Machine Learning Engineer Challenge

Introduction:
Information overload is real and it takes quite a bit of time to sort out a lot of unstructured data. To make our movie nights easier, it would be useful to be able to understand at a glance what a movie is about.

The challenge:
We would like you to develop a simple command-line application: given a title and a short movie description it should return an appropriate genre.

Input Arguments

We expect the application to take these arguments:
`movie_classifier --title <title> --description <description>`

Input constraints:
```
--title: the movie title. A mandatory non-empty string.
--description: the movie description. A mandatory non-empty string.
```

Usage example:
```
movie_classifier --title "Othello" --description "The evil Iago
pretends to be friend of Othello in order to manipulate him to serve
his own end in the film version of this Shakespeare classic."
```

Output format
```
{
    "title": "Othello",
    "description": "The evil Iago pretends to be friend of Othello in
order to manipulate him to serve his own end in the film version of
this Shakespeare classic.",
    "genre": "Drama"
}
```

Guidelines:

- Feel free to use any programming language, framework and library you want.
- Make it concise, readable and correct.
- The performance of the model are not our main concern: we want to have a peek at how you structure an ML workflow, so don't spend too much time tuning your classifier but focus on delivering a reasonably production-ready system.

- We do not prescribe a genre taxonomy: feel free to use whatever genres you believe to be adequate for classifying movies.
- You can use any dataset to train your classifier and complete the assignment, as long as it's publicly available so that we can reproduce your work. We suggest the MovieLens dataset, but the final choice is yours.
- Write automated tests for your solution!

Please describe in `README.md`:

- How to run it (don't assume anything already installed);
- What libraries/programming language/algorithms you used and why;
- How to reproduce your work (training included).

Bonus points for:

- Good comments;
- Dockerfile.

Have fun, take your time. When you are done, please send a link to your public Github repo at recruiting@truelayer.com