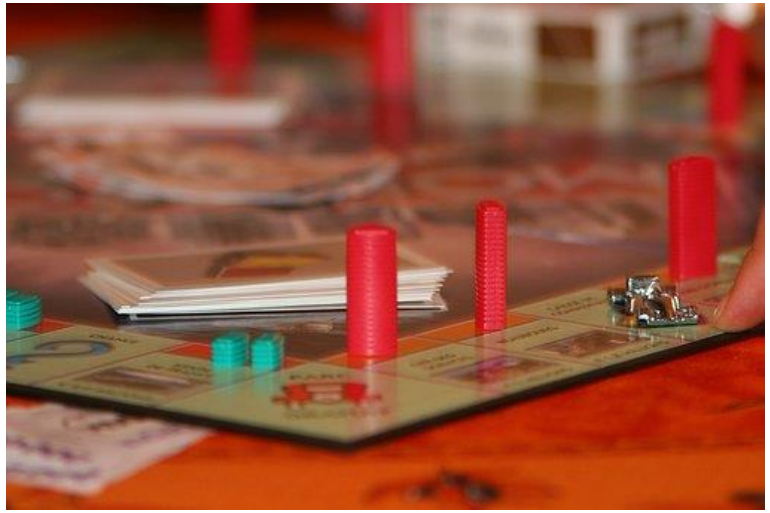
# Capstone Project 2

**Prediction of board game ratings based on their reviews**

**Diane Deroualle – October 2019**

# Problem Statement



- Board games have regained popularity in recent years

- **Purpose: build a model with machine learning and natural language processing to predict the ratings of board games considering the reviews of players, the number of players, the average time of a game, the number of rates… ….**

# Dataset

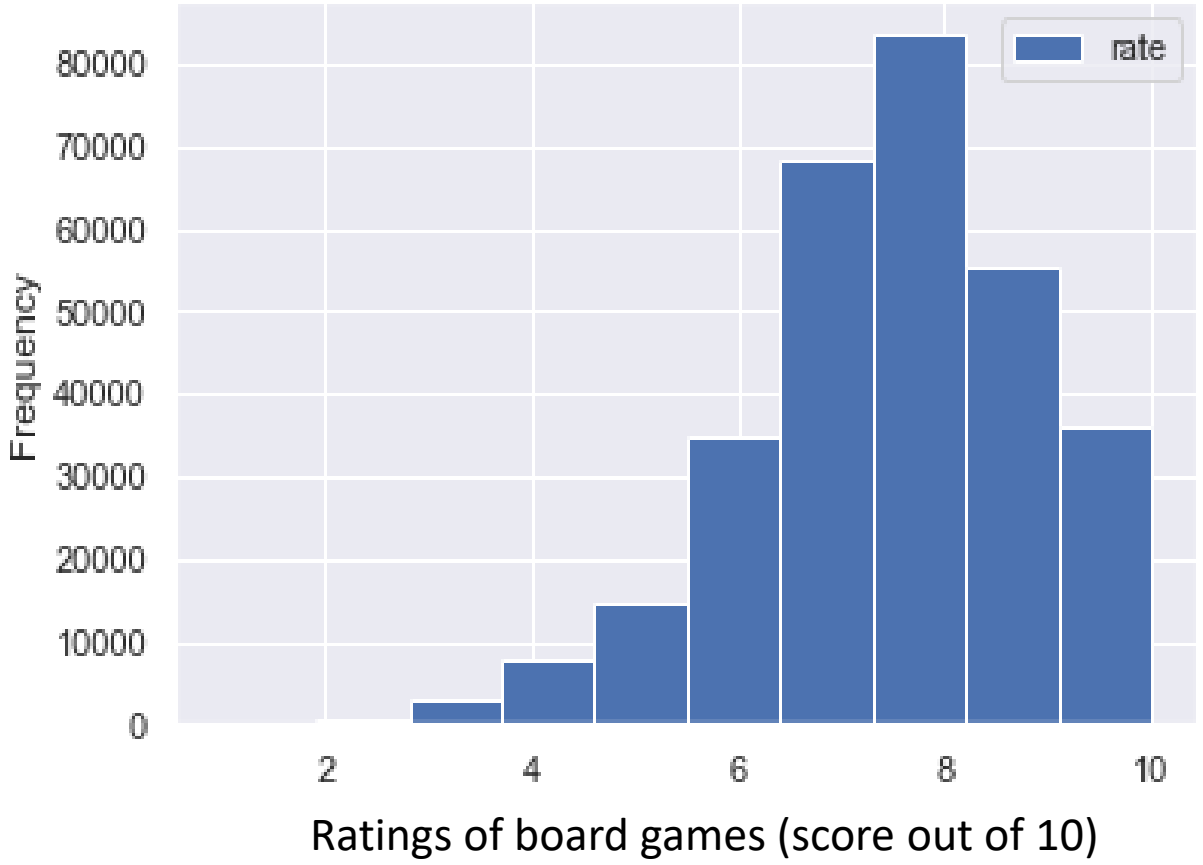
- Scraping and APIs on the website: https://boardgamegeek.com

- For each of the 50 most rated boardgames:

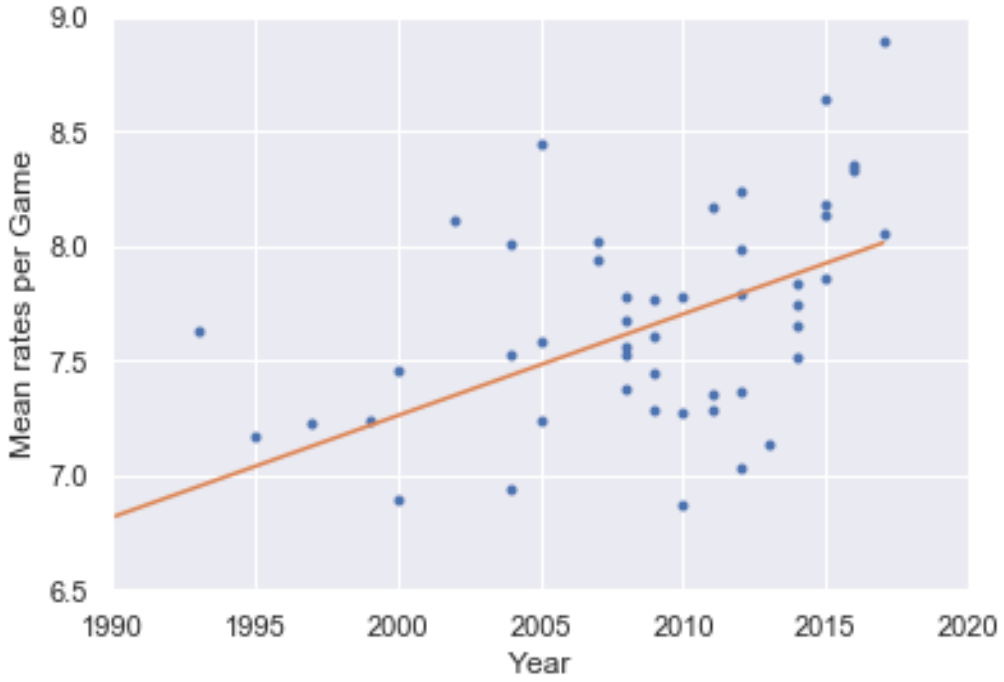
  - ID
  - Name
  - Year of design
  - Minimum and maximum number of players required
  - Minimum and maximum number of minutes required to complete the game
  - Minimum age required
  - Category
  - Number of rates
  - Username of players
  - Reviews
  - Ratings (score out of 10)

- Dataset with 304864 raws

# Distribution of all the rates

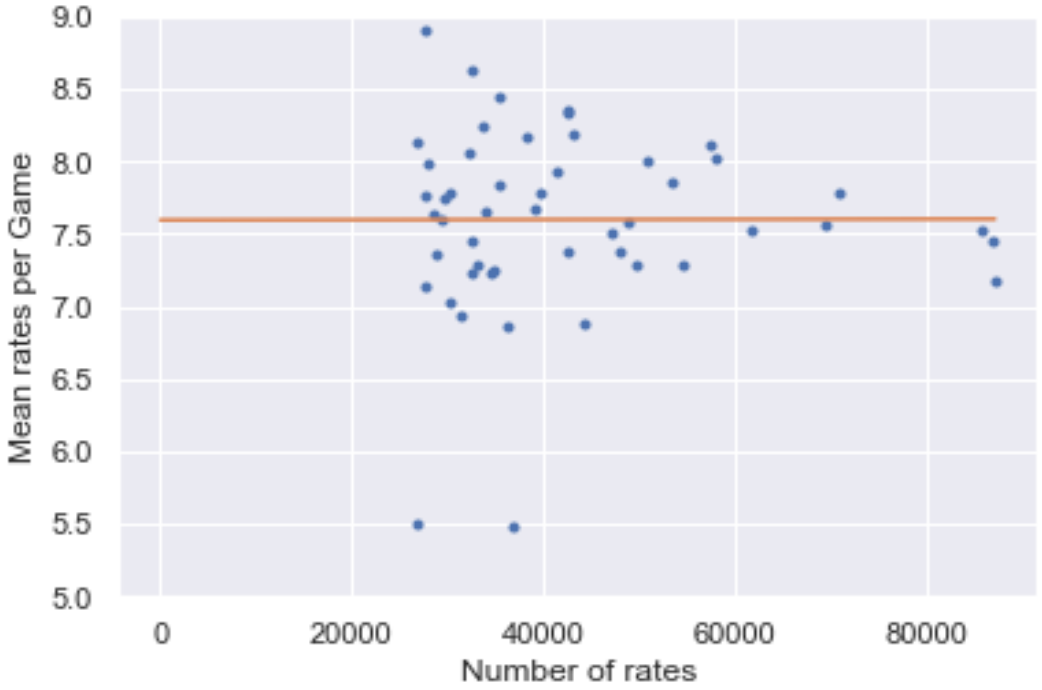


# Relationship between year of design and mean rates per game




- **Positive correlation between the *year of the design of a game* and the *mean of ratings of this game*. More a boardgame is recent and more it seems to have higher mean ratings.**

# Relationship between the mean rates per boardgame and the number of rates per games




- No correlation between the *number of ratings of a game* and the *mean of ratings of this game*.
- No bias in the rates related to the fact we took the 50 most rated games from boardgamegeek.com.

# Linear Regression Model

New dataset: **per boardgames** without the “review” column

➔ Predict the **mean rating** of boardgames

$R^2 = 0.636$

**Most significant features ($p < 0.05$):**

| Features | + / - | Coefficients |
|---|---|---|
| **Year** | + | 0.043 |
| **Max_play** | - | 0.099 |
| **Min_age** | + | 0.095 |
| **Nb_rate** | + | $8.42^{e-06}$ |

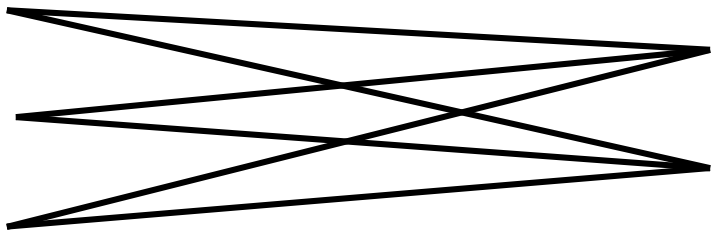
# Predict ratings of boardgames

## FIRST STEP

Convert only “review” column to a matrix of token counts:

**Natural Language Processing**

- CountVectorizer
- TdidfVectorizer
- Doc2Vectorizer



## SECOND STEP

Predict ratings of boardgames with all the columns of dataset:

**Machine Learning Algorithm**

- Random Forest Regressor
- Support Vector Regressor

# Predict ratings of boardgames

## FIRST STEP

Convert only “review” column to a matrix of token counts:

**Natural Language Processing**

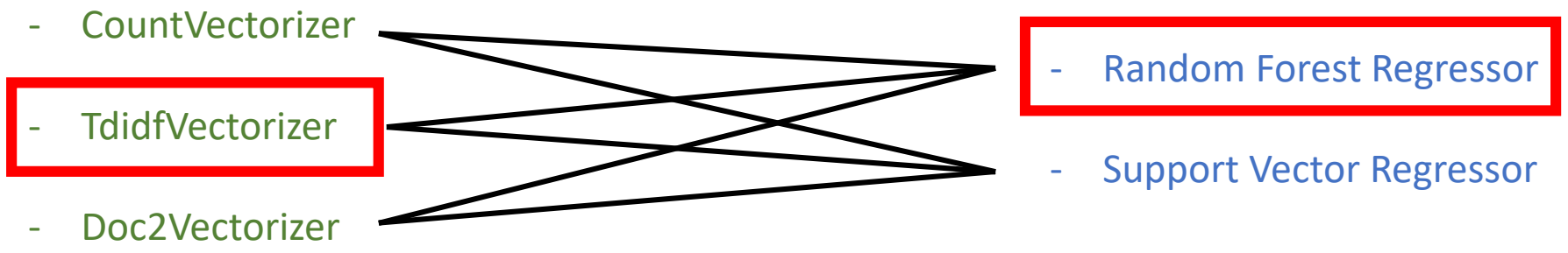
- CountVectorizer
- TdidfVectorizer
- Doc2Vectorizer

## SECOND STEP

Predict ratings of boardgames with all the columns of dataset:

**Machine Learning Algorithm**

- Random Forest Regressor
- Support Vector Regressor



In a pipeline and using a cross-validation with 5 folds:

**$R^2$ training data = 0.21**
**$R^2$ test data = 0.21**

# Best predictor words

=> New model with only reviews as features

| ‘Good words’ | ‘Bad words’ |
|---|---|
| Love | Precio |
| Favorite | Better |
| Great | Long |
| Best | Like |
| Perfect | Felt |
| Excellent | Bad |
| Fantastic | Maybe |
| Amazing | Wa |
| Awesome | Random |
| Firm | Boring |

# Conclusion

➔We choose the model with TfidfVectorizer to process the ‘review’ column and Random Forest Regressor as machine learning algorithm.

➔Managers of gaming shops or department stores should focus on more recently created games, games with a minimum age not too low and a maximum number of players not too high.

➔We need to improve the natural language processing of the ‘review’ column with a more powerful computer to improve the model.