



Data Analytics

Unit 7 - Storytelling with Data, Web Scraping, APIs, AB Testing

NOV - DEC 2020 | BERLIN

What will I learn in this unit?

Unit #8



Python

HTML

Tableau

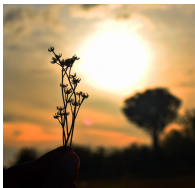
Presenting

The aim of this unit is to polish their data analytics and engineering skills by performing an end-to-end data product: we will create a program that takes an input from the user and automatically collects data from the internet through web scraping and APIs; then it goes through a clustering model and finally returns an output back to the user.

They will implement agile methodologies to develop the product and finally they will “sell it” with an engaging presentation

A large group of approximately 40 people, mostly young adults, are posing for a group photo in front of a modern building with a grid-like facade. They are arranged in several rows, with some people kneeling in the front. Most of them are wearing blue t-shirts with the 'IRON HACK' logo, which consists of the words 'IRON' and 'HACK' inside a hexagon. The entire image has a blue color overlay. A white-bordered box is centered over the group, containing the text 'Fun day - Monday'.

Fun day - Monday



Morning lecture

LFB Best of class dashboards

Why do we tell stories?

Zoom in Zoom out

Data storytelling

Narrative Arc

Tips on Tableau Story setup

--Project intro--

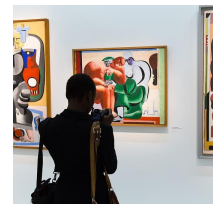
(split into groups)



Afternoon Session

Group Project

**Covid-19 and Human
Movement**



Gallery

4.30-4.45 Break

4.45 Gallery of Data Stories



Why do we tell stories?

Value and meaning

Oldest tradition

Learn its important in childhood

To be remembered - emotion causes memory

Makes us human - relate the story of everything - how we relate to things- impact!

Connection - is a story - make something relevant - you feel involved

Communicate ideas - shared reality/ history

Tells you who you are

Explain our world

Distinctly human trait - religion, nationhood

Identifies us and other

Information - warnings - morality



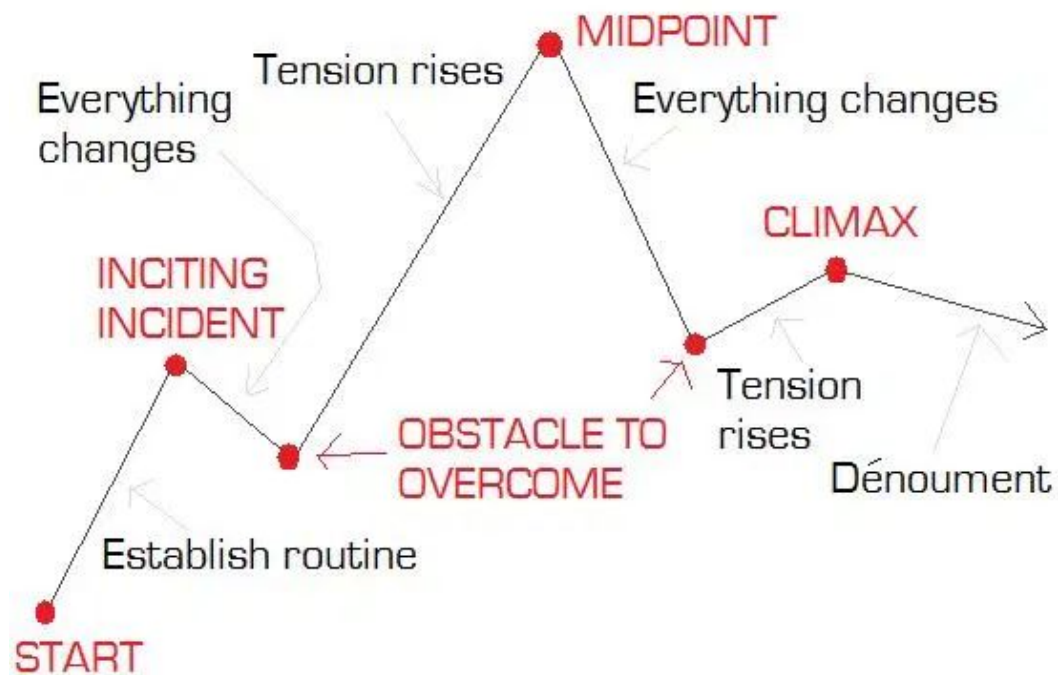
What makes a good story?

- Clarity
- Visuals are vivid
- Tone
- Humour
- Strength in storyline - peak
- Structure
- Common thread
- Relatability
- Shock , Surprise, unexpected
- Elicit an emotional reactions

Zoom in
Zoom out







THE STORY ARC



Movement Range Maps

Movement Range Maps inform researchers and public health experts about how populations are responding to physical

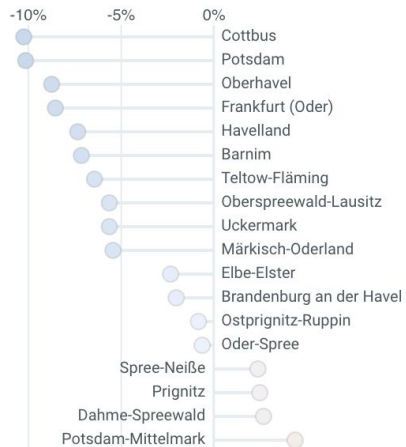


Change in Movement ⓘ

% of People Staying put ⓘ

Change in Movement (Regions of Brandenburg)

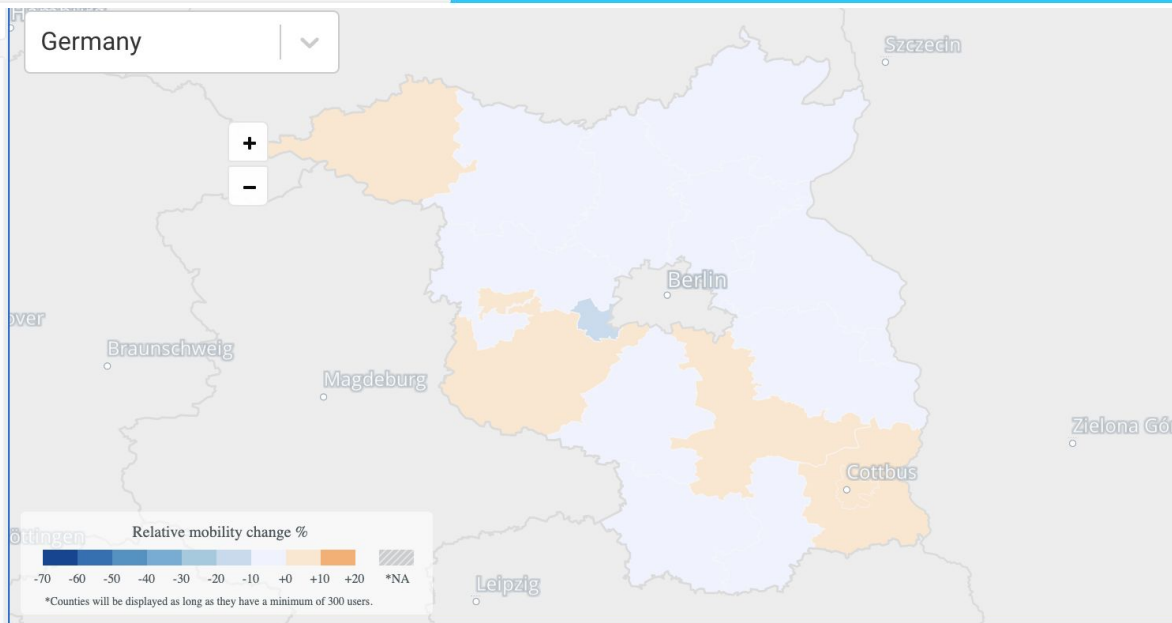
1 ↓



Germany

+

-



Regions of Brandenburg

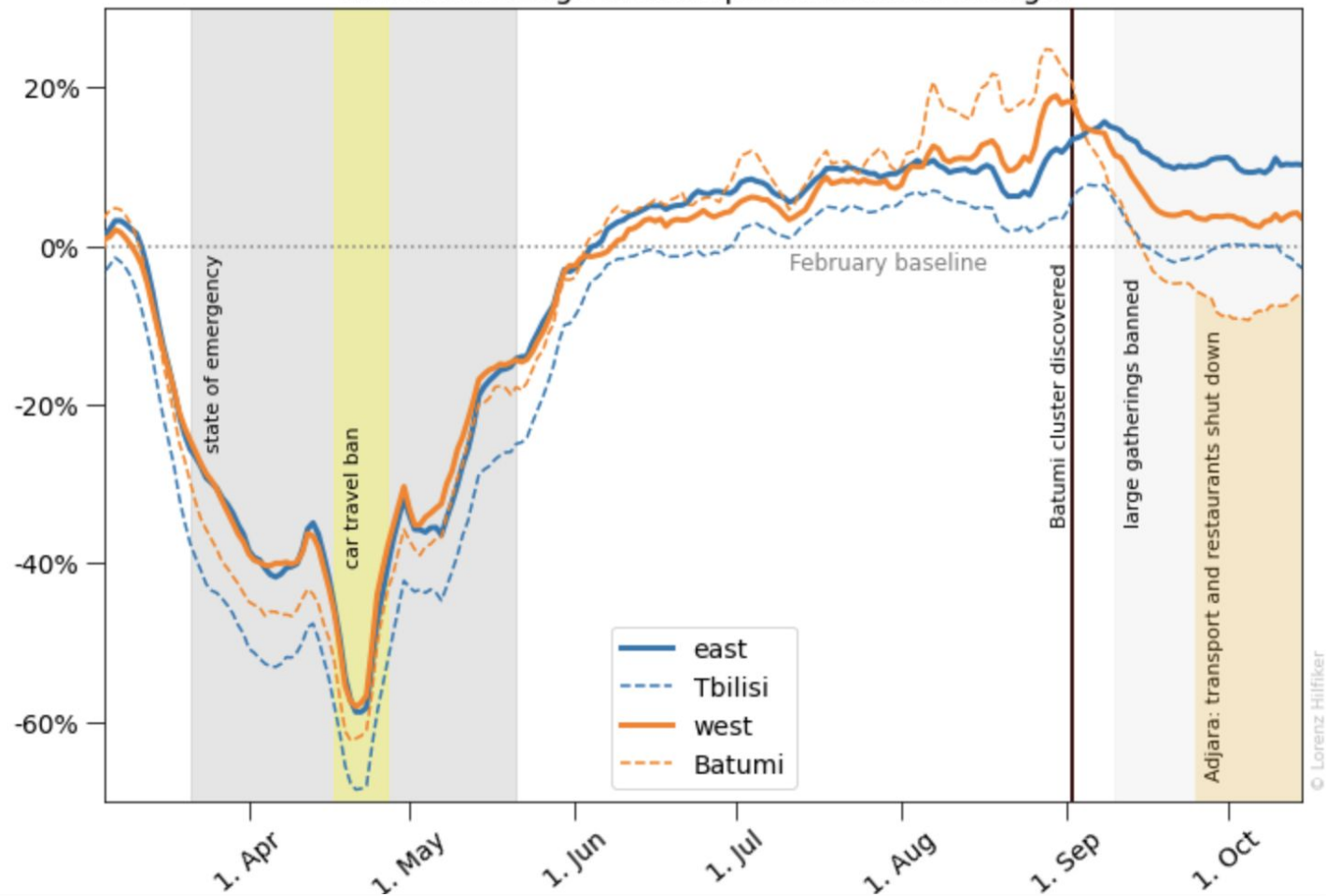
Brandenburg —

1W 1M 3M ALL

[View as separate charts](#)

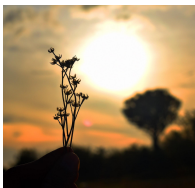


Movement range of smartphone users in Georgia



A large group of approximately 40 people, mostly young adults, are posing for a group photo outdoors. They are arranged in several rows, with some standing and some kneeling or sitting in the front. Most of them are wearing blue t-shirts with the 'IRON HACK' logo, which consists of a hexagon containing the words 'IRON' and 'HACK'. The background features a modern building with large windows and a brick facade, and a body of water is visible on the left. The entire image has a light blue overlay.

Two Day - Tuesday



Morning session

Introduction to WebScraping

Case Study explained

When do we need it? 8.01.1

Html basics

- Tags, structure, inspect
- Next steps for newbies

Beautiful Soup & Parsing 8.01.2

Scraped data & pandas

Andres presentation- final
project - and deep learning



Afternoon Session

Lunch 12:30 - 1:30

Review of Pandas and Getting
started with Web Scraping
with Flo



Lab Session

->TA assisted Labs from 16:15 -

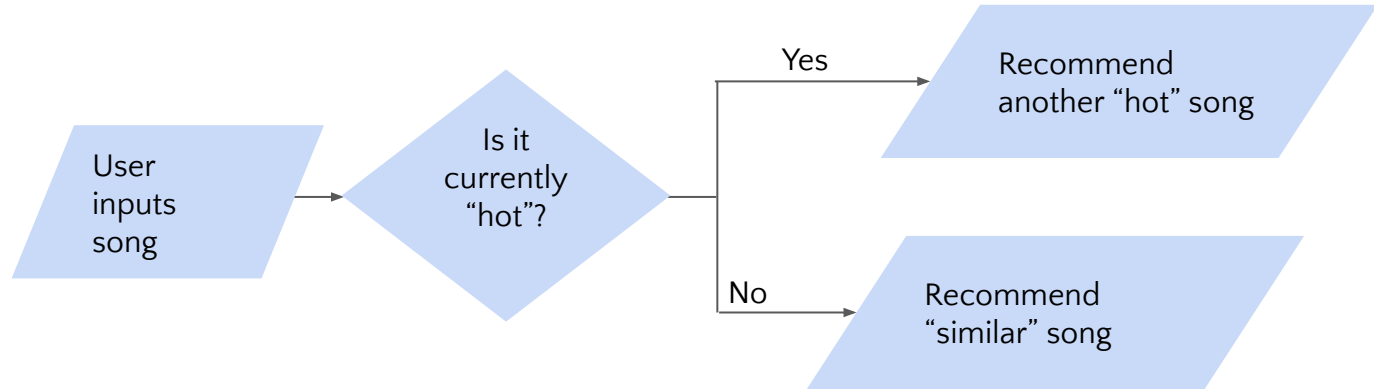
8.01 (inside lesson unit in Day 2
of student portal) **HTML Web
Scraping**

Web Scraping -optional

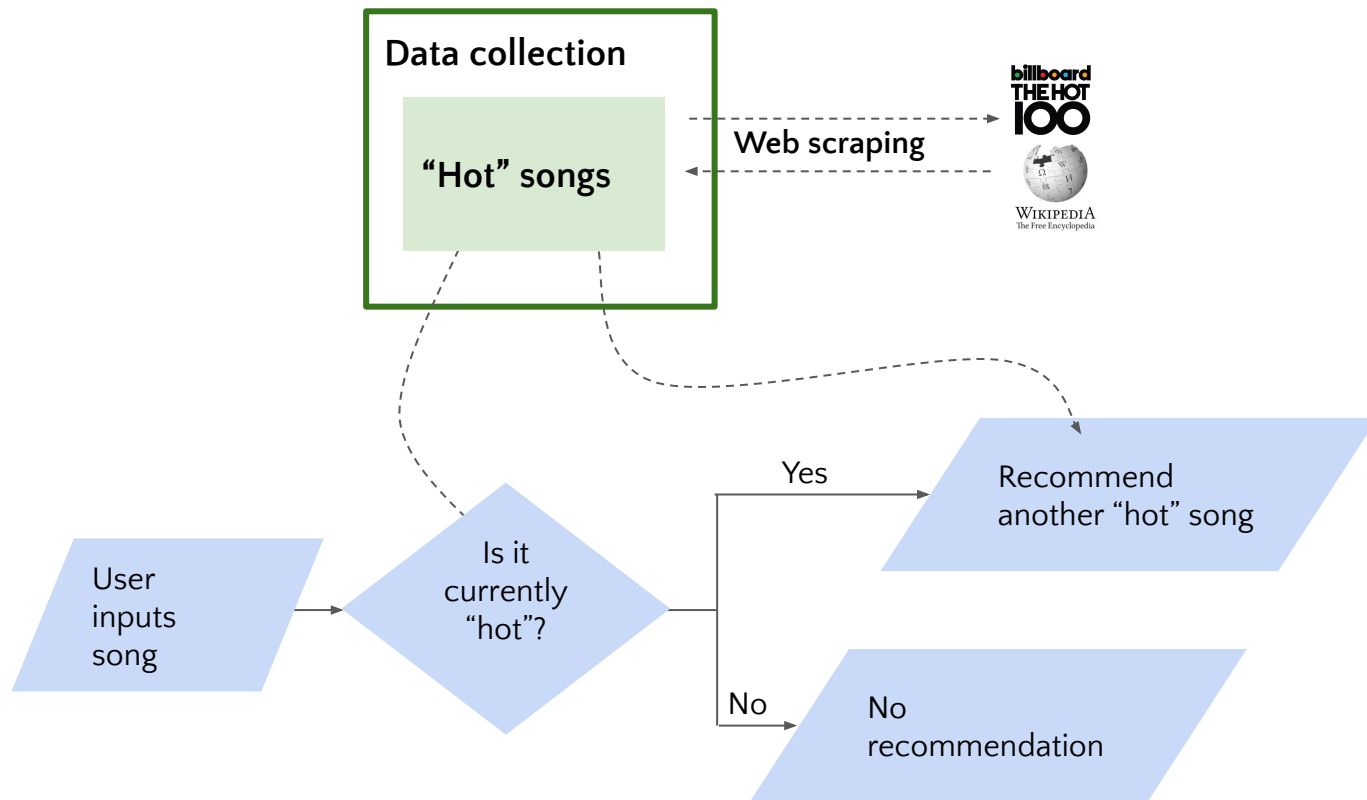
HTML Tutorial - optional

CSS Level 32 - optional

Project flowchart - GNOD Case Study



1st prototype



When to use web scraping

No API - if API is available, normally better to use it

Automation Needed - of course we can copy + paste but ... ugh

Less Restricted - eg no API account required, less rules to follow (eg limit on # requests)

ISSUES

- You depend on the structure of the site being scraped

 - Can be messy

 - Can change overnight

 - Website protections

When to use web scraping

Ideas for sites and use cases

Yellow pages - addresses of companies in a city

Reddit

Asos - images of menswear

Social networks

Amazon - prices of products

Bbc news - see how countries are described

Airbnb - apartments and room prices / sizes / locations - impact

Twitter - Bit coin all time high - look for acronyms

Skyscanner - demand forecasting - prices - best times to book

Linkedin - for filtering jobs

[Web scraping
slides](#)

Basic html (tree) structure

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
    <p>My second paragraph has a <b>bold<b> word!</p>
  </body>
</html>
```

- An **HTML element** is defined by a start tag, some content and an end tag. When web scraping we will mostly be interested in the content, but the tag will be crucial in locating the content.
- **Tags** are just keywords that encapsulate some content. They tell the web browser how to display the content. Some examples of common tags are:
 - `<!DOCTYPE>` and `<html>` define the document type
 - `<head>`, `<title>` and `<body>` define the main parts of the document
 - `<h1>` to `<h6>` define headings
 - `<p>` defines a paragraph
 - `` will make its contents bold

Tags , attributes and value pairs

```
<a href="https://www.ironhack.com/">a data  
bootcamp</a>
```

Attributes you need to know

- The **id** attribute: unless the creator of the site has broken basic conventions, id's are unique. That makes them the best attributes for locating data in a site. If you discover that the piece of information you're trying to collect is an element that has an id, your job will be SO EASY. Bad news though: that doesn't happen often.
- The **class** attribute: it's often used to give style to multiple elements. For example, go to <https://xkcd.com/>. Notice how there are elements like "boxes" or "buttons" that are styled similarly in a site. Instead of defining the style for each one of these elements, the style for all the "boxes" might be defined in a different script (a CSS document), and it just points to all elements with `class = "box"`. This is often a useful way to locate content inside of an HTML script.

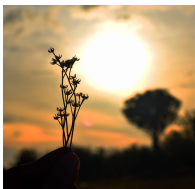


The dormouse's story

[Wikipedia - languages](#)

A large group of approximately 40 people, mostly young adults, are posing for a group photo outdoors. They are arranged in several rows, with some standing and some kneeling or sitting in the front. Most of them are wearing dark blue t-shirts with a white hexagonal logo that says "IRON HACK". They are standing on a paved area in front of a modern building with large windows and a brick facade. There are trees and a body of water visible in the background. The entire image has a purple tint.

‘Hump day’ Wednesday



Morning session

Fresh Brain Lab time -10:30

- Finish working on web scraping labs from Tues

Web scraping extended - multiple pages

Project definition & data assessment

Intro to APIs 8.04.2, 3

Lunch 12:30 - 13:30



Afternoon Session

Mapping using Folium with Brecht

Consulting skills workshop with Sian (small group) - I will run two separate sessions

Defining your final project & MVP as a brief - due PM friday



Lab Session

->TA assisted Labs from 15:00-

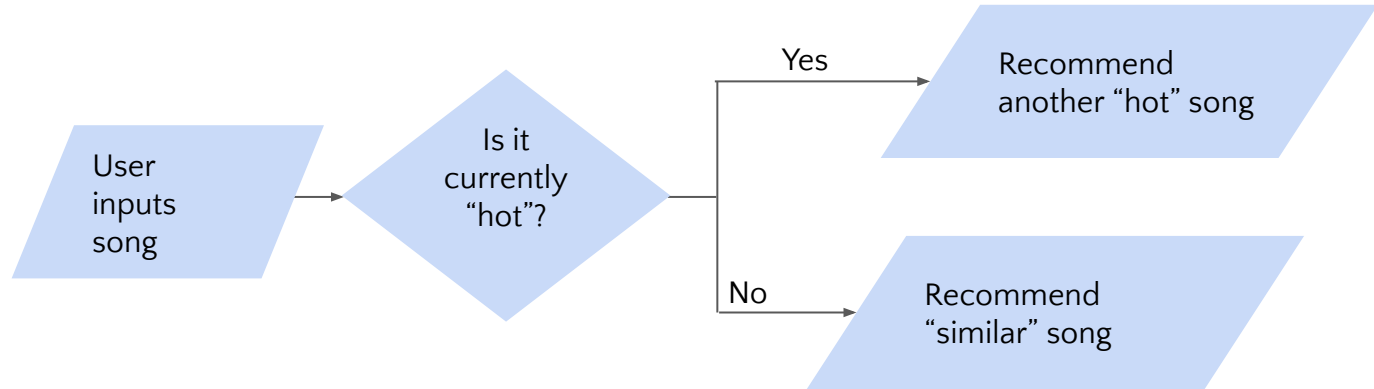
8.02 Web Scraping extended

8.03 Intro to APIs - optional

(Both are inside respective lesson unit in Day 2 of student portal)

[LAB] Advanced Web Scraping - optional see student portal

Project flowchart - GNOD Case Study



Scraping multiple pages



IMDB - notebook provided

Assemble urls to send multiple requests

Using sleep()

Pull 631 movie titles and synopsis

US presidents - your turn

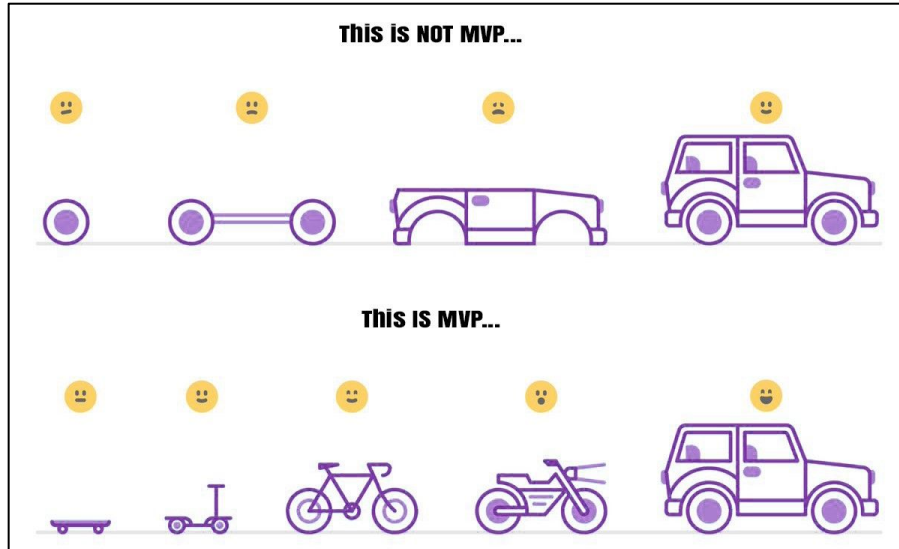
Collect all wikipedia links

Scrape each page

Organise in data frame

Extract some facts

Using error handling



Your first prototype

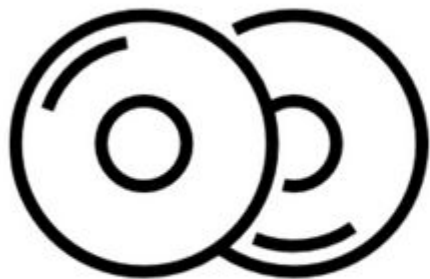
Specifics of your GNOD product

What's your MVP?

Create your python pipeline

- User experience
- Architecture
- Scheduling
- Testing





Prototype



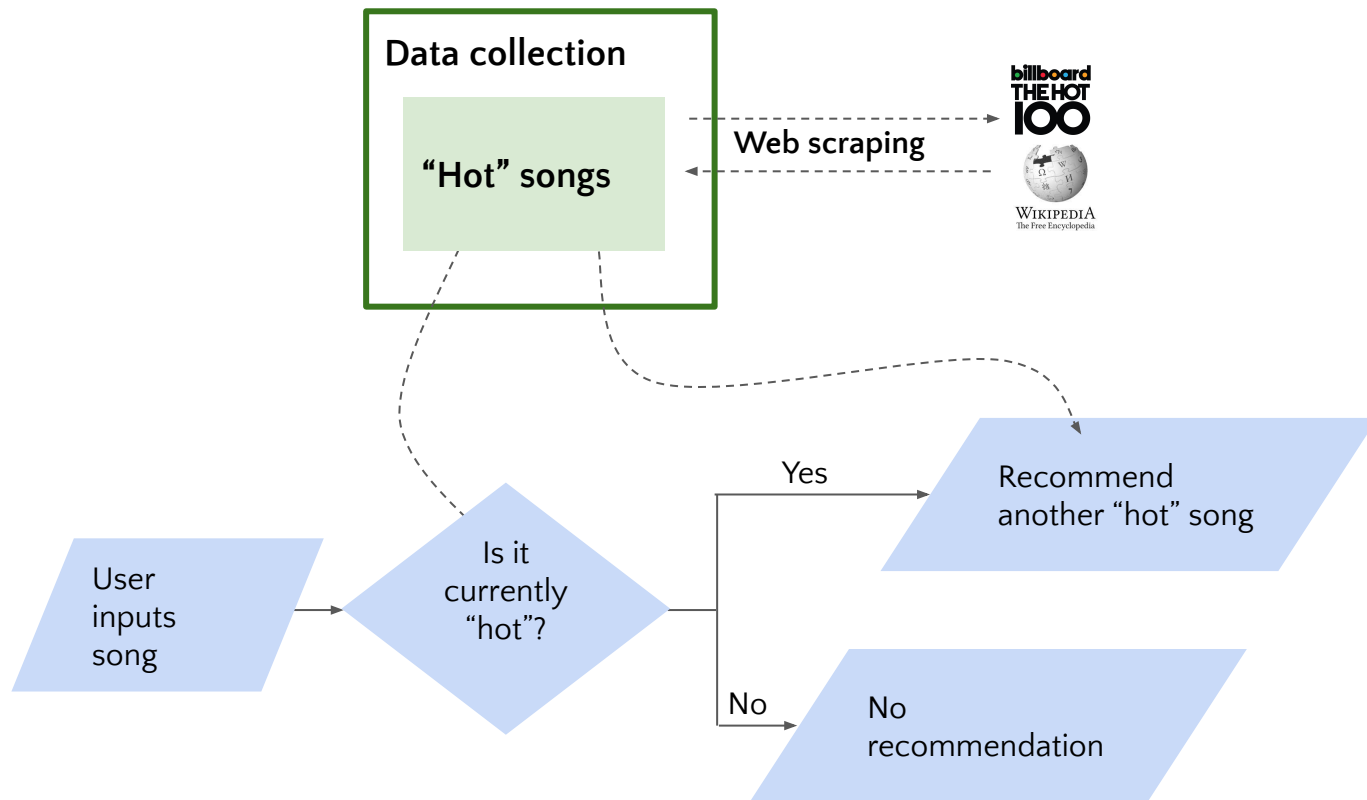
MVP



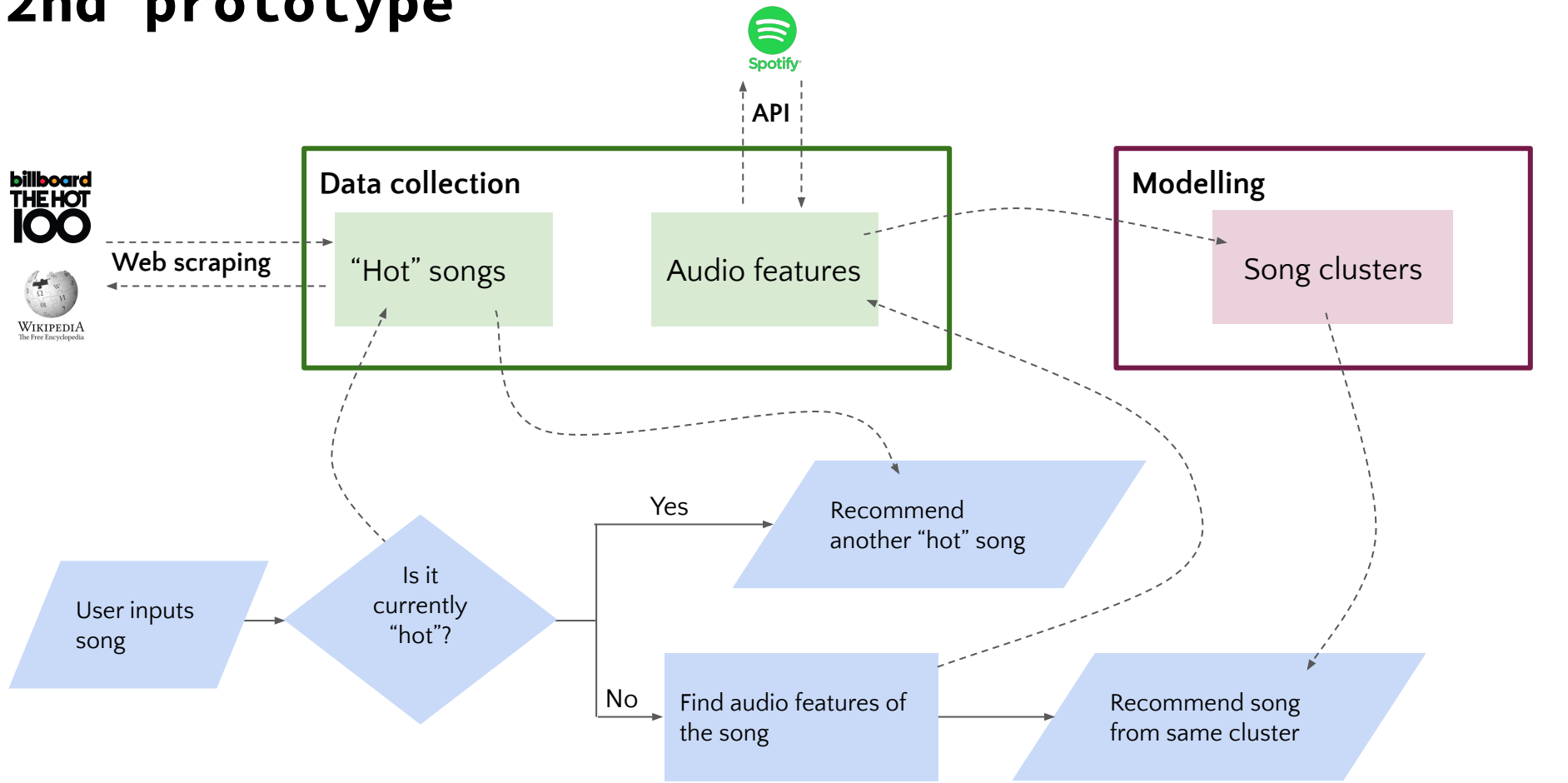
Product

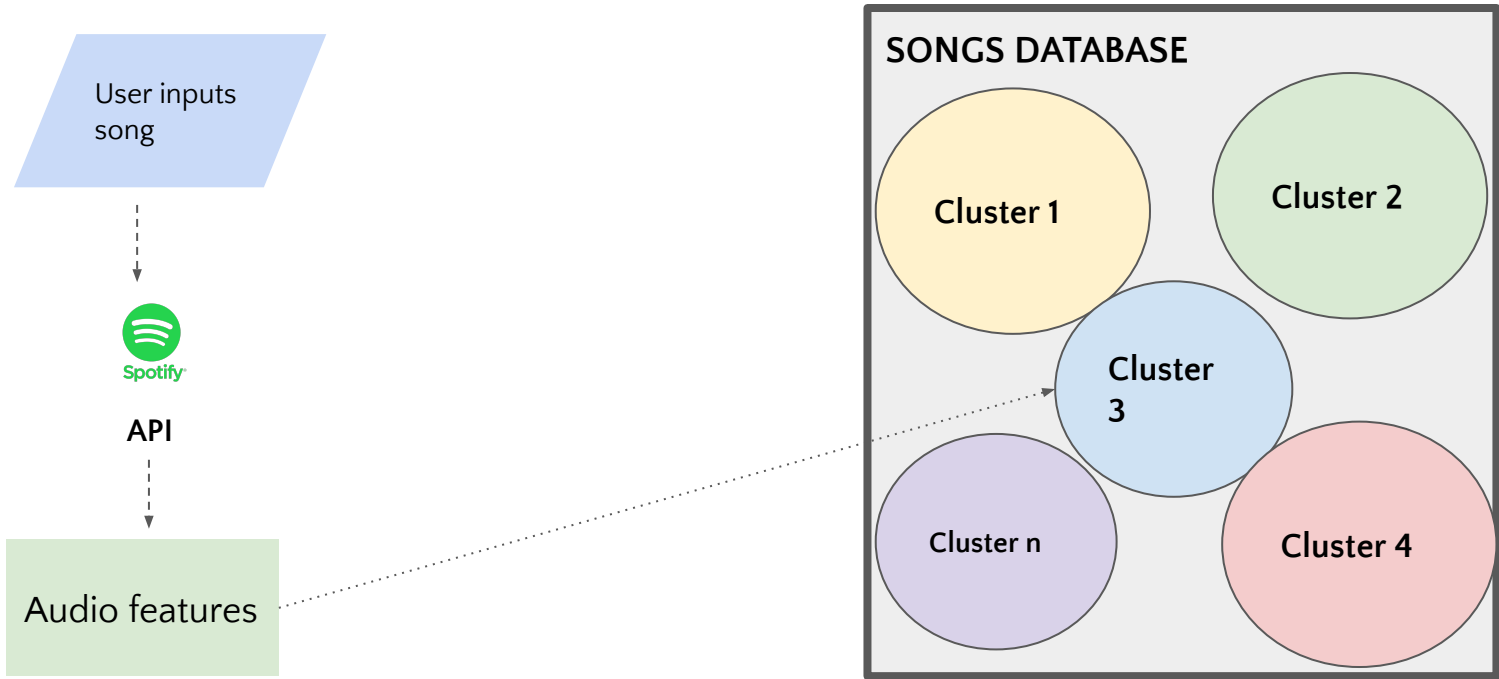
[Medium article - step by step](#)

1st prototype



2nd prototype





User experience:

- What happens if the user inputs a song that doesn't exist?
- What do we do with songs that have the same name, but a different artist?
- How do we deal with typos?

Architecture:

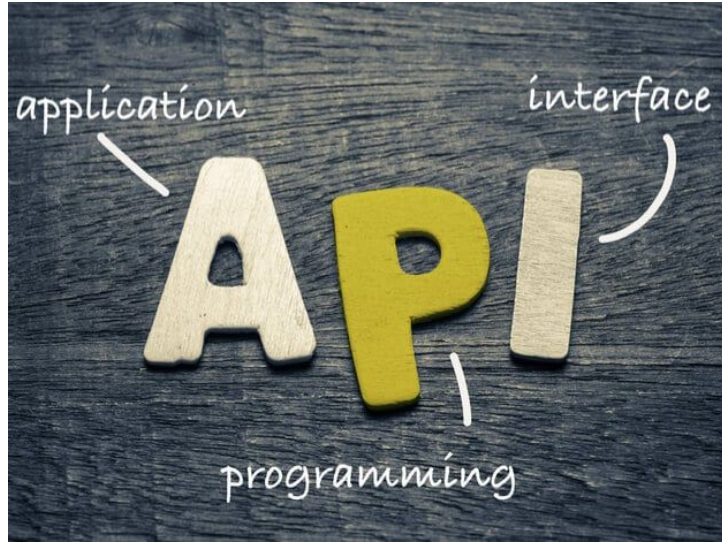
- Do we build the interaction with the user in the same notebook as the web-scraping?
- Where do we store the scraped songs?

Scheduling / Automation:

- Should we scrape billboard / wikipedia every time a user sends a request?

Testing:

- Does it work when you test it with a real user (a classmate or colleague or relative)?



Intro to APIs

What is it - in plain english

REST API

Requests Library - try 3 web pages

Status Codes review

Skyscanner API- sign up and use
Understanding JSON

Apply this to the Skyscanner API

Intro to APIs

-----Why do we need this for the Gnod Case study?

Our company Gnod is interested in grouping clients by their musical interests.

To do that we will need to collect data from musical sites so we can understand better this topic.

We have heard that the Spotify **API** is pretty good for this kind of data collection so we will take a look at it.

If it's not enough or we need more information, we will try other data collection methods.

Intro to APIs

API stands for **Application Programming Interface**.

At some point or another, most large companies have built APIs for their customers, or for internal use.

Every page on the internet is stored somewhere on a remote server. ...you can spin up a server on your laptop capable of serving an entire website to the Web

When you type www.facebook.com into your browser, a request goes out to Facebook's remote server. Once your browser receives the response, it interprets the code and displays the page.- the browser interacts with a remote server's API.

Intro to APIs

An API isn't the same as the remote server — rather **it is the part of the server that receives requests and sends responses.**

You've probably heard of companies packaging APIs as products. For example, Weather Services like OpenWeather sells access to [APIs](#)

In technical terms, the difference is the format of the request and the response- your browser expects html, and gets data eg json

When a company offers an API to their customers, it just means that they've built a set of dedicated URLs that return pure data responses

Some APIs dont need access tokens - eg [github](#)

APIs- better than a csv file?

- **The data changes quickly.** Stock price data or betting houses are a perfect example for it. It doesn't really make sense to regenerate a dataset and download it every second. It is incredibly expensive and wouldn't be efficient nor effective at all, as it would be not just expensive but also really slow.
- **You want just a piece of all your data.** Imagine you want to download your facebook pictures. Without an API you would need to download the entire Facebook dataset, and that doesn't really make a lot of sense since we have APIs to filter that information.
- **Repeated computation involved.** The Spotify API that can tell you the genre of a piece of music. You could theoretically create your own classifier, and use it to compute music categories, but you'll never have as much data as Spotify does, saving a lot of space.

RESTful APIs

REpresentational State Transfer

- Frequently used web service method
- There are some [Architectural constraints](#).
- This generalizes the use of **HTTP**, making requests to specific URLs
- We will use this tool to interact with an API
- Using **GET** and **POST**

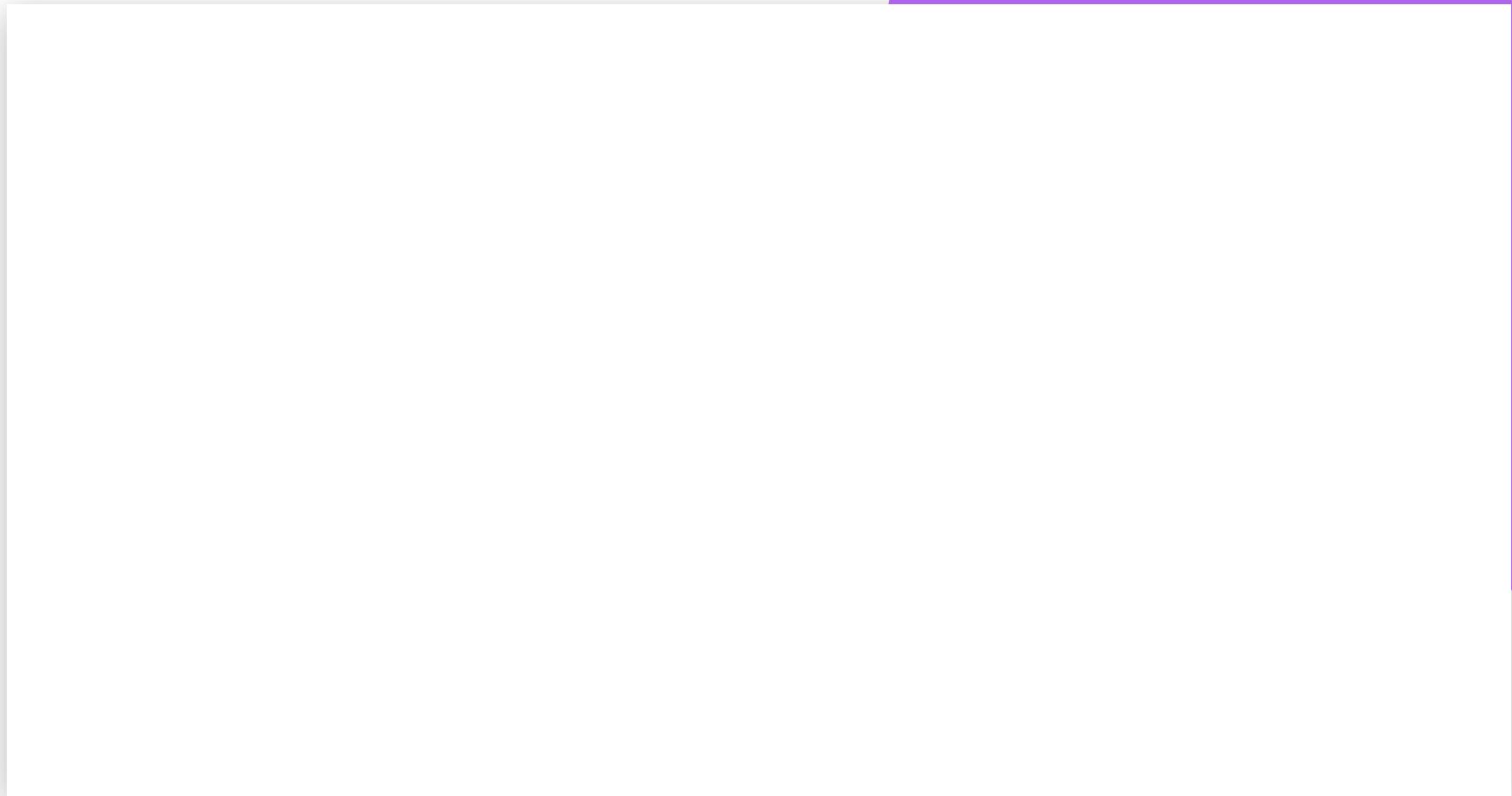
Lets try web pages to retrieve json - which status code comes back?

Import requests and run a url request to the below webpages

- [google](#)
- [NBA](#)
- [rottentomatoes](#)

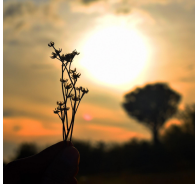
Request status codes

- 200: Everything went okay and the result has been returned (if any).
- 301: The server is redirecting you to a different endpoint. This can happen when a company switches domain names, or an endpoint name is changed.
- 400: The server thinks you made a bad request. This happens when you don't send along the right data, among other things.
- 401: You are not properly authenticated.
- 403: The resource you're trying to access is forbidden: you don't have the right permissions to get it.
- 404: The resource you tried to access doesn't exist.
- 503: The server can't handle the request. </details>



A large group of approximately 40 people, mostly young adults, are posing for a group photo outdoors. They are arranged in several rows, with some standing and some crouching in the front. Most of them are wearing dark blue t-shirts with a white hexagonal logo that says "IRON HACK". They are standing on a paved area in front of a modern building with large windows and a grid-like facade. There are trees and a body of water visible in the background. The image has a dark blue overlay, and the text "Nearly there Thursday" is centered in a white box.

Nearly there Thursday



Morning session

Lunch 12:30 -



Afternoon Session



Lab Session

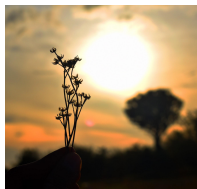
->TA assisted Labs from 16:00

Coingecko (optional)



A large group of approximately 40 people, mostly young adults, are posing for a group photo outdoors. They are arranged in several rows, with some standing and some kneeling or sitting in the front. Most of them are wearing dark blue t-shirts with a white hexagonal logo that says "IRON HACK". The background features a modern building with large windows and a brick facade, and some greenery. The entire image has a warm, yellowish-orange tint.

TFI Friday



Morning session

Lunch 12:40 - 1:45



Afternoon Session