# *Glassdoor Rating Prediction*
# Machine Learning for Natural Language Processing 2022

**Nathalie Bou Farhat**
Ensae
Nathalie.Boufarhat@ensae.fr

**Josephine Gilbert**
Ensae
Josephine_Gilbert@ensae.fr

**David Gauthier**
Ensae
David.Gauthier@ensae.fr

## Abstract

We use different NLP models to predict firm ratings from their employees' comments. We find that models following simple rules with feature engineering perform well for the prediction task, even relative to more complex approaches. This suggests that predicting simple outcome from complex text can be achieved by simple models.

## 1 Problem Framing

How do employees rate their firms? In this project, we study how employees' ratings and statements on their firms relate to each other. We apply a collection of NLP prediction models to Glassdoor data we scraped from the web. These data contain ratings from employees ranging between 1 to 5 stars and commentaries separated in headlines, pros, and cons rubrics.

Selecting textual features that best explain the ratings is crucial for numerous applications. For instance, if analysts want to predict how a firm's image will evolve in time. Yet, doing so is a daunting task given the comments' lexical and syntactic diversity. Accordingly, the models we select to predict ratings from textual data aim to leverage different types of information extracted from the commentaries, from basic word counting to more advanced syntactical analysis.

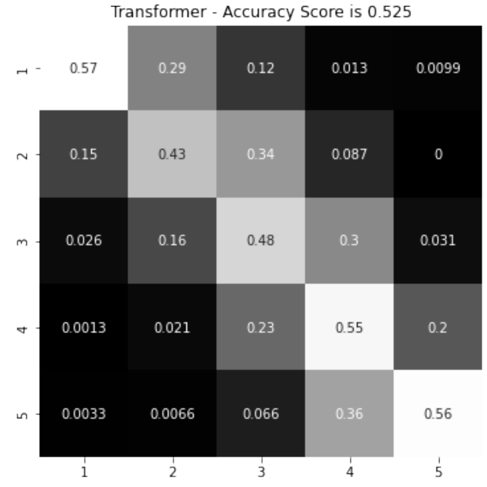
We find that basic models relying only on simple features exhibit a degree of efficacy similar to, or sometimes higher than their more complex counterparts. Our result underlines the potential for simple rule-based models to predict simple ratings from complex textual data.

## 2 Experiments Protocol

We can divide our models into two categories :

- *non deep learning models :* this includes very basic models such as logistic regressions but also more complex models including Random Forests or XGBoost. The inputs of these models were both textual (the different text fields of the dataset) and numerical (features such as the length of the '*pros*' and '*cons*' fields, for example).
- *deep learning models :* we experimented two types of deep models : an LSTM-based neural network and a transformer. Theses two models only used the three textual fields '*headline*', '*pros*' and '*cons*'.

In all those models, we treated the problem as a classification task and used the cross-entropy loss. We tried to think of a better metric, one that could take the ordinal aspect of ratings into account, but we could not improve on cross entropy loss. Furthermore, the confusion matrices of our different models show that when they make mistakes, the models generally predict a rating that is close to the expected one, so using cross entropy was not really a problem (see the figure below for the confusion matrix of the transformer).



## 3 Results

Simpler models seemed to perform better than more complex ones. In particular, despite many attempts to improve the training procedure or the architecture of the model, the LSTM never reached a satisfying level of accuracy (its validation score rarely got above 0.4). For the simpler models we extracted from the reviews 5 different types of feature-set that we used to build our models.

Below is a summary table showing the accuracy scores for different models and feature-sets. Word2Vec , TF-IDF and the combined features turned out to be most useful. Whereas Logistic Regression with Word2Vec features was the best model for this problem. This clearly shows the power of word embeddings in dealing with NLP problems.

| | | vector space | | | | | |
|---|---|---|---|---|---|---|---|
| | Models | All Features Combined | Polarity & Subjectivity | Sentiment Intensity Analyzer | TF-IDF | Word2Vec | Doc2Vec |
| | Random Forest | 46.2% | 30.1% | 33.0% | 43.3% | 45.5% | 38.7% |
| | Logistic Regression | 46.4% | 35.6% | 36.0% | 46.9% | 47.0% | 42.9% |
| | XGBoost | 46.3% | 34.0% | 35.5% | 44.7% | 44.9% | 39.7% |