Network based prediction of movie Nomination to the Oscars and the Golden Globes

Alice Bizeul, Gaia Carparelli, Antoine Spahr, Hugues Vinzant January 9, 2020

1 Introduction

On January 5th and February 9th respectively the Golden Globes and Academy Awards (also known as Oscars) ceremonies take place and reward some of 2019's best movies. Indeed, these ceremonies assign the most prestigious Awards in the film industry. Because of this renown, there is a big business around getting insights on which films could potentially compete and even win these rewards. A simple example could be the interests of a movie director that wants to understand and anticipate whether the casting he recruited could potentially make his movie compete to win an Award. Another scenario would be a film corporation which could be interested on whether a movie it is funding is actually worth the effort and investments.
Of course the prediction of the chances of a movie to get a Nomination to an Award such as a Golden Globe or an Oscar taking into account its specific characteristics is not straightforward. In fact as we can imagine, understanding the patterns behind a movie winning a reward is highly complex and a lot of factors have to be combined and considered in order to have a robust guess about a movie success. The subjective aspect in the voting in order to attribute an Award has to be taken into account since each member allowed to vote gives her/his preferences according to her/his own movie tastes.
In the present report we will investigate and try to predict if a given movie has a chance of getting nominated to either the Oscars or the Golden Globes. To answer this study question we strongly believe that it is not sufficient to look trivially at actors or the movie director and for this we chose to explore movies and their associated characteristics using a Network Graph. In this way, a lot of the too highly dimensional information around a film could be visualized in an easier format and more features could be considered at the same time. Moreover, this network organization of data was also exploited for the prediction of movie Nomination which could have not been feasible with classical machine learning techniques.

2 Data Acquisition and pre-processing

In order to answer our study question, the subset of the IMDb data from Kaggle[1] was used. This dataset contains information about 4803 movies released mostly between 2000 and 2018.
The IMDb dataset does not provide information about Nominations and Awards neither for Oscars nor Golden Globes. To counter this, extra data has been scrapped from the web with the goal to combine the collected information to the existing movie dataset. Specifically, the Oscars[2] and the Golden Globes[3] nominees and winners were scrapped directly from the corresponding websites.
At this point, some data cleaning was performed and one movie (*America Is Still the Place*) was removed from the dataset since it had most of its features missing leaving a remaining of 4802 movies. Then, two movies (*Chiamatemi Francesco - Il Papa della gente* and *To Be Frank, Sinatra at 100*) had a missing *runtime* that has been filled in manually using a web research.
After that, the three distinct datasets (IMDb, Oscars and Golden Globes) were merged based on the movie name and the year. Adding the year as a merging criterion was done to avoid confusion when movies had similar names (e.g. there are two *Titanic* movies who got Oscars, one in 1953, the other in 1997).
Finally, for each movie the following features were selected and added in the final dataset: list of crew members, list of cast members, list of keywords, list of genres, runtime, budget, revenue, popularity, vote count and vote average. The labels were instead: Nominations (sum of the Golden Globes and Oscars Nominations) and the corresponding number of Awards.

3 Network Construction

In order to proceed with the network construction, four different features were used to connect each movie: the list of cast and crew members, the list of keywords characterizing the movie and the list of genres. Those categorical features are high dimensional and very sparse having 54'202 different cast members, 52'235 different crew member, 9814 different keywords and 21 different genres. Therefore, using a network, where each movie is a node, and an edge weight reflecting the similarity between two movies based on the four already mentioned features could help mitigate the high dimensionality and sparsity of data. In this way the information of the high dimensional features is kept and transposed in the network structure.
In practice, the adjacency matrix was obtained by computing cosine similarity between every possible pair of movies based on a vector of cast, crew, keywords and genres. Finally, it was decided to keep only edges with a weight of 0.25 or greater, meaning that the weights of all the edges with a similarity below this threshold were set to 0. This was done to sparsify the network and remove connections of poorly similar movies.
To summarize, the network is thus composed of 4802 nodes, each representing a movie. The nodes are connected based on their similarity of cast members, crew members, keywords and genres. Aside from that, six classical signals (movie characteristics) are used as features for each node: revenue, budget, runtime, vote score, vote average and popularity. Moreover, the genres were also

[1] https://www.kaggle.com/tmdb/tmdb-movie-metadata#tmdb_5000_movies.csv
[2] https://www.oscars.org/oscars/ceremonies/
[3] https://www.goldenglobes.com/winners-nominees/

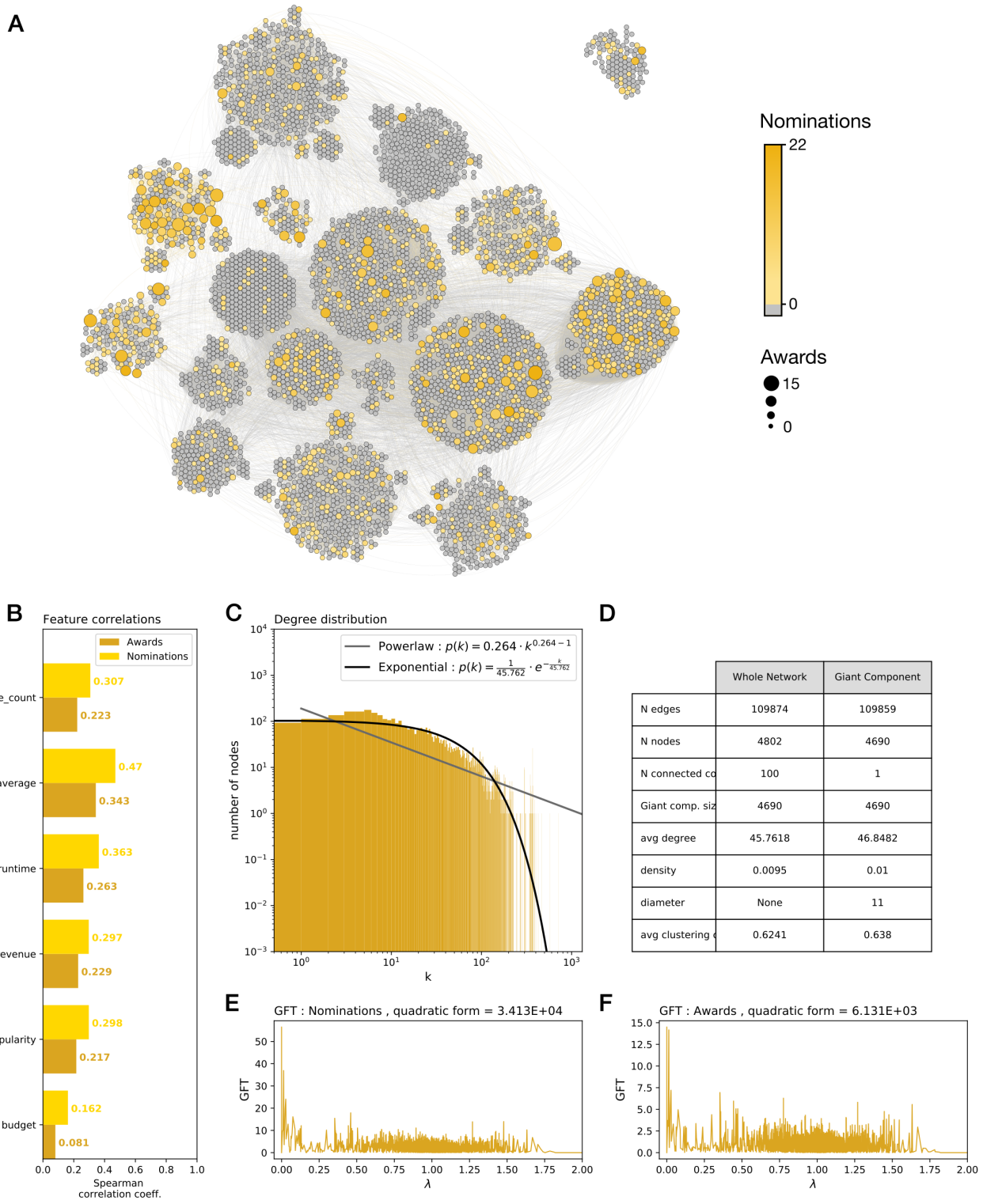

Figure 1: Network Summary. The network structure and properties are presented. **A** A visualisation of the network in a circle pack layout based on the connected component, the modularity and the clustering coefficient (done using Gephi visualization platform). The color represents the number of Nominations to an Oscar or a Golden Globes. The size of the nodes indicates the number of Awards obtained (Oscars and Golden Globes combined). **B** Bar plot of Spearman correlation coefficients between the six features and the total counts of Nominations and Awards. **C** Degree distribution in a log-log scale. Two fit are presented : powerlaw and exponential. **D** Summary table of the network properties. **E, F** Graph Fourier transform (GFT) respectively of the Nomination and Awards signal on the network. The quadratic form ($f^T L_N f$ where L_N is the normalized Laplacian and f the signal) of the signals is also given in the title. An interactive visualisation of the network is available at https://antoine-spahr.github.io/Movie-Network-Visualisation/.

added to the feature list as dummy variables (1 if the movie if of this genre, 0 otherwise) and each node exhibits labels corresponding to the number of Nominations and the number of Awards. Finally, in order to facilitate further learning, self loop have been added to each node.

4 Exploration

4.1 Prior Data Exploration

First of all, the data was explored independently of the network structure in order to gather some information on the different signals correlation with the labels Awards and Nominations) and the labels distribution.
The Spearman correlation coefficient between the classical graph signals and the labels were computed to get insights of the relevance of the available features to predict if a movie had or not a Nomination or an Award. The coefficients are presented on figure 1B as a bar plot. Overall, the correlations do not exceed 0.47 and are weaker with the Awards than with the Nominations. The vote average presents the strongest correlation while budget is the less correlated. In addition the results indicate that movies with higher average vote or a longer duration tends to be more nominated.
In brief, the features seems to carry information that may enable to predict whether a movie is nominated for an Awards or not and training an algorithm on those should yield above random results.
In addition, it was found out that the dataset did not have a balanced distribution of Awards nor Nominations. Only 22.4% of the movies have one or more Nomination and 8.6% of the movies have one or more Awards.

4.2 Graph Exploration

4.2.1 Graph Properties

First of all, some properties of the graph were explored such as the number of edges, the number of nodes, the number of connected components, the giant component size, the average degree, the density, the diameter of the giant component and the average clustering. These numerical values are presented in a table on figure 1D.
The network is composed of 109'874 edges and is therefore considerably sparse (density of 1%). Most of the nodes are connected to the giant component (97.7% of the nodes). The average clustering coefficient is rather high (0.6241) and indicates that, on average, the neighbors of a nodes are also connected which support the presence of a small-world property on this network. The diameter of the giant component is quite large (11) and suggests that the network's hubs can be quite distant and there are no central movies connected to most of the others. Finally, the average degree of a node shows that movies are on average connected to many others.

4.2.2 Graph Type

In a second step, to get more insight on the connection structure of the network, the degree distribution is observed (figure 1C) and two curves are fitted to it: a power law and an exponential. It appears that most of the nodes have a rather small degree but some are connected to many others (up to almost a thousand). The power law curve does not provide a satisfying fit of the degree distribution while the exponential fits it quite well. It suggests that the network is a scale-free network in a sublinear regime, meaning that there are smaller and fewer hubs than in linear (power law) scale-free networks. This observation is consistent with the high diameter of the giant component (1).

4.2.3 Visualization and Nodes Properties

The network structure is further investigated by the mean of a graphical visualisation. Figure 1A shows the network displayed using a circle pack layout based on the components, the modularity classes and the clustering coefficient. The modularity enables to detect communities within the network and it appears that the network does form communities of movies. A deeper investigation of the genres distribution over the network highlights that the network tends to group movies that share similar genres. The similarity in cast member, crew member and keywords may increase the clustering by genres since most the actors and producers tends to work on movies of similar genres (Hugh Grant played in many different comedies for example) and movies of the same genre are likely to share similar descriptive keywords.
Then, to explore whether the network structure can improve the prediction of movie Nomination or Awards, the nodes on figure 1A are colored according to the number of Nominations and sized according to the number of Awards a movie has got. One can observe that movies of some communities such as *Not another teen movie, yes man, 17 again* corresponding to comedies are rarely nominated for an Oscars or a Golden Globes while other communities (containing movies like *American beauty* and *Pursuit of happiness* which are dramatic and more engaged movies seem to present a high concentration of Nominations. By consequence, the network structure seems to carry information that could improve the prediction of Nominations or Awards. Note that some movies are not connected to the giant component (upper right of the visualisation) and among them, some have been nominated and have received Awards such as *Tom Jones* or *A Passage to India*.

4.2.4 Attributes Analysis

The distribution of the Nominations and Awards signals on the network is further investigated by computing their graph Fourier transform (GFT) and their quadratic form ($f^T L_N f$ where f is the signal and L_N the normalized Laplacian). The latter is very useful to determine the smoothness of a signal over the graph (the higher the quadratic form the more noisy the signal). The two signals GFT and their quadratic forms are presented on figure 1E and 1F. The quadratic form of a constant signal of ones is equal to 9.584E+02. As a results, the Awards signals appears to be smoother than the Nominations one since the quadratic form is smaller. This behavior may be due to the small number of movies that have been awarded, the signal is then rather smooth because most of the nodes have a value of zero. The GFT of both signals highlight a rather noisy signal as many peaks are present at eigenvalues between 0.5 and 1.5.

After all of these considerations, for simplicity sake and

especially to counter the strong imbalance on the Awards distribution, it was decided to put the focus only on the binary prediction of a movie Nomination to either the Oscars or the Golden Globes.

5 Exploitation: Nomination predictor

At this point, it was possible to proceed with the implementation of a predictor able to discriminate between movie having received at least one Nomination (to either an Oscar or Golden Globe Award) from a movie which did not receive any.
We started by considering 7 features (namely : budget, popularity, revenue, runtime, vote average, vote count and genres) and the labels: Nominations vs not. At this point, it appeared that some features (budget, popularity, revenue and vote count) had a very skewed distribution and were thus log-transformed.
Afterwards the dataset was split into train, validation and test set (respectively 64%, 16%, 20%) by keeping the same class proportion in all of the three subsets.

5.1 Models

To be able to find the best predictor for the binary classification class (where class 1 is represented by the movies having one or more Nominations and class 0 being the movies not nominated at all), different classifiers were tested. To have a sort of baseline, a random classifier was implemented at first. After that, several classification methods were implemented among which simple classifier and Graphical Convolutional Network (GCN):

1. Simple Logistic Regression (LR)
2. GCN based Laplacian polynomial filter + LR
3. GCN architecture

If the first method do not take advantage of the Graph representation of the data, the last two specifically rely on that for the learning. In the second method, the algorithm uses GCN to learn coefficients of a polynomial Laplacian filter and apply them for learning using Logistic Regression. The third method is based on a GCN architecture containing 3 linear layers (one at the beginning and two at the end) and 11 Graphical Convolutional (GC) layers. The number of GC layers has been chosen from the diameter of the giant component in order to convolve until the 11th-order neighborhood of every node (i.e. to reach all nodes) (2). The chosen activation function was ReLu for the hidden layers and log-softmax for the last one. For both model 2 and 3, the chosen loss was the Cross Entropy (which suited very well for the binary classification task), and Adam was used as optimizer.
Since classes are very unbalanced, an appropriate weighting was added in the computation of the cross-entropy loss function during the learning of methods 2 and 3. For the same reason the classification metric chosen for the optimisation processes was the F1 score (3).

5.2 Hyper-parameters Tunning

Model 2 hyper-parameters (i.e. the learning rate, the polynomial order, the dropout probability, the weight decay and standardization) were tuned over 1000 epochs on the validation set. The best performance was obtained with standardization and with a learning rate of 0.3, a polynomial order of 3, a dropout probability of 0.2 and a weight decay of 5e-5.
Model 3 hyper-parameters (i.e. learning rate, hidden size of the first layer, hidden size of the last layer, weight decay and with standardization) were also tuned using the same procedure. The best performance was obtained with standardization and with a learning rate of 0.001, a weight decay of 0 and hidden sizes of 16 and 512 for the first and the last layers respectively.

5.3 Results

All models were compared based on their performance on the test set. More precisely, their accuracy, recall, precision and F1 score (for class 1) were used as metrics and can be seen in figure 2.
From the results, it is clear that all of the three models (Logistic Regression, Laplacian Polynomial and Graph Neural Network) are doing better than the Random classifier. Due to class imbalance, accuracy may not be an appropriate metric to pay attention to since a classifier always predicting class 0 would already reach almost 80% of accuracy. Knowing this, one can observe that model 2 is the best one exhibiting the highest recall, precision and F1 score for class 1. More precisely, model 2 is capable of retrieving 82% of nominated movies (recall) with a precision of 45%. This is evident by looking at the confusion matrix for this model where it is also noticeable that 72% of class 0 samples (true negative rate) are correctly classified meaning that this model performs well in both the classification of class 1 and class 0. The Graph Neural Network has less good results in terms of predictions of class 1 samples (i.e. nominated movies) and in fact the recall is lower (72%) and less precise (37%).
Comparing model 2 to the results of the simple logistic regression it is evident that in the case of the LR the recall is much lower (41%) even if the precision is higher (65%) meaning that the model finds less nominated films but more precisely (an hypothesis could be that this model only detects "easy-to-discriminate" nominated movies). By looking at the confusion matrix in this case one can see that this model predicts often the label 0 (i.e. movie not nominated) having a true negative rate of almost 94% and this explains also the high value of accuracy (82%) almost corresponding to the class 0 proportion.
Therefore, we can see how using a Network really does help detecting nominated films, and in fact the true positive rate drastically increases for models 2 and 3 with respect to model 1.

6 Conclusion

The initial study question, namely being able to determine if a given movie will get nominated for either an Oscar or a Golden Globes has been partially answered. In the exploration part of the analysis it was possible to visualize that through the network structure of movies we could gather many relevant information. The concentration of nominated movies is higher in certain clusters (highly correlated to a specific genre) this already showed that some

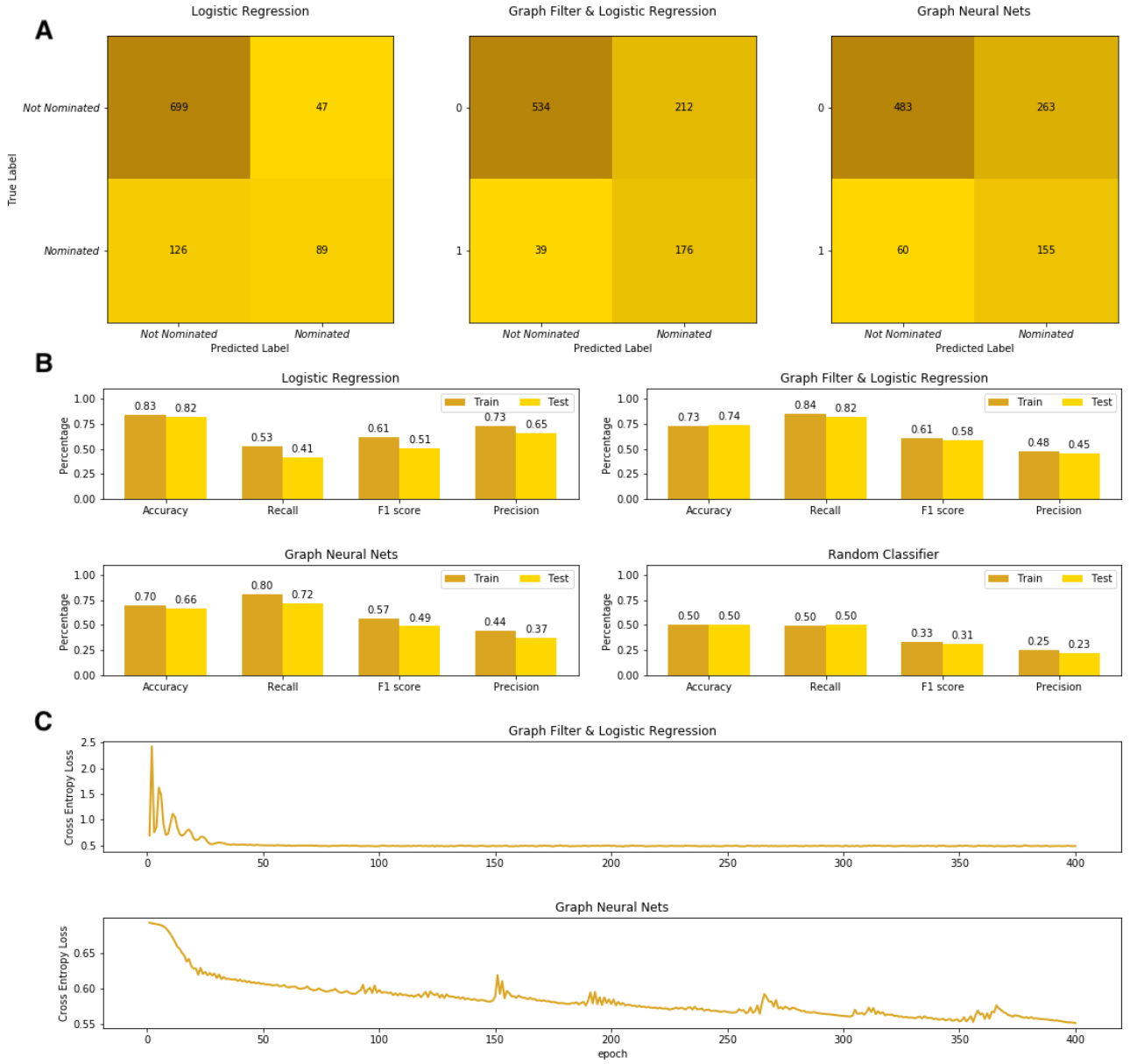

Figure 2: Performance of Classifiers. Performances of various classifiers tested **A** Confusion Matrix of Logistic Regression, Laplacian Graph Filter & Logistic Regression, Graph Neural Network on test set **B** Accuracy, Recall, F1-score, Precision of train and test set for Logistic Regression, Laplacian Graph Filter & Logistic Regression, Graph Neural Network, Random Classifier **C** Cross Entropy Loss evolution across epochs for Graph Filter & Logistic Regression, Graph Neural Network during learning.

specific movie characteristics permitted to discriminate between the two classes. Then, during the exploitation phase a real classifier was implemented. In this second phase, even if the classifier performance was showing results clearly above random, these are still far from perfect since no clear separation between classes has been reached. This can be because of many factors. First of all, information used to construct the graph was very sparse and hard to handle. As further improvement more movies features could have been scraped from the web in order to complete the already existing data. A track to follow would be to add features related to the advertising of a movie or its impact on social networks (using tweet sentiment analysis for example) in order to gather insights on its expectancy of affluence. Else adding more artistic features such as the main colors used on the movie set, the color contrast or the illuminance of the scenes could also help enrich the dataset. Other practical features as for example the film shooting duration or the number of languages in which it has been translated can be indicator of success.

Another idea would have been to give different weight and importance to specific subcategories of features used for the graph construction. We could have given more importance to the main actors or to the director.

Then, going back again to the learning part, the selected network signals did not show an incredibly high correlation with the Nominations meaning that the predictor performance was somehow limited by this factor. Finally, another aspect playing a role could have been the fact that the nominated movies were a minority of the total dataset, maybe a way to rebalance a bit the classes should have been explored by getting more examples of class 1 samples (i.e. nominated movies).

References

[1] A-L. Barabasi, Network Science, the scale-free property, 2014.

[2] https://tkipf.github.io/graph-convolutional-networks, visited on 8th January 2020.

[3] S.Kotsiantis et al., Handling imbalanced datasets: A review, 2006.