Alma Mater Studiorum - University of Bologna
LM Informatica

Data Analytics Project

Cotugno Giosué [TOADD] - Pruscini Davide [1007343]

July 7, 2022

Contents

Chapter 1

Introduction

In this report we will discuss about the whole data pipeline of a project that uses different techniques of machine learning to classify some tabular data. In particular, we choose the first project proposal that aims to predict the average mark of a film from its features. The first step of the data pipeline requires to download, save and load in memory the MovieLens [1], TMDB [2] and IMDb [3] datasets. Consequently, the datasets were elaborated with the objective to generate a unified dataset that can be used as an input for some machine learning algorithms. Afterwards, during the modelling phase we built an MLP model thanks to the PyTorch framework [4]. In addition, we used other techniques like SVM, tree methods and naive bayes methods that are available into the Scikit-Learn library [5]. In order to find a good configuration during the performance analysis, it was mandatory to define a large enough hyperparameters space for all the models that we defined. Moreover, in the last phase, the cross validation technique was used to obtain more robust statistics results that has been compared between the trained models.

Chapter 2

Methodology

In the next chapter there will be the explaination of the data pipeline that the project followed. In particular, each subsection will focus on a specific task, except for the data visualization that has been used only when needed.

2.1 Data Acquisition

The used datasets are downloaded at runtime directly from the sources. These datasets come directly from MovieLens' page and provides 6 different files. More informations about the nature of the data are available here.

Dataset	**Features**
ratings.csv	userId, movieId, rating, timestamp
tags.csv	userId, movieId, tag, timestamp
movies.csv	movieId, title, genres
links.csv	movieId, imdbId, tmdbId
genome-scores.csv	movieId, tagId, relevance
genome-tags.csv	tagId, tag

Most of these datasets provides information for approximately 60000 films. The links dataset provides two identifiers that allow to collect information from the IMDB and TMDB databases. Thanks to the links' features, it has been possible to collect some more information on the running times of the films that could provide more insight into them. Talking about the ground truth of the supervisioned models, the rating mean is missing. So the target feature will be computed during the pre-process phase thanks to the ratings dataset. Further information about the features usage and the cardinality of the datasets will be discussed in the Pre-Process section.

2.2 Data Pre-process

In this section, will be discussed the pre-process phase for each of the above presented datasets. In order to achieve a major clearity, the work on each dataset will be discussed in a specific subsection where the operation computed on them will be explained. At the beginning of each subsection the cardinality will be reported.

Movies - movies.csv

Cardinality: 58098×3

Inside this dataset, the title and generes features contains multiple information. In particular, the title has been splitted in two part, where on one hand there's only the title name, and on the other, there's the year of the film production. Since the title name is a string, that doesn't add more information to a classic machine learning model, this feature has been converted into its length meanwhile the production year just constitutes a new feature. The genres feature contains a pipe separated list of a fixed possible values. Since the list is just saying if a film has a specific genre or not, to each film, all the fixed values has been added as a feature, and if a genre appears into the genres feature, that column will result into 1 that indicate the presence of that genre, 0 otherwise. At the end of this initial phase, the first sample of movies dataset looks like:

movieId	**title_len**	**year**	**action**	**adventure**	**...**	**Western**	**(no genres listed)**
1	16	1995	0	1	...	0	0

In order to finish the data pre-process on this dataset, the data cleaning is required. First of all, there are some films where the year of the film production is missing. Analyzing the distribuition of these, as showed in Figure 2.1, it is possible to see that the distribuition is right skewed, so the missing values can be filled with the **median**, as suggested during the lectures.

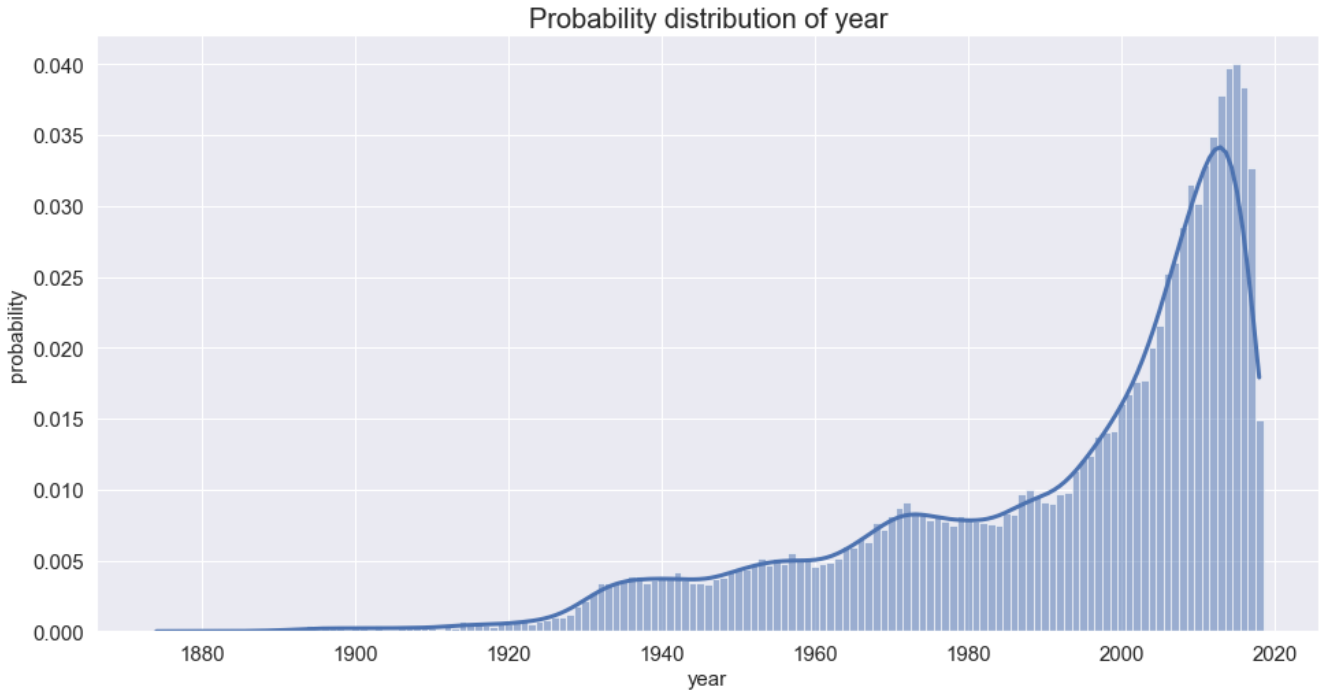

Figure 2.1: Right skewed probability distribuition of the year feature

Since the feature (no genres listed) and the films that has no genres provide no information about the film, they will be removed obtaining a final cardinality of 53832×22.

Tags - tags.csv

Cardinality: 1108997×4

In order to use the relevant data from this dataset the timestamp and userId features have been deleted because they don't explain anything more about the films. The information provided by each sample in the dataset are not very meaningful, for this reason it was decided to count the number of tags associated with each film, in order to understand how much interaction that film generated. When a film doesn't have any related tag, the tag_count can be setted to 0 because no users were interested on that film. After these operations the cardinality was reduced to 45981×2.

Ratings - ratings.csv

Cardinality: 27753443×2

The features in this dataset were necessary to find the target column. In order to compute it, the average of each films' ratings has been calculated. In addition, the number of the rating on each film has been counted. However, the task of this project is the classification, for this reason the rating_mean has been discretized in 10 bins, where each of them covers a rating range of 0.45 as showed in Figure 2.2.

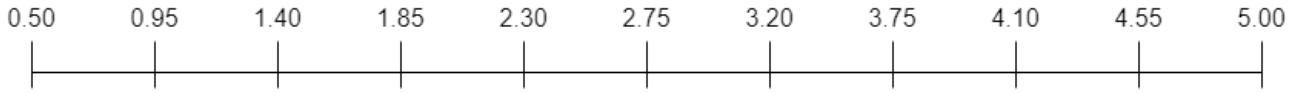

Figure 2.2: Discretization of the rating_mean feature

At the end, the final cardinality is 53889×4.

Genome - genome-scores.csv and genome-tags.csv

Cardinality scores: 14862528×3
Cardinality tags: 1128×2

Talking about these two datasets, the merge operation over the tagId was needed because it was interesting to associate the tagId with its string name. After this union, on each sample there is the correspondence between a movieId, the tag name and the relevance of that tag. The final step consist on relate every single film to its relevance, using the pivot function.

movieId	title_len	year	action	adventure	...	Western	(no genres listed)
1	16	1995	0	1	...	0	0

TMDB - tmdb.csv

IMDb - imdb.csv

2.3 Modeling

2.4 Performance Analysis

Bibliography

[1] GroupLens. MovieLens Latest Datasets. https://grouplens.org/datasets/movielens/latest/.

[2] TMDB Community. Api Overview. https://www.themoviedb.org/documentation/api/.

[3] IMDb.com. Imdb datasets. https://www.imdb.com/interfaces/.

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.