

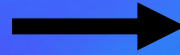


Case Introduction



What is the case about?

The goal of this case is to build a Movie Recommender!



What data will you use?

You will use IMDb data that contains information about movies, like:

- Title
- Description
- Language
- Popularity
- Runtime
- Genre
- Release date
- Revenue
- ...

Title	Description	Release Date	Revenue
Avatar	In the 22nd century...	12/10/2009	2.79 billion
Star Wars: The Force Awakens	Thirty years after defeating the Galactic Empire...	12/15/2015	2.06 billion
Forrest Gump	A man with a low IQ has accomplished...	7/6/1994	677 million

There is a lot of data present, you don't need to use it all!

What will you do in the case?

- **Make recommendations using the movie descriptions!**
- **You will split into teams of 3 or 4 people.**
- **You will have access to a Colab Python Notebook.**
- **Some code already written for you, but some you have to yourself:**
 - Descriptive analysis
 - Data preprocessing
 - Improving the recommendations
- **Afterwards, you will need to shortly pitch your findings to a jury**

What do you need to pitch?

In your pitch (3 minutes maximum) you should:

- Show what descriptive analytics and data preprocessing steps you took
- Share your insights on the recommendations:
 - Share you **chosen movie** and its **recommended movies**
 - Tell which **improvements** you made to the recommender
 - Discuss whether the recommendations **make sense**, and if they are logical given the improvements you made
- Convince us of your ideas for the future, think for example about:
 - **Suggestions** that could further improve the recommender
 - Suggest **other recommender methodologies** that could be used



Background knowledge: NLP basics



Three basic steps in NLP



**1. Text
Preprocessing**



**2. Calculating
Embeddings**



**3. Calculating
Similarity**



Preprocessing text to only get relevant information

I'm looking for a
movie, it's about
the Superheroes.,
and the avengers
with Thanos but not
Inf. War movie,
popular



looking movie
superheroes
avengers thanos not
infinity war movie
popular

Embedding to convert text into a numerical format

looking movie
superheroes
avengers thanos not
infinity war movie
popular



1.009
5.826
7.518
3.027
⋮
9.241
4.745



Calculating cosine similarity to get similarity score

looking movie
superheroes
avengers
thanos not
infinity war
movie popular


$$\begin{bmatrix} 1.009 \\ 5.826 \\ 7.518 \\ 3.027 \\ \vdots \\ 9.241 \\ 4.745 \end{bmatrix}$$

After the devastating
events of Avengers:
Infinity War (2018),
the universe is in
ruins. With the help
of remaining allies,
the Avengers...


$$\begin{bmatrix} 1.119 \\ 4.826 \\ 7.523 \\ 3.027 \\ \vdots \\ 2.241 \\ 5.045 \end{bmatrix}$$


**Cosine
Similarity:
0.78**



Embeddings can also be illustrated graphically

A paraplegic Marine
dispatched to the moon
Pandora on a unique mission
becomes torn between...

After the devastating events
of Avengers: Infinity War
(2018), the universe is in
ruins. With the help of
remaining allies, the
Avengers...

looking movie
superheroes
avengers thanos
not infinity war
movie popular



Background knowledge: NLP

We will generate our NLP recommendations based on the **movie description** column. We will do this by determining which descriptions are similar to the base movie description

You need to perform 3 steps that are commonly used in NLP to arrive at recommendations:

1. Text Preprocessing

- You need to 'clean' the movie descriptions such that we only keep relevant text.
- Think of removing characters/words that do not add to determining similarity between descriptions


2. Calculating Embeddings of the Descriptions

- Embedding a text means to convert it into a **numerical representation**
- This numerical representations is a vector which represents the description
- The model we use has 384 dimensions, so our movie description will be transformed into a vector with 384 numbers!

3. Calculating Similarity between Embeddings

- After we have calculated the embeddings of all the movie descriptions, we need to see which embeddings are similar
- A commonly used metric for this is the **cosine similarity**.
- A cosine similarity of 1 implies the vectors are identical.
- A cosine similarity of 0 implies the vectors are very dissimilar.

Let's start with the case!

- You will have **60 minutes** for the case
- 3 things for you to **improve**:
 - Descriptive Analytics
 - Text Preprocessing
 - Improving Recommendations
- Pitch your findings to us in **3 minutes**
- **Scan** the QR code to open case repository
- To start the case, **open** “NLP Case 2024.ipynb”
and click on  [Open in Colab](#)
- **Slides** are also available behind link!



me-qr.com/MdaPgkLJ