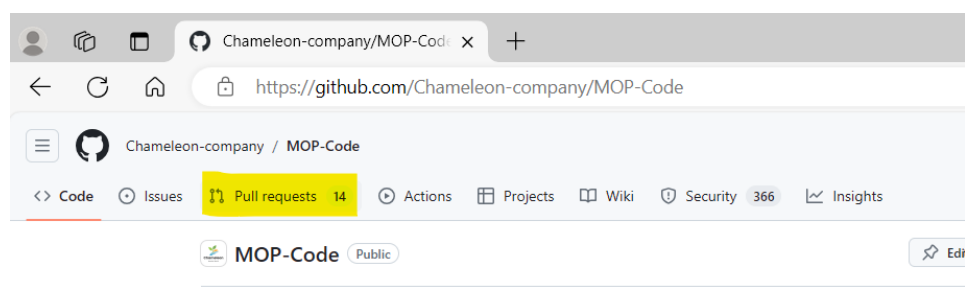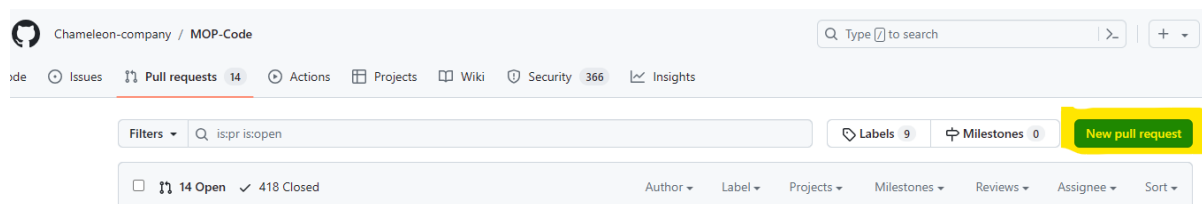The following documentation is a pictorial guide of how to make and how to review a pull request.

Making a Pull Request

Select the pull request tab
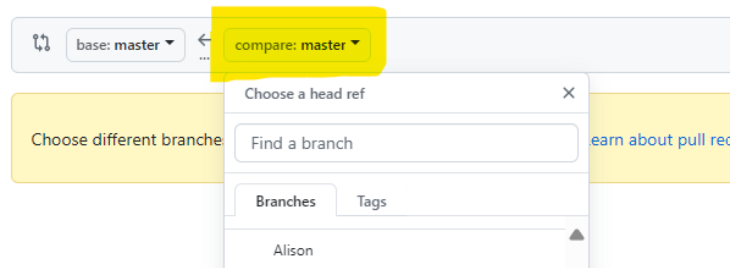

Select open new pull request

Select your branch from the drop down list in "compare"

Select "create pull request"

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks or learn more about diff comparisons.

base: master ▾ ← compare: Alison ▾ ✓ **Able to merge.** These branches can be automatically merged.

Discuss and review the changes in this comparison with others. Learn about pull requests **Create pull request**

⦿ **3 commits** ⊞ **3,425 files changed** ⦿ **1 contributor**

Add title and description for your pull request

Title must include which team you are (DATA SCIENCE or WEBDEV)

Name of what you are working on (This should match what is on your Trello card)

Percentage of completion of your use case from Trello

Description must include reason for pull, e.g. use case completed, Sprint 1 pull request.

Description must include a list of the changes that have been made to your folder since last making a pull request.

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. Learn more about diff com

base: master ▾ ← compare: Alison ▾ ✓ **Able to merge.** These branches can be automatically merged.

Add a title

DATA SCIENCE TEAM - Argyle Square Bin Fill Level New Use Case - 80% complete

Add a description

Write | Preview | H B I ≔ <> 🔗 | ≝ ≔ ☑ | 📎 @ 🗗 ↩ ⬚

Week 5 pull request for review from Sprint 1

Use case includes:
- Use case introduction
- Import data using API v2.1 GET request with /exports extension
- Filtering of dataframes
- Use of date time to resample data by different time increments
- Box plots from a dataframe to view datasets
- Scatterplots from a dataframe to identify variable relationships
- Pairplots from a dataframe to identify variable relationships
- Use Pearson correlation heatmap to identify linear variable relationships
- Use distance calculation to identify non-linear variable relationships
- Linear regression for predictive analysis
- Random Forest for predictive analysis
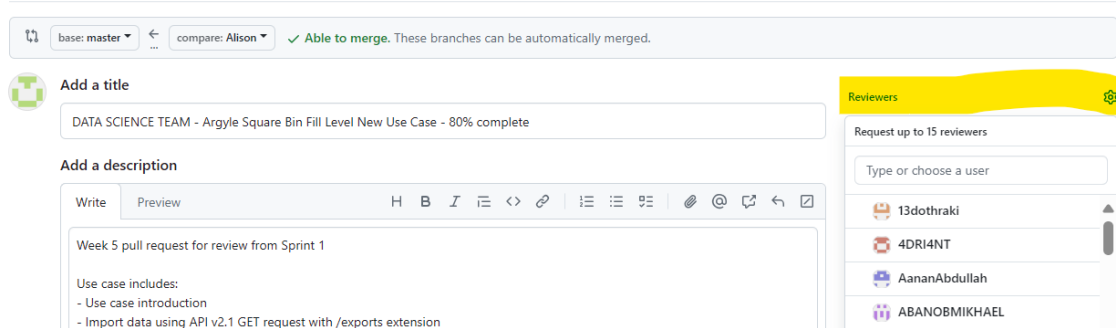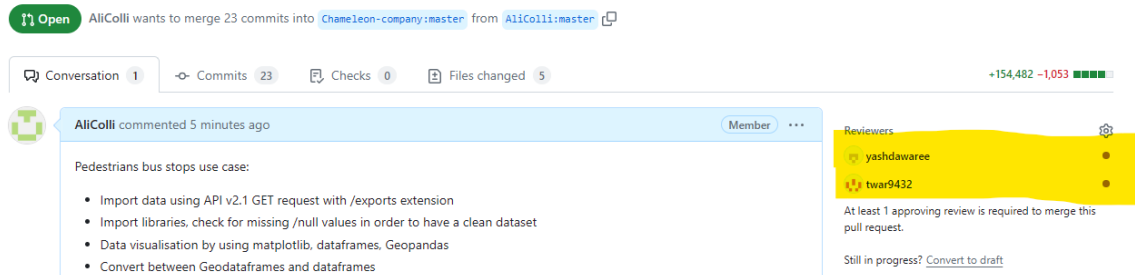- Calculate feature importance, r-squared value and accuracy from predictive analysis

⊞ Markdown is supported | 🖼 Paste, drop, or click to add files

Select reviewers and choose 2 reviewers

One reviewer can be any member of your team, the other MUST be a leadership student.

You should message students fist to check if they are happy to review for you. You can find a list of GitHub user names to add reviewers on the Melbourne Open Playground team list in Teams.

Check you have added two reviewers

Create your pull request

Create your pull request by pressing the green "create pull request" button

After Making a Pull Request

Check your emails

You will be emailed when your pull request receives comments or approvals.

[Chameleon-company/MOP-Code] Use case updated (PR #476)

This sender notifications@github.com is from outside your organisation. Attachments and pictures have been blocked.
Block sender | Show blocked content

Check GitHub Conversation for Feedback

After reviewing, your team members will leave you feedback in GitHub.

Open your pull request, and scroll down to the comments section to see this. If you need to make changes you will need to do so in your IDE and then push to GitHub again.

DATA SCIENCE Pedestrians Bus Stops Use Case 100% completed #489

Edit | <> Code ▾

Open AliColli wants to merge 23 commits into Chameleon-company:master from AliColli:master

💬 Conversation 4   ⟲ Commits 23   ☑ Checks 0   ⊡ Files changed 5

+154,482 −1,053

AliColli commented 2 days ago   Member ···

Pedestrians bus stops use case:

- Import data using API v2.1 GET request with /exports extension
- Import libraries, check for missing /null values in order to have a clean dataset

Reviewers   ⚙
twar9432   ⟲ ⊡
yashdawaree   ●

Requested changes must be addressed to merge this pull request.

twar9432 requested changes 2 days ago

View reviewed changes

twar9432 left a comment   Member ···

Overall, the use case looks good. There's clear comments of the code and good analysis.
Note: There's a plot shown as a 'map' with the axes as longitude and latitude, however without any background it's difficult to

After making changes

Once you have updated your code and pushed again you will see a new comment automatically generated

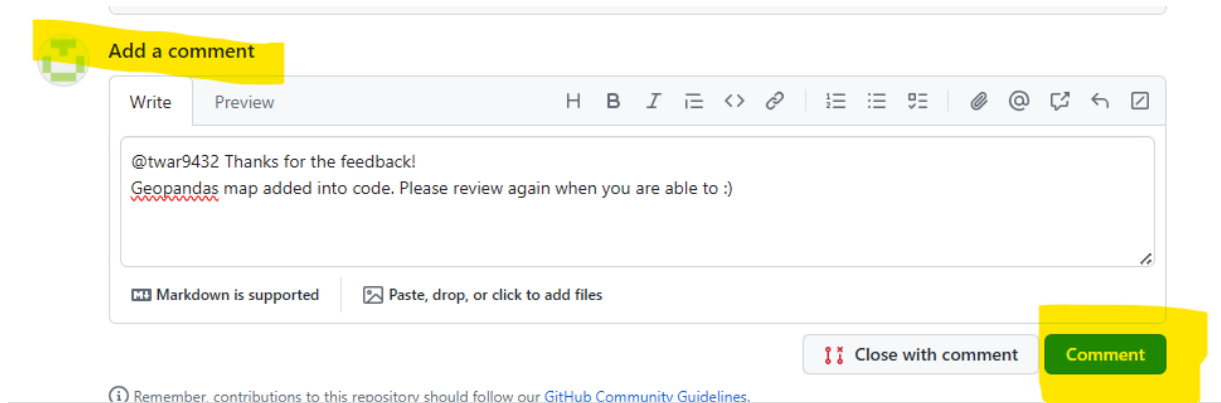AliColli and others added 2 commits 1 minute ago

○ 🟢 Pedestrian Counts and Bus Stops API update - folium map of overlap lo... ···   ● 115bcc6

○ 🟢 Merge branch 'Chameleon-company:master' into master   Verified   ● 324790d

You will need to also add a comment to email your reviewers so that they know you have updates your work and that it is ready to be re-reviewed.

Be sure to write @ next to their GitHub name so they receive an email notification.

Be sure to press the green "comment" option to keep your pull request open.

Check review status

Check both reviewers have approved your pull request. Look for green ticks against reviewers names

Merge pull request

Once you have two approvals, Merge your pull request!

Reviewing a Pull request

View open pull requests

On GitHub, select "Pull requests"
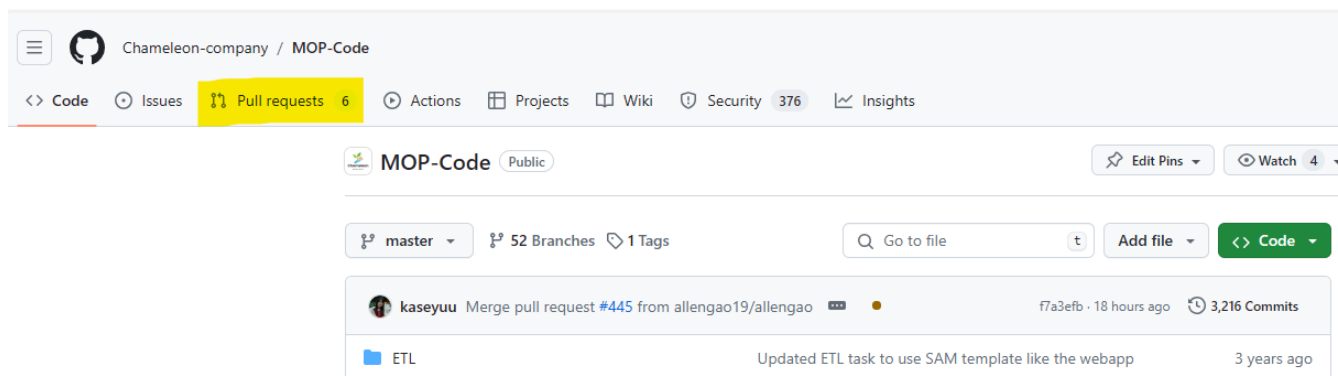
Select a Use Case to review

Select the "Reviews" tab and the "Awaiting review from you"
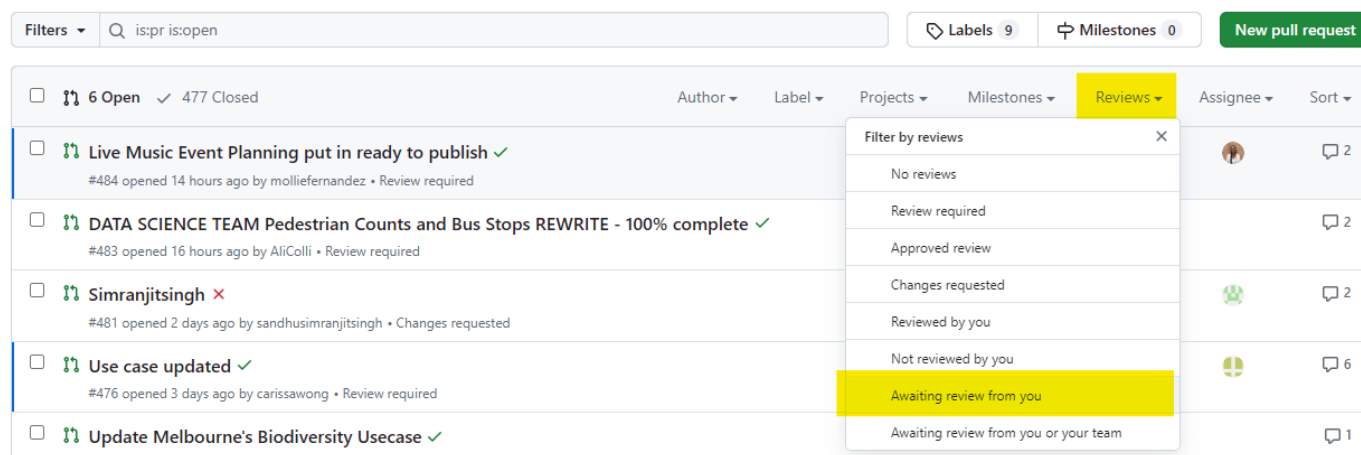
OR
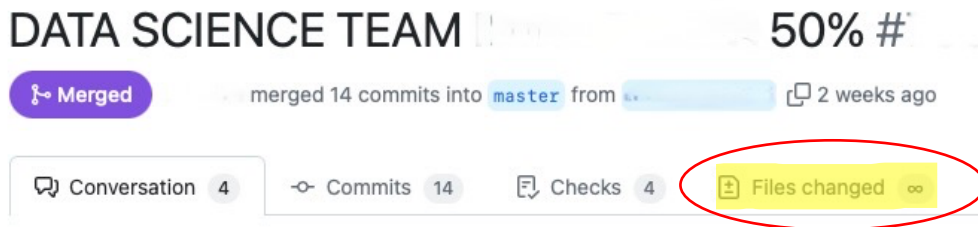
Check your emails for review requests

Follow the link to the use case from your email

City of Melbourne
Open Data

Chameleon
Smarter World

DATA SCIENCE Team

DEAKIN
UNIVERSITY
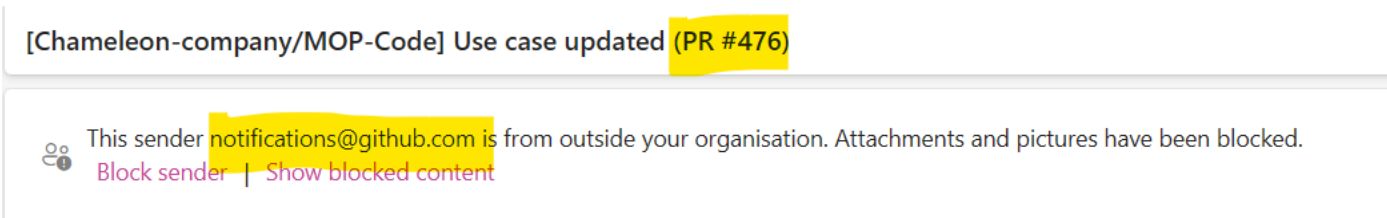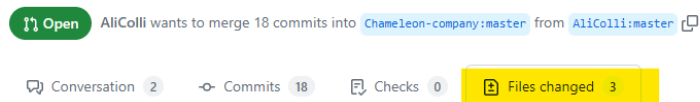
View the Use Case

When the use case is open, first thing, and the most important thing is to check is whether there is an infinity number of files being changed. If the request is for an infinity number of files to be merged, please do not approve this pull request, and let the requestor know to review this and redo the pull again.

A correct pull should look like below, showing a reasonable number of files changed.

When the use case is open, select the files changed tab. If the use case is large, you will get a load diff alert. In this case, select the three dots on the right hand side and go to view file.

Things to check for in the "Preview" view:

Dataset import

– Must be API v2.1

– API key must NOT be visible

IMPORT DATASETS

Import datasets using API v2.1 GET request with /exports/

```python
from io import StringIO

# Function to collect datasets using API
def datasetcollect(dataset_id):
    base_url = 'https://data.melbourne.vic.gov.au/api/explore/v2.1/catalog/datasets/'
    #apikey = " "
    dataset_id = dataset_id
    format = 'csv'

    url = f'{base_url}{dataset_id}/exports/{format}'
    params = {
        'select': '*',
        'limit': -1,  # all records
        'lang': 'en',
        'timezone': 'UTC',
        #'api_key': apikey
    }

    # GET request
    response = requests.get(url, params=params)

    if response.status_code == 200:
        # StringIO to read the CSV data
        url_content = response.content.decode('utf-8')
        dataset = pd.read_csv(StringIO(url_content), delimiter=';')
        return dataset
    else:
        print(f'Request failed with status code {response.status_code}')
```
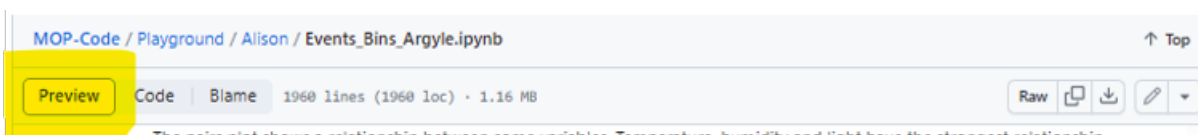
```python
# Import stage activity dataset
dataset_id = 'meshed-sensor-type-3'
stage_activity_all = datasetcollect(dataset_id)
print(len(stage_activity_all))
stage_activity_all.head(3)
```

Code sections

– Code must be clear and easily readable

– Comments must have a space and capital letter for first word

– Explain code in a way that can be easily followed

```python
# BIN DATASET PREPROCESSING
# Filter unwanted values from bin dataset

# Keep only rows with bin sensors in the stage area
filtered_bin_sensor = bin_sensor_cols[bin_sensor_cols["dev_
filtered_bin_sensor.head(3)

# Check max and min values in bin fill levels
# Max and min of filllevel column
print(filtered_bin_sensor['filllevel'].agg(['min', 'max']))
```

Markdown style, colours in graphs and tables

– Adhere to Chameleon Company requirements

– Details in use Case publishing guide on page 3

MOP-Code/datascience/documentation/Use case publishing guide.pdf at master · Chameleon-company/MOP-Code (github.com)

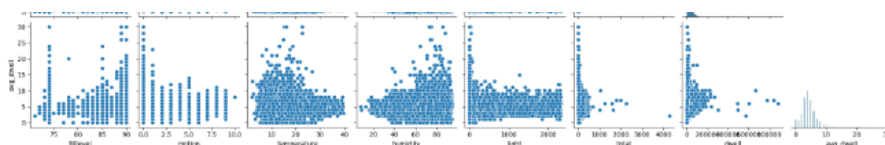**Written explanations**

– Reason for methods chosen

– Explanations of output

Remember: the use case is written

*to teach people how to do this!*

So explanation is important!

The pairs plot shows a relationship between some variables. Temperature, humidity and light have the strongest relationship. The strongest linear relationship is between total and dwell. The pairs plots only compare numerical data.

Pearson correlation is used to determine extent of linear relationship between each pair of variables.

```python
# Calculate correlation heatmap for all columns
matrix = fmd.corr(method = 'pearson').round(2)
mask = np.triu(np.ones_like(matrix, dtype=bool))
sns.heatmap(matrix, annot=True, cmap='viridis_r', mask=mask)

plt.show()
```

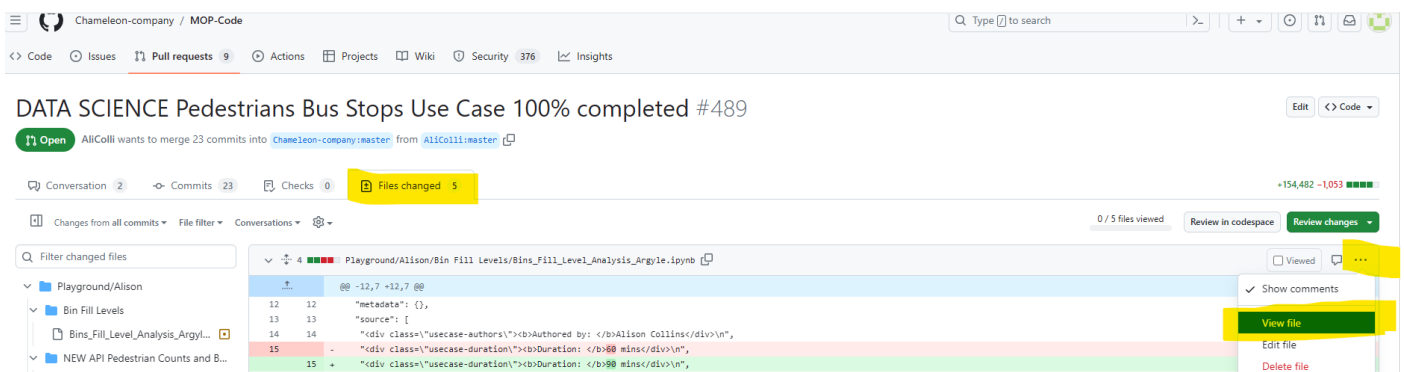## Things to check for in the "Code" view:
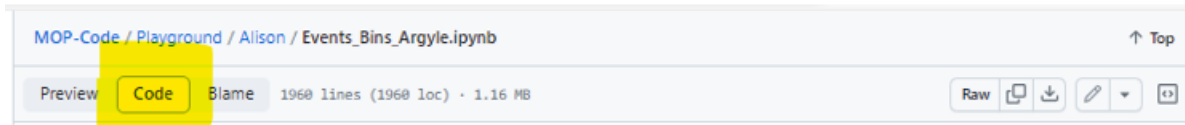
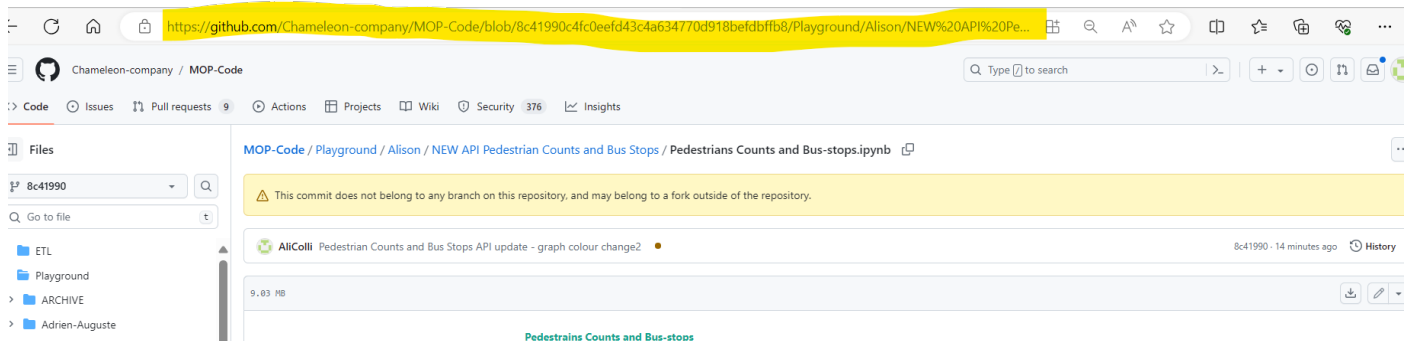**HTML in code**

– HTML as per usecase TEMPLATE

Troubleshooting viewing:

If the Use Case or parts of it (graphs, tables, folium maps) will not render on GitHub, copy the link from the GitHub and open in https://nbviewer.org/

Select the files changed tab, three dots on the right and then view file

Copy the address bar

Paste into https://nbviewer.org/ and press go

nbviewer

A simple way to share Jupyter Notebooks

Enter the location of a Jupyter Notebook to have it rendered here:

https://github.com/Chameleon-company/MOP-Code/blob/8c41990c4fc0eefd43c4a634770d918befdbffb8/Playground/Alison/N   Go!

Further reading

Capstone Pull Requests: 5. Navigating Pull Requests on GitHub | Capstone (verdant-raindrop-f3e404.netlify.app)

Capstone Reviewing Pull Requests: 6. Reviewing Pull Requests | Capstone (verdant-raindrop-f3e404.netlify.app).

Author

Alison Collins, 2024

Katrine Chan, 2024 v2