Predicting movie success and its determinants

Adin Gojak, Héloïse Hulstaert

Table of content

Introduction

During this semester, we had the occasion to discover or learn Python more in depth. We could already exercise ourselves thanks to the different assignments that we had to submit all along. To integrate the different concepts seen in class, we were asked to complete a project composed of data scraping, data visualization and finally machine learning. In this report, we will explain first our motivations regarding the choice of subject. From this general topic, we will develop a question that we will try to answer thanks to the different analyses we will conduct. Then we will detail the different methods used and finally we will conclude by linking our initial question/ topic research and the results we found.

Motivations

First of all, we wanted to find a topic linked to the interests of the two of us. One of us was more interested in cultural-related topics (museums, films, books etc.). The other member of the group would rather work on politics but was mainly interested in the predicting aspect and vetoed museum related topics. We first considered working on French politics as we were both following it closely and would thus be interested to know if we could either predict the future presidential elections results or the political belonging of a party based on text (NLP). However, we feared for the first option to lack information. Indeed after having searched on the French national database we did not find lots of information at the national level (most information was at the communal one) and we did not really see where to go and what to do. Concerning the text analysis topic, even if we were really motivated, we feared it would be too difficult as Adin had friends doing it as a thesis subject. Moreover, we saw we would have to see it by ourselves which highly discouraged us.

We thus kept looking for a subject. As one of the members of our group had done during the COVID a presentation on the cultural sector, more particularly the film industry, we both knew that since the crisis (and even before), it has been difficult for the cinemas, the film producers etc. We thus thought that if we arrived to predict which type of films were appreciated by the watchers, we could help them cope with nowadays difficulties. Both of us were onboard as it was linked to the cultural sector and offered a real possibility to develop machine learning methods.

Managerial question

As already announced in the motivations, we are convinced that predicting if a thing will be successful in the eyes of the audience could be really useful for movie producing firms and cinemas. As such we will try to answer through our work to 2 main questions:

- Can the success, in the viewers' eyes, of a film be predicted ?

- What are the factors influencing the success of a film?

We chose to orient our question from the viewers' point of view and not the one of the press. Indeed, even if reviews made by experts in journals such as Telerama etc. can have an impact on the films their readers will go and see, at the end of the day, it is individuals going to see the film and not the experts noting it that will ensure that the film is profitable. As we all know, the bottom line in the film industry where one film can cost a lot is highly important and can be tight.

Data Scraping (notebook **Data webscraping)**

In order to answer our question, we first need to create a database of films. We decided to source the data from the website of Allociné which is one of the main French film databases and the one we use whenever we want to get rating and information on a film. Moreover, Allociné is quite comprehensive as it provides a lot of information on the films, on the actors, on the directors etc.

Once we decided to use this film, we researched it in detail to know which data should be webscrapped. We chose to keep the following features:

- name of the film (to be able to identify it)
- release date (as we might want to compare films in time and to see if the year of creation might have an impact on success)
- length of the film (one could ask oneself whether really long film will have as positive ratings)
- genres. Here we decided to keep the 3 first ones as a film can have multiple genres
- directors (2 of them)
- actors (3 of them)
- ratings press
- rating viewers
- realisator (first one)
- nationalities of the film (2 first ones)
- nomination of the film
- budget (if available)
- box office in France (if available)
- languages
- nominations of the 2 first actors
- nomination of the first director

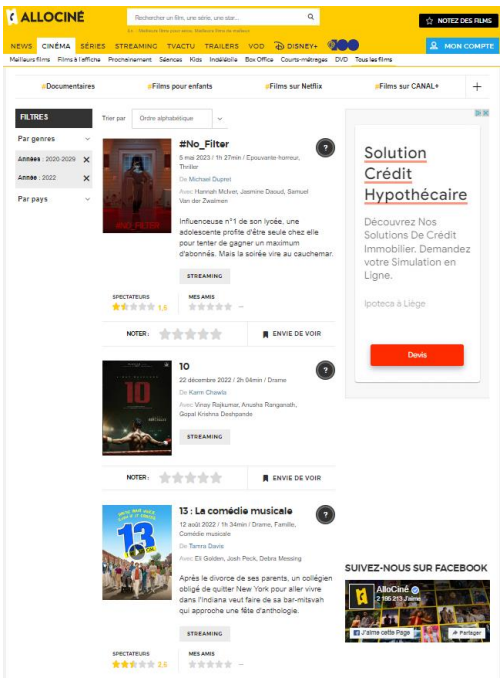
We hesitated in taking information such as short or long movies but did not take it as it was correlated with film length.

It was now time to begin web scraping. In class, we studied two methods: Selenium and Beautiful soup. We tried both with some features to see which one to use. We could easily

see that Beautiful soup was way more efficient for a high quantity of data as it was faster. Indeed, as Selenium is replicating a user going from the different pages, it automatically necessitates way more time. It would thus have been more logical to use Beautiful Soup. However, Allociné apparently had some strong “robot detectors". As such, by using Beautiful Soup, some lines of codes that should have worked were not working. By looking at the developer mode, we could see that lots of warnings were displayed. By reading them, we understood that Allociné detected that we were webscraping and was “blocking” the information. We had a look on the internet to see what to do to counter this. One solution was to add time between the requests to seem more “natural”. However, this did not work. Other solutions implied using IP rotation, webscraping out of the Google cache etc. We were not feeling at ease implementing any of these solutions. We thus decided to revert to Selenium even if it would take way longer to later retrieve the data.

We began by initializing all the packages, methods we will need. Then, we coded the Chromedriver. We decided to first go to the main page of Allociné to be able at this point to accept the cookies. Indeed, if we began directly by web scraping the pages we are interested in and put the cookies acceptance in the loop, we would have later on a problem as cookies enabling only happen once when loading the first page of the website.

As we would always webscrape the same data from different pages, we created a function. This function will then be used in a loop with the different URL of the pages. We decided to webscrape the pages classed in alphabetical order for **specific years**. We restrained ourselves to the 3 last years. We wanted to have enough data to construct a robust model later on in our machine learning step and first thought we would scrape all the films. However, retrieving all the data (15 films per page and 2589 pages) would be way too long and huge (plus, Allociné even with Selenium is sometimes spotting us and preventing us from getting too much data). We thus decided to retrieve all the films for the 3 past years which amounted to around 4000 films.



Each time we use our function, we first retrieve the information on the page with 15 different films. At this point, we go through each film to web scrape its information.

We can then get the name of the film, its release date, its length, its genres, the directors, the actors and finally rating by the press and viewers. We also get the links to the specific page for the film, for the two main actors and the director. We decided not to go directly to these pages but to first get the information for this page for the 15 films as it would otherwise lose the information already collected. Once the loop has been through the 15 films, we create another loop to go through the links of the films. On the new pages, we get more specific information on the films such as realisator, nationality, awards, budget, boxoffice, budget and languages. These elements demanded lots of caution when webscraping them as they did not always exist. More specifically, all the

information except realisator are stored in div. However if the information does not exist (such as awards), it is not written - (or 0) in front of awards but awards is simply removed from the div. As we used relative XPATH to get the information, this caused the whole code not to work anymore. We thus had to use lots of if and elif to take the necessary precautions.

This led me to a remark I would like to make on the whole webscraping experience on Allociné. When I began web scraping it, I was expecting something looking like the assignment we did but with more information and a more complex structure. However, that is not what happened. Indeed, I could really see that Allociné is not adapted to webscraping. Lots of different div have the same class name. As such, it was sometimes impossible to use CLASS_NAME and I had to use XPATH. Next time, I will have to webscrape. I will thus spend more time analyzing the website structure in terms of class name etc. to avoid webscraping such messy websites. Some research on the internet confirmed that other people had the same problem and switched to other film databases websites.

Once the loop with the link for more information on films is ended, we will implement a new loop going through the actors link. Here we had to take into account that some films did not have actors at all and thus we had to input NA when taking the link from the main film pages. We also had to account for actors having no awards. Indeed, if actors have no award, they have no div called award. As we used XPATH, we had to be cautious and check using if and elif etc.

Finally, we created a last loop going through the director link. As for the actors, we had to be cautious in case it did not exist at all. We also encountered the problem of the director having no award at all.

At this point, we had all the information from the first main page and went to the second selected one (the 101 thus) to again apply our function until the end of the loop.

We stocked the different pieces of information in a panda dataframe and then in a csv file as we wanted to stock the information and thus not have to go back each time through the whole process of collecting the data with Selenium. In the following steps, we will thus use the csv file we created.

NB: Even with Selenium, we can sometimes have bugs coming from the fact that Allociné detects that we are not real humans. It is sufficient to run the code again for it to work.

Data Cleaning (notebook **Data cleaning)**

Even if we got all the data from Allociné, some of them had to be transformed. For example, in terms of release date, we thought it might be more interesting to transform it in day - month - year. Secondly, we transformed the length to have it as an int in minutes. Then,we had a look at the null values (i.e. NaN) per feature. This made it possible for us to understand which feature would be problematic. Indeed, we strongly thought that having more than 2500 missing values for a dataframe of approximately 3700 observations would not lead to interesting conclusions. We thus removed the second realisator, the third genre, the boxoffice and the second nationality. We also removed all the observations having no value for note_spectators

(rating viewers) as it is our target variable and we would not be able to make models (decision tree, regression) if we have no target value. We also removed the budget variable as it was very often set to 0 or NaN. The note_spectators and note_presse were converted to float and the "," as a separator for decimal was replaced by the "." . We also created a new variable for the ratings. We created it thanks to a function.

Value note_spectators	New variable : note_spectator_word
<=1	catastrophic
< 1 and <=2	bad
< 2 and <= 3	soso
<3 and <= 4	good
< 4 and <= 5	excellent

Finally, we had to make some transformations for the number of nominations. Indeed, on the Allociné website, there was some information about awards and nominations (for example: "3 prix et 2 nominations"), when we web scrapped, we took the information as 3 2. We thus had to create a new variable "nomi" which gave the sum of awards and nominations.

Data Visualization (notebook **Data vizualization)**

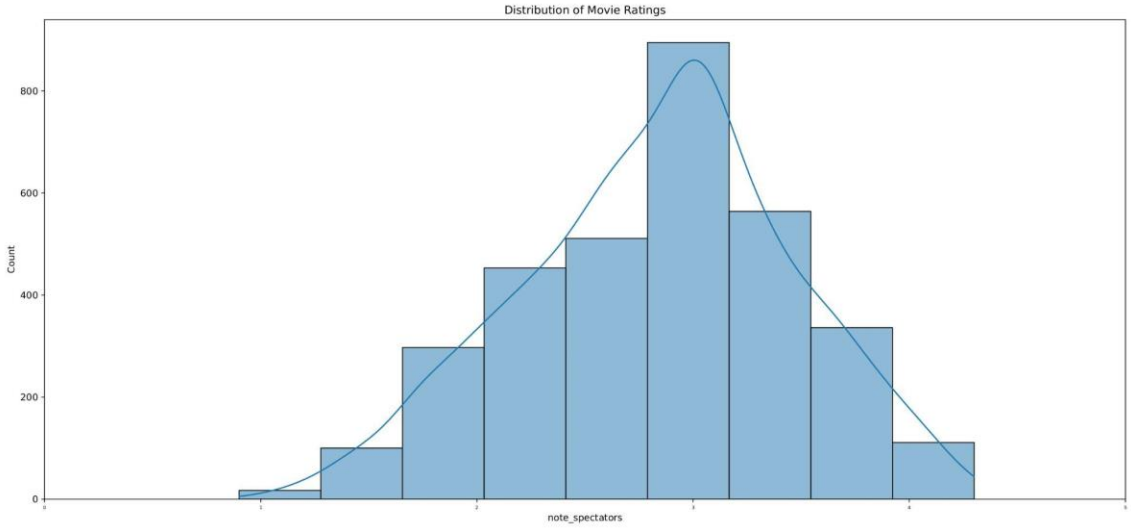
Thanks to the data visualization, we will get a first grasp of the data. This will allow us to already make some assumptions on whether one can predict the success of a film and whether some factors are highly explanatory.

We will first construct different graphs linked to the nationality. As one could expect, the majority of the films originated from the United States of America **fig 1**. The second producer country is France. Together, they account for more than 50% of the films produced in the last 3 years as can be seen in **fig 2**. As can be seen, we also computed some graphs with only the 10 most producing countries as only the first 5 give insightful information and that our graphs were hard to read with all the information.

We then computed a map to show on the world map where most films were produced. This can be seen in **fig 3**. As expected, only the U.S.A and France really stand out from the other countries.

Departing from the nationalities, we also decided to compute some graphs linked to the length, the genres and the actors. In this report, we will mention length and genre. **Fig 4** shows an histogram with the distribution of length across the dataset. One can see that most films have a length around 1 hour 30 min which corresponds to what we observe when watching films on the television. In terms of genres, **fig 5** makes it possible to understand that drama, comedy and action are the 3 main genres. If we have a look at second genres, we can conclude that drama is still the most represented followed by romance and thriller.

After this first analysis of the dataset, we will analyze more precisely the distribution of ratings.



As can be seen, few ratings fall below ⅕ and few (even if still more) after ⅘. The bulk of the distribution is between 2.5 and 3.5. This seems coherent with what one might expect as lots of films are good or at least not bad but few are catastrophic and few are excellent. This is backed up by an analysis on note_spectator_word.

soso	49.847747
good	35.505481
bad	12.515225
excellent	2.040195
catastrophic	0.091352

Now that we have a good idea of what our dataset is (and how the target variable is distributed), we will design some correlation and heatmaps to visualize the correlations. This will help us to make some assumptions about which features are explanatory and which are not. Indeed, this will allow us to see which factors are more correlated to the target variable (which is spectator ratings). However, correlation can only be computed between numerical values. We thus first did an analysis with only the numerical features. After that, we used OrdinalEncoder to try to see whether some categorical variables had some correlation with the target variable. The first heatmap **fig 6** (only with numeric variables) allows us to understand that the rating of the viewers is mainly correlated with the rating of the press, the length of the film and the nomination of the film. The second heatmap **fig 7** (with the categorical variables included) only adds some information about language. However, we must take these results with some caution as we had to go through a “trick” to get correlation and some other form of test such as ANOVA or Chi2 would certainly have been more suited.

Finally, the heatmaps give some information about correlation inside the feature variables. If we had 2 variables too highly correlated, this could cause problems later on. However, as we can see it is not really the case here. Obviously, the different spectators ratings are highly correlated but only one of this variable type will be kept at the end to be our target variable in machine learning anyway. We can also see that Scenar (screenwriter) is quite correlated with réalisateur (so director) as sometimes these two functions are filled by the same person.

Machine Learning

We applied two types of machine learning methods. We used some logistic regression and decision trees. The different logistic regression allows for prediction while the decision tree allows to know which factors are most important.

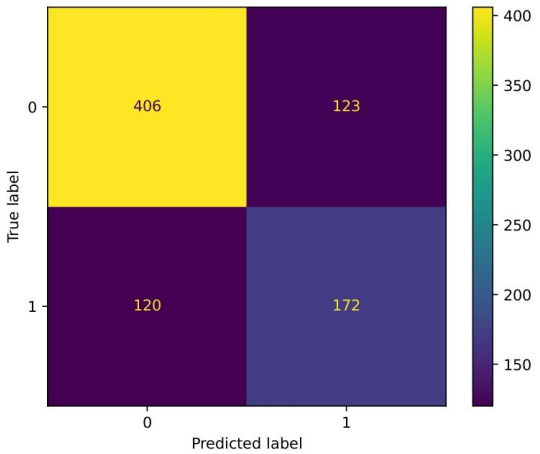
Decision Trees (notebook **Data ML Decision tree)**

Regarding Tree Analysis, we approached it in a similar way to other models. Our objective was twofold: to predict outcomes (success or failure) and identify important factors.

As we already cleaned the data, the work left wasn't so huge. We first selected which variables we would keep (the ones that were correlated with the target variable and some others that we thought had some impact). In terms of target variable, it is important to note that, for the purpose of simplification in this exercise, we considered a film to be successful if it had a rating above 3/5. We also observed that the choice of threshold value affected our confusion matrix.

We used a pipeline for our work to make it clearer. We used DecisionTreeClassifier as we are predicting classes (belonging to good films or not). We also limited the depth to have a readable tree. Moreover, we tried before without imposing the limit and it did not give better results. It was even worse as each leaf went to a pure node. We also imputed a preprocessor to deal with the data. Indeed, we had both numerical and categorical data. We standardized the numeric variables and used OneHotEncoder to create dummy variables for the categorical ones. For both types of variables, we also had NaN which we decided to transform thanks to SimpleImputer (the mean for the numeric and "missing" for the categoric). We thus obtained a model (De_tree_M). We tried several optimization techniques (GridSearchCV) but they did not give "interesting" results as they had a really low sensitivity (the ratio of true positives over the sum of true positives and false negatives). Which was not interesting for us as it implied that it predicted quite bad good films…

After completing this exercise, we focused on interpreting the results.



We got quite good results considering the small amounts of observations available. We can see that in terms of prediction our model predicts better the true negative (bad films) with a specificity of 0.767 than the good films (true positive) with a sensitivity of 0.589. However, we think that it is reasonable as our dataset is unbalanced and thus it is easier for the machine learning algorithm to predict the main class (bad films).

Finally, we get an AUC and accuracy of respectively 69.65% and 70.4%

In terms of explanatory power, so which factors are important to the success of a film, we found thanks to the decision tree **fig 8** that the first question to ask is related to note_presse. At the second level, questions regarding the awards and the length are asked. At the 3rd level, one question is asked about nationality. This seems logical with the correlation analysis we did before.

Logistic regression (notebook **Data ML regression)**

For the logistic regression part, we tested different models (LogisticRegression, optimization thanks to LogisticRegressionCV, Lasso and XGboost). For each of them, we created a pipeline as we did for the decision tree. We just had to adapt some parameters to the model under consideration.

For the first model, LogisticRegression, we tried two different datasets (one with only the 4 factors with correlation to the target class (M1) and another one with these factors and some others , the same that we used for the decision tree (M2)). We can easily see that the M1 is not suitable as it does not predict the true positive very well as can be seen in the confusion matrix **fig 9**. In terms of other models, we have quite close results as can be seen in this table:

Model	accuracy	AUC	specificity	sensitivity
M2 logi	0.7393	0.7658	0.85255	0.5342
M2 logit (LogisticRegressionCV)	0.6724	0.6886	0.7316	0.5651
lasso	0.7308	0.7565	0.8431	0.5274
Xgboost	0.7357	0.7799	0.845	0.5377

We can thus see that the results of the models are quite close. M2_logi so the one with LogisticRegression performs better in terms of accuracy and AUC. However, M2_logit (so LogisticRegressionCV) performs better in terms of sensitivity. We will still opt for M2 logi.
One must observe that we get better results for sensitivity with the decision tree method and better results for accuracy and AUC with logistic regression.

Conclusion

We began this work by asking ourselves two questions:

- Can the success, in the viewers' eyes, of a film be predicted ?
- What are the factors influencing the success of a film?

Thanks to the whole analysis we did, we can now answer them:

The success of a film can indeed be predicted as proven both by the Logistic regressions and the decision tree algorithm.

We could also find the factors explaining the success the most. These are note_presse, length, awards for the film, nationality and language.

However, some caution must be taken from these assertions. We did not always achieve really good results. We think that if our dataset had more variables and especially more observations, we could have potentially improved our models. Indeed, models should be trained on larger datasets to really recognize some patterns. In our case, one can ask oneself whether the algorithm could detect the importance of some actors with only 3 years of films. In terms of our willingness to extend the number of variables, the main challenge lies in the availability of such data. As mentioned earlier, websites are designed to block web scraping, and some data is not publicly accessible. Although we could have used an existing dataset (given that this question has been explored in several academic papers, including at our sister university, UCL), we decided to engage in the exercise and explore the type of data we could gather and the insights it could provide.

Appendices

fig 1:

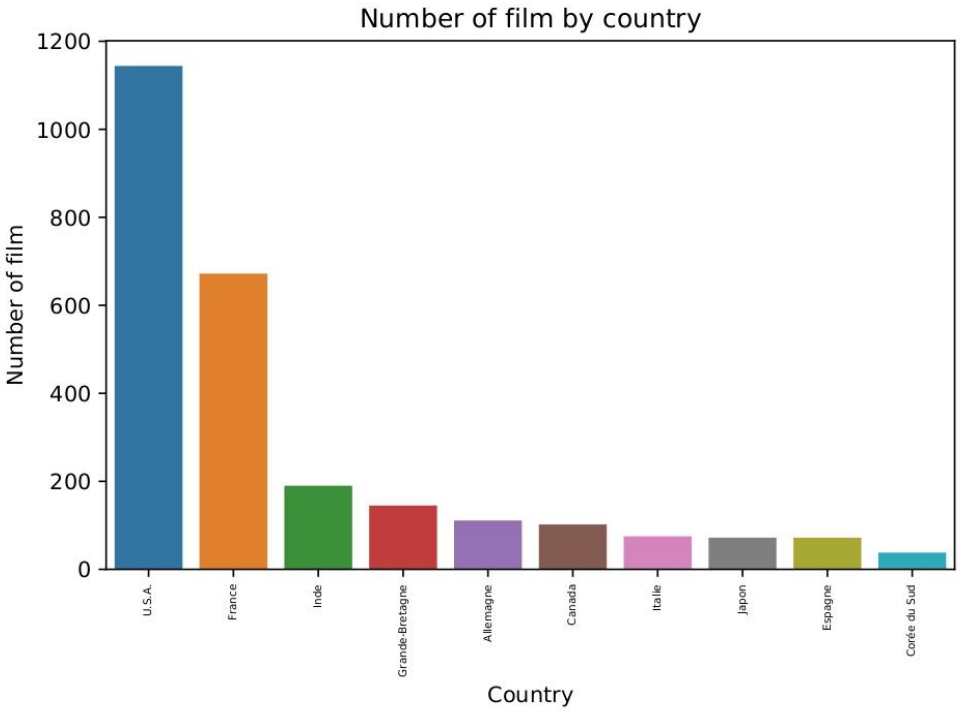


fig 2

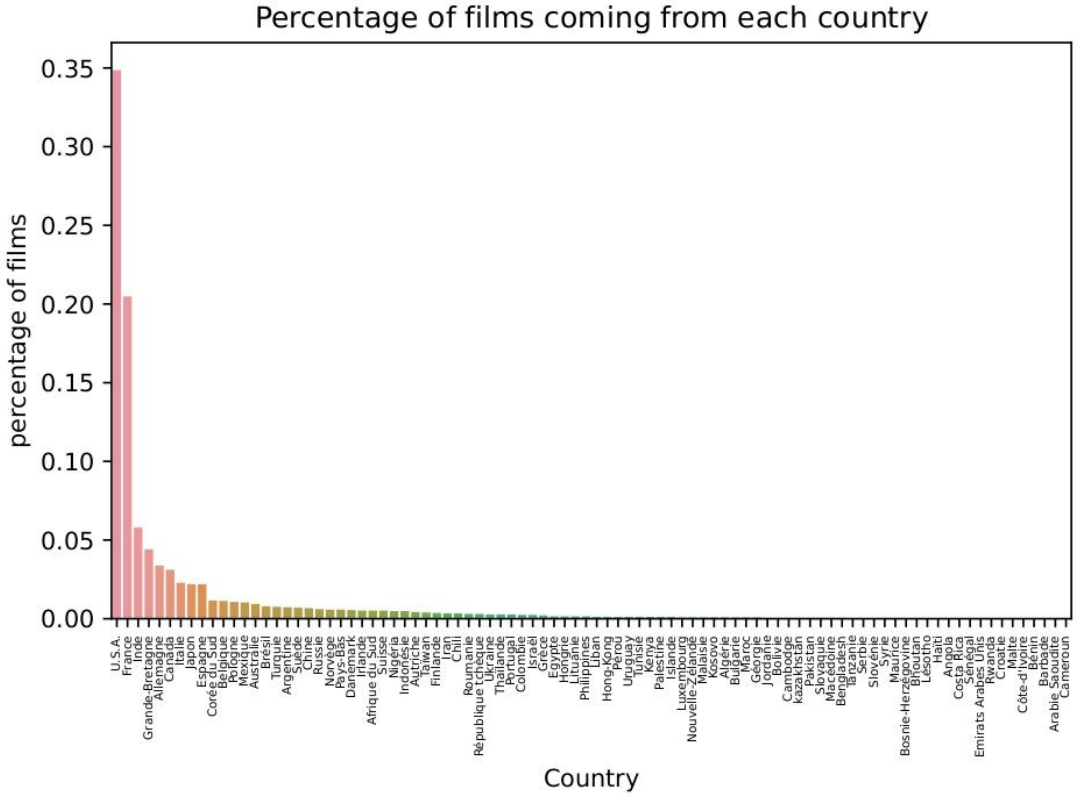


fig 3

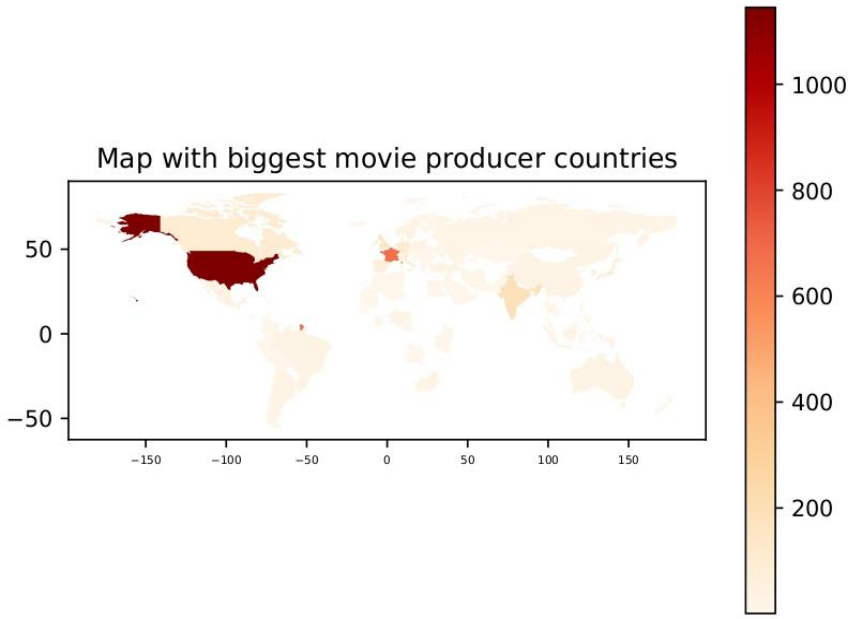

Map with biggest movie producer countries
50
0
−50
−150
−100
−50
0
50
100
150
1000
800
600
400
200

fig 4

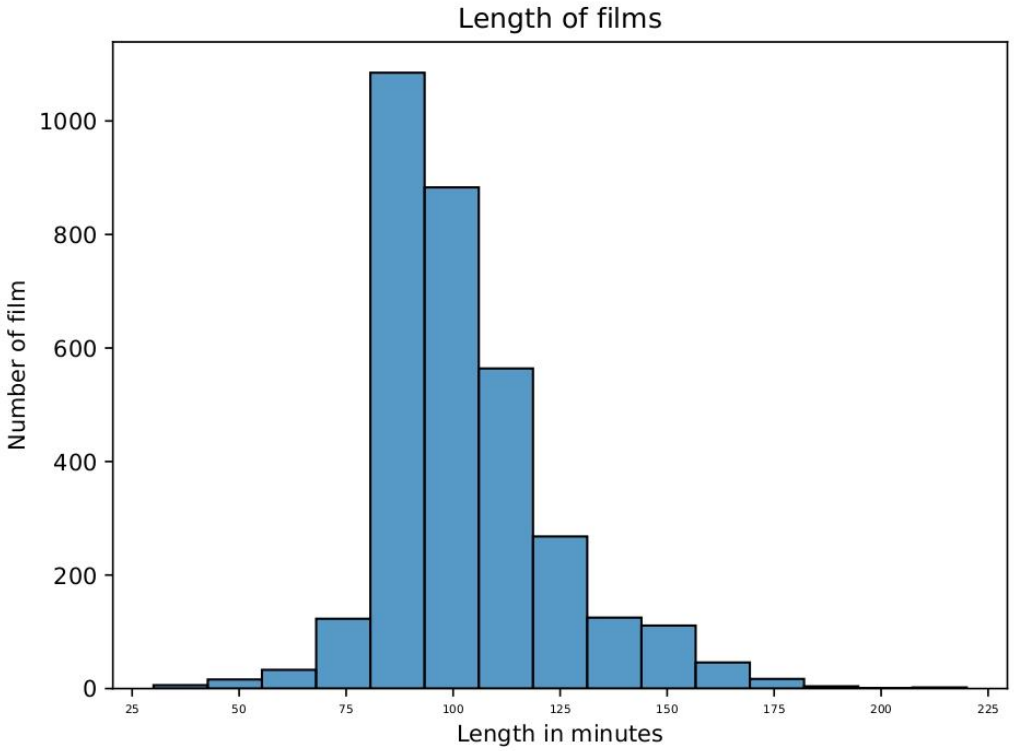

Length of films
Number of film
1000
800
600
400
200
0
25
50
75
100
125
150
175
200
225
Length in minutes

fig 5

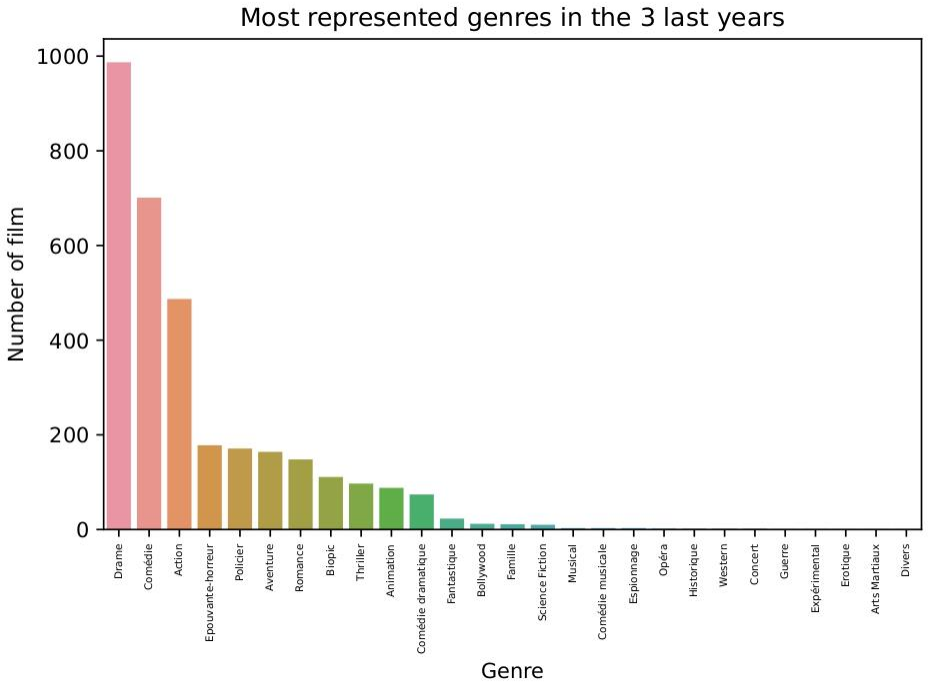

Most represented genres in the 3 last years
Number of film
1000
800
600
400
200
0
Drame
Comédie
Action
Epouvante-horreur
Policier
Aventure
Romance
Biopic
Thriller
Animation
Comédie dramatique
Fantastique
Bollywood
Famille
Science Fiction
Musical
Comédie musicale
Espionnage
Opéra
Historique
Western
Concert
Guerre
Expérimental
Erotique
Arts Martiaux
Divers
Genre

fig 6

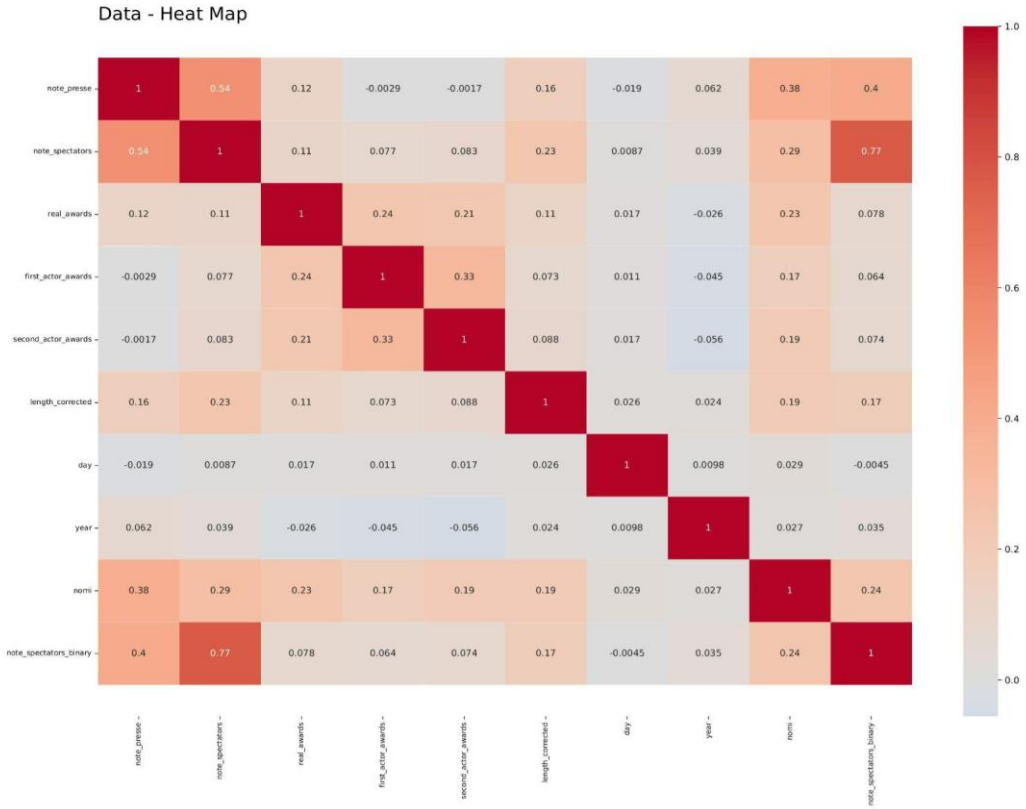

Data - Heat Map
note_presse
note_spectators
real_awards
first_actor_awards
second_actor_awards
length_corrected
day
year
nomi
note_spectators_binary
1.0
0.8
0.6
0.4
0.2
0.0

fig 7

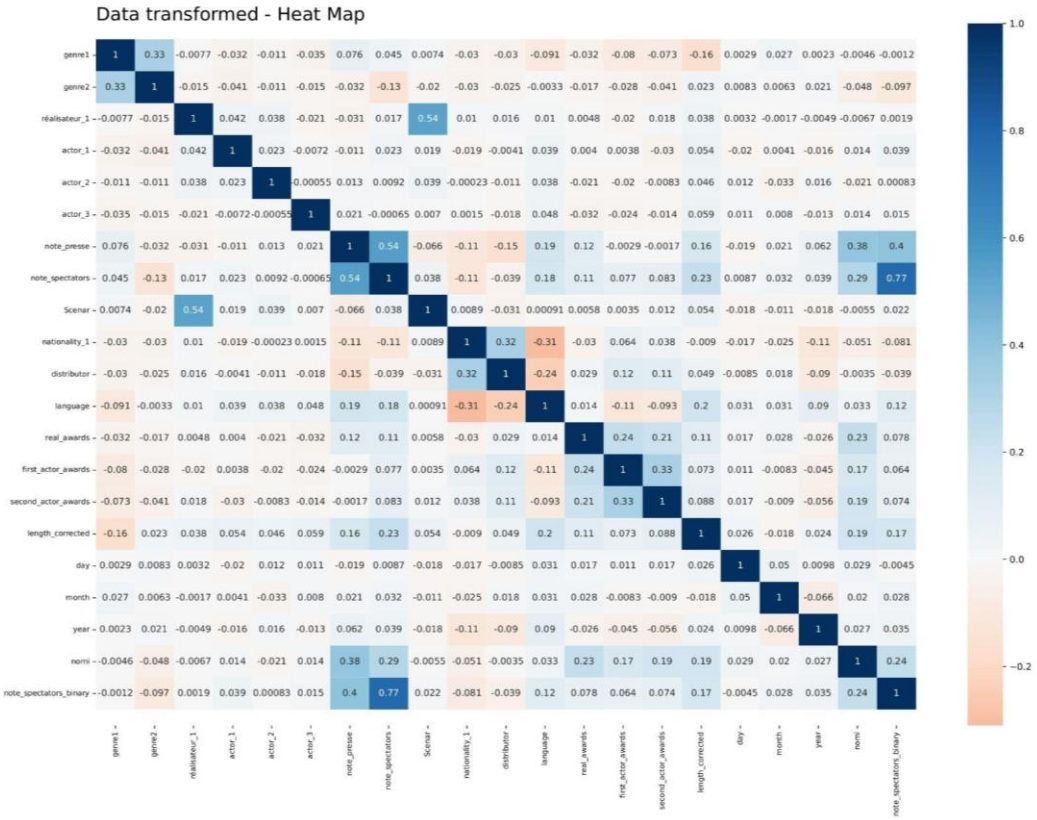

Data transformed - Heat Map
genre1
genre2
réalisateur_1
actor_1
actor_2
actor_3
note_presse
note_spectators
Scenar
nationality_1
distributor
language
real_awards
first_actor_awards
second_actor_awards
length_corrected
day
month
year
nomi
note_spectators_binary
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

fig 8

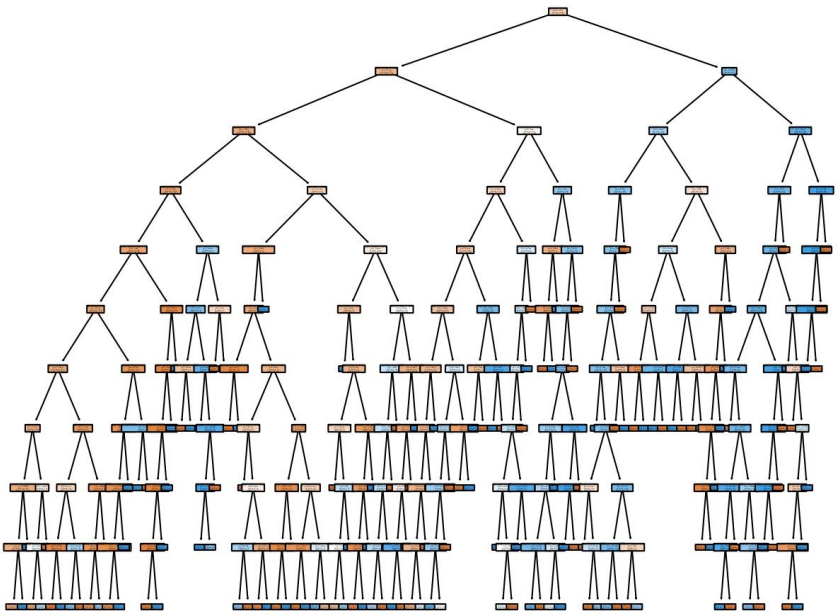
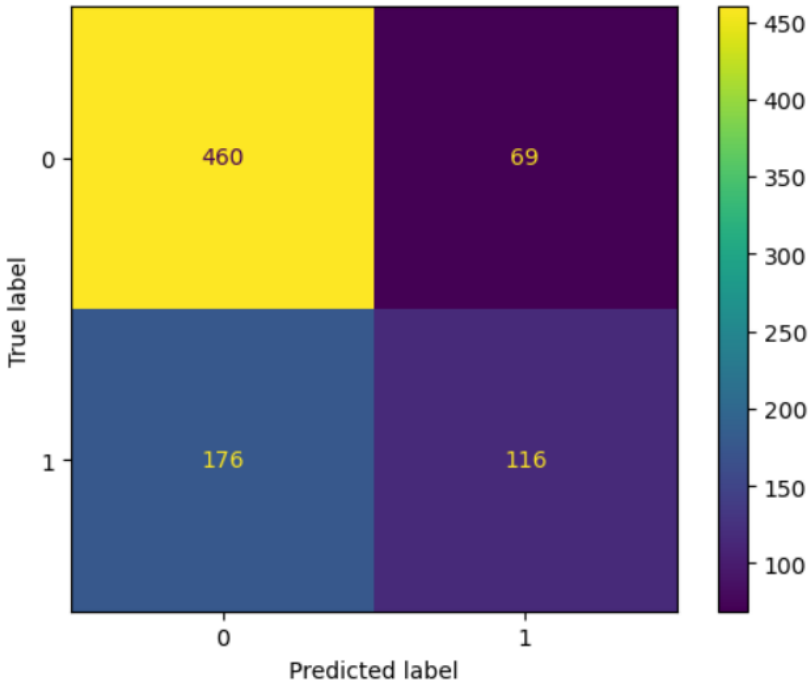


fig 9


460
69
176
116
0
1
True label
0
1
Predicted label
450
400
350
300
250
200
150
100