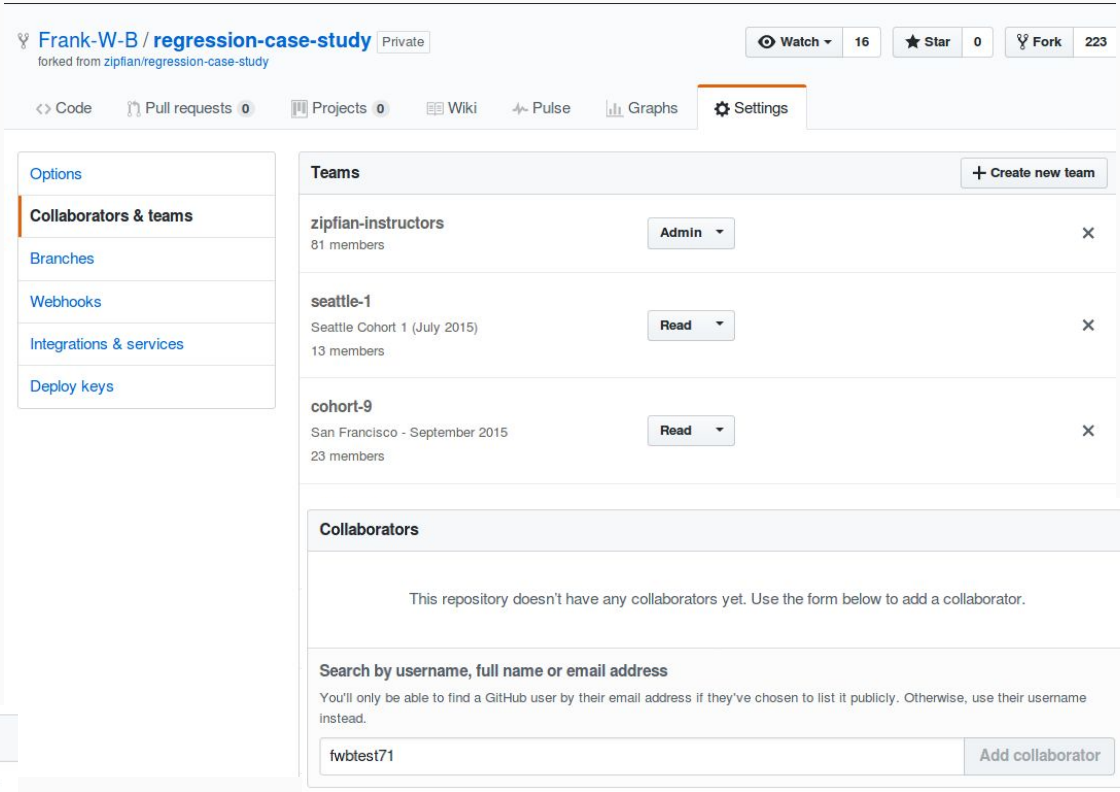
data	Add solution files	3 months ago
images	minor changes	2 years ago
soln_fb	Add fb solution files	9 hours ago
src	Add solution files	3 months ago
.gitignore	create .gitignore so we don't track Train.csv	6 months ago
README.md	Updating Scary Wording about submission	5 months ago
config.yaml	remove default channel/username in config.yaml	8 months ago
score_model.py	Initial commit.	2 years ago

README.md

1. Upstream owner adds collaborators

a. Settings
b. Collaborators & Teams
c. Scroll down to bottom
d. Add username(s) of collaborators
e. Send invitation (permission level Write)
f. Person invited needs to check their email associated with GH account to accept invitation.
g. When they accept invitation will look like this in upstream repo:





2. Collaborator forks upstream repo

fwbtest71 / regression-case-study Private
forked from Frank-W-B/regression-case-study

Unwatch 1 | Star 0 | Fork 224

Code | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

Case study for regression (week 4).

Edit

Add topics

16 commits | 1 branch | 0 releases | 6 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

This branch is even with Frank-W-B:master. Pull request Compare

Frank-W-B Add fb solution files — Latest commit 7f39063 9 hours ago

data	Add solution files	3 months ago
images	minor changes	2 years ago
soln_fb	Add fb solution files	9 hours ago
src	Add solution files	3 months ago
.gitignore	create .gitignore so we don't track Train.csv	6 months ago
README.md	Updating Scary Wording about submission	5 months ago
config.yaml	remove default channel/username in config.yaml	8 months ago
score_model.py	Initial commit.	2 years ago

README.md

3. Everyone clones their repo (and non-main people add remote URL)

galvanize

`$ git remote -v` to see the remote, called origin, associated with the Github repo they cloned the repo from. It should show origin and nothing else (e.g., no web address listed)..

Everyone else should `$ git remote add upstream (main person's repo URL)` to set the URL for your repo to the main person's repo URL.

```
[mbp:~ frank.burkholder$
[mbp:~ frank.burkholder$ git clone https://github.com/fwbtest71/regression-case-study.git
Cloning into 'regression-case-study'...
remote: Counting objects: 67, done.
remote: Total 67 (delta 0), reused 0 (delta 0), pack-reused 67
Unpacking objects: 100% (67/67), done.
[mbp:~ frank.burkholder$ cd regression-case-study/
✔ ~/regression-case-study [master {origin/master}|✔]
[00:24 $ git remote -v
origin  https://github.com/fwbtest71/regression-case-study.git (fetch)
origin  https://github.com/fwbtest71/regression-case-study.git (push)
✔ ~/regression-case-study [master {origin/master}|✔]
[00:24 $ git remote add upstream https://github.com/Frank-W-B/regression-case-study.git
✔ ~/regression-case-study [master {origin/master}|✔]
[00:27 $ git remote -v
origin  https://github.com/fwbtest71/regression-case-study.git (fetch)
origin  https://github.com/fwbtest71/regression-case-study.git (push)
upstream        https://github.com/Frank-W-B/regression-case-study.git (fetch)
upstream        https://github.com/Frank-W-B/regression-case-study.git (push)
```

4. Everyone makes their own branch to work in

```
[00:34 $
✔ ~/regression-case-study [master {origin/master}|✔]
[00:34 $ git branch
* master
✔ ~/regression-case-study [master {origin/master}|✔]
[00:34 $ git branch frank
✔ ~/regression-case-study [master {origin/master}|✔]
[00:34 $ git branch
  frank
* master
✔ ~/regression-case-study [master {origin/master}|✔]
[00:34 $ git checkout frank
Switched to branch 'frank'
✔ ~/regression-case-study [frank L|✔]
[00:34 $ git branch
* frank
  master
```

For this example...

- git branch frank creates the branch
- git checkout frank lets you work on that branch
- git branch shows what branch you're on

5. Do your work!

```python
import numpy as np

from sklearn.datasets import load_boston
from sklearn.ensemble import RandomForestRegressor
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import Imputer
from sklearn.model_selection import cross_val_score

rng = np.random.RandomState(0)

dataset = load_boston()
X_full, y_full = dataset.data, dataset.target
n_samples = X_full.shape[0]
n_features = X_full.shape[1]

# Estimate the score on the entire dataset, with no missing values
estimator = RandomForestRegressor(random_state=0, n_estimators=100)
score = cross_val_score(estimator, X_full, y_full).mean()
print("Score with the entire dataset = %.2f" % score)

# Add missing values in 75% of the lines
missing_rate = 0.75
n_missing_samples = np.floor(n_samples * missing_rate)
missing_samples = np.hstack((np.zeros(n_samples - n_missing_samples, dtype=np.bool),
                             np.ones(n_missing_samples, dtype=np.bool)))
rng.shuffle(missing_samples)
missing_features = rng.randint(0, n_features, n_missing_samples)

# Estimate the score without the lines containing missing values
X_filtered = X_full[~missing_samples, :]
y_filtered = y_full[~missing_samples]
estimator = RandomForestRegressor(random_state=0, n_estimators=100)
score = cross_val_score(estimator, X_filtered, y_filtered).mean()
print("Score without the samples containing missing values = %.2f" % score)

# Estimate the score after imputation of the missing values
X_missing = X_full.copy()
X_missing[np.where(missing_samples)[0], missing_features] = 0
y_missing = y_full.copy()
estimator = Pipeline([("imputer", Imputer(missing_values=0, strategy="mean", axis=0)),
                      ("forest", RandomForestRegressor(random_state=0, n_estimators=100))])
score = cross_val_score(estimator, X_missing, y_missing).mean()
print("Score after imputation of the missing values = %.2f" % score)
~
~
"imputing_values.py" [New] 43L, 1863C written
```

6. Push your branch to your fork, but WAIT!

Before you add, commit, and push you should really make sure your repo is current with the upstream master branch! Basically, always do a git pull before you do a git push to avoid errors/merge conflicts.

```
✔ ~/regression-case-study [frank L|…1]
01:45 $ ls
README.md          config.yaml        data               images             imputing_values.py score_model.py     soln_fb            src
✔ ~/regression-case-study [frank L|…1]
01:45 $ git pull upstream master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/Frank-W-B/regression-case-study
 * branch            master     -> FETCH_HEAD
 * [new branch]      master     -> upstream/master
Updating 7f39063..9028a12
Fast-forward
 data_cleaning.py | 4 ++++
 1 file changed, 4 insertions(+)
 create mode 100644 data_cleaning.py
✔ ~/regression-case-study [frank L|…1]
01:45 $ ls
README.md          data               images             score_model.py     src
config.yaml        data_cleaning.py   imputing_values.py soln_fb
```

When pulled upstream master branch it downloaded a data_cleaning file that the upstream owner merged onto master branch

6. OK, now push your branch to your fork

```
✔ ~/regression-case-study [frank L|…1]
[01:53 $ git status
On branch frank
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        imputing_values.py

nothing added to commit but untracked files present (use "git add" to track)
✔ ~/regression-case-study [frank L|…1]
[01:53 $ git add imputing_values.py
✔ ~/regression-case-study [frank L|●1]
[01:53 $ git commit -m "Add imputing values file"
[frank 8d622f7] Add imputing values file
 1 file changed, 43 insertions(+)
 create mode 100644 imputing_values.py
✔ ~/regression-case-study [frank L|✔]
[01:53 $ git push origin frank
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 924 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/fwbtest71/regression-case-study.git
   9028a12..8d622f7  frank -> frank
✔ ~/regression-case-study [frank L|✔]
01:54 $
```

Add, commit like usual.

Push the branch you made to origin

AHHHHH I GOT A MERGE CONFLICT!

```
windows@WIN81 /c/Projects/resolve-conflicts-demo (master)
$ git checkout branch-b
Switched to branch 'branch-b'

windows@WIN81 /c/Projects/resolve-conflicts-demo (branch-b)
$ git status
# On branch branch-b
nothing to commit, working directory clean

windows@WIN81 /c/Projects/resolve-conflicts-demo (branch-b)
$ git merge branch-a
Auto-merging demo-file.md
CONFLICT (content): Merge conflict in demo-file.md
Automatic merge failed; fix conflicts and then commit the result.

windows@WIN81 /c/Projects/resolve-conflicts-demo (branch-b|MERGING)
$ git status
# On branch branch-b
# You have unmerged paths.
#   (fix conflicts and run "git commit")
#
# Unmerged paths:
#   (use "git add <file>..." to mark resolution)
#
#       both modified:      demo-file.md
#
no changes added to commit (use "git add" and/or "git commit -a")

windows@WIN81 /c/Projects/resolve-conflicts-demo (branch-b|MERGING)
$ notepad demo-file.md
```

- Take a deep breath. You got this.
- This just means that two people edited the same file and it's confused about which one to keep.
- If it's on a file that you swear you didn't change and you just want the master version to overwrite yours…
  - git checkout filename
  - git pull upstream master
- If you made some changes on this file and you want to keep your changes AND get the other person's changes…
  - git stash
  - git pull upstream master
  - git stash pop
  - Look at the file and fix any overlapping changes

If you get stuck in the world of Vim...

```
apone #3.0 vi (0)
14          - present
15          - user: root
16          - enc: ssh-dss
17
18 <<<<<<< HEAD
19     thatch:
20       ssh_auth:
21         - present
22         - user: root
23         - source: salt://ssh_keys/thatch.id_rsa.pub
24
25 =======
26     me@mykey:
27       ssh_auth:
28         - present
29         - user: root
30         - enc: ssh-dss
31         - source: salt://keys/mykey.pub
32 >>>>>>> wip: Use the file server to transfer ssh keys
33 '''
34
35 # Import python libs
36 import re
37
38 def _present_test(user, name, enc, comment, options, source, config):
%1 s/s/ssh_auth.py[+] [python] <utf-8,unix> [Git(8ebcf66b)]          17,0-1 of 228  6%
[0] 0:vi- 1:django-admin.py  2:sync  3:vi*          @apone 2012-04-21 Sat 18:13 4.76 3.42 1.62
```

- The <<<< HEAD text indicates where you have conflicting text in your document that you need to address
- You can switch to INSERT mode (so you can edit), by typing :i
- When you are done, press ESC to return to normal mode
- Type :w and hit enter to write your changes
- Type :q to quit Vim
- Pro tip: Learn Vim basics OR learn the basics of another text editor and set that as your default (emacs, nano, etc.)

6. See that your branch exists on your fork

galvanize

New branch on GH

New files

fwbtest71 / regression-case-study Private
forked from Frank-W-B/regression-case-study

Unwatch 1 | Star 0 | Fork 224

Code | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

Case study for regression (week 4).

Edit

Add topics

18 commits | 2 branches | 0 releases | 6 contributors

Branch: frank | New pull request | Create new file | Upload files | Find file | Clone or download

This branch is 1 commit ahead of Frank-W-B:master. Pull request Compare

Frank-W-B Add imputing values file — Latest commit 8d622f7 a minute ago

data	Add solution files	3 months ago
images	minor changes	2 years ago
soln_fb	Add fb solution files	11 hours ago
src	Add solution files	3 months ago
.gitignore	create .gitignore so we don't track Train.csv	6 months ago
README.md	Updating Scary Wording about submission	5 months ago
config.yaml	remove default channel/username in config.yaml	8 months ago
data_cleaning.py	Add data cleaning file	an hour ago
imputing_values.py	Add imputing values file	a minute ago
score_model.py	Initial commit.	2 years ago

7. Collaborator issues a pull request to upstream

Frank-W-B / regression-case-study Private
forked from zipfian/regression-case-study

Unwatch 17 | Star 0 | Fork 224

Code | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs

Collaborator's repo

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base fork: Frank-W-B/regression-case-... | base: master | ... | head fork: fwbtest71/regression-case-st... | compare: frank

Able to merge. These branches can be automatically merged.

Upstream/ main person's repo

Add imputing values file

Write | Preview

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Allow edits from maintainers. Learn more

Create pull request

Reviewers
No reviews—request one

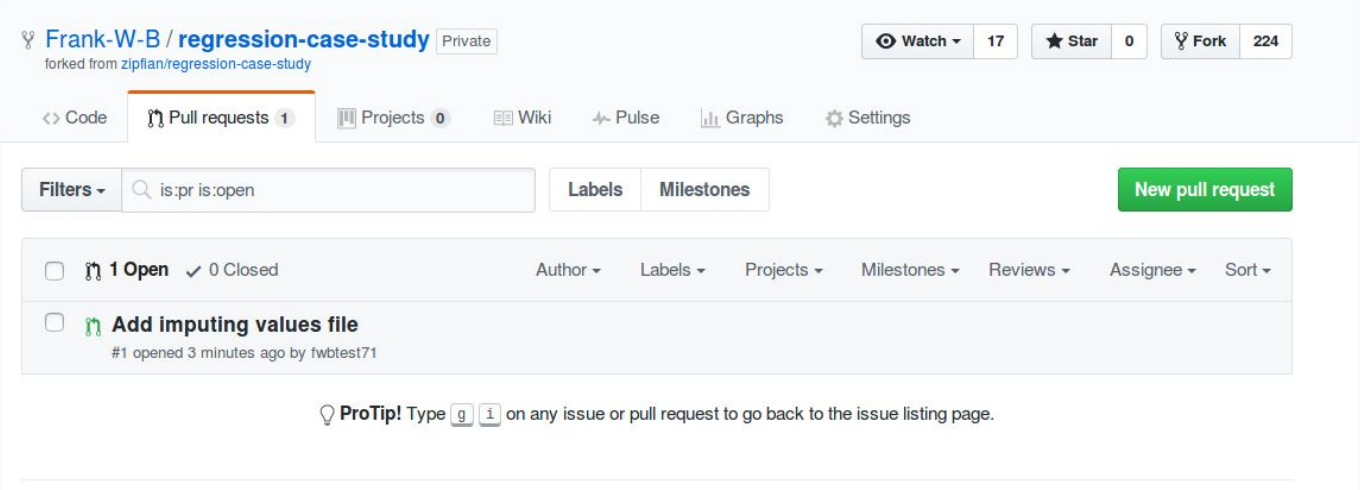
Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

7. Upstream sees the pull request...



7. And chooses to merge it

Frank-W-B / **regression-case-study** Private
forked from zipfian/regression-case-study

Watch 17 | Star 0 | Fork 224

Code | Pull requests 1 | Projects 0 | Wiki | Pulse | Graphs | Settings

Add imputing values file #1

Edit

Open **fwbtest71** wants to merge 1 commit into `Frank-W-B:master` from `fwbtest71:frank`

Conversation 0 | Commits 1 | Files changed 1 | +43 −0

fwbtest71 commented 4 minutes ago

No description provided.

`Add imputing values file` 8d622f7

Add more commits by pushing to the **frank** branch on **fwbtest71/regression-case-study**.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request | or view command line instructions.

Reviewers
No reviews—request one

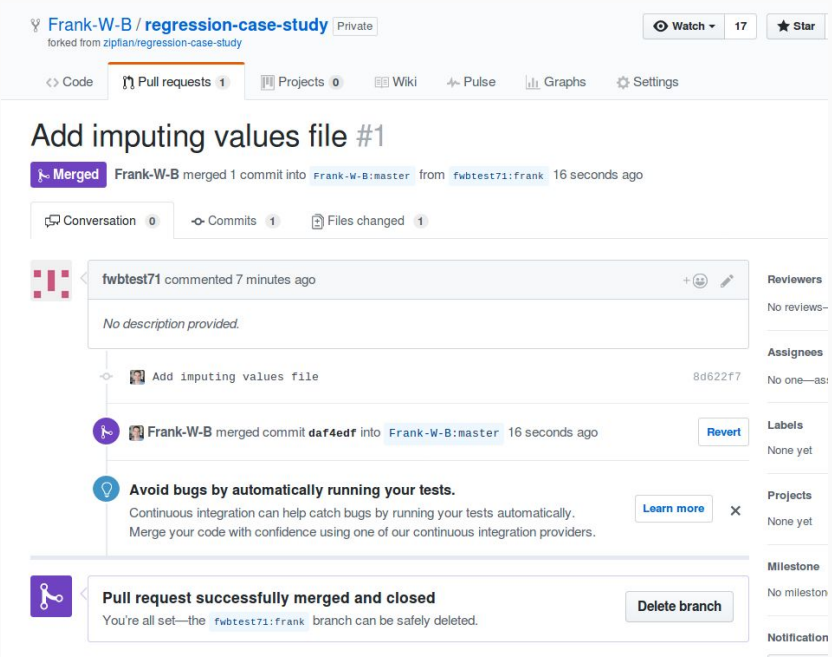
Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone

End of the Day



- Push all your changes
- Open pull request & merge
  - Base fork = main person's master repo
  - Head fork = your branch
- After seeing confirmation page, delete your branch
- For collaborators to get all the final changes, both locally and remotely…
  - `$ git pull upstream master`
  - `$ git push origin master`
- We will have 5 minute presentations on your process, your outcomes, what went well, learning experiences, etc.
- **Reconvene at 4pm to present!**

More Git Resources

How to collaborate on git:
https://code.tutsplus.com/tutorials/how-to-collaborate-on-github--net-34267

The difference between origin and upstream on github:
http://stackoverflow.com/questions/9257533/what-is-the-difference-between-origin-and-upstream-on-github

Resolving merge conflicts using the command line:
https://help.github.com/articles/resolving-a-merge-conflict-using-the-command-line/

Resolving merge conflicts on Github:
https://help.github.com/articles/resolving-a-merge-conflict-on-github/