Introduction

The main objective of this document is to outline the data structure and features used for the classification and the methodology that will be pursued to overcome the two main hurdles: dealing with missing data and multi-label genre classifications. It also includes an outline of the modeling approach that will be performed

The Data

The data will be mainly collected from the TMDb database based on a list of 12,000 movie id numbers that we acquired. These id numbers are also linked to IMDb id numbers in order to complete our data acquisition. Some additional information, specifically the movie director, will be acquired from the IMDb database. We pulled the 12,000 movie id's and randomized their order. In order to create the training and test set partitions, we will randomly sample here as well. There are lots of instances where not all of the prediction features are available for each observation, so our method for dealing with this missing data will be to just eliminate those observations because we have such a large pool to draw from. We will examine the eliminated observations in order to ensure that there is not a systematic reason for missing data that could skew our model by its removal.

The data will be partitioned as follows (before missing-data removal):

- Two training data sets each includes 5000 movies
- A test data set of 2000 movies

Features

There are many predictors available on the TMDb site that we could utilize, but it seems that there are a few that will be most predictive for our models. The features used for the classification will be as follows:

- Title
- Budget
- Revenue
- Director
- Keywords (and Overview)
- Runtime
- Poster (converted to grayscale and PCA transformed)

We can utilize a bag of words approach with the Keywords and do a similar approach with the overview if the keywords are missing. The quantitative variables of budget, revenue, and runtime will be simple measures that can be included directly. And, we hope, the directors will be used to identify certain areas of expertise that will be predictive.

One additional feature that we will also include is basically the average and the variance values of all colors in the RGB vector format of the poster. The reason for this additional data about the poster is that the poster color composition and intensity variability conveys a message about the movie content and hence is a direct indicator to the movie genre. We believe that a lot of the movie mood can be conveyed by color schemes where our PCA analysis of the posters may be too coarse to pick up on small details that would otherwise indicate this mood.

We selected these features for a few main reason. First of all, they are available for the majority of movies and missing data should not be much of an issue. Second, we believe these are the best predictors when our final goal is classification of genres. Third, these categories cover a lot of the major components of movies so even if our intuition about classification is incorrect, the model should be able to learn differences between the genre classes.

Y Labels

Our Y data will be constructed as a matrix of genre multi-labels with rows equal to the number of movies (observations) and columns equal to the number of unique genres (classes). For this purpose, a unique list of the available genre is extracted from the TMDB database. This list will constitute the label matrix to which each movie's classification will be performed.

id	genre
0	Action
1	Adventure
2	Animation
3	Comedy
4	Crime
5	Documentary
6	Drama
7	Family
8	Fantasy
9	History
10	Horror

11	Music
12	Mystery
13	Romance
14	Science Fiction
15	TV Movie
16	Thriller
17	War
18	Western

Each movie will be assigned a binary Yes / No that corresponds to each unique genre. These binary indicators are not mutually exclusive to accommodate the possibility of a movie having more than one genre classification. In fact, most movies have multiple genre classes, so we will try to capture this behavior.

Genre Imbalance

When breaking down each movie into a vector of its genres, we find that certain genres occur with much higher frequency than others. With 10,000 training data points, we should be able to adequately identify the traits that differ between genres. However, we want to make sure that we do not overload the training data with a certain genres and make it so that these genres are the default prediction without ample cause.

What we may do with our data is under-weight the high frequency genre types in order to make the genre weights more even. We can accomplish this by randomly removing observations (movies) that fit an over-represented genre category. Another way we could balance our data is by replicating the sparse genre types, but this seems problematic when we are dealing with poster visuals and movie-specific data because it could skew the true parameters for the low-frequency genre type. We seem to have enough data at our disposal that paring down the data is a reasonable option.

If this approach does not seem to work for us, we may go toward a solution that balances the data by combining the under-represented categories into one “Other” genre type. If we find that our under-weight model does not end up predicting these low-frequency genre types, we can combine a number of low-frequency types into one category so that it is more balanced with the other data. This is a last resort as it could distort categorization more if the obscure types are

very different but are artificially put into the same categorization bin. This is something we will have to address later on.

Modeling

In general two types of classifiers will be considered. The first is the MLP deep learning model and the second is a conventional classifier

In the case of the conventional classifier and to address the multi-label nature, a separate model will effectively be implemented to each genre in a model bank anatomy. Each individual model will yield a binary prediction on its respective class to indicate its predicted membership status.

The overall selected features can be categorized in three groups:

- The pixelated poster
- The movie overview bag of words
- All other selected features

A preprocessing will be performed by reducing the dimensions of the bag of words and the pixelated poster as a first step. This dimensionality reduction will be performed by PCA decomposition focusing on components contributing by 90% of the data variability.

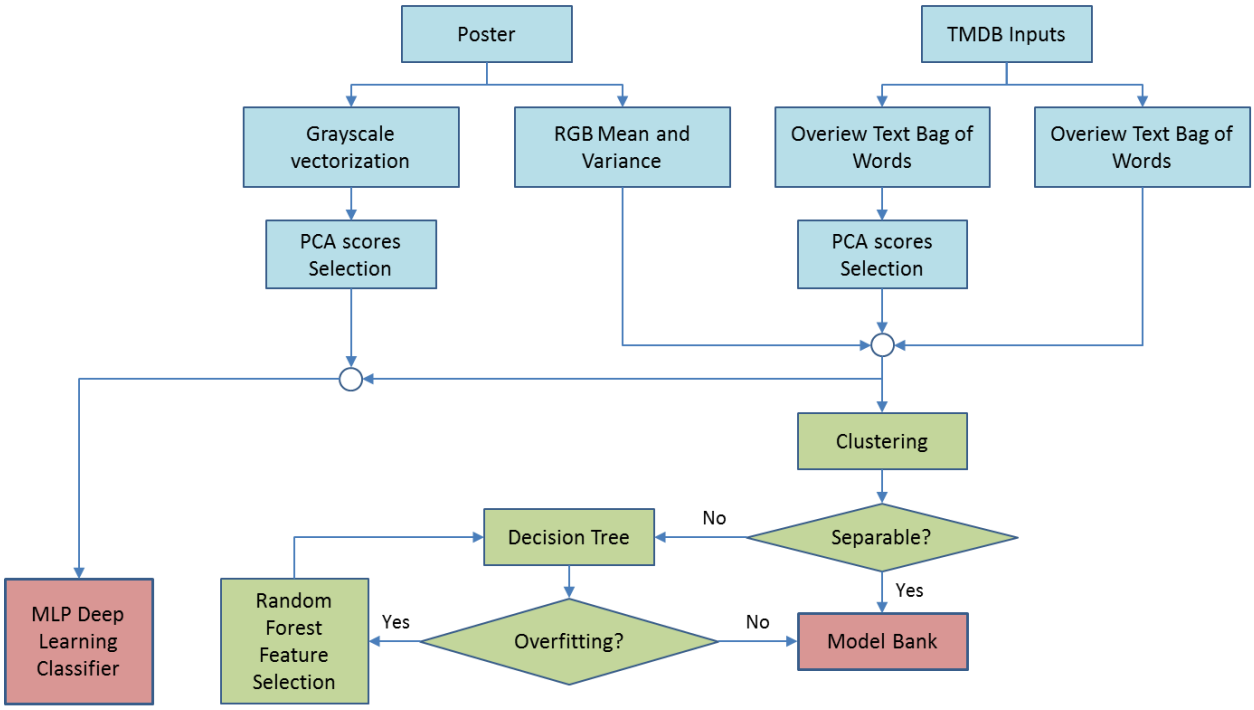
All of the three groups will be used for the MLP deep learning model which will have a separate output node for each unique genre class as an accommodation to the multi-label nature of the classification.

For the more conventional classifier, only the reduced bag of words features and the other inputs will be considered in order to optimize the feature space dimension and avoid overfitting. A data clustering method will be performed to assess the class separability. If it is determined that the data classes constitute distinct clusters, a logistic regression will be implemented for each movie genre as per the architecture described above. The logistic regression is selected based on the binary classification nature of each model and also the fact that its tuning is minimal compared to the SVM. If the data clustering reveals that the feature scatter represents a challenging class separability, a decision tree with bagging will be be considered in lieu of the logistic regression.

In both cases and for each individual model in the model bank, a random forest will be implemented to evaluate the feature importance for each movie genre classification. Accordingly, a subset of the feature space could be determined for each movie genre classifier. However, this approach could represent an added complexity to the model bank structure that can exceed the complexity of the deep learning classifier. This approach is considered as a last resort if overfitting cannot be avoided.

The following Figure illustrates the modeling approach described above:

Movie Genres Classification Modeling Process



Lastly, here is the link to our github repo: https://gitlab.com/cs109bFinalProject/dataexplorations

It is setup to allow those with the link to request access. With this, you can feel free to examine our most recent patch to see our code for scraping the data.

Group 26
Nisreen Shiban
Hany Bassily
Nina Iftikhar
Dominick DeLucia