A-Team (Group 1)

Kushal Agrawal, Joshua Levitas,
Akangkshya Pathak, Milind Thummala

IMDb Movie Genre Classification

October 09, 2022

I. Overview

IMDb (an acronym for Internet Movie Database) is an online database of information related to films, television programs, home videos, video games, and streaming content online – including cast, production crew and personal biographies, plot summaries, trivia, ratings, and fan and critics' reviews.

We intend to use the following dataset (freely available on Kaggle) to predict the primary genre of a movie, relying only on the natural language description of the movie on IMDb.

Genre Classification Dataset IMDb | Kaggle

II. Motivation

Our model will be able to predict genres from natural language descriptions, but that is not the only way it can be used. The knowledge extracted from this task can be transferred with little effort to more impactful problems. A few examples of the same are:

1. **Improving search results:** The trained language model can be incorporated into a search engine, to display highly relevant results to users looking for movies with specific themes or tropes. This is hard to do with simple text matching.
2. **Recommending similar movies:** The trained language model can be used to provide richer context to the recommender systems at IMDb or streaming platforms.

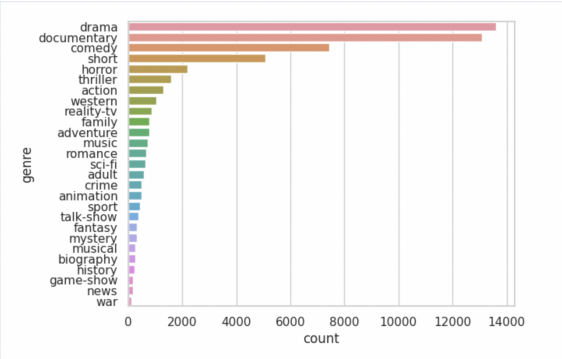
III. Dataset

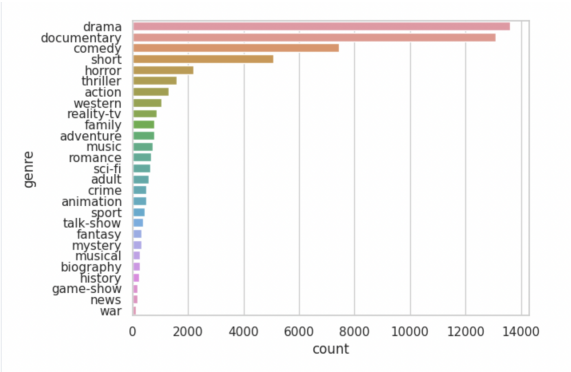
The dataset is split up into a training set and a test set, each with 54,000 examples. Each example has a numerical identifier, a movie title, a description, and a genre label. The first 5 examples are depicted in the following image (with truncated descriptions):

id int64	title object	genre object	description object
1	Oscar et la dame rose (2009)	drama	Listening in to a conversation...
2	Cupid (1997)	thriller	A brother and sister with a past...
3	Young, Wild and Wonderful (1980)	adult	As the bus empties the students for...
4	The Secret Sin (1915)	drama	To help their unemployed fath...
5	The Unrecovered (2007)	drama	The film's title refers not only to...

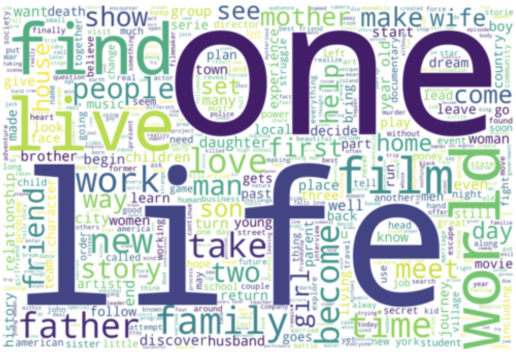
As part of Exploratory Data Analysis, we discovered the following key insights:

1. The dataset has no null, missing, or duplicate values.
2. As can be seen in the figures below, the distribution of movies over genres is almost identical in both the training and test sets, so we will not have to subsample from either dataset in order to equalize the distributions.





3. There are certain key words which occur much more frequently than others in movie descriptions, which weakens their predictive power. We generated word clouds (shown below) from all the descriptions (with stop words removed) for the training and test datasets to get an idea of what the distribution of common words looked like. We observe that both these datasets have similar words in their description and a few words like “one”, ”life”, ”find” occur frequently in their descriptions.



IV. Anticipated Method and Outcome

By the end of the project, we expect to employ a *pre-trained language model* (like BERT) as the workhorse for our system, while also incorporating engineered features extracted from the text. As a baseline for performance, we can use a simple TF-IDF based model.

For *text preprocessing* in advance of using the language model, we plan to tokenize the text description by word (and, perhaps, bi-words, tri-words, sentences, etc.), while also removing some subset of stop words. We would also like to explore the text fields by looking into grammatical structure and statistics (such as part of speech), or through extracting sentiment.

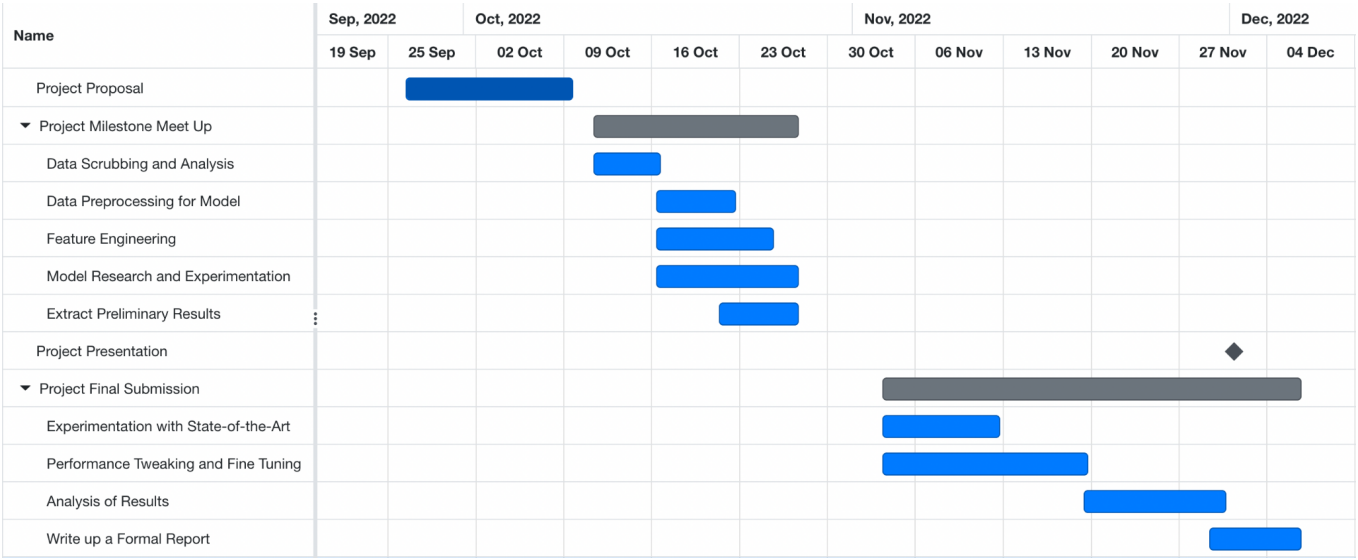
For *fine-tuning the language model,* we will need to embed it at the core of our model and train all the free parameters on our training set, with the model error determined by its performance on our task of movie genre classification.

For *model training*, we plan to first use a small section of the dataset and see if we are able to extract the desired features correctly and then use the full dataset. We will also experiment with changing the hyperparameters for the model using grid search.

V. Teamwork and Project Timeline

We envision the timeline to be as follows:

Task	Start Date	End Date
Project Proposal	Sep 26, 2022	Oct 09, 2022
Project Milestone Meet Up	Oct 11, 2022	Oct 27, 2022
Data Scrubbing and Analysis	Oct 11, 2022	Oct 16, 2022
Data Preprocessing for Model	Oct 16, 2022	Oct 22, 2022
Feature Engineering	Oct 16, 2022	Oct 25, 2022
Model Research and Experimentation	Oct 16, 2022	Oct 27, 2022
Extract Preliminary Results	Oct 21, 2022	Oct 27, 2022
Project Presentation	Dec 01, 2022	Dec 01, 2022
Project Final Submission	Nov 03, 2022	Dec 06, 2022
Experimentation with State-of-the-Art	Nov 03, 2022	Nov 12, 2022
Performance Tweaking and Fine Tuning	Nov 03, 2022	Nov 19, 2022
Analysis of Results	Nov 19, 2022	Nov 30, 2022
Compilation of Formal Report	Nov 29, 2022	Dec 06, 2022



A simple look at the Gantt Chart will reveal that some tasks are planned to occur parallely. This is because specific tasks will be delegated to members of the team. The initial division of labor has been planned as follows:

- **Data Analysis/Cleaning:** Milind and Akangkshya
- **Data Preprocessing:** Joshua
- **Feature Engineering:** Kushal
- **Research and Scope Analysis:** All members are expected to contribute
- **Development and Experimentation:** All members to run their own experiments before we settle on a common codebase

These responsibilities will rotate after the milestone meet-up, to allow every member to contribute to every facet of the project.

VI. Afterword

Some details regarding the exact methodology of the project have been intentionally left vague, to allow for flexibility during the execution phase of the project.

In the same vein, we have added time buffers to our task estimates, to accommodate contingencies. Many tasks in the timeline will probably take less time than has been allotted to them.

We are confident that with these considerations in mind, we will be able to deliver the project within the allotted time.