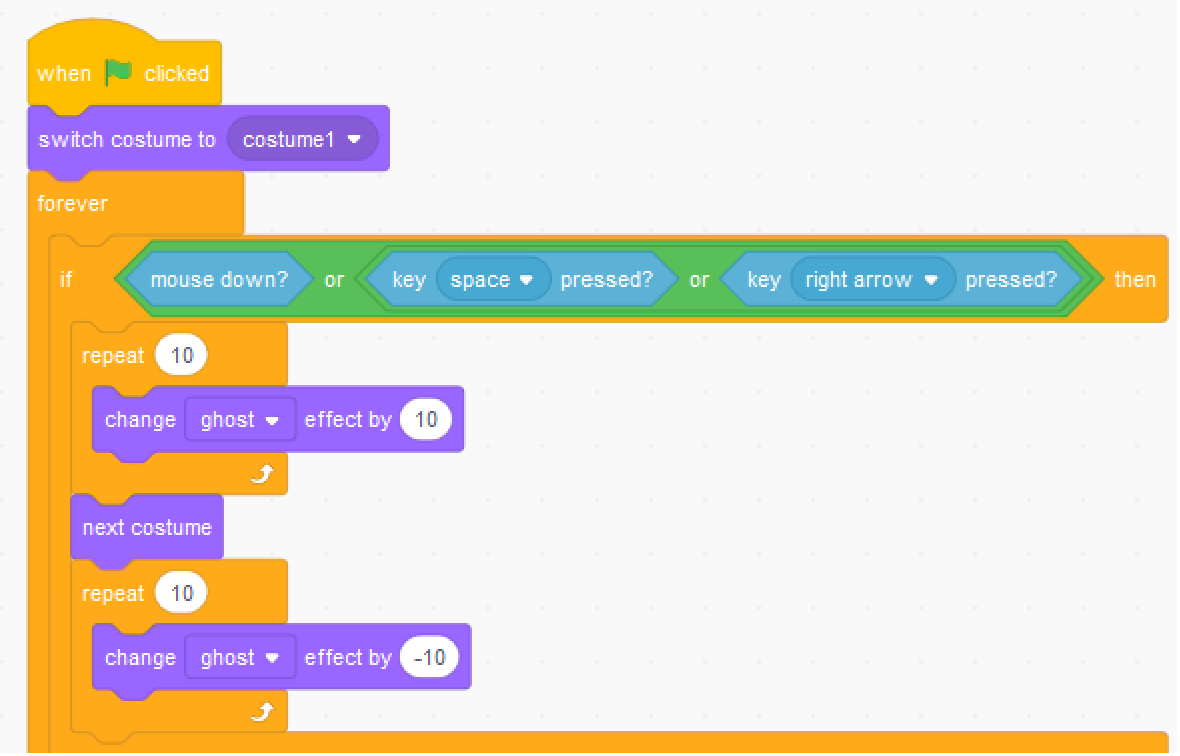
SCRATCH EXPLORER

BY

MATT DAVIDSON, JACOB COHEN, CRYSTAL YU, AND FAISAL ALSALLUM

BACKGROUND:

- Scratch is a block-based visual programming language
- It is a project of the Lifelong Kindergarten Group at the MIT Media Lab. It is provided free of charge.
- Can be used to program a user own interactive stories, games, and animations



```
when ⚑ clicked
switch costume to (costume1 ▾)
forever
  if <<mouse down?> or <<key (space ▾) pressed?> or <key (right arrow ▾) pressed?>>> then
    repeat (10)
      change (ghost ▾) effect by (10)
    ↺
    next costume
    repeat (10)
      change (ghost ▾) effect by (-10)
    ↺
```

DATA USED

- 250K projects from 100K different authors scraped from the Scratch project repository
  - Blocks used (including inputs)
  - Popularity metrics (views, favorites, loves, remixes)
  - Computational Thinking scores (e.g. abstraction, control flow)
- Limitation: the data set is not recent (2016) so it has some unavailable projects as will be shown in the demo
- Limitation: data is “big” (~3.6GB) so hard for many users to access/analyze

data URL: https://github.com/TUDelftScratchLab/ScratchDataset.

Paper published about the dataset: E. Aivaloglou, F. Hermans, J. Moreno-Leon and G. Robles, "A Dataset of Scratch Programs: Scraped, Shaped and Scored," *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, Buenos Aires, 2017, pp. 511-514, doi: 10.1109/MSR.2017.45.

USE CASES

Use case 1: A middle school CS teacher

Use case 1

Teacher

- She wants to introduce Scratch to her students

Use case 1

Teacher

- She wants to introduce Scratch to her students
- She is looking for good example projects to show in class

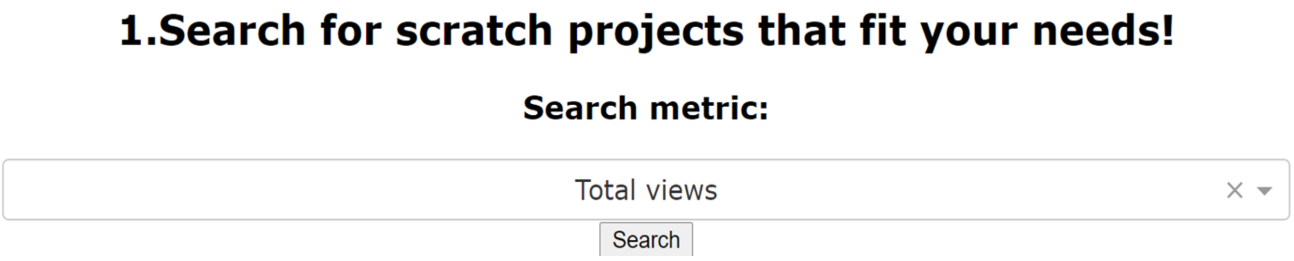
Use case 1

Teacher

- She wants to introduce Scratch to her students
- She is looking for good example projects to show in class
- The dataset is hard to look through because of its large size

Use case 1

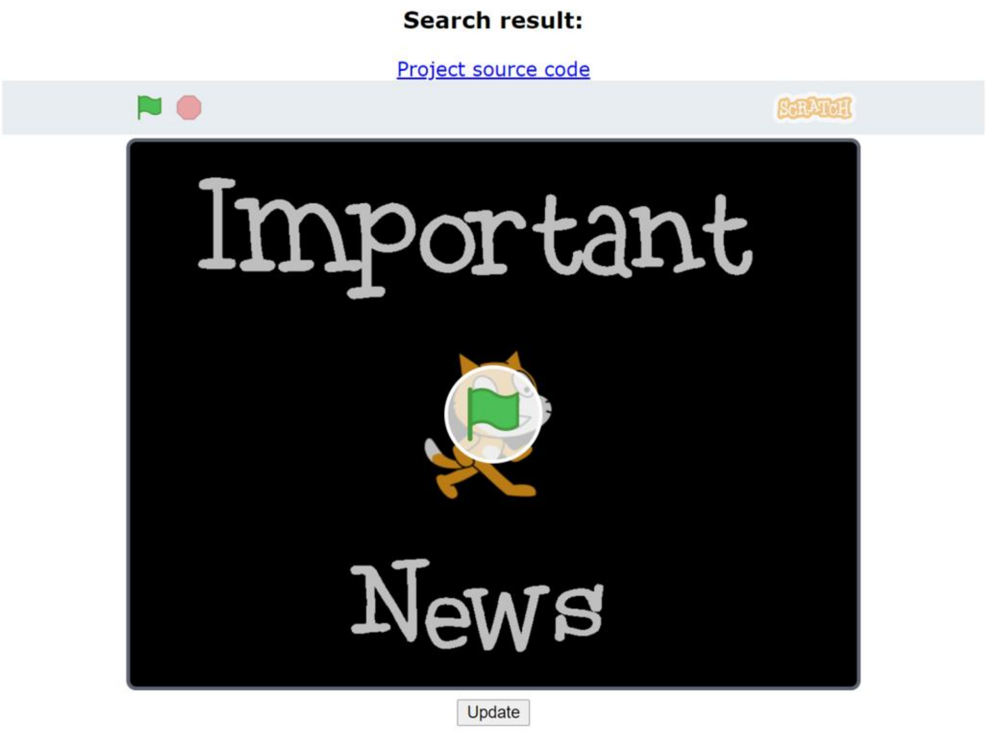
- Our tool will search through the Scratch dataset and returns relevant projects on any of fourteen metrics.
- If the project is still hosted on Scratch, the project is directly displayed and the user can simply click on the project to see it.
- The source code for the project is also linked.
- The results can be updated to the next best project.



1.Search for scratch projects that fit your needs!

Search metric:

Total views



Search result:

Project source code

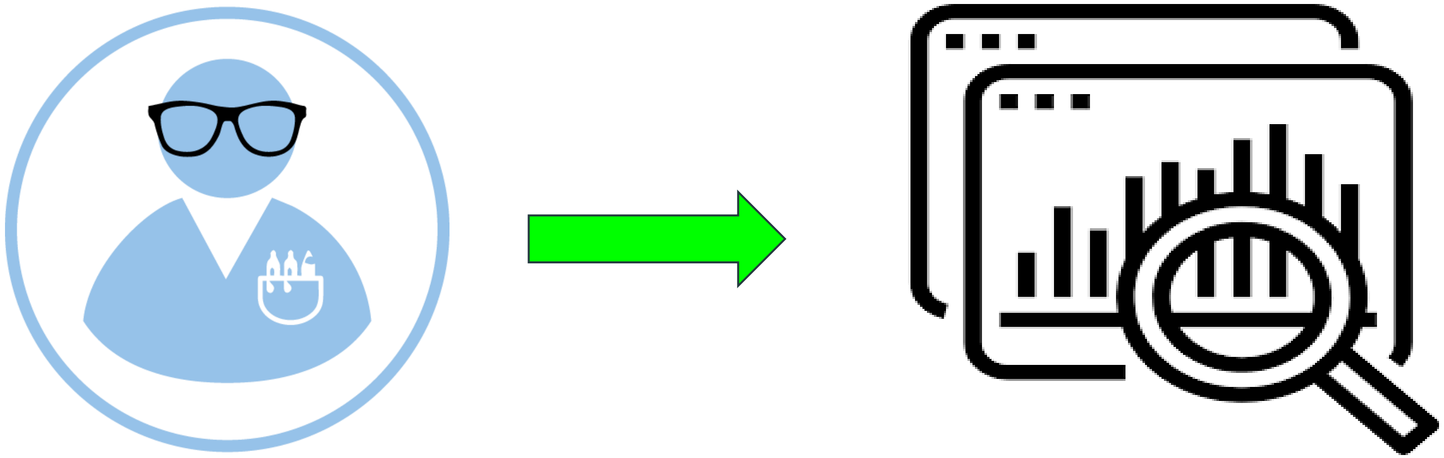
Use case 2: A researcher

Use case 2

Researcher

- A researcher has found the scraped Scratch project repository

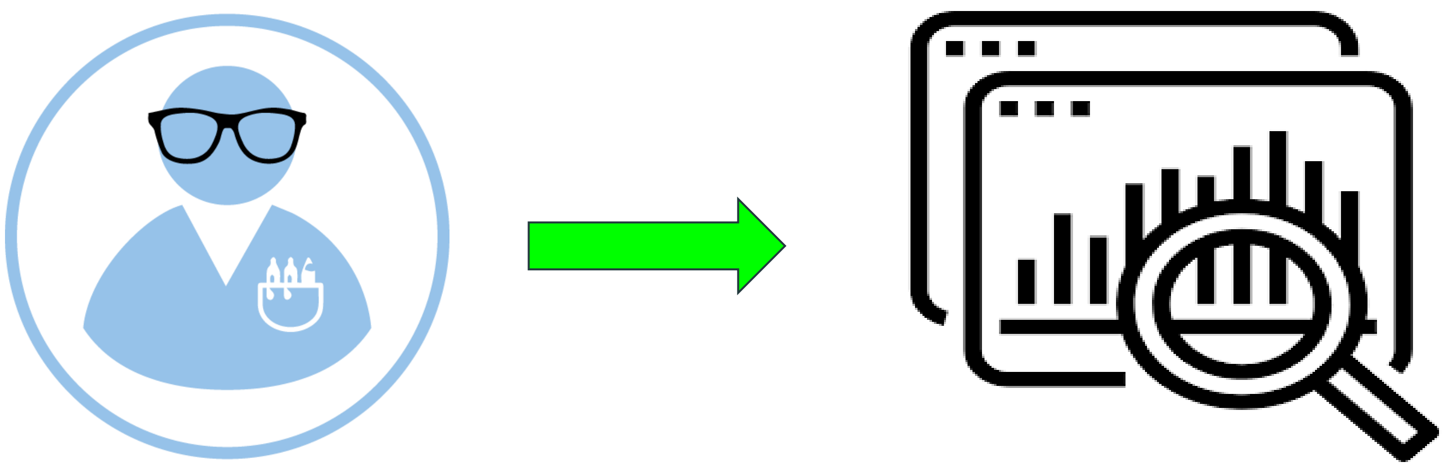
Use case 2



Researcher

- A researcher has found the scraped Scratch project repository
- Wants to use it to understand what kinds of Scratch projects get the most views, favorites, loves, or remixes.

Use case 2

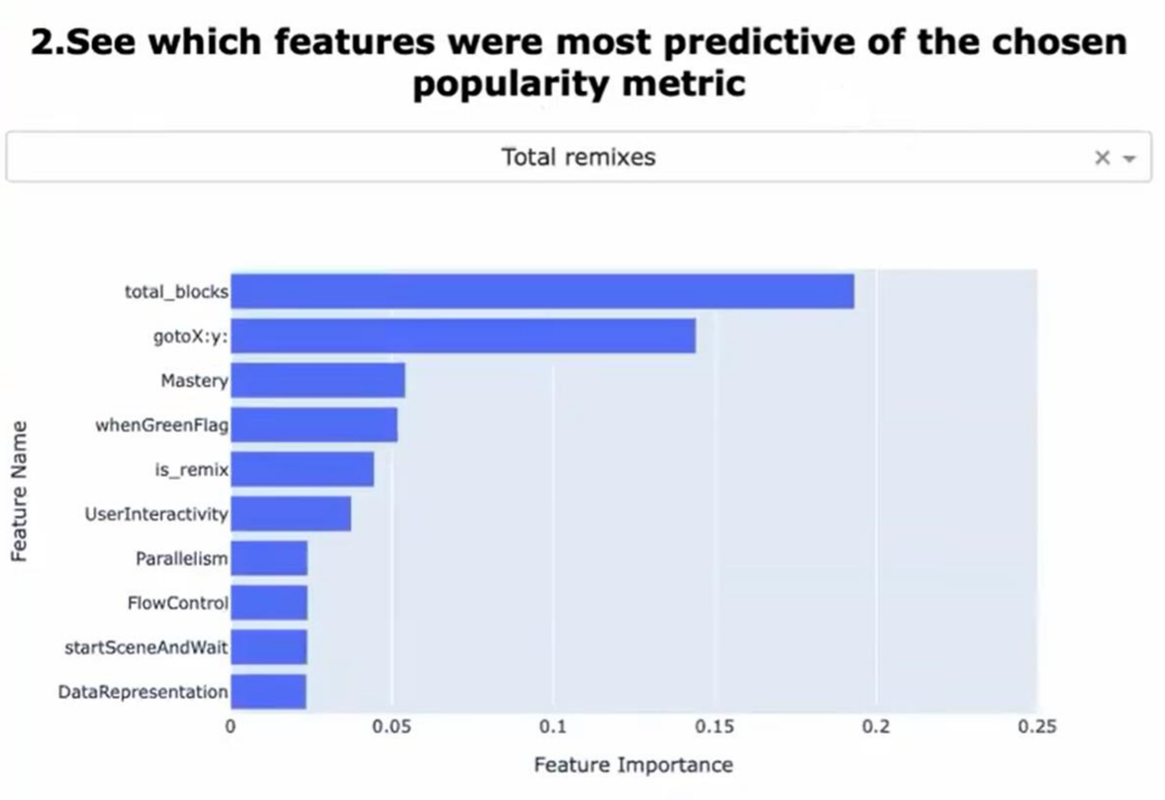


Researcher

- A researcher has found the scraped Scratch project repository.
- Wants to use it to understand what kinds of Scratch projects get the most views, favorites, loves, or remixes.
- The data file is too large for Excel, and he doesn’t have any experience using other tools like Python to examine the data.

Use case 2

Our tool analyzes the Scratch dataset and displays a graphic illustrating the top ten features that a popular project often has, along with the feature's importance. Users can hover over each bar in the graphic for more information.

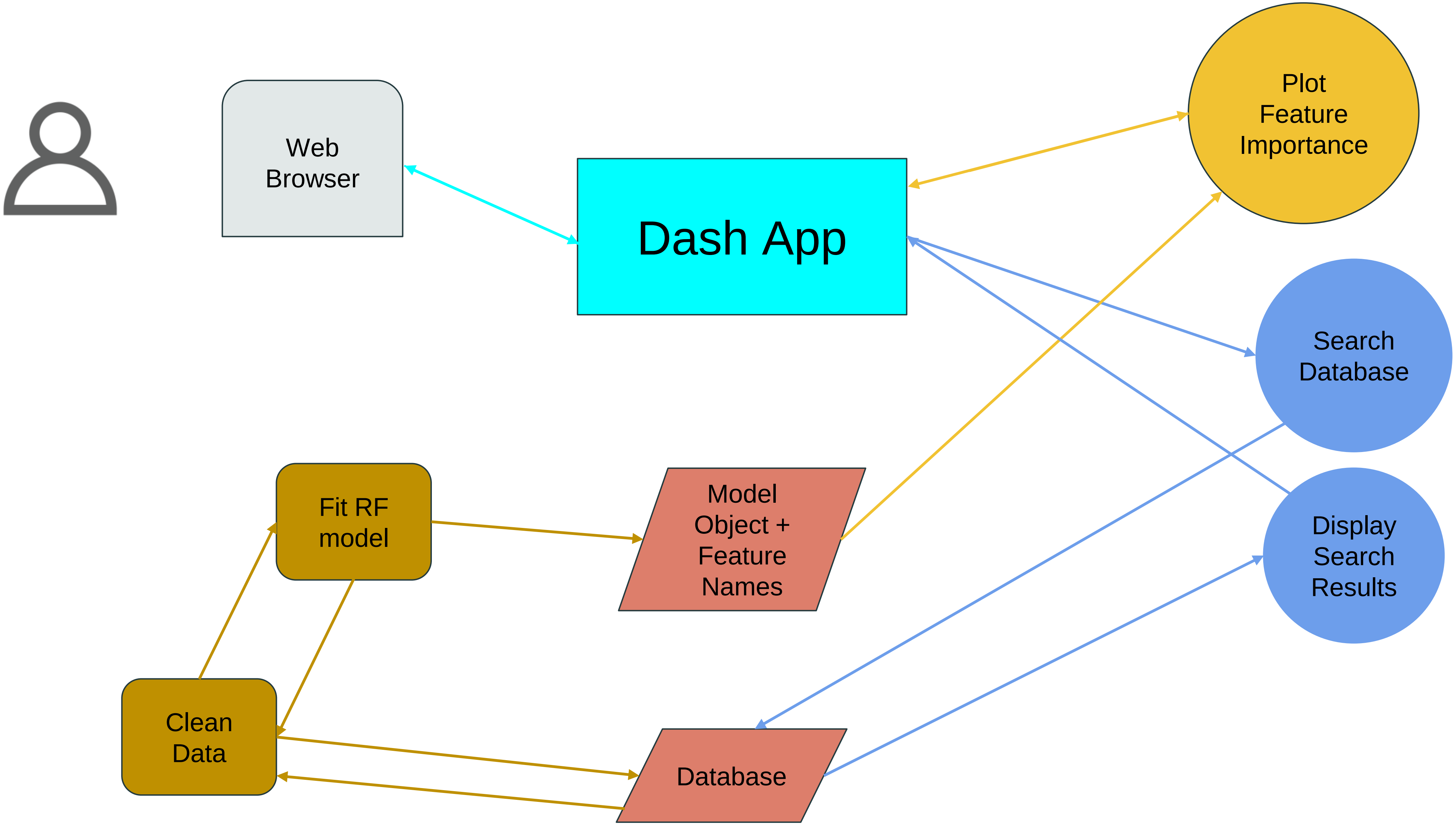


2.See which features were most predictive of the chosen popularity metric

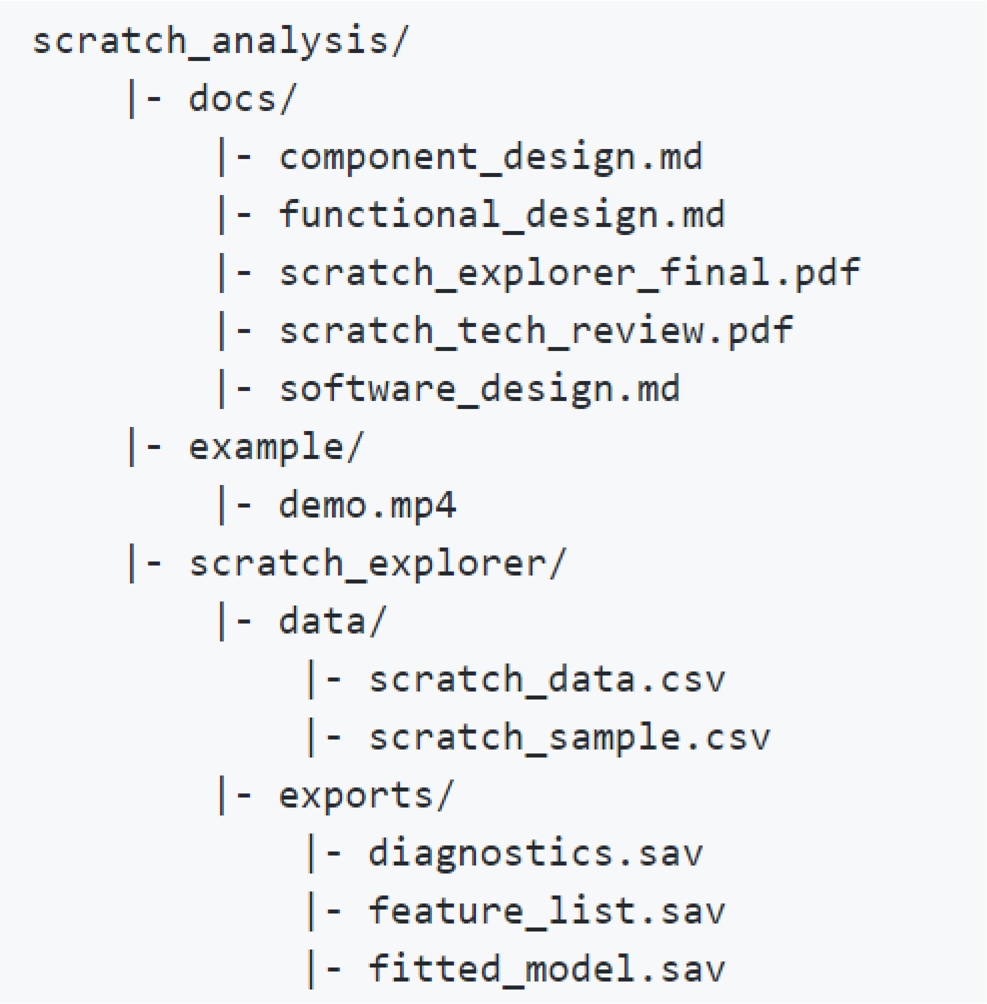
Total remixes

DEMO

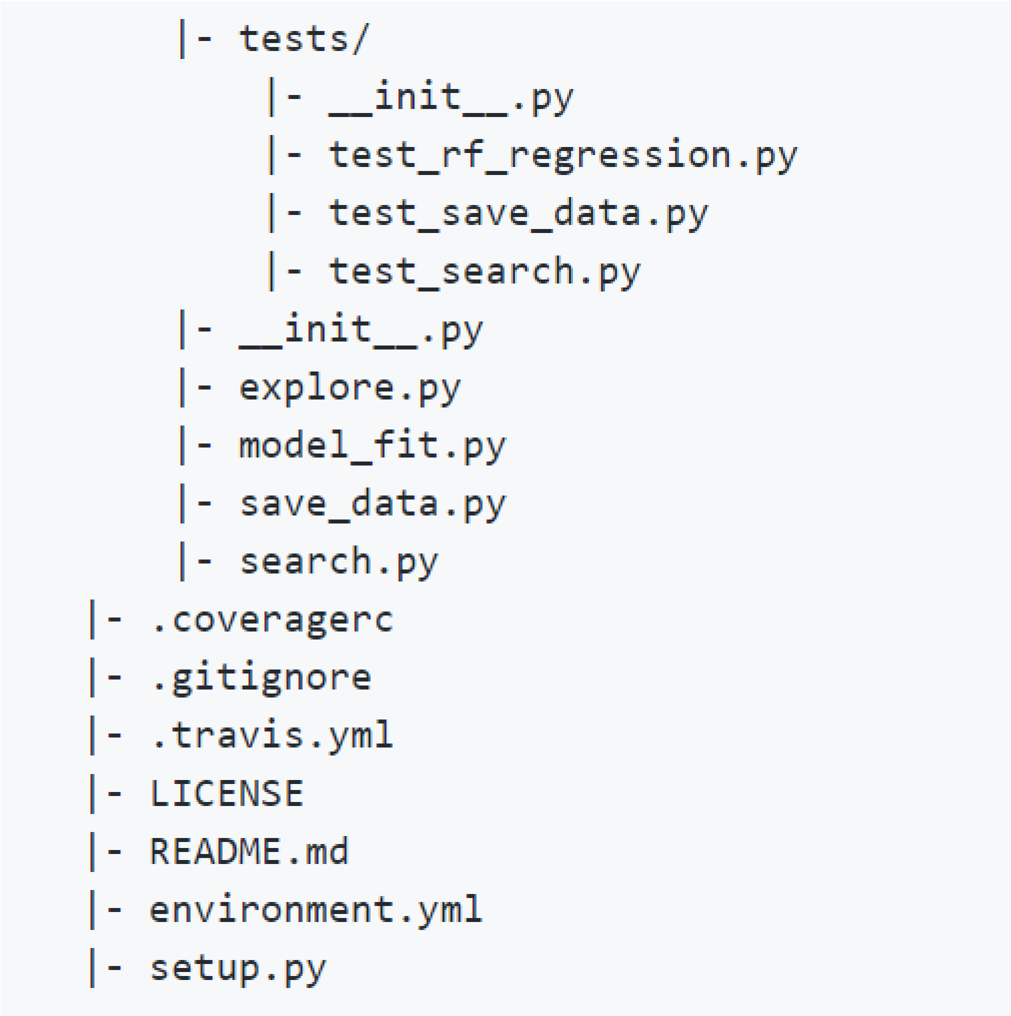
DESIGN


Web Browser
Dash App
Plot Feature Importance
Search Database
Display Search Results
Fit RF model
Model Object + Feature Names
Clean Data
Database

Project Structure: Github repository



```
scratch_analysis/
    |- docs/
        |- component_design.md
        |- functional_design.md
        |- scratch_explorer_final.pdf
        |- scratch_tech_review.pdf
        |- software_design.md
    |- example/
        |- demo.mp4
    |- scratch_explorer/
        |- data/
            |- scratch_data.csv
            |- scratch_sample.csv
        |- exports/
            |- diagnostics.sav
            |- feature_list.sav
            |- fitted_model.sav
```



```
        |- tests/
            |- __init__.py
            |- test_rf_regression.py
            |- test_save_data.py
            |- test_search.py
        |- __init__.py
        |- explore.py
        |- model_fit.py
        |- save_data.py
        |- search.py
    |- .coveragerc
    |- .gitignore
    |- .travis.yml
    |- LICENSE
    |- README.md
    |- environment.yml
    |- setup.py
```

Lessons Learned

- Define specifications well beforehand
- Git:
  - Be careful with merges
  - `git pull` for the latest code, and to avoid conflicting code
- Testing:
  - More is better
  - Ideally, write tests while writing code
  - Mocks are a great way to test computationally expensive code
- Working with large datasets can be challenging

THANK YOU