**Predicting Movie Recommendations by Leveraging Deep Learning and MovieLens Data**

by Annie Phan, Drew Solomon, Yuyang Li, and Roma Coffin

**Project Overview:**

Recommendation systems use consumer data to develop personalized preferences to customers. Tech companies have honed in on this strategy as it has proven to be successful in enhancing user-experience, popular examples include restaurant suggestions on Grubhub, playlist suggestions on Spotify, product suggestions on Amazon, or movie suggestions on Netflix. Recommendation systems typically use clustering, nearest neighbor, or matrix factorization techniques. Deep learning models have recently increased in popularity though to overcome limitations of these methods and increase prediction accuracy.

By accessing the MovieLens dataset which consists of 1,000,209 ratings on 3,900 movies from 6,040 MovieLens users and leveraging deep learning, one goal is to build better movie recommendation systems that more accurately provide personalized content for the modern consumers. Or find new applications…

While our main objective is to predict movie recommendations using MovieLens data, our aim is to replicate a previously created model and improve upon it and also find new applications. This first blog post will aim to shed light on the data collection process, exploratory data analysis, model methodology, and next steps for this project. Please do take note that although changes were made, information was used from the original research project site to assist with our analysis.

Paragraph about James Le’s project - who is he, why did he do the project, overall approach - he has his linkedin profile , why did he choose his code and not other people (there are many people who work on this dataset) - it’s because his approach is clean, applicable, but also very good results, etc - fits both the “state of the art” requirement but also practical for us to do since we are newbies, have little time frame, don’t know much about recommendation system yet

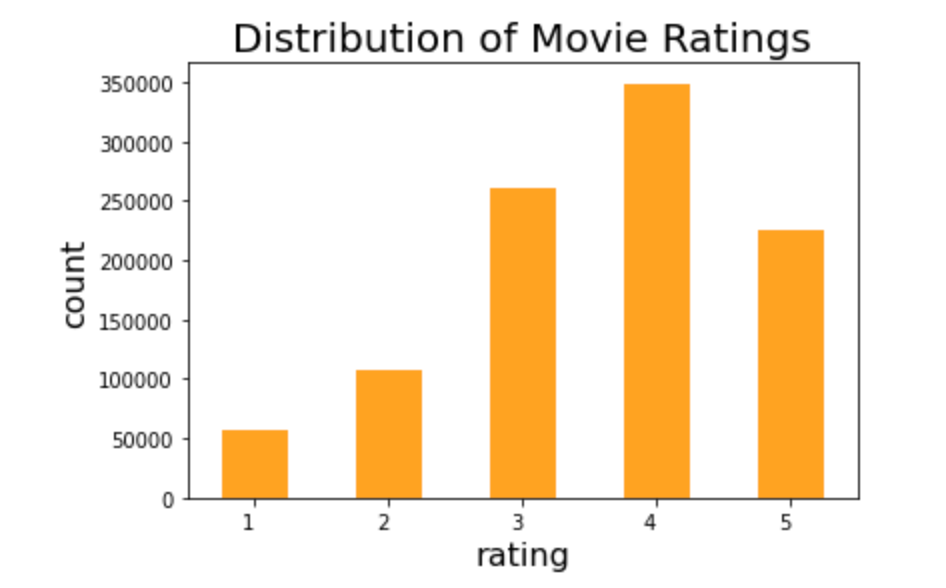
**Our rationale for choosing the project:**

* Talk about how recommendation systems are right now
* Talk about how we are challenging ourselves because Recommendation System is the last topic to be covered in our class, so we are using all the DL models we have been taught and thinking ahead to improve it, etc
* Challenges we faced: his

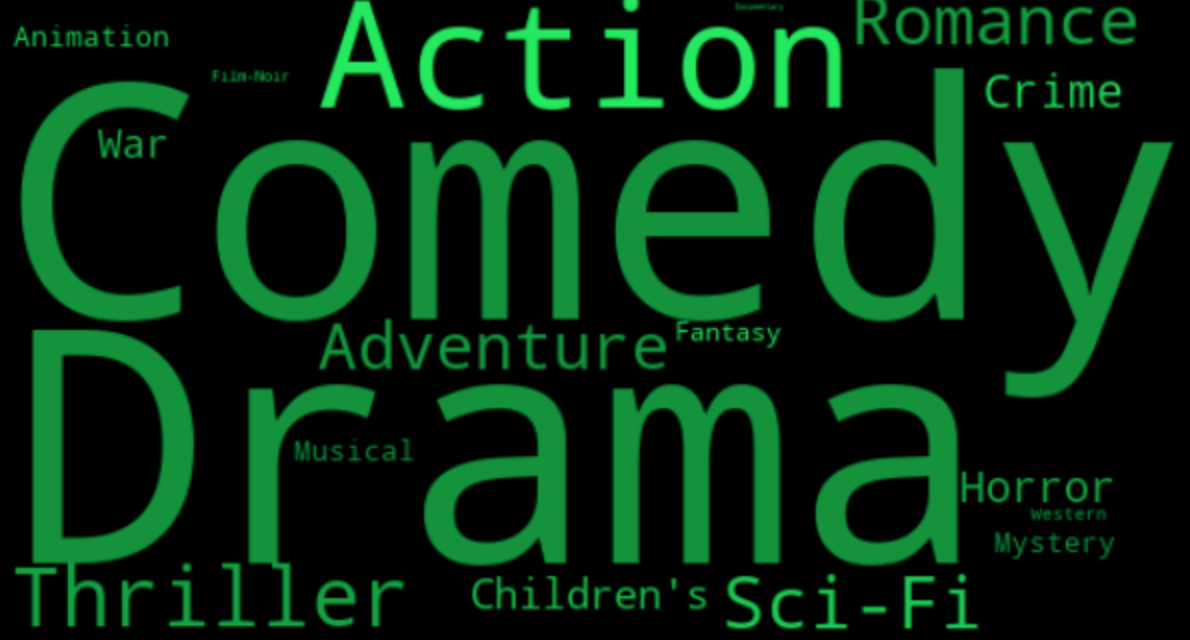
**Exploratory Data Analysis:**

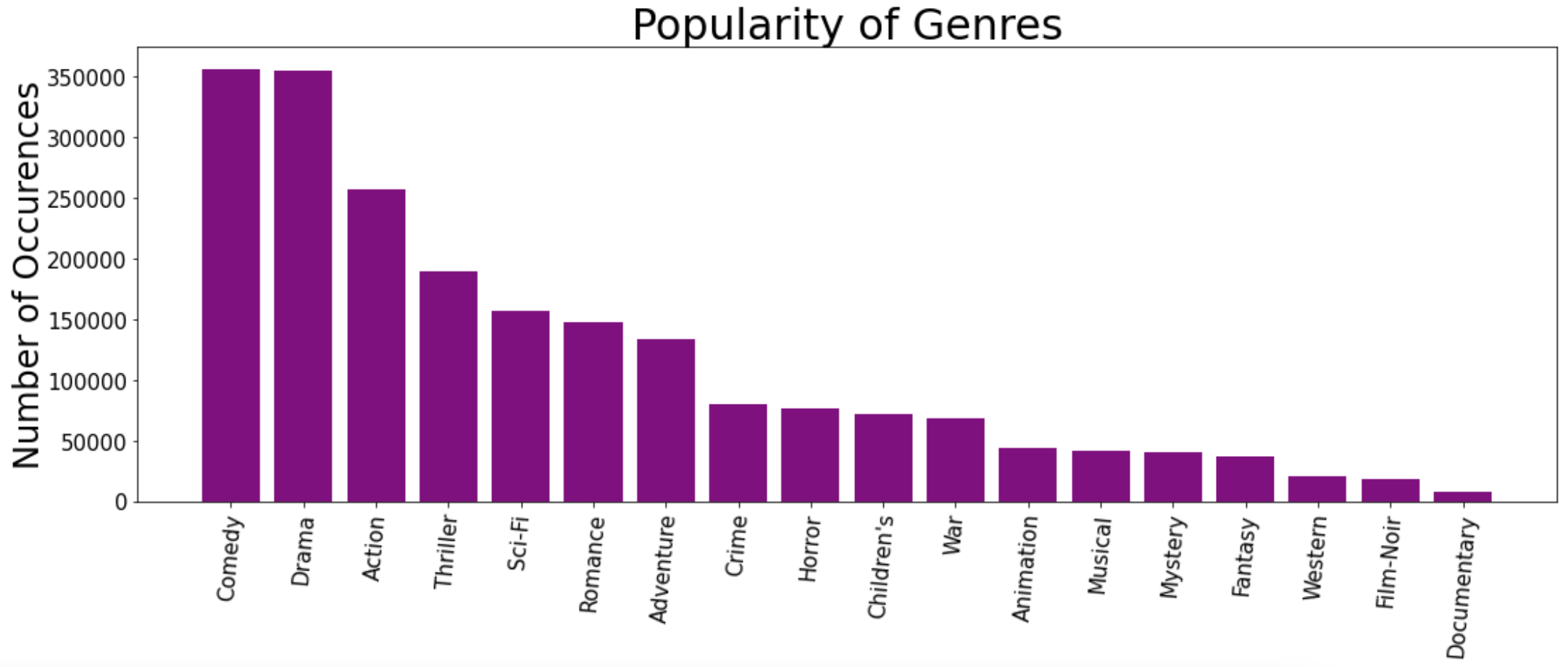
For us to better understand the features in the MovieLens dataset, exploratory data analysis (EDA) was performed. EDA helped to familiarize us with the three datasets used (Movies, Ratings, and Users) and when the three datasets were merged by capturing common data patterns and to help us with data visualization. Additionally, our EDA process will help us discover if the deep learning approaches being applied such as recommendation systems that are being considered are applicable.

We started by checking to see if any of the three datasets had missing or confusing information, as we would desire to remove missing values or fix any faulty or corrupted aspects of the data. Luckily, no unusable items were found, but we did notice that the genres feature had 301 different types which we will discuss in more detail later on in this blog post. Next, we wanted to get a sense of the distribution of ratings in the dataset. This bar chart reveals the dataset is skewed to the left, slightly imbalanced, and the rating 4 was the most dominant. This is important to know so in the future we can potentially stratify (e.g. using stratified K fold) to ensure all of the ratings are adequately represented across training, validation, and testing datasets.



After checking the ratings, we decided to build a Wordcloud visualization and bar chart to see the number of occurrences for each genre. Genre output was edited because there were 301 different categories for genre. Since, some movies had multiple genres (ex. romantic comedy would count as romance and comedy) they were counted as each genre separately. We can see that comedy and drama are the most present genres.This could give us a better idea of what potential bias we might have in the training set. Therefore, we can try to eliminate the bias during our model constructing stage.





**Baseline Model Methodology & Results:**

These **systems** rely on both implicit data such as browsing history and purchases and explicit data such as ratings provided by the user.

* Talk about the author’s preprocessing methods: explain his reasons and critique it → anything we can do better, especially from EDA?
* Talk about how the author implemented 4 models but we are only focusing on the deep learning approach
* Brief explanation of collaborative factoring and matrix factorization in recommendation systems
* Talk about the structure of his model, include the photo for the plot of the model below plot(model) in the colab, talk about how it has user and movie vector, dot product, etc → this is a vanilla implementation of CF in deep learning (I think it’s called Embedding model but please check to be sure) → refer to his medium post: <https://le-james94.medium.com/the-4-recommendation-engines-that-can-predict-your-movie-tastes-bbec857b8223>
* Talk about training performance, RMSE score, loss, training time (he used CPU and took 3 hours, we used GPU and took ~45 minutes for 30 epochs) → things like this we need to keep in mind moving forward

**Next Steps:**

Our next step is to explore 2 methods to further this project: the “Algorithm” approach and the “Application” approach.

Might decide to narrow down to 1 approach later if more promising…

With the “Algorithm” approach, …. improve upon the results from the baseline model. This will be accomplished by further looking into developing more complicated deep learning models to attempt to estimate movie recommendations more accurately. Examples: Dense Layers, using Transfer Learning, adding regularization, changing the learning rate schedules, changing the optimizer, take a look into these articles and name some of the approaches in here as examples: <https://medium.com/@jdwittenauer/deep-learning-with-keras-recommender-systems-e7b99cb29929>, <https://www.onceupondata.com/2019/02/10/nn-collaborative-filtering/>

By adjusting the hyperparameters and building and refining our model structure we can improve upon our performance and achieve a higher accuracy score. We will also explore different models for transfer learning, and experiment with hyperparameters via cross-validation.

With the application approach, we will pick a Deep Learning application and explore how best to solve it using different models, or focusing on improving one model. the author currently implements an Embedding model for Collaborative Filtering (look it up), we need to do a different type of DL models, such as transformers, RNN, CNN. From our research, some applications that has great results or interesting or whatever adjectives are Variational autoencoder: <https://github.com/noveens/svae_cf>, Transformer: Kera’s transformer implementation on the MovieLens dataset: <https://keras.io/examples/structured_data/movielens_recommendations_transformers/>

In order to achieve these technical approaches, we are planning to do more research, reaching out to the author, looking at other people’s implementation

**Citations:**