Optimising Dating

with Web Drivers and Machine Learning

https://github.com/yohanderose/Dat3Bot

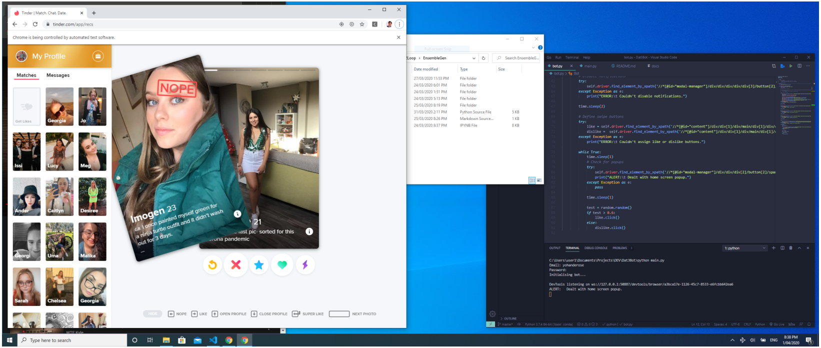
What is the aim?

- True love uwu
- More realistically though – some booty
- Even more realistically, just optimising emotional investment

You'll use selenium to interact with the tinder web app, and train a machine learning model on your attraction preferences to (hopefully accurately) decide on which way to swipe.

Live Demo

Ooh, ambitious.



```
        try:
            self.driver.find_element_by_xpath('//*[@id="modal-manager"]/div/div/div/div/div[3]/button[2]
        except Exception as e:
            print("ERROR:\t Couldn't disable notifications.")

        time.sleep(2)

        # Define swipe buttons
        try:
            like = self.driver.find_element_by_xpath('//*[@id="content"]/div/div[1]/div/main/div[1]/div/
            dislike =  self.driver.find_element_by_xpath('//*[@id="content"]/div/div[1]/div/main/div[1]/
        except Exception as e:
            print("ERROR:\t Couldn't assign like or dislike buttons.")

        while True:
            time.sleep(1)
            # Check for popups
            try:
                self.driver.find_element_by_xpath('//*[@id="modal-manager"]/div/div/div[2]/button[2]/spa
                print("ALERT:\t Dealt with home screen popup.")
            except Exception as e:
                pass

            time.sleep(1)

            test = random.random()
            if test > 0.6:
                like.click()
            else:
                dislike.click()
```

```
C:\Users\user1\Documents\Projects\DEV\Dat3Bot>python main.py
Email: yohanderose
Password:
Initialising bot...

DevTools listening on ws://127.0.0.1:50887/devtools/browser/a3bca17e-1126-45c7-8533-e6fcbb642ea6
ALERT:   Dealt with home screen popup.
```

Dependencies

General requirements

- Python >=3.6
- requests, pillow
- Selenium and chromedriver executable
- Scikit, numpy and pandas
- deepface, face-align, dlib and OpenCV

Personal stuff

- Add your login details to **credentials.py**
- Highly recommend you build your own dataset. I used a selection from Liang et al. and Kaggle to start, but bot will continue to scrape and add to dataset.

Selenium

```
# Clicks big blue button
webdriver.find_element_by_id('bigBlueButton').click()

# Returns referential list of all buttons
all_buttons = webdriver.find_elements_by_tag('button')
```

Selenium cont.

- Install the package, and put relevant driver into path

  https://chromedriver.chromium.org/downloads
- Make app modular or object oriented from the start
- Develop and debug with interactive flag

  ```
  python3 -i main.py
  ```
- Use the inspector in browser dev tools to find web element info

Making Models

Jupyter Prototyping

- Creating a sensible numeric representation of a face (feature vector)
- build_dataset.py

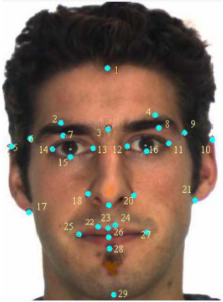


Figure 1: Facial Landmarks (Schmid et al.)

Making Models cont.

Scikit

- Training and comparing a variety of models on a bunch of labelled feature vectors
- train_model.py train.log
- test_model.py

Deploying

- Bot.run()
- Extensions: PCA, Deep Learning
- Considerations:
  - Bot detection countermeasures
  - Social stigma and ethics
  - Security