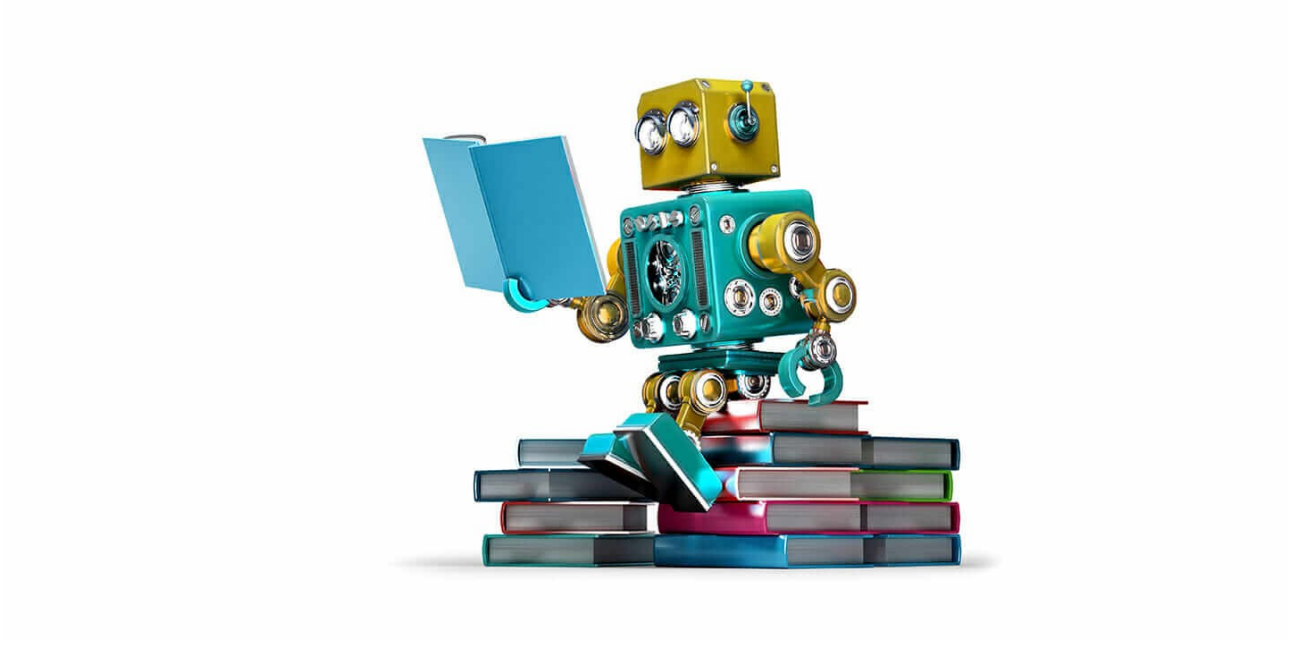
Project : Assigning genres to movies with Machine Learning implementation in Python

Etienne Raveau,
Master's Engineering student specializing in Mathematics and Computer Engineering at IMT Atlantique



I. Presentation

The project in hands consists in assigning genres to movies using machine learning methods in python environment.
The data-set is composed of approximately 9700 entries of movies with following features: A unique ID, representing the movie, a title with the date of the movie mentioned into brackets, and a list of genres, separated by '|'.

189043	Boundaries (2018)	Comedy|Drama
189111	Spiral (2018)	Documentary
189333	Mission: Impossible - Fallout (2018)	Action|Adventure|Thriller
189381	SuperFly (2018)	Action|Crime|Thriller
189547	Iron Soldier (2010)	Action|Sci-Fi
189713	BlacKkKlansman (2018)	Comedy|Crime|Drama
190183	The Darkest Minds (2018)	Sci-Fi|Thriller
190207	Tilt (2011)	Drama|Romance
190209	Jeff Ross Roasts the Border (2017)	Comedy
190213	John From (2015)	Drama
190215	Liquid Truth (2017)	Drama
190219	Bunny (1998)	Animation
190221	Hommage à Zgougou (et salut à Sabine Mamou) (2002)	Documentary
191005	Gintama (2017)	Action|Adventure|Comedy|Sci-Fi

Figure 1: Initial state of the DataSet

Therefore, the purpose is to be able to predict genres related to a movie, given its title and date.
To do so, machine learning methods have been implemented on Python.
This report will describe step by step the overall method used to implement the model. Afterwards, the methodology used to clean the data-set in order to match it with algorithm input requirements is depicted. Finally, the results will be discussed regarding either their performance and accuracy.

II. Implementation of the model

The model is divided into 5 main steps:

- Data Cleaning
- Creating train and test sets and encoding them to fit algorithm's required input
- Choosing and training classification algorithm
- Predicting output given the test set
- Plotting results and performance

Each of these five steps is a crucial part of the model, but obviously some of them are more important and time-consuming. As a consequence, only the data cleaning, the choice of the algorithm by comparing performances and the analyses of the results will be thoroughly described.

III. Data cleaning

First of all, data cleaning had to be processed in order to improve and facilitate the following learning and predicting phases.
The cleaning has been done by implementing the following steps:

- Reshaping the titles by separating the actual title and the date of the movie: this is done by replacing directly on Excel each parenthesis by a comma and then splitting the movie string on the commas it contains.

- Removing instances with mistakes in the entry (title badly written, date missing etc..) but manually fixing when possible.

- Removing movies with no genre mentioned.

- "Replicating" movies so that every entry only has one genre: if a movie had initially Action| Adventure as genres, it will now be represented by two rows in the table: one with the genre Action and the other one with the Adventure genre.
(in a second implementation, this phase has been replaced by creating a list of genres by splitting the given string on '|')

- Removing genre column from the data tables to be stored in a "target" table, meaning that the classification will consist in trying to find the genres the movie is more likely to be related to, based on its name and date.

- Removing ID column, unnecessary for the study.

Finally, around 40 entries have been removed and more than 30 mistakes have been corrected.
After this process, the data-set was all set for further processing.

```
>>> (executing file "text_Classification.py")
[['1950', 'Rashomon (Rashōmon) '], ['2008', 'Front of the Class '], ['1994', 'Africa: The Serengeti '], ['200
2', 'Babylon 5: The Legend of the Rangers: To Live and Die in Starlight '], ['1997', 'Shadow Conspiracy ']]
['Crime', 'Drama', 'Documentary', 'Sci-Fi', 'Thriller']
```

Figure 2: The five first entries of train set and train target after the cleaning

IV. Results and algorithm comparison

A performance of at best 30% for the single label model and 8% for the multi label model are obtained (based on the score given by the score() method implemented by the used classifiers). This relatively poor performance could be due to many different factors.

On one hand, the single label model duplicates entry in order to tackle with the presence of several genres for a single movie. Therefore, the algorithm could grant a movie with an actual genre, but still being considered to have badly evaluated. For example, let's take the movie Jumanji (1995). This movie is initially related to the genres: Adventure, Children and Fantasy. After cleaning, the table will have 3 entries: Jumanji, Adventure; Jumanji, Children; Jumanji, Fantasy. When it comes to evaluate, Jumanji & Adventure could be picked among others. Then, maybe the classifier will successfully relate Jumanji to the genre Children or Fantasy and the evaluation will be considered wrong while it was expected by the scoring function to be Adventure. This obviously heavily drives the performance downwards, as a large majority of the movies are related to more than one genre. Using another scoring method should drastically boost this performance indicator.

On the other hand, the multi label method is way more complex as this time it is supposed to grant a movie with several labels. Really poor performance are obtained if using the same algorithms as the single method. But by using multilabeling classification tools, performance can be raised up to 8%. Likewise, the low performance is mainly due to the scoring calculation. Most of the time, at least one genre is granted to the movie, and for some movies, 4 genres are successfully granted. But as some movies remain without genres and that for a majority of the movies, at least one is lacking, the performance plummets as only total match are considered.
Finally, given the number of entries, the calculation relies almost exclusively on the date of the movie (as the title gives at first glance few information regarding the genre), thus the model has a strong tendency to assign same genres at every movie from a same publishing date.

Algorithm	Time (s)	Score (%)
SVC(gamma='scale', C=1)	22	20
SVC(gamma='scale', C=10)	24	10
KNeighborsClassifier(5)	1	16
DecisionTreeClassifier(max_depth=5)	0.31	29
RandomForestClassifier(max_depth=10,	0.25	27
AdaBoostClassifier	5.7	28
MLPC(Alpha=1)	222	29
(Multi) MLkNN(k=5)	13.9	6
(Multi) BrkNNa(k=1)	2.14	2.8

Figure 3: Comparison of time and performance of different classification algorithms (multiclassifier are preceded by ‘multi’)

However, based on graphical representation, the hypothesis of inadequate scoring method is further discussed. Indeed, it seems that the higher the score, the fewer genres and movies taken into consideration (cf. Figures 4-11)

Furthermore, the comparison with expectations must be done. Figure 12 represents the result of SVC(C=10) compared to the expecting results. One could easily see that the match is pretty good, even if some genres have been unnecessarily added. As the comparison with expected results is most of the time unreadable on a same plot, only this best scenario has been stored. Otherwise, Figure 13 gives the expected results alone, if one wants to compare with the other prediction plots.

It should be noted that those figures have been created using algorithms with their relative best parameters, these having been determined before through comparative analysis.

Anyway, the results could be analyzed observing that some genres remain common through the years, while some are more related to punctual artistic or fashion movements and only appear at some precise periods.

Finally, comparing the better performing models, namely SVC(C=10) for single labeling and BrkNNa(k=1) for multi labeling, it seems that good results could be obtained using the two different implementations.

V. Conclusion

All in all, the implementation of machine learning could be a long process with many difficulties along the way. However, this project succeeded in assigning genres to a given set of movies, using and comparing different machine learning algorithm. The model was built by cleaning the data and extract relevant features, choosing and comparing algorithms based on their performance and finally printing and analyzing the results by comparing predictions with expectations.

From these steps, it has been withdrawn that a data cleaning is unavoidable in order to enhance performance and facilitate either the learning and predicting phases. Besides, as every machine learning is inherently specific, it is indispensable to define key performance indicators in order to measure the performance of different algorithms and ensure the efficiency of the model by choosing the best-suited one. However, even if such indicators have been defined, it is often difficult to measure them conveniently when dealing with such amounts of data.

Finally, some satisfactory results have been shown under the single label classification decision with SVC(C=10) as well as with the multi label classification with BrkNNa(k=1).

The method used in this study could be applied to every machine learning problem, even if the data cleaning step should be adapted from one to another.

Further improvements of the model could be in the definition of more precise indicators of performance, allowing to compare easier and more efficiently the prediction to the expectation.

Besides, more features should be added to the data table, as it is obviously really difficult to produce an accurate estimation of a movies' genres, based only on its date and name (which provides in itself very few information). Further studies may take into consideration to perform a text analysis on the movies titles, in order to withdraw genre-related information.

Appendices:

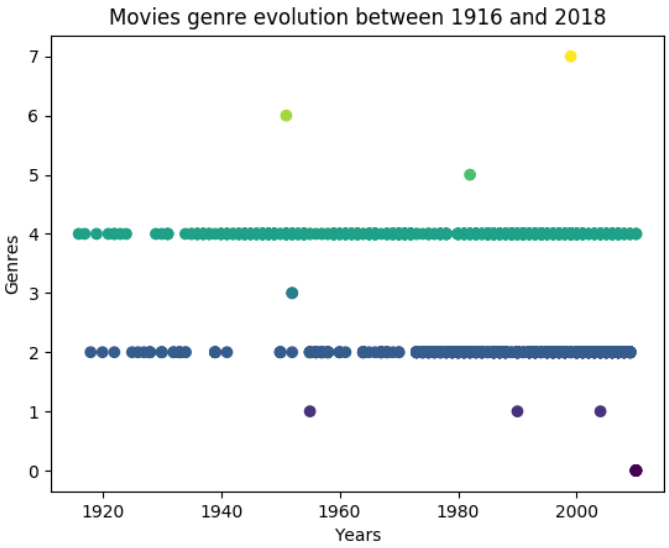


Figure 4: SVC (C=1)

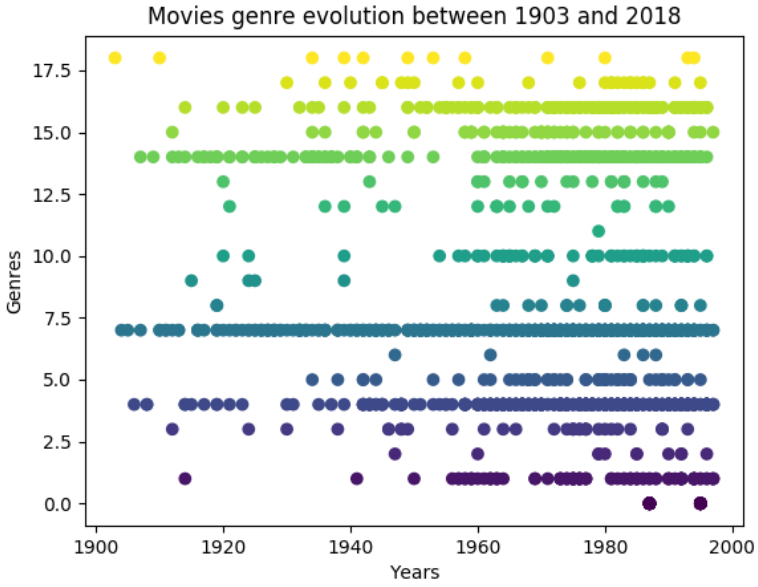


Figure 5: SVC (C=10)

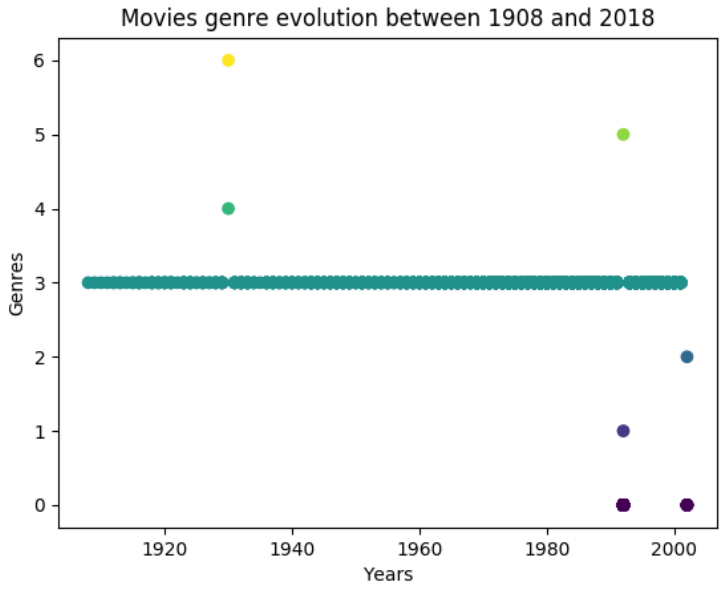


Figure 6: DecisionTree(max_depth=5)

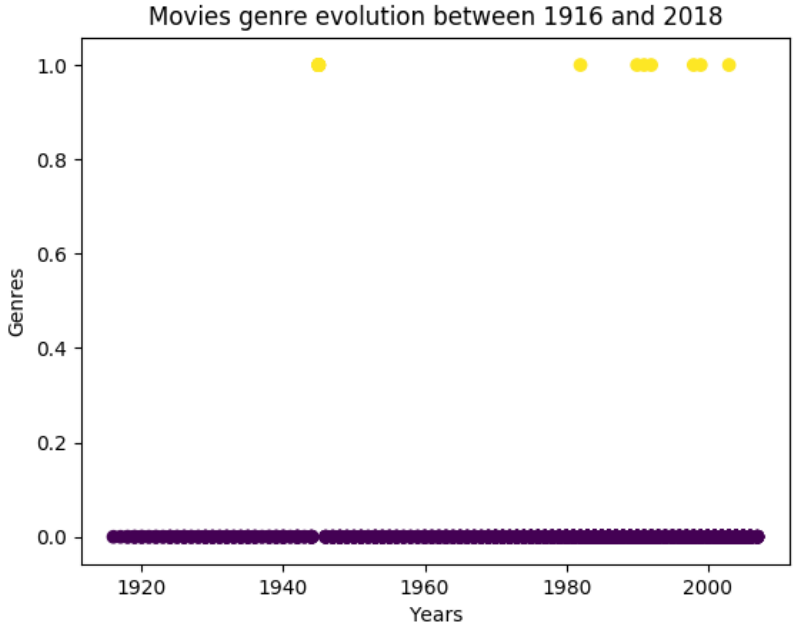


Figure 7: RandomForestClassifier(max_depth=10, n_estimators=20, max_features=2)

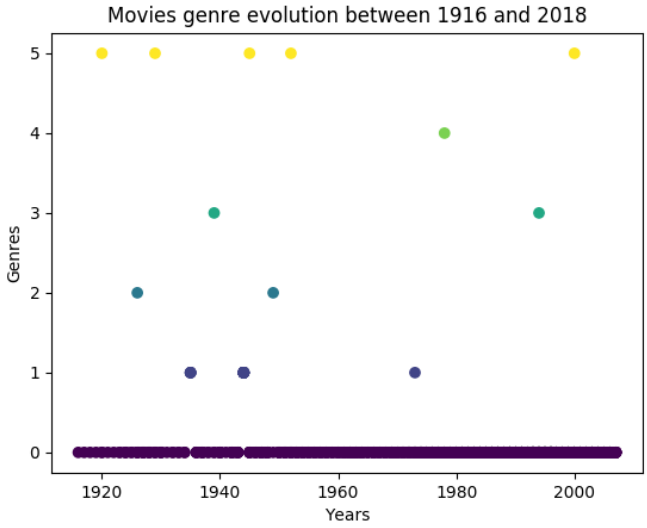


Figure 8: AdaBoostClassifier

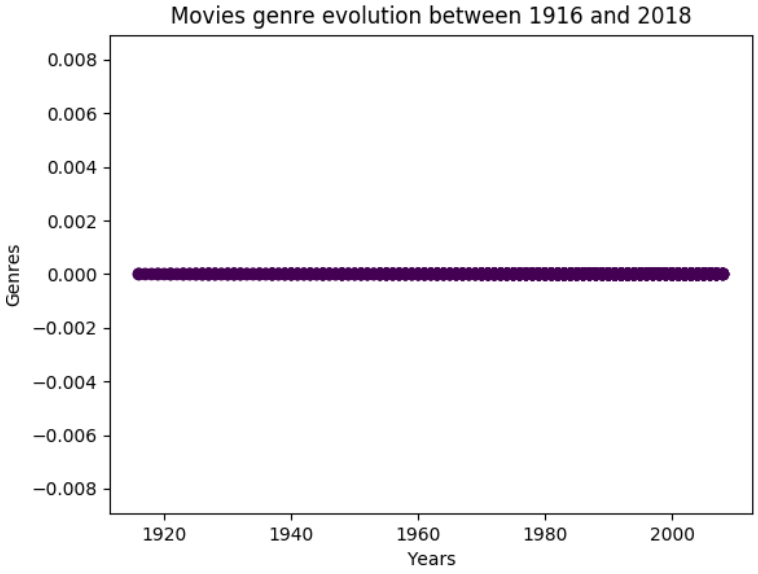


Figure 9: MLPC(alpha=1)

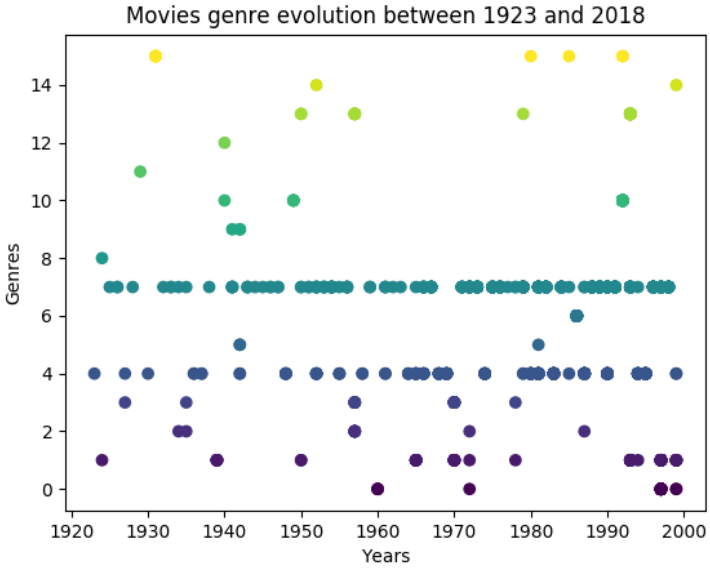


Figure 10: MLkNN(k=5)

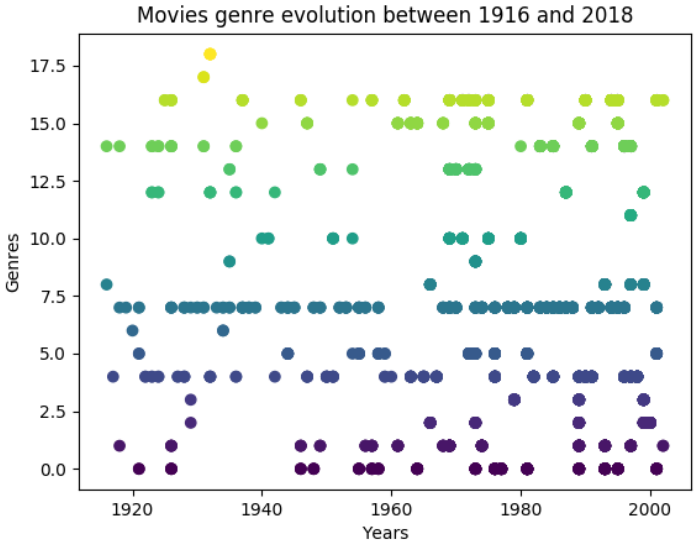


Figure 11: BrkNNa(k=1)

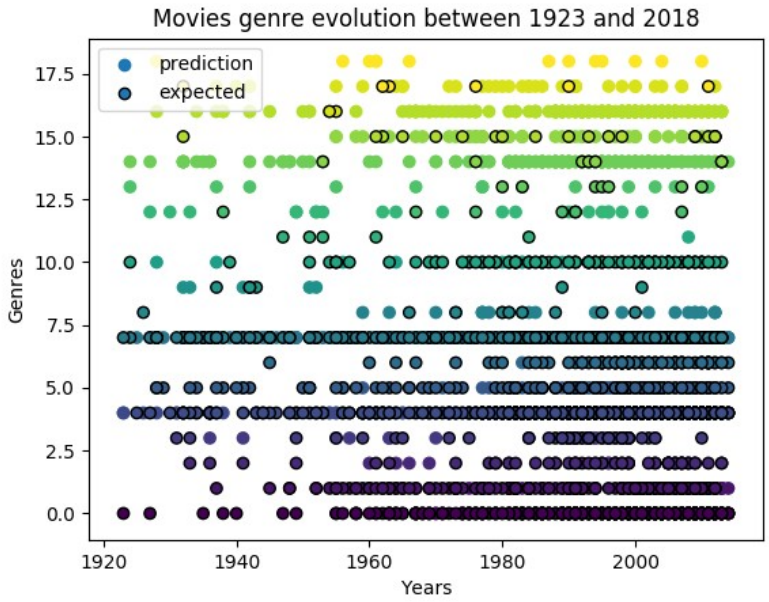


Figure 12: Comparison Expectations-Predictions with SVC(C=10)

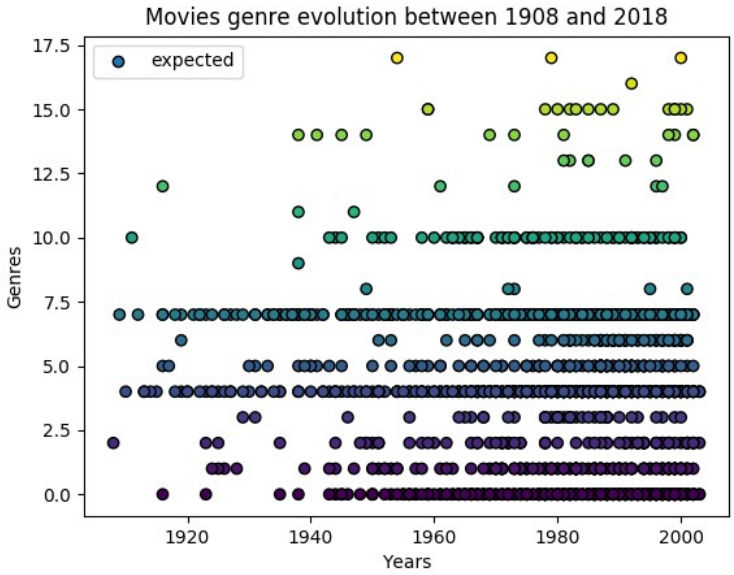


Figure 13: Expectations for genres