* Went through this tutorial <https://levelup.gitconnected.com/using-tensorflow-with-flask-and-react-ba52babe4bb5>
* Trying to implement my own movie recommendation model in the above tutorial
  + After installing new dependencies flask server is up and running with new code
  + First attempt to call flask server returns

Text

Description automatically generated

* + - Errors on lines 60 and 33 of my code
    - Ending the server after this error did return with some data successfully:

Text

Description automatically generated

* + - Uncommented out ‘knn\_model.fit(movieUser\_scipy\_df)’. Was commented out because I wasn’t sure if the imported model had saved the fit from the jupyter notebook file
  + Second attempt to run returns error:

Text

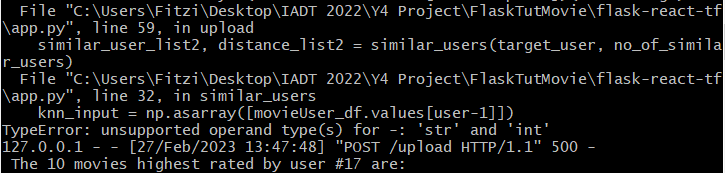
Description automatically generated

* + - The error was because I had left an old class name from the original toturial at the end, no mention of errors from first run attempt, looks like uncommenting the model.fit line had not been applied to the model before exporting as H5
    - Ending the server returned all the information I was looking for, promising

Text

Description automatically generated

* + Trying to pass variables from react to flask
    - target\_user2 = request.json = Error 400 bad request
    - target\_user2 = request.json.target\_user2 = Error 400 bad request
    - target\_user2 = request.target\_user2 = Error 500 Internal Server Error on ‘target\_user2 = request.target\_user’
    - target\_user2 = request.json["target\_user"] = Error 400 bad request
    - target\_user2 = request.form["target\_user"] = Error 400 bad request
    - target\_user2 = request.form("target\_user") = Error 500 Internal Server Error on ‘target\_user2 = request.form("target\_user")’ - TypeError: 'ImmutableMultiDict' object is not callable
    - target\_user2 = request.form.target\_user = Error 500 Internal Server Error on ‘target\_user2 = request.form.target\_user’ - AttributeError: 'ImmutableMultiDict' object has no attribute 'target\_user'
    - Removed ‘body: formData’ from fetch request and ‘e.preventDefault();’ from onSubmit
    - Added ‘const data = new FormData();’ and ‘data.append("target\_user", document.getElementById("target\_user").value)’ and added ‘body: data’ to fetch request
    - Flask server successfully receiving form data
  + Flask not receiving data in correct format, only accepts int or str but what it’s receiving should already be being sent as an int from the React form



* + - Added int() around the incoming data before being stored in variable for Flask



* + - Couldn’t take in any other form input other than ‘target\_user’
    - Tried to isolate where problem was before realising I needed to append the other inputs in the React app
  + Trying to send data back from Flask server to React
    - Needed e.preventDefault(); in onSubmit, not sure why yet
    - Imported StringIO and sys to store printed data
    - Sent stored printed data to React app
  + Trying to format data sent from flask to react in a way that it can actually be interacted with in react
    - Created empty arrays and then filled them with data as the back end app goes through it
    - Tried passing one of the arrays from flask to react but received error:



* + - Trying to format data as JSON now, imported jsonify, a flask library for translating data to JSON
    - Received same error
    - Tried going back to the lists and change them into JSON rather than arrays
    - Data is now being sent from flask to react as JSON, but not displaying on the page. Also the data received is a bit… odd. I don’t know why it’s using a mix of “ and ‘, and it only shows the beginning and end of the data and doesn’t expand.

Graphical user interface, text

Description automatically generated with medium confidence

* + - Tried receiving ‘data.data’ in React instead of just ‘data’ as I’m now sending the data under a spesified data tag, but there was no change.
    - Console logged data.data[0], or any other placement, it returns single characters rather than entire entries as I would have expected
    - Fixed data received by React app not showing on page, had to change prediction.message to prediction.data
    - Trying to use dictionaries instead of lists to store data and use json.dumps from flask instead of jsonify
    - Can’t store multiple entries with the same key in dictionary, it just overrides what was already there under that key
    - Figured out I can just add entire lists under one key
    - Trying to translate the dictionary to JSON and send to React, but am getting the same JSON error as above when using jsonify or json.dumps



* + - After trying every possible way I could think of, some ridiculous ways even, I discovered that there’s actually a python bug causing this: <https://bugs.python.org/issue24313>. Ways I had attempted and eventually succeeded below:

Text

Description automatically generated

* + - Used a function that converts any np.int64 variables to just int
    - Data is now being sent and logged by React app, but doesn’t show on page and crashes the display
    - Needed to target the specific data categories in new format, so ‘predicition.data.data’ becomes ‘prediction.data.Recommendations’
  + Trying to format output so that each individual movie can interact with it’s IMDb entry from the IMDb API
    - Trying to use a for loop to loop through the output and put each data point on its own line

Text

Description automatically generated

* + - Was receiving error: Cannot read properties of null (reading 'data')
    - Needed to add:



* + Ran out of free IMDb API requests, the app was automatically sending requests and I hadn’t noticed, I had thought I had 1000 free requests a day, 100 will be far too few, might need to either look for alternatives or temporarily buy access to more requests for the duration of the project

Text

Description automatically generated

* Downloaded some official and unofficial IMDb datasets with user ratings and IMDb IDs
  + Using pandas and numpy to format the data so that it can be used by my AI model(s)
    - After removing all episodes, shorts, series etc. brings the dataset down from 9.5m+ entries to just under 700,000. 700,000 is still a lot though, so I will try to merge with average ratings dataset and remove movies with less than x amount of ratings.
    - The merge and then removing titles with 0 ratings lowers the dataset to 325,000 rows.
    - Split the genres column so that each string of genres for each row was broken up to an array of strings with each string being a single genre, each unique genre is then turned into a column, every row has a Boolean value to determine if each genre column applies to it.
    - Removing rows with less than 10 ratings lowers it to 290,000.
    - Removing rows with less than 100 ratings lowers it to 132,000.
    - Removing rows with less than 1,000 ratings lowers it to 42,000.
    - Removing rows with less than 10,000 ratings lowers it to 10,500.
    - Removing rows with less than 100,000 ratings lowers it to 2,250.
    - Each of these filter options have been turned into their own csv file for future use.
    - Merged 700,000 just movies dataframe with individual user ratings dataframe (4.6m ratings)
    - Needed to pivot newly merged dataframe
      * Got error: ‘IndexError: index 875914235 is out of bounds for axis 0 with size 875909652’. Apparently it’s a Pandas bug with trying to pivot very large datasets, it’s been known about for over 5 years but no solution.
      * Started over with the previously exported dataframe of movies with only 100,000+ ratings, lowering list of 700,000 movies to 2,250.
      * When trying to pivot this one, received error: ‘ValueError: Index contains duplicate entries, cannot reshape’
        + The dataframe did not contain duplicates. Fixed by changing ‘pivot()’ to ‘pivot\_table()’
      * Everything else went fine and the dataframes were fed through the AI model and received an output for similar users to target user and movie recommendations for target user, but it did not output the top rated movies by target user.
      * Exported multiple sizes of dataframes
    - Even the smallest dataframe was too large, the exported csv file was over 4GB and crashed my PC any time I tried to read it in another python file.
      * Made a substantially smaller dataframe by filtering the users list by removing users with less than 15 ratings and merged it with movies with over 500,000 total ratings. This gave me a dataframe with about 300 movies and 20,000 users.
      * The exported CSV file was implemented into the react/flask app and works fine
* Setting up Express back end, connecting with MongoDB
  + Created registration and login functionality
  + Added email verification
  + Added password authentication using jsonwebtoken and bcrypt
* Expanding React app
  + Added a simple navbar
  + Added register/login form
  + Added page not found functionality for missing/wrong URLs
  + Created cards for the movies
* Imported my CSV of the 300 movies that have over 500,000 total user ratings to my MongoDB database.
  + Tried using MongoDB Database Tools/mongoimport
    - Errors saying mongoimport is not recognised as a command
    - Added mongoDB Database Tools to system path, no fix
    - Restarted PC, mongoimport now a recognised command
    - Attempt to import the CSV file using mongoimport in cmd prompt
    - Various errors
  + Tried using MongoDB Compass
    - Worked fine, added the CSV to my MongoDB database
* Expanding React/Express/MongoDB systems
  + Connected the Express app with the database of movies
  + Connected the movies to the cards
    - React didn’t like that it was returning as objects
    - Just minor formatting change where I call the card fixed it
  + Created a list collection with functionality in Express/MongoDB to store user’s personalised lists
  + Created a listContent collection with functionality in Express/MongoDB to as movies to user’s created lists
  + Added populate to get requests to receive more information on each call.
  + Login functionality stopped working, one of the variables was undefined. After some troubleshooting discovered that some password stuff was commented out in the user schema, not sure how, why or when, but uncommenting it fixed it.
  + Adding a link to each movies IMDb page in the movie cards, but the variable holding the IMDB ID was returning as [[object object]], even though it was displaying the data if the same variable was called display, it just wouldn’t use the data as a reference in ‘href’ for some reason. Just using the data from props worked instead, but it’s not ideal.
  + Added the functionality for users to add movies to their lists in the React app.
  + Saved user information to local storage on login
  + Getting, displaying and adding interaction with user lists based on user id in local storage
    - Need two calls to the API for this, the second call to the API is requires data from the first API call, but by the time the second API call Is run, the variable storing the first API call’s data is null
      * Tried adding an if statement that required the data from the first API call before running, which worked, but then the second API call would continuously make requests to the API multiple times a second, not ideal.
      * Tried adding a second condition to the if statement by declaring a variable and giving it a value of 0, then giving the condition of only running the if statement if the variable was > 1. It continuously called the API regardless, even though console log showed the variable as > 1.
      * Tried adding the second API call in the same useEffect() as the first API call, but seeing as I needed data from the first API call to make the second API call, it wasn’t working. The variable storing the data was again null at the time the second request was being made.
      * Tried making a second useEffect() just for the second API call, but same problem with variable storing data from the first API call being null when the second API call was run.
      * Tried adding the second useEffect() in an if statement, but conditions aren’t allowed with useEffects apparently.
      * Tried adding if statement inside the second useEffect(), it worked
    - Trying to display the data from the second API but it’s always received as an array of multiple [object object]s.
      * Trying to push the title of each movie to a list using a for loop, but it returns with movieID being undefined, even though if I console log the exact same line, it works

Text

Description automatically generated

* + - * Sss
  + Adding rating capability
    - Created a star rating component from MUI’s rating library
    - Created backend functionality for adding to a ratings collection on MongoDB
    - Tied the frontend component with the backend functionality, a user clicking anywhere on the row of 5 stars should add that movie along with the user who rated it and the rating given to the ratings collection
      * Got it all working except for storing and sending the rating info in time, when a user clicks on the rating it’s setting a variable to the rating chosen and at the same time trying to send data to the collection, so it’s sending the data before the rating value has been set. It works when trying to send the data the second time, because the rating value is already set then.
      * Tried to add a .then to the onChange functionality, but didn’t work
  + Adding ability for user to change which list and its items are displayed on their profile page
    - Added a dropdown menu from MaterialUI
      * It was using typescript, looked into ways to translate typescript to javascript
      * After a few minutes saw that they actually have javascript option as well as typescript, so just used that
      * Connected the list names taken from the API call to be displayed as options in the list
      * Got and stored the currently highlighted list option so that an API request can be made to get the list items for a specific list whenever the user selects it
* Sending new user ratings to the flask backend to be integrated with the AI model
  + Can’t find how to get index of users added to the system, am trying to get the index in a pivoted table from the userID. Can get the data based on userID, but can’t seem to get the index value.
  + Figured it out, had to use numpy not pandas to do this
  + Set up flask backend to receive data and add newly rated movies to the csv file so that the AI model can use it
  + Sent the data to flask from React, but having trouble sending the rating score, it tries to send before the rating variable has been set. It creates and adds a row with a rating of 0, but if clicked a second time when the rating variable has been set, does send the correct rating and creates and adds a row with the correct rating scrore.
* Lost all my insomnia requests for some reason, they did a new UI change recently that I haven’t learned, cant find anyone else with this problem, have emailed Insomnia support to see if it can be recovered any way.
* Getting and displaying movie information from IMDb API for single movie pages
  + I previously left off here having successfully setup the IMDb API to call a single time with movie ID taken from a previous API call to my own backend
  + Getting errors trying to display the data from the second API call as it’ trying to display the data before the data exists.
  + Turned out to be a simple solution where I just needed to add a question mark after the variable which makes it only be called if not null
* Adding search/filter functionality
  + Checked if MUI had any filter/search functianlity in their library, they did, but it was designed for tables mostly and didn’t seem suitable for my movie cards
  + Tried using useMemo from React to create responsive search bar, but I couldn’t use it conditionally, in the way I was trying anyway (outside functions)
    - Reformatted some functions and consts inside another function with useMemo so that it could be used conditionally
    - Partial success, the search box is there and does filter when anything is typed in, but it tries to filter on the first character typed and doesn’t save that character. Need to either only filter when a submit button is clicked, or keep the characters on typing and have results dynamically update on adding characters. As it is, copying and pasting a full title/word does work.
    - Have solved the issue of characthers typed not being saved in the search bar, but it’s still not ideal, have to reclick the search bar every time you want to add a character
  + Tried adding filter with hardcoding in some functions with if/else statements, but it wasn’t working and was getting rather confusing
    - Tried chaining the search bar function with a similarly formatted filter function with useMemo so that they can work simultaneously.
    - Tested it on average rating of movies and it surprisingly worked first time, search and filter works simultaneously (with just average rating filter at least.)
    - Set up filter for genres as well, but ran into problem of needing to access data like movie.drama which would return 1 if it is drama, 0 if not, but I need to be able to access it using a useState variable that has a string set to the genre, so something like movie.<useStateName> when the useState is set to ‘Drama’ would be functionally the same as movie.drama, but I don’t know how to do this and am struggling to know hot phrase this problem to even look up solutions.
* Adding pagination for movie homepage
  + Looking for libraries/frameworks that I can use for this, found one in chakra-ui, but when I tried to implement it, it broke the App. It also had a clashing library for a ‘Box’ component with an MUI ‘Box’ component, not sure if they would have been able to work together.
  + Found an example of pagination with just MUI on codesandbox for list items, tried to implement it with my movie cards and got it working. Also works with filters and search bar
* Added different colours for the star rating system. Stars are dark blue when hovered, light blue when rated by user, and remains yellow/gold when unrated by user. Still have problem sending rated movies to Express backend on first click, as the value is null when trying to send, but value is set on second click and sends the data. Double clicking also removes the users rating visually on the page, the stars go back to yellow/gold and are the same score as the average rating, not the current user’s rating.
* Trying to fix the rating not working on first click
  + Tried adding the function calls inside the setState, but apparently that’s deprecated and doesn’t work anymore
  + Tried removing the function calls in the star rating tags and adding a useEffect that runs the functions on value change. It half works, it successfully adds the movie rating to the Express/Mongo collection, but is an ‘unprocessable entity’ according to the Flask backend.
    - Not sure exactly what happened/what I did, but it’s working now with the useEffect
* Need to pair Flask’s output recommendations with Express/Mongo’s collection of movies on tconst (the IMDb ID)
  + Added Express functionality to get movie with specific tconst
  + Need to store x amount of get requests depending on how many movies are being recommended, to get each individual recommended movie from the Express/Mongo backend
    - Made a useEffect for the axios requests that waits until recommendations have been received from the Flask backend, and then loops through the individual movie get requests with each item brough back as recommendation from Flask.
    - Tried using a useState to store the Express/Mongo data, but on each looped request the data overrode the previously set data
    - Tried declaring an empty array and then pushing the response.data to that array, then once the loop has ended, set a useState to the filled array
      * Setting the useState outside the for loop wasn’t working, the useState remained empty, had to chain an additional .then to the axios request after pushing the response to the array, and the state is update on each request, containing all information after last request.
        + This made the function that was to assign the data to movie cards get called too early, it was waiting for the useState to be not null, so when the useState was update the first time with the first axios request, it was calling the movie card function. To prevent this, I just made the function wait until the useState?.length was => than the amount of movies being recommended
  + Trying to map the movie cards but it’s saying I need unique keys, but as far as I can see, they are unique. It’s using the MongoDB \_id as key, which can’t not be unique.
    - It was taking the key of the first object every time. Needed to add a [0] in the variable path
  + The recommendations are being received from Flask backend, the tconst being saved, then used to retrieve the tconsts movie information from Express/MongoDB backend, then that information is mapped to moviecards to be displayed. Only problem is that it only displays the first card of returned movie info.
    - Removed the if statement around the mapping function and put it around a middleman setState, which set a new state with the entire list of movies returned from Express/Mongo, that new state is then used to map only when all objects are present
      * React crashed as this created an infinite loop
      * Enclosed the if statement in a useEffect, and it’s working. Maybe.
      * The recommendations themselves are not changing, they are not updating when the user rates more movies, as it’s supposed to.
        + Trying to figure out what could be causing this. If I use Jupyter Notebook and run the python code with the dataset with updated user ratings, it gives the same recommendations as the app, and they’re the same no matter what user ratings are there. But if I use the old dataset, that doesn’t have new user ratings, and I manually add a rating in python for a ‘new user’, it gives back different recommendations.
        + I tried exporting the old dataset with manually added new user rating to compare with the dataset being used by the application, and they both have the exact same, single new user rating, but the exported dataset is 50% larger in file size and row size. I am clueless as to how. It just be the exact same size.

Just noticed that about 8k ratings got dropped from the larger dataset when I was testing with it, not sure what caused it, haven’t been able to replicate it so far

And now it is updating on most ratings, really not sure what’s going on, why it wasn’t working, or if it’s going to stop working again

* Adding functionality to display user ratings on movies they’ve already rated
  + Added functionality to the Express backend to send a get request for user ratings based on userID and movieID.
  + Added the call to the API to the movie card, took the rating value and and set the same useState as is set when a user rates a movie, so that the if/else statement would return the blue stars if it was already rated, just as they would change on new user rating. This triggered the post requests to both Flask and Express backends causing duplicate entries.
    - In the useEffect where the post requests are triggered when the value state is filled, I added a second constraint of requiring a second state to be null to trigger, this second state is given a value when a movie already has a rating and the get request to the API returns with data.

Text

Description automatically generated

* + - The onRate and onSubmit functions are the post requests to Flask and Express backends
* Added a heart icon for the movie cards for adding to favourites
* Added frontend registration functionality
* Tried adding backend functionality so that on new user registration, a favourites list is automatically created
  + Getting errors, it’s saving the newly created mongoDB ‘\_id’ when a new user registers so that a new list can be made with that ID, but when it trys to create that list the user hasn’t been added to the user collection yet, and I get objectID error.
* Added functionality to allow users to choose how many movies they see per page
* Adding edit functionality for users to edit their ratings
  + Added this functionality in Express backend
  + Added in React frontend, but not without problem. For some reason when it updates a rating, it also adds a new entry to the collection with the old rating details
* Showing all of a user’s rated movies on their profile page
  + Added functionality in Express to get all ratings by specific user
    - Was getting errors about the ObjectID for userID, saw it was saying the error was at a different function in the controller.js file, remembered having a similar problem like this before, just needed to reorder the routes in server.js
  + Added functionality in React to request, receive and display users rated movies
* Fixed genre filters being applied on selection
* Fixed selected filter settings not showing in the select input boxes
* Added option to reset filters
* Add second recommendation method in the form of similar movies on individual movie page
  + Ran through some tests on Jupyter Notebook and got the python code working with a dataset of the same movies I was already working with
  + Added functionality in Flask backend to receive request from front end and output similar movies in JSON format
  + Added request functionality in React to call and receive the data
  + Added a for loop inside and if statement that requires the JSON data from Flask, and loops through requests to the Express backend for the designated similar movies by tconst (IMDb ID)

To be done:

Fix homepage not showing recommendations unless React saves/refreshes

ix search bar losing focus on each character typed

Create favourites list for new user on signup

Create favourites list for new user on signup

Tried, but can’t seem to do simultaneously

Second Flask request with movieCard

* Fix the genre array thing for filtering

Check why rating some end movies doesn’t work the first few clicks

* Add second AI model

Get it working with cardsAdd show x movies per page

Let users edit ratings

Can edit, but also creates a new copy of the old rating.

* Show user ratings on cards
* LoginRequired in Express not working
* Show users rated movies on profile

Remove unnecessary navbar components

* Fix selected option not showing in input window when selected on movies page
* Add option to clear filters

? Give dropdown of current user lists when adding to list

? Multiple lines in the Home.js return ‘if else’ thing

? Fix changing list in profile page being delayed one action (maybe its because of setOptionIndex in handleMenuItemClick)

At end of report make sure:

Full stops at end of all captions

All front-end and back-ends are formatted the same

Line spacing between titles and paragraphs

Comment code (and python files)

Make sure dataframes and datasets are used appropriately

Break up researched part where researched by is said 3 x

Implementation chapter a bit more drawn out

Write some paragraphs on similar applications

Heading 2s on page breaks

in Design: Program Design: group techs and talk about groups

describe design pictures

Heading 2 bold

Break out longer figure captions to paragraphs

Resize comments in testing

Rewrite most of personal reflection part

Format reflection under one heading

Move reflection to conclusion chapter

Add to the future of recommender systems