

Project "Mood" is supposed to be a web application wherein user gives us information how is he feeling right now. On base of that we match a movie or book that will be ideal for his current mood.

In our project we are using Django framework which is written in Python. Django will help us (or make it easier at least) create such things like logging, logging out and registration for user. To do that we needed a HTML-s help. That's where we realize, that not only writing a code is a problem but also connection all elements in the way that everything works. Django Auth has got ready templates to logging and registration, that's why it was not such a big deal for us. Next step which we took is that user after logging could see all movies that our application can propose and rate them or their level of matching into his current mood. That's where we made HTML file which takes from our base titles of the movies and shows it to user. In the tab for film evaluation we gave 3 possibilities (like, neutral, unlike) in our program we wrote values 1, 0, -1. Another important element that we learned when creating the project is validation. It is a check that the data complies with the application rules and user requirements. It protects us against program errors. For example, not to rate a movie that isn't there or assign it to a non-existent user. Then we added questions and several options for answers. Based on them, we have developed a function that will assign a weight to each question and a value to the answer. Thanks to this, after receiving a response by the user, the function will return a value from 0 to 1, which will correspond to our user's satisfaction (the higher, the happier)

While we were creating the project, we met the git consoles in detail. Until now, most of us have worked individually and rather on simple task, so we didn't pay attention to other 'branches' or functions. Unfortunately ,git' wasn't our friend and he caused us a lot of problems... sometimes more than writing a program. However, we understood that without him we couldn't work in a group.

Django is very well developed and has many built-in files. Initially, we underestimated its potential. Now, when we begin to understand why we need all this, we know how much work Django does for us.

Dataset:

I used is movie_metadata.csv from <https://data.world/data-society/imdb-5000-movie-dataset>

after some basic elaboration I created movie.json file that contains this attribute
"title","overview","genres","vote_average","release_year"

Sentiment analysis :

for getting the sentiment from movies I used "overview" , this was possible thanks to the tool vaderSentiment this is a description :

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. It is fully open-sourced under the MIT LICENSE
<https://pypi.org/project/vaderSentiment/>

For future realise of our project we want change out analyzer because we want make our one using other technique

Scrip:

1. sentAnalyzer.py:
 - python scrip that create our movie list with sentiment,rate,title,and genres , extract list of genre from our dataset

2. rate.py:
 - python script that take "rate" and the name of films that user want rate and update the rating
3. analyzer.py:
 - python script takes text and that return a sentiment value from 0 to 1
4. filtermovie.py :
 - python script that takes anserwe from user and genre of movies calculate a sentiment of user and give back a list of movie with " \geq " sentiment point
 - this script also update a filterhistory.json taking a history of user choise