Università degli Studi di Milano Bicocca

IMDB REVIEWS ANALYSIS

TEXT MINING & SERACH PROJECT

MARCO DONZELLA: 829358
REBECCA PICARELLI: 834286

Tables of contents

Key points of the presentation

Introduction

Goal 1: Building a binary text classification model to detect the type of sentiment of the reviews (positive or negative)

Goal 2: Getting the main topics of the reviews

Method:
- Download of the IMDb reviews dataset
- Implementing some text representation techinques after data preprocessing
- Use of machine learning models for the classification task
- Use of topic modelling techniques for the second task

Programming language used: Python

IMDb (Internet Movie Database) is the world's most popular source for movie, owned by Amazon. It contains and manages lots of data and information about movies and actors and represent a free source of information, perfect to perform some text mining tasks

Link: https://www.imdb.com/

Data

The data is free and it's possible to download it here: http://ai.stanford.edu/~amaas/data/sentiment/

All the reviews are contained in files .txt and are perfectly balanced for the sentiment. These files have been uploaded and merged with pandas, in order to simplify the analysis.

Therefore, the datasets are:
- unsup : 50 000 reviews unsupervised, without the target (sentiment)
- train_pos : 12 500 positive reviews
- train_neg : 12 500 negative reviews
- test_pos : 12 500 positive reviews
- test_neg : 12 500 negative reviews

The structure of the datasets consist of two columns: the text of the review and the related sentiment (positive or negative) for the labelled data.

Preprocessing

The following steps of preprocessing have been used to clean and prepare the data:

- Case folding (lower case)
- White spaces, punctuations and emoji removal
- Tokenization
- Lemmatization
- Stop words removal

Preprocessing

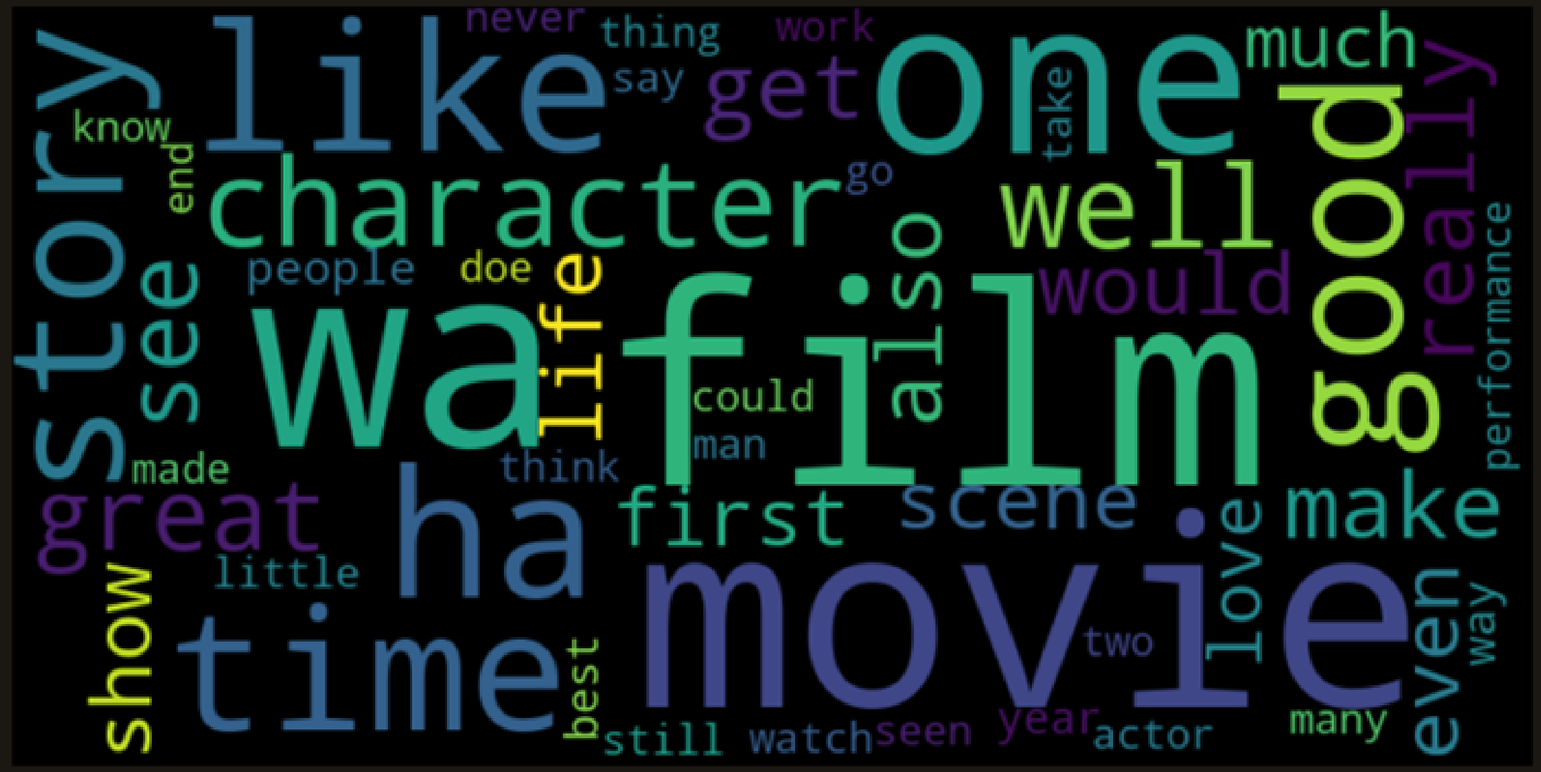
Here is an example of the data (train_pos) before and after the 5 preprocessing steps:

0	If you want mindless action, hot chicks and a ...
1	Director Kinka Usher stays true to his own cre...
2	The novel is easily superior and the best part...
3	Silly movie is really, really funny. Yes, it's...
4	Jesse and Celine (Ethan Hawke and Julie Delpy)...

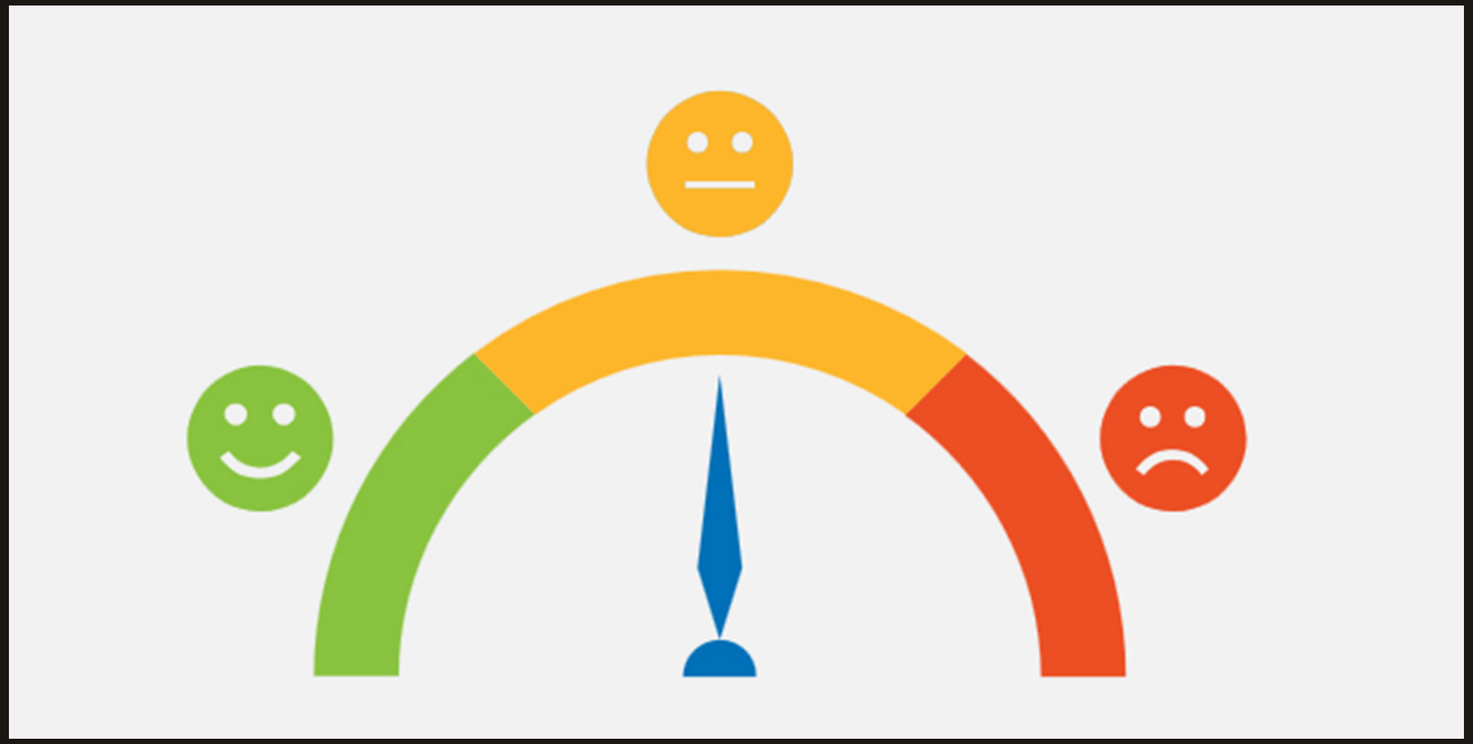
0	[want, mindless, action, hot, chick, post, apo...
1	[director, kinka, usher, stay, true, credo, pl...
2	[novel, easily, superior, best, part, film, di...
3	[silly, movie, really, really, funny, yes, got...
4	[jesse, celine, ethan, hawke, julie, delpy, tw...

WordCloud

To have a clearer view of the most important tokens, 5 WordClouds were created (train_pos, train_neg, test_pos, test_neg, unsup). They are consistent with each other and show the same words. Here is an example (train_pos):



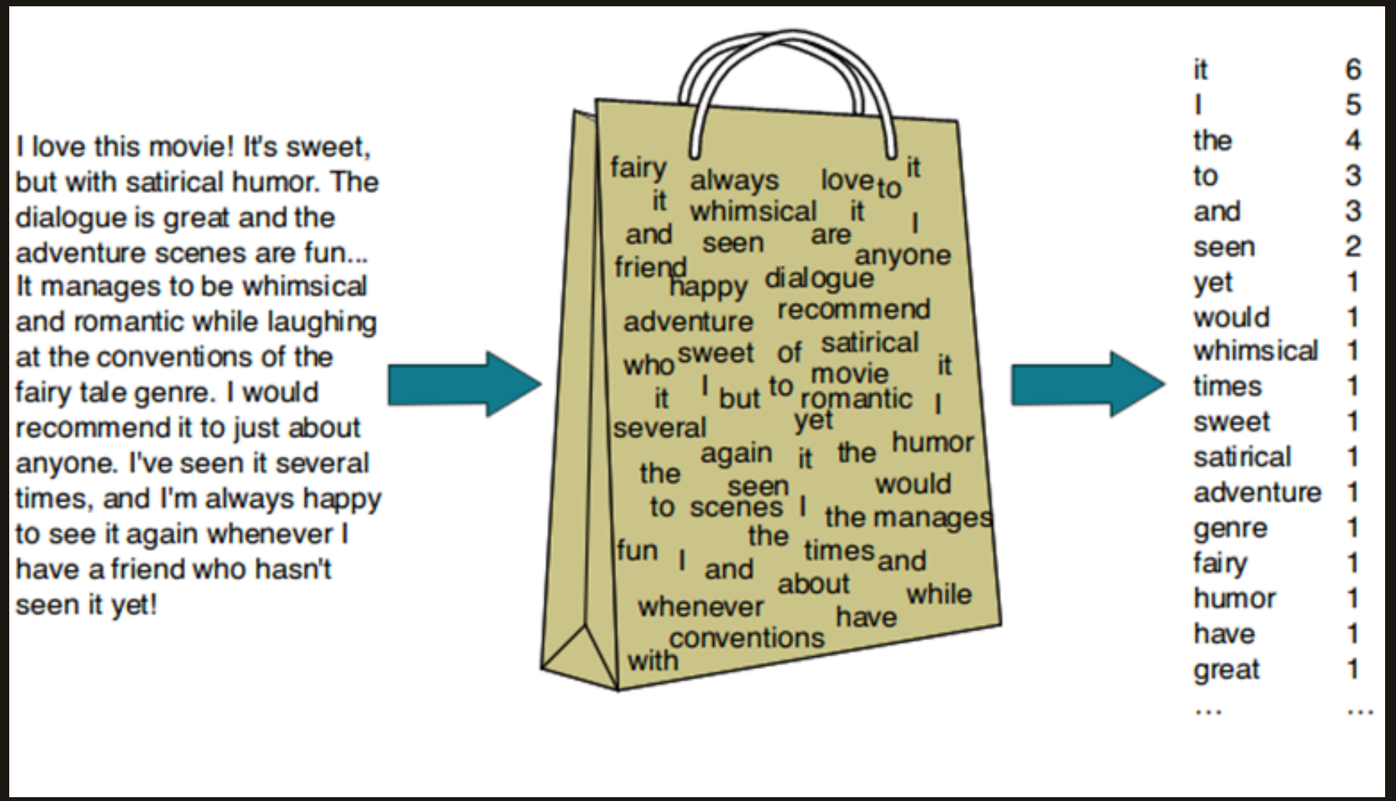
Text representation & Classification



Text representation

In order to manage the textual data preprocessed and to use machine learning algorithms for the classification task, it was necessary to convert the data into a numerical form.

5 types of textual representations were used:
1) Binary Bag of Words
2) Bag of Words
3) Bigram Bag of Words
4) TF-IDF
5) Bigram TF-IDF



Models

The following machine learning models were used:

- Multinomial Naive Bayes
- Linear Support Vector Machine
- Logistic Regression
- Random Forest
- AdaBoost Classifier

All data obtained from the 5 text representations was used with all these models.

Classification - results

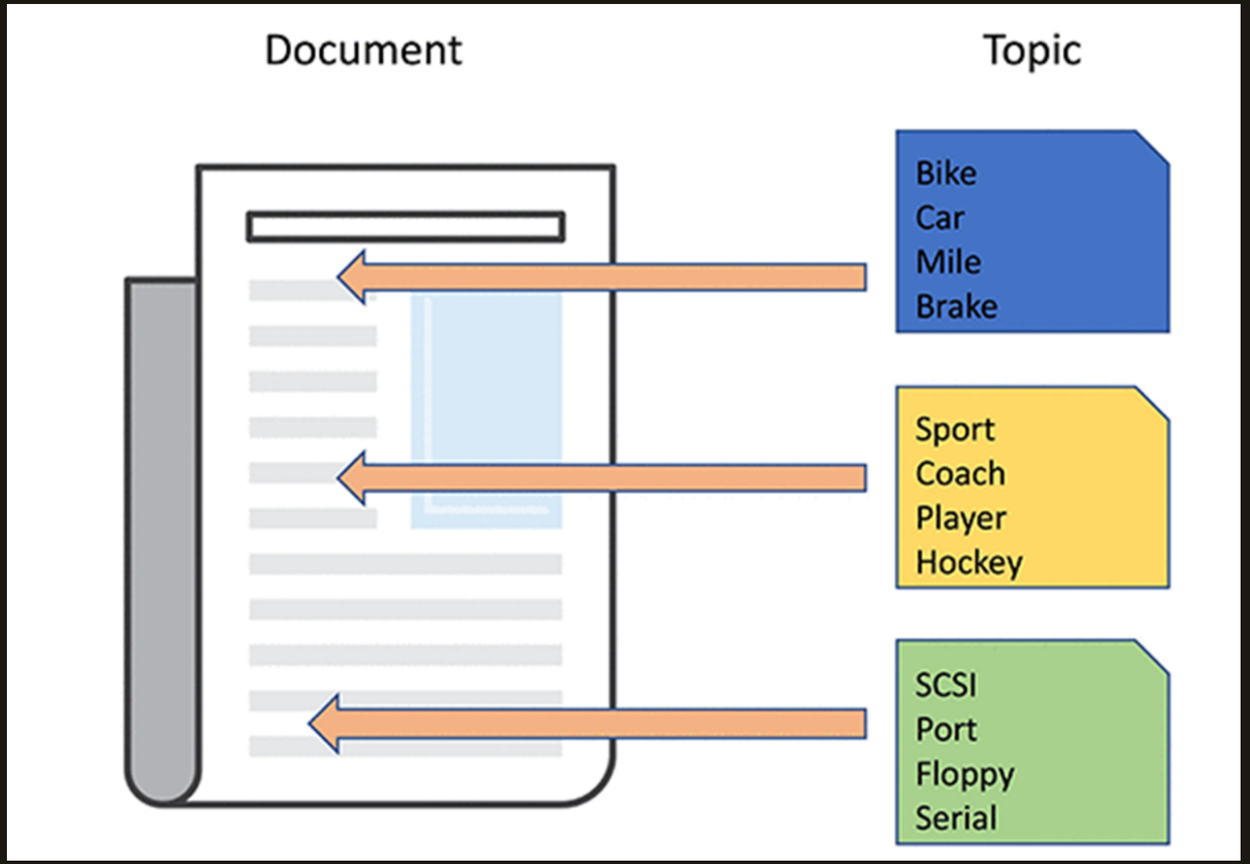
The best model is the Logistic Regression with TF-IDF as text representation with an Accuracy of 0.86.

All models have good performance, with an average Accuracy of about 0.80. Also the other metrics calculated (Recall, Precision, F1, F2) are very similar beeteween the models.

The Random Forest model overfits the data.

The N-gram parameter doesn't lead to significative improvements.

Topic Modelling



Topic Modelling

- Unsupervised task performed with the LDA model, since the Dirichlet model allows to describe the distribution and patterns behind frequently co-occurring words

- Analysis conducted with different numbers of topics (5,8,10,15,19) and after removing too unfrequent (in less than 30% of documents) and too common words (in more of 70% of documents)

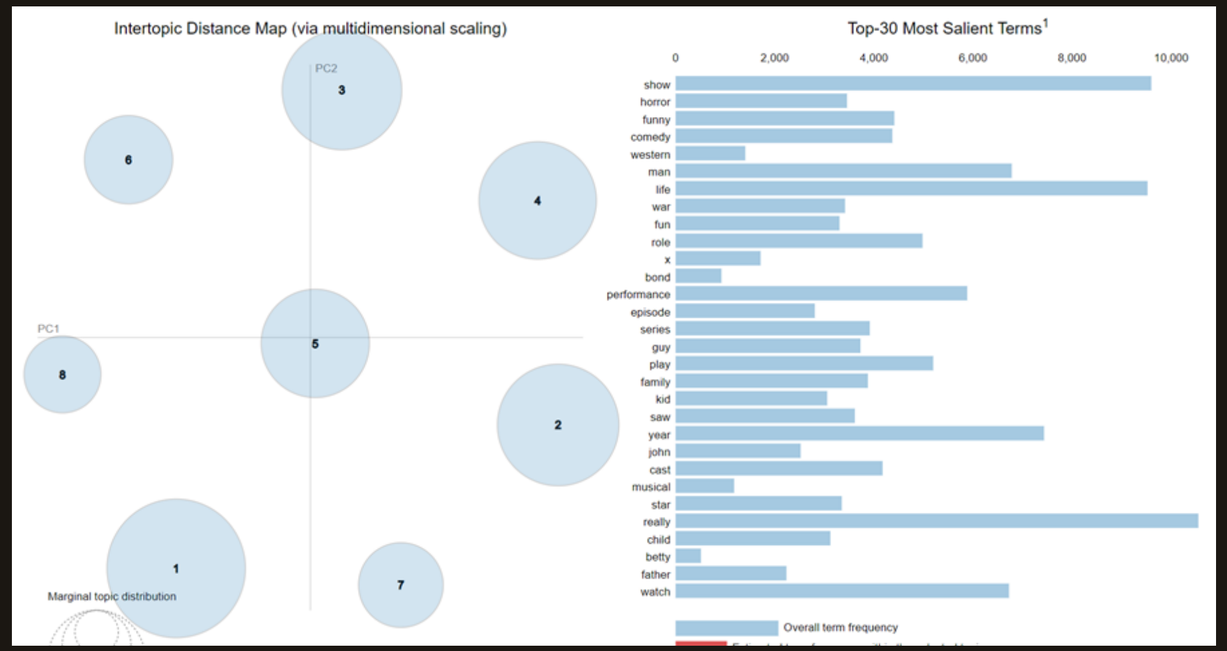
Evaluation of the results

- **Human evaluation** : topics sometimes tend to be confused even if the domain of the documents is overall evident

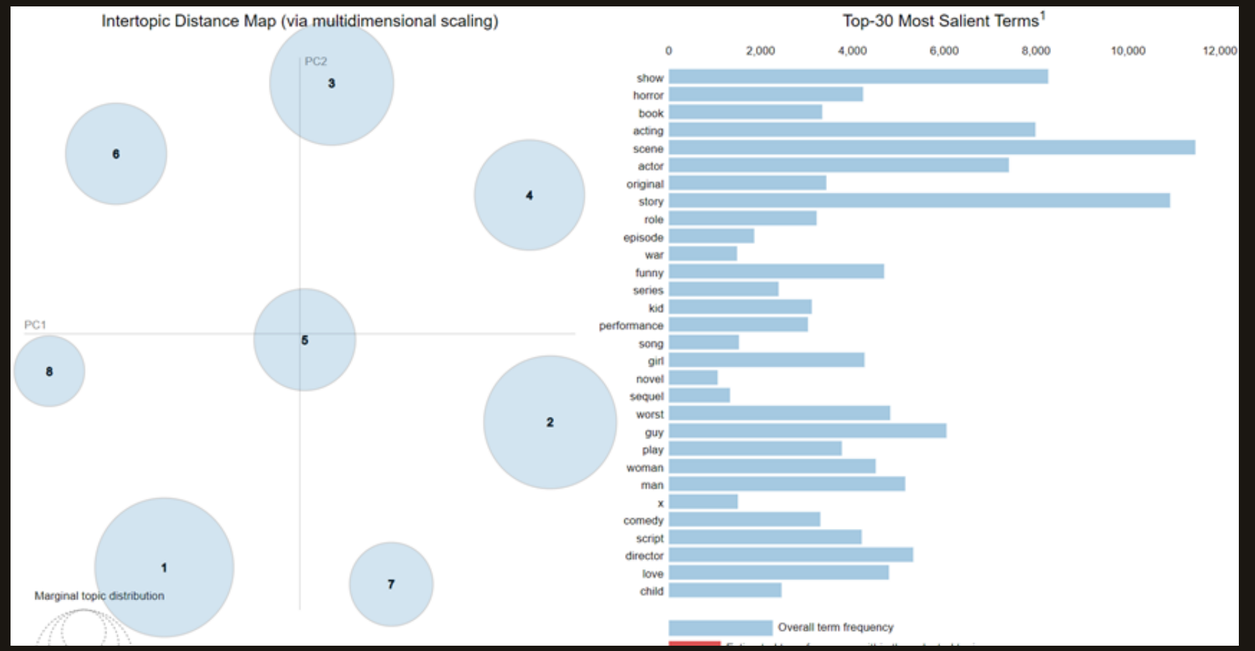
- **Extrinsinc evaluation metrics** : Perplexity to see how the model handles new data and Coherence (c_v version) for semantic similarity of topic top words

	Positive reviews	Negative reviews	Mixed reviews
Perplexity (8 topics)	-8,393	-8,321	-8,410
Coherence (5 topics)	0,293	0,283	0,293
Coherence (8 topics)	0,282	0,281	0,286
Coherence (10 topics)	0,290	0,282	0,318
Coherence (15 topics)	0,294	0,281	0,339
Coherence (19 topics)	0,303	0,284	0,337

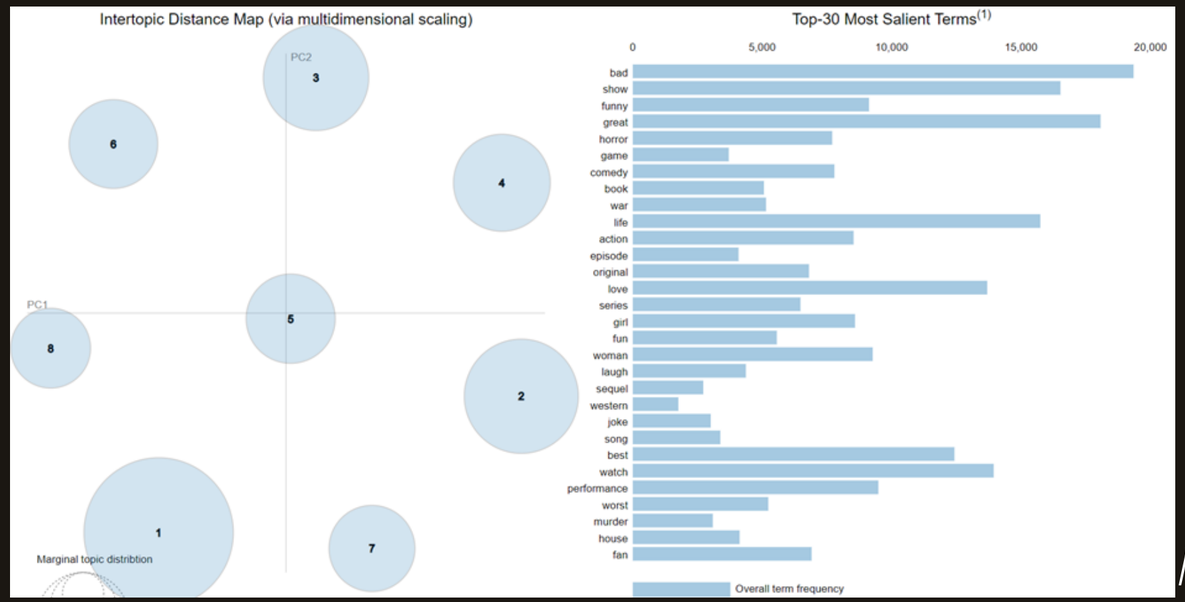
Intertopic Distance Maps of LDA outputs



Positive reviews



Negative reviews



Mixed reviews

...

Conclusions

- For the classification task: the best model is the Logistic Regression with TF-IDF as text representation with an Accuracy of 0.86. In general all models have good performances, with an average Accuracy of about 0.80
- For the topic modelling task: the LDA model with 8 topics was chosen according to the human interpretability of the topics and the graphical output (well-separated and quite big bubbles), even if the Coherence scores weren’t too satisfying in any case.

Future developements

Classification ➔ It would be interesting to use other techniques of text representation and machine learning models. Furthermore, with more powerful processors, it would be possible to use Deep Learning techniques and models.

Topic Modelling➔ It would be interesting to try with more accurate preprocessing and parameter tuning and possibly using bigrams or trigrams as input for the model.

Thanks for the attention