

# 2202DSA | Unsupervised Sprint

## Deploying a streamlit app on AWS...

October 2022



# “in a Cloud somewhere...”

- Google's Eric Schmidt (2006), describing Google's services



# 1 / **Learning Objectives**

The what and how of deployment - AWS, and streamlit

## 2 / **Requirements**

## 3 / **Setting up streamlit**

## 4 / **Using streamlit**

## 5 / **AWS set up**

## 6 / **Deployment on AWS**

## 7 / **Concluding deployment notes**

## 8 / **Final exams**

# Our learning objectives

We are covering the **why** and **how** of deployment for the unsupervised sprint

## Model Deployment

IN ONE  
SENTENCE

Model deployment is the process of putting machine learning models into production.

IN DETAIL

- Allows unrestricted use of your model.
- Audiences unfamiliar with Machine Learning get to see it in action.
- Ultimately your product will be used by non-technical people.



## Streamlit

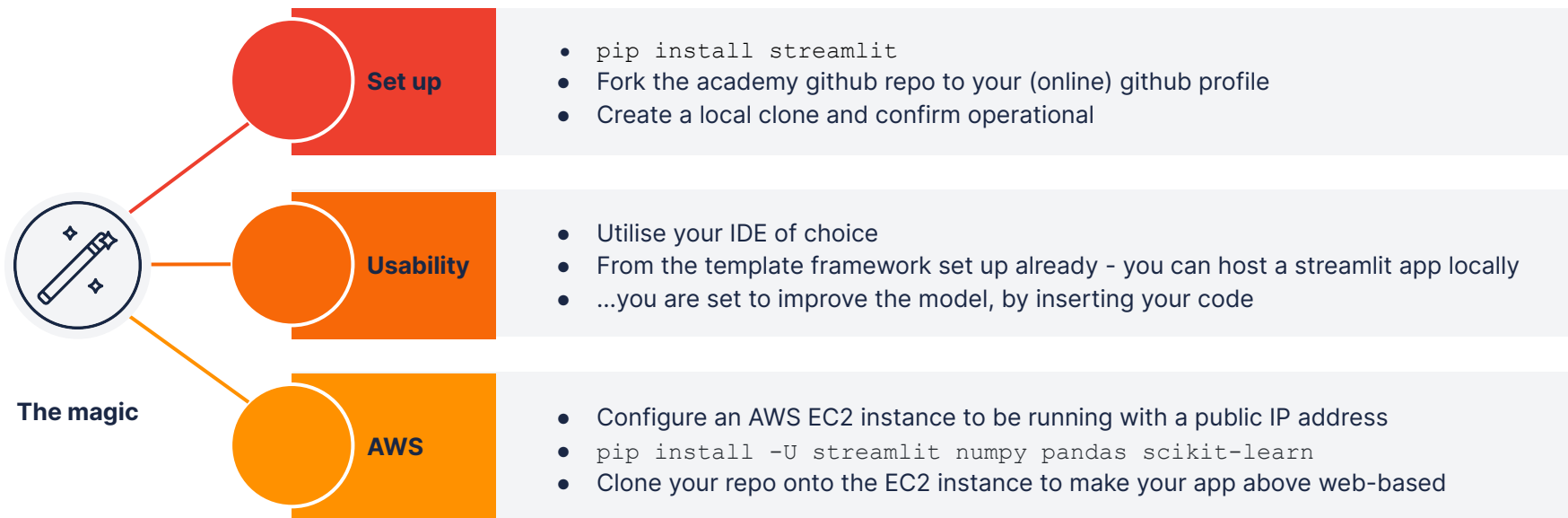
Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science.

- Takes the work out of building a user interface
- Data Science friendly
- See real-time changes while you code (helpful if you've never developed a front-end UI before)



# There are 3 chapters to make the magic happen

The rest of this deck is dedicated to describing these processes in detail



1 / **Learning Objectives**

2 / **Requirements**

3 / **Setting up streamlit**

4 / **Using streamlit**

5 / **AWS set up**

6 / **Deployment on AWS**

7 / **Concluding deployment  
notes**

8 / **Final exams**

# Requirements

Before you get started, here are a few things you will need to do:



## IDE

An interface to code. Jupyter not recommended - be able to deal with a python script (.py file), with, for example, atom or VS code,



## Streamlit

Have a look at the Streamlit documentation



## Git repo

Save a trained model, one has already been created, and you will start from this Git repo (from Explore) and update different pieces. Check out the Building a recommender system Train.



## AWS credentials

Deployment will occur be local first... then up to AWS. Do host a model on AWS, we will use an EC2 instance, and need an AWS account to create, launch, and use the instance.



1 / **Learning Objectives**

2 / **Requirements**

3 / **Setting up streamlit**

4 / **Using streamlit**

5 / **AWS set up**

6 / **Deployment on AWS**

7 / **Concluding deployment  
notes**

8 / **Final exams**

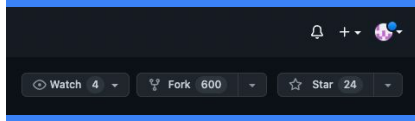


# Setting up streamlit

Here we are going to make sure we have access to all the required code (the streamlit package, and associated files, which are currently on the academy's github page).

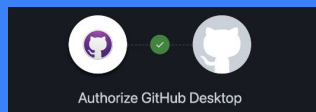
01.

Navigate to github repo  
[link here](#); **fork the repo.**



02.

Clone to your local github  
desktop repository.



03.

In Anaconda prompt / command prompt  
`pip install streamlit`  
`pip install -U streamlit numpy pandas scikit-learn`  
`conda install -c conda-forge scikit-surprise`

```
((base) jamescombrink@james-MacBook-Air ~ % pip install streamlit
Collecting streamlit
  Downloading streamlit-1.14.0-py2.py3-none-any.whl (9.2 MB)
    |#####| 9.2 MB 3.6 MB/s
Collecting semver
  Downloading semver-2.13.0-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: importlib-metadata>=1.4 in ./opt/anaconda3/lib/python3.8/site-packages (from streamlit) (3.10.0)
Collecting altair>=3.2.0
  Downloading altair-4.2.0-py3-none-any.whl (812 kB)
    |#####| 812 kB 2.8 MB/s
Requirement already satisfied: tornado>=5.0 in ./opt/anaconda3/lib/python3.8/site-packages (from streamlit) (6.1)
```

02.

**1 / Learning Objectives**

**2 / Requirements**

**3 / Setting up streamlit**

**4 / Using streamlit**

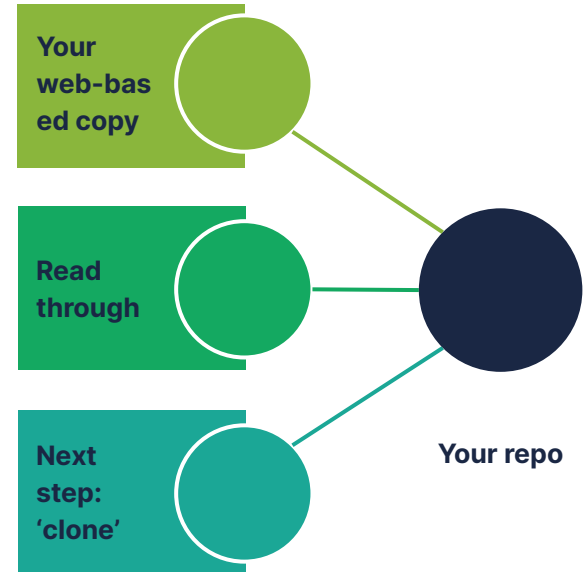
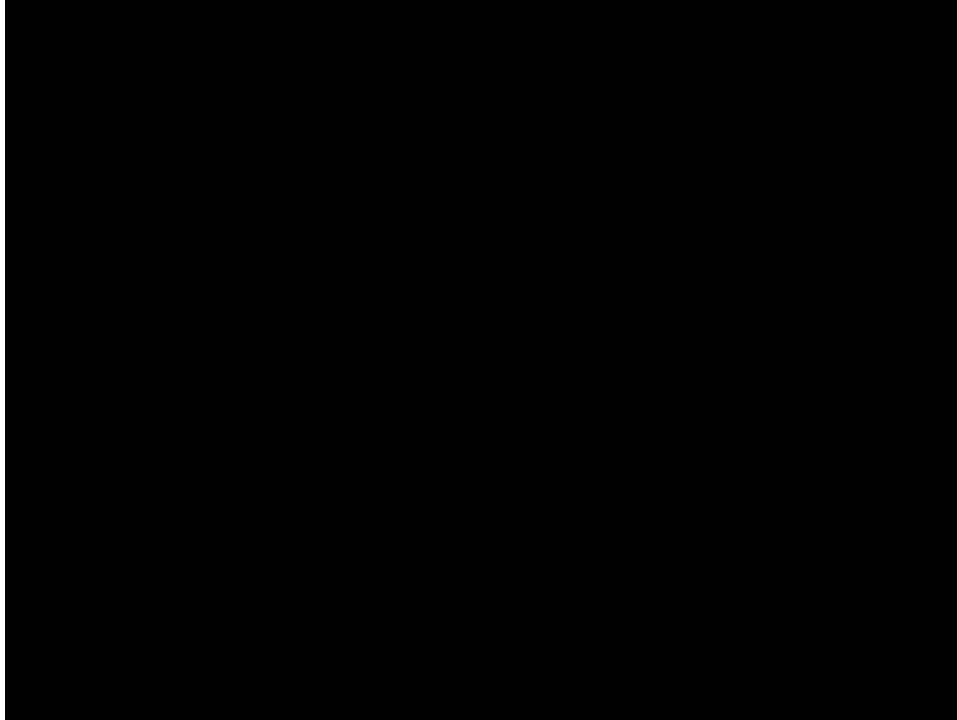
**5 / AWS set up**

**6 / Deployment on AWS**

**7 / Concluding deployment  
notes**

**8 / Final exams**

# [1] To fork the repo...

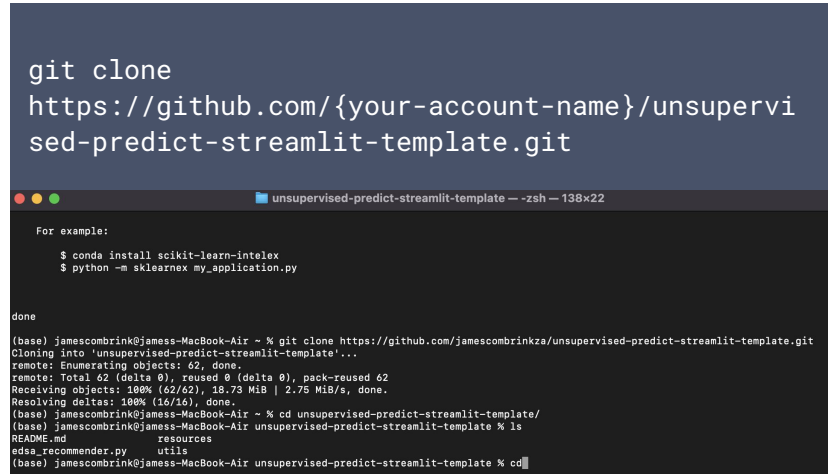
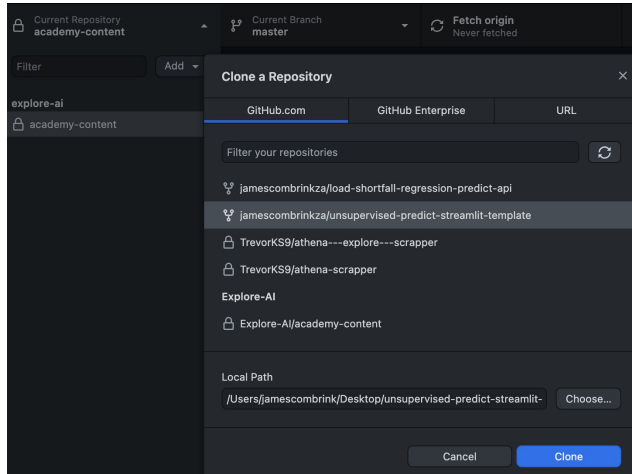
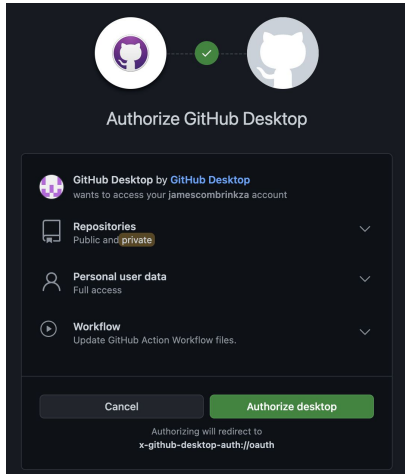


# [2] Clone to local github repo desktop

Use the guide on git & github in the python sprint, if you do not have github desktop

Can be done on github desktop....

...or with anaconda prompt (or git bash)



# [3] Install packages

This will enable you to run everything locally

All in anaconda prompt

In browser - navigate to the local URL

03.

In Anaconda prompt / command prompt

```
pip install streamlit
pip install -U streamlit numpy pandas scikit-learn
conda install -c conda-forge scikit-surprise
```

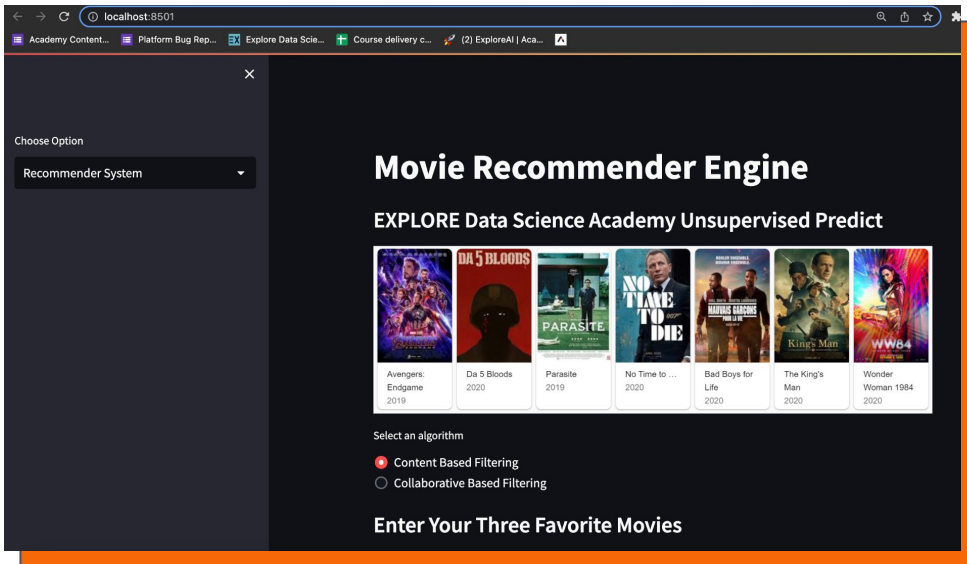
```
((base) jamescombrink@james-MacBook-Air ~ % cd unsupervised-predict-streamlit-template/
((base) jamescombrink@james-MacBook-Air unsupervised-predict-streamlit-template % streamlit run eds_a_recommender.py
```

Welcome to Streamlit!

If you'd like to receive helpful onboarding emails, news, offers, promotions, and the occasional swag, please enter your email address below. Otherwise, leave this field blank.

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8501>  
Network URL: <http://10.0.0.102:8501>



**1 / Learning Objectives**

**2 / Requirements**

**3 / Setting up streamlit**

**4 / Using streamlit**

**5 / AWS set up**

**6 / Deployment on AWS**

**7 / Concluding deployment  
notes**

**8 / Final exams**

# Using streamlit

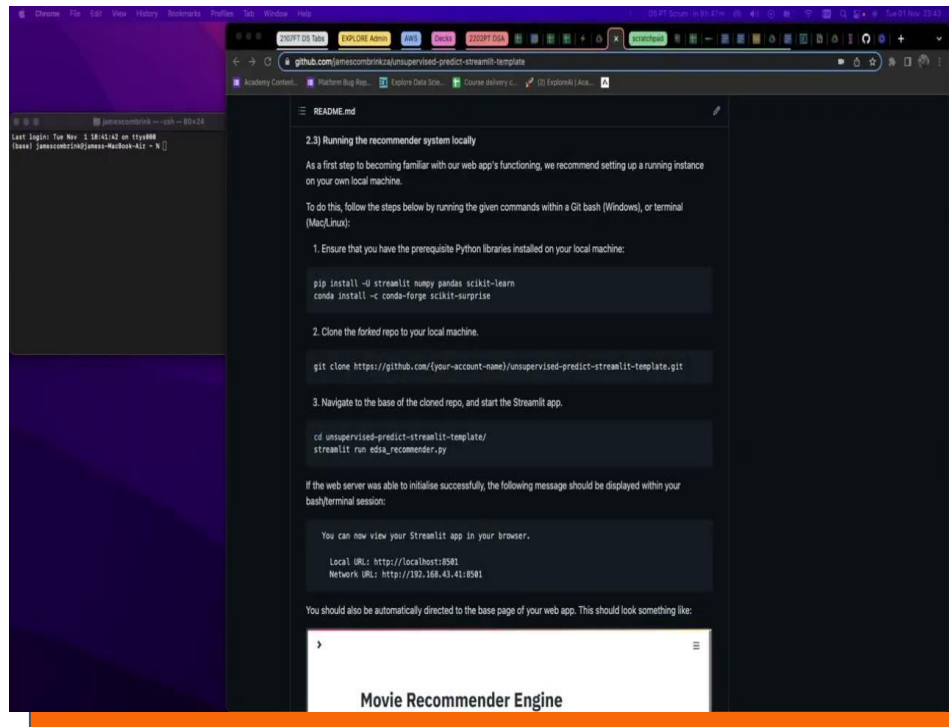
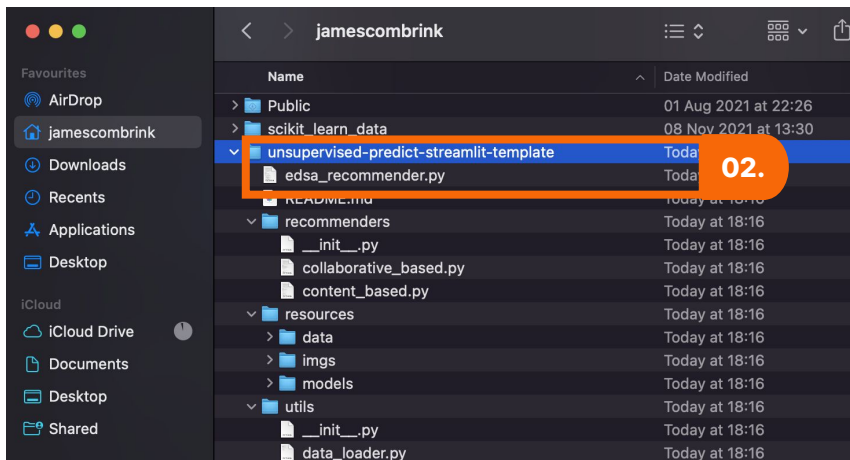


01.

So we can run the python code, through streamlit.

02.

Navigate in the cloned repository, and find: `edsa_recommender.py`



## ...actually using streamlit

Let's look at the difference after a few minor edits - adding some more text, and switching images



- `st.write()` - any argument. Add a graph, a variable, a heading!
- `st.image()` - for a PNG to pop up!
- `st.dataframe()` upgrade to `dataframe`, or a matplotlib graph with `st.pyplot()`
- `st.text()` or `st.markdown()` for text and LaTeX
- `st.checkbox()` and `st.selectbox()` for layers, and further pages
- Streamlit [documentation](#) for a full list of what you can do - but for now, this is enough to get you started.



**1 / Learning Objectives**

**2 / Requirements**

**3 / Setting up streamlit**

**4 / Using streamlit**

**5 / AWS set up**

**6 / Deployment on AWS**

**7 / Concluding deployment  
notes**

**8 / Final exams**

Unlike *data science* where some insight is better than no insight - *data engineering*, is binary.



# On AWS we need to get a few things set up

If we page back to the Regression predict notes - we find an introduction to set-up an EC2

You will need an EC2 instance. In Regression, we introduced how to create this.



▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

**My AMIs** | Quick Start

☒ Owned by me | ☐ Shared with me

[Browse more AMIs](#)  
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

**Explore-DS-course-basic**  
ami-05d0a4e31516e1999  
2022-10-24T04:27:57.000Z | Virtualization: hvm | ENA enabled: true | Root device type: ebs

Description

ExploreAI Academy's main AMI

Architecture	AMI ID
x86_64	ami-05d0a4e31516e1999



**1 / Learning Objectives**

**2 / Requirements**

**3 / Setting up streamlit**

**4 / Using streamlit**

**5 / AWS set up**

**6 / Deployment on AWS**

**7 / Concluding deployment notes**

**8 / Final exams**

# Deployment on AWS

This is the part where we make it globally accessible

You will need an EC2 instance. In Regression, we introduced how to create this.

Step 1

Once your instance is running - you will connect remotely into it.

Public IPv4 address  
34.243.110.255 | [open address](#)

Instance state  
Running

Private IPv4 addresses  
172.31.33.228

Public IPv4 DNS  
ec2-34-243-110-255.eu-west-1.compute.amazonaws.com | [open address](#)

Step 2

Ensure that you have access to a running AWS EC2 instance with an assigned public IP address.

`ssh explore-student@ec2-34-243-110-255.eu-west-1.compute.amazon`

Step 3

Install the prerequisite python libraries:

`pip install -U streamlit numpy pandas scikit-learn`

`conda install -c conda-forge scikit-surprise`

Step 4

Clone your copy of the API repo, and navigate to its root directory:

`git clone`

<https://github.com/{your-account-name}/unsupervised-predict-streamlit-template>  
`cd`

`cd unsupervised-predict-streamlit-template/`

Step 5

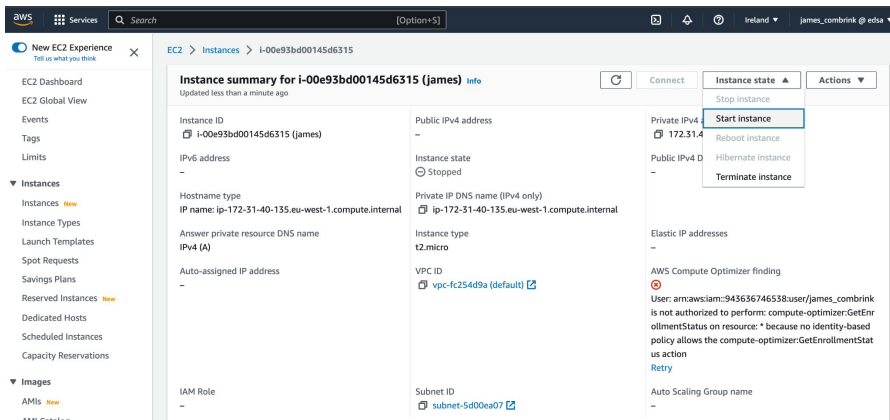
Run

`tmux`

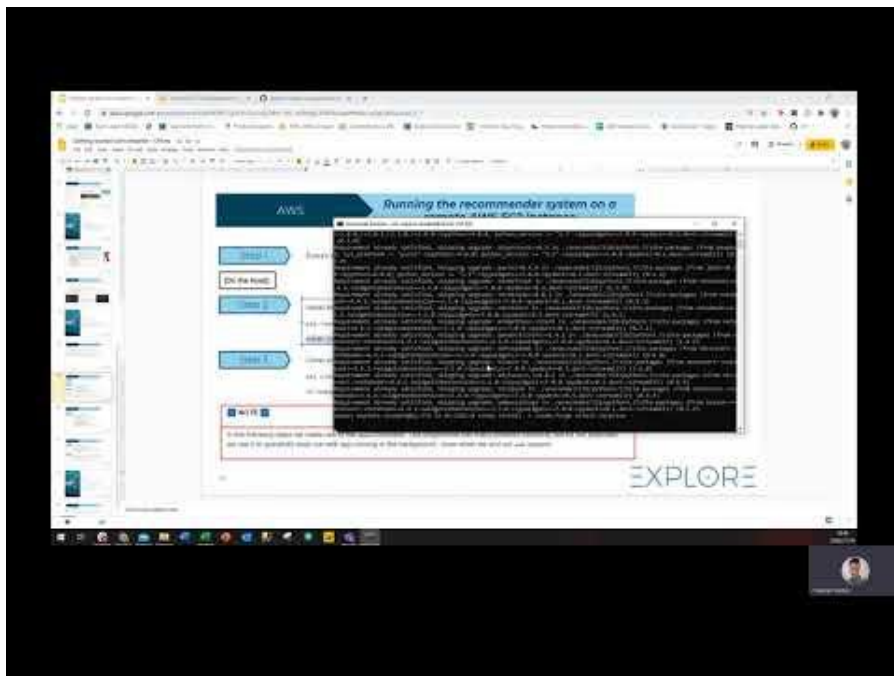
`streamlit run --server.port 5000 edsa_recommender.py`

Step 6

Get, and use, the external URL in browser. Now you can use the web-based streamlit app from anywhere in the world!



# Deployment with Damian



# Deployment on AWS

Let's do it



**1 / Learning Objectives**

**2 / Requirements**

**3 / Setting up streamlit**

**4 / Using streamlit**

**5 / AWS set up**

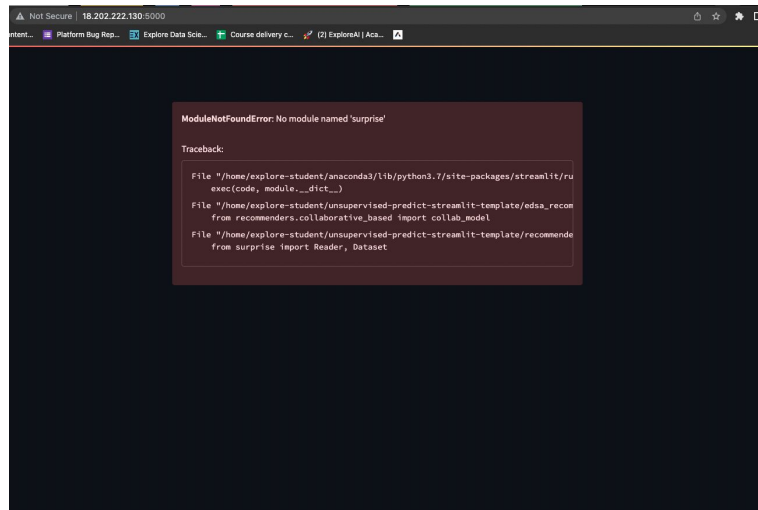
**6 / Deployment on AWS**

**7 / Concluding deployment  
notes**

**8 / Final exams**



# A few errors you may encounter on route

A screenshot of a web browser window displaying a Streamlit application. The browser's address bar shows the URL '18.202.222.130:8000'. The application interface is dark-themed. A red error message box is centered on the screen, containing the text 'ModuleNotFoundError: No module named 'surprise''. Below this, a 'Traceback:' section lists three file paths and the corresponding code lines that caused the error. The first path is '/home/explore-student/anaconda3/lib/python3.7/site-packages/streamlit/runners.py' at line 111. The second path is '/home/explore-student/unsupervised-predict-streamlit-template/eda\_recommender.py' at line 10, showing the import of 'collab\_model' from 'recommenders.collaborative\_based'. The third path is '/home/explore-student/unsupervised-predict-streamlit-template/recommender.py' at line 10, showing the import of 'Reader' and 'Dataset' from 'surprise'.

To get the final output we want...

Git push



**1 / Learning Objectives**

**2 / Requirements**

**3 / Setting up streamlit**

**4 / Using streamlit**

**5 / AWS set up**

**6 / Deployment on AWS**

**7 / Concluding deployment  
notes**

**8 / Final exams**

# Let's broach the discussion of the exams

First and foremost... These are untimed, and open book. Retries are allowed (***post-completion date***)

*There are 6 exams. The details about exactly what is contained will be mentioned at a high level here; BUT will be able to be perused, paused and investigated. We will open the exams early(ish) and close it ealy(ish) to allow for resubmissions to start from as early as December.*

## Machine Learning Exams

Advanced Regression

Advanced Classification

Unsupervised

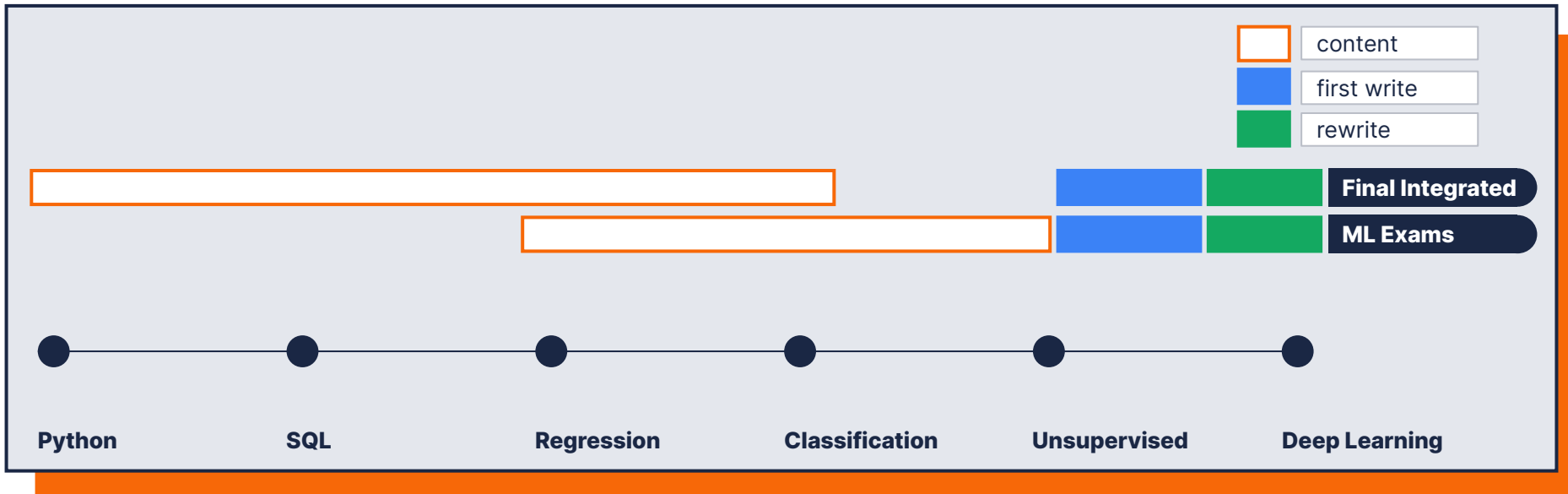
## Final Integrated Exams

Final Integrated Exam - Part 1: Fundamentals

Final Integrated Exam - Part 2: Machine Learning

Supervised (Python + Regression + Classification)

# Final exams timeline



# DON'T PANIC

In the wise words of Douglas Adams...

01. % of class now freaked out
02. Average mark in first-write of worst exam ('supervised')
  - Content:** Material from python, to Unsupervised is examinable open-book and untimed.
  - First write:** Early December (optional), early deadline to allow faster rewrites.
  - Rewrite:** The higher of the initial submission (per exam) and rewrite counts in your portfolio.
03. % class *still* panicked even though there is no reason

