Thrive Market Take Home Test

- Use any programming language of your choice.
- Provide clear instructions to build & run your code.
- Provide some basic documentation around your API endpoints.
- Dockerize your code if possible

Problem:
We are going to be building a product -> product recommender. A product to product recommender is a system that returns products similar to a given product. We have provided mock data for your convenience. Each line in the file is a valid json object.

As an example, a single line in the file looks like :


```
{"productid":19,"recommendations":[8234,229,2031,3914,225,14349,8894,8227,14656,44,459,10337,8635,8230,10335,10336,10334,10332,462,10324],"scores":[1.0,0.7071067812,0.5773502692,0.5,0.4472135955,0.4082482905,0.377964473,0.3535533906,0.3333333333,0.316227766,0.3015113446,0.2886751346,0.2773500981,0.2672612419,0.2581988897,0.25,0.242535625,0.2357022604,0.2294157339,0.2236067977]}
```


Productid -> The source product id of this line.

Recommendations -> all product ids that are similar to the given product id

Scores -> Affinity score of the recs product to the source product. (recommendations and their scores are in the same order) (productid 8234 has a score of 1.0, 229 has a score of 0.707)

Load the provided recommendation data file into a database of your choice.
Build a web-service with a GET API end-point : **/recs** that takes productid as a parameter.
So in this case **/recs?productid=19**
Should return a JSON payload like:


```
{
“status”: 200,
“message”: “recommendations for PID : 19”,
“data”:[8234,229,2031,3914,225,14349,8894,8227,14656,44,459,10337,8635,8230,10335,10336,10334,10332,462,10324]
}
```

Every productid provided in the file should receive a 200 response from the web service. For any other product ids, the service should return a 404 with an error in the message field and an empty list for data.

For other errors, the service should return a 500 response with an error in the message field and an empty list for data.

An easy solution would be to load the data provided into a key-value store like Redis database and building a small python flask-app for the web service.

You **don't** need to build a visual interface for this.
You **don't** need to worry about network security / access control for this.

Submission Instructions :
. Create a private git repository with your code and add @mihirkelkar as a contributor.
OR
. Zip your source code and email it to us.

Required

1. Functional Code.
2. Instructions to build / run code
3. Returns a response for every productid present in the provided file.

Bonus Points

1. Application is dockerized.
2. Swagger docs are available.

This should take around 3-5 hours to complete. Provide credit for all code that isn't yours. Use of the internet as a reference is allowed and encouraged as long as due credit is provided.