Capstone Project 2: Final Report

Board Game Review

Problem Statement

Board games have regained popularity in recent years, they make it possible to entertain many afternoons with family or evenings with friends during hours. There are more and more of them and different kinds of games ranging from collaborative games to strategy games and more family games.

The purpose of this project is to construct a model using machine learning and natural language processing to predict the rating of board games based on the reviews of clients and some other features as the number of players, the average time of a game, the number of rates…

Potential clients of this project could be the board game designers so that they know the favorite game characteristics of the clients or they could be the managers of gaming shops, board game bar or larger department stores so that they know which games they will be most sensitive to sell.

For that, I will use scraping and APIs on the website: https://boardgamegeek.com following commands on this website: https://boardgamegeek.com/wiki/page/BGG_XML_API2. The website boardgamegeek.com is a reference in terms of board games to receive advices and explanations from a large community.

Description of the dataset

We extracted from the fifty most rated boardgames on boardgamegeek.com, the following features for each of these boardgames:

- The ID
- The name
- The year of design
- The minimum number of players required
- The maximum number of players required
- The minimum number of minutes required to complete the game
- The maximum number of minutes required to complete the game
- The minimum age required to play
- The category

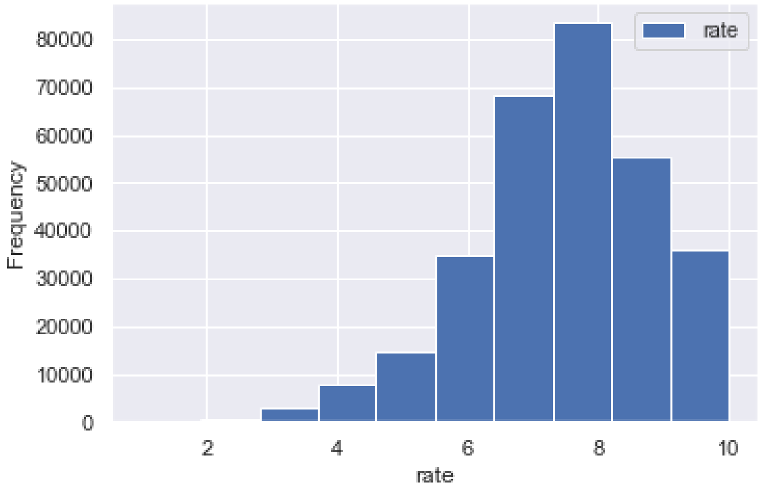
- The number of rates
- All the reviews about the game with the username of the person who wrote it and the associated note score out of 10.

Then, we registered the new dataframe in csv format.

In a new notebook, we loaded the dataframe, we shuffled it because it is sorting by games and then, we deleted the rows containing missing values in the ‘review’ column. The final dataset contains 304864 reviews.

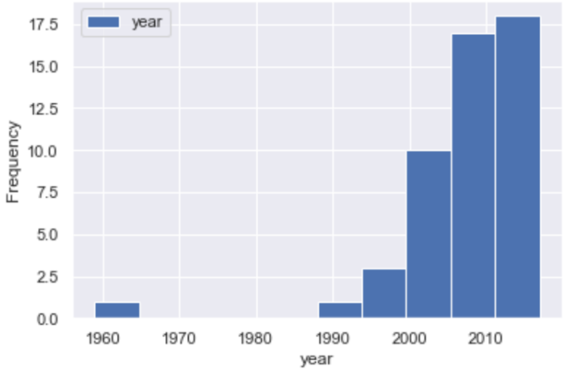
Visual exploratory data analysis

The following histogram shows us the distribution of all the rates gave by players for the 50 more rated games of the website boardgamegeek.com. We can see a normal distribution of the rates distributed around 7.



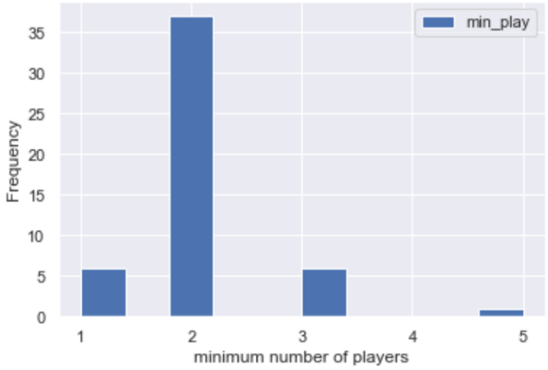
Then, I created a new dataframe with one boardgame per row and the average of all the rates given for each game in the ‘rate’ column using ‘groupby’ method. From this new dataframe, we have drawn several plots.

Distribution of years of design of the boardgames



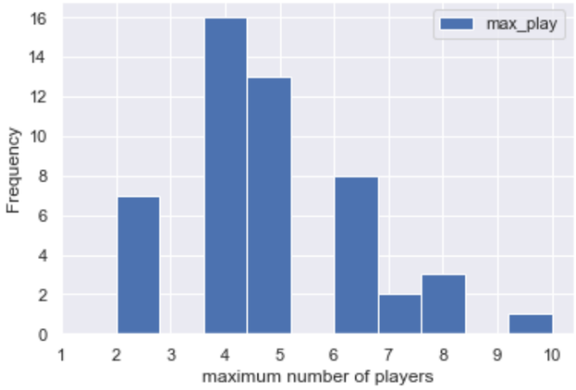
This histogram shows us that most of the games have been created during the last 20 years. There is one outlier game created in 1960.

Distribution of the minimum number of players required for a boardgame



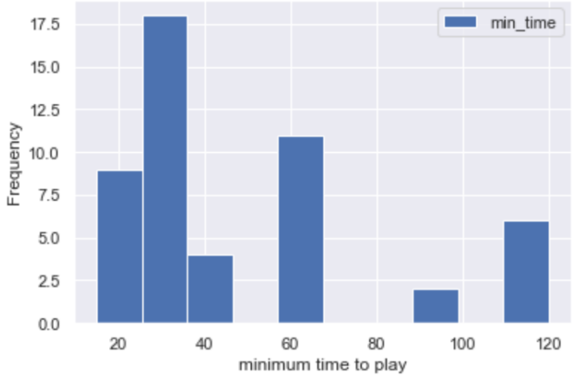
We can see that the distribution of the minimum number of players is between 1 and 5 with a big majority of 2 players.

Distribution of the maximum number of players required for a boardgame



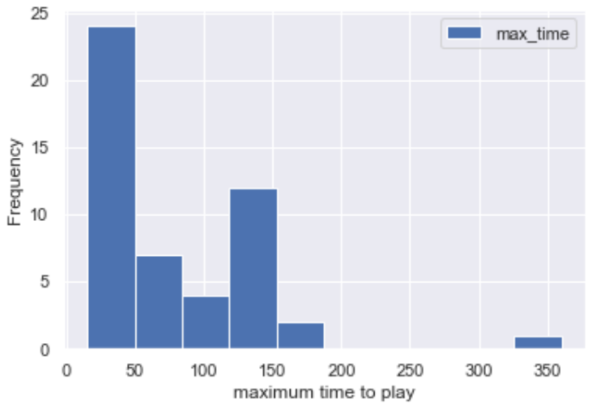
We can see that the distribution of the maximum number of players is between 2 and 10 with a majority of 4 players.

Distribution of the minimum number of minutes required to complete a game



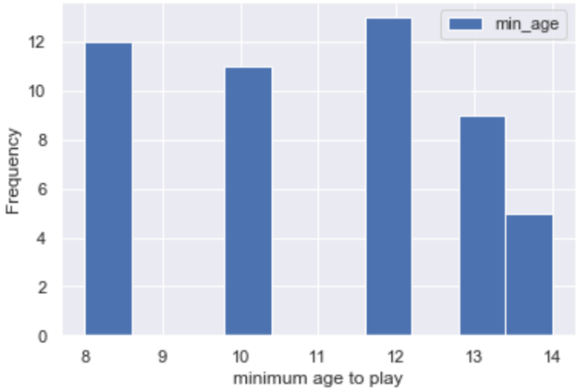
This plot shows us that the distribution of the minimum number of minutes for a play is between 20 and 120 minutes with a majority of 30 minutes.

Distribution of the maximum number of minutes required to complete a game



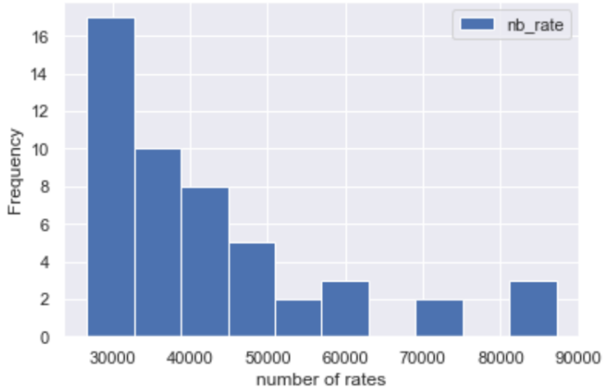
This plot shows us that the distribution of the maximum number of minutes for a play is most often less than 50 minutes.

Distribution of the minimum of age required to play



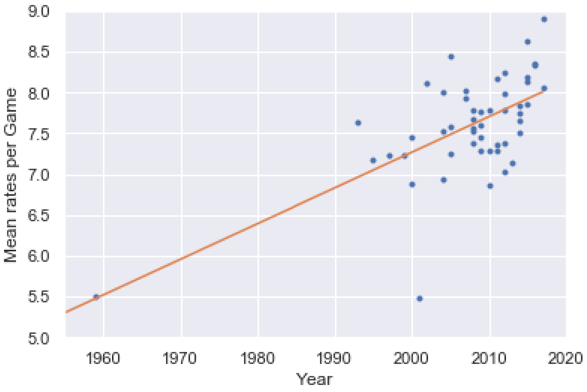
This histogram shows us that the distribution of the minimum age required to play is between 8 and 14 years old.

Distribution of the number of rates by boardgame

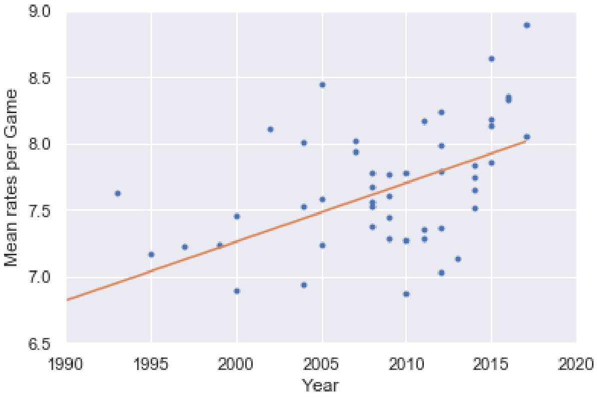


This histogram shows us that the distribution of the number of rates by game is between 30000 and 85000 rates.

Relationship between the mean rates per boardgames and the years of the design

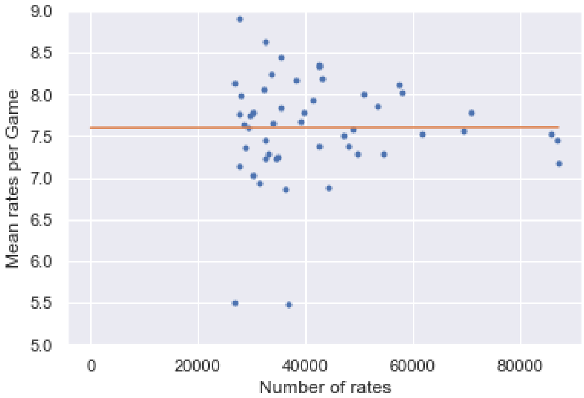


We can see that seems to have a positive correlation between the year of the design of a game and the mean of rates of this game. More a boardgame is recent and more he seems to have higher mean rates.



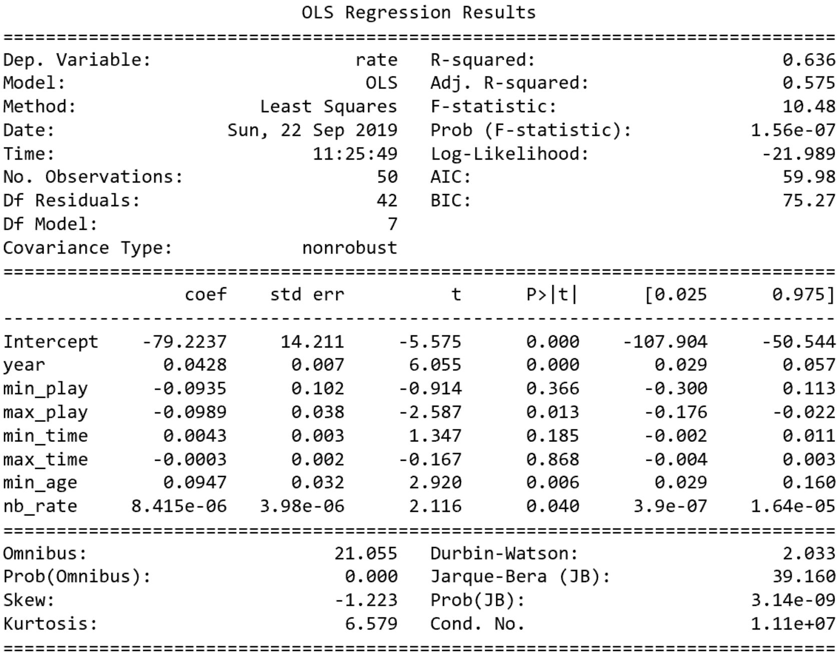
We can observe on the previous plot an outlier value which is a very old boardgame compared to the others (designed before 1960). We removed this value in this second plot to compare the slop without the outlier value. We can see that the slop is similar to the previous plot, it seems to have a positive correlation between the year of the design of a game and his average rate.

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



We can see that seems to have no correlation between the number of the rates of a game and the mean of rates of this game. So, even if we chose to take the 50 most rated games from boardgamegeek.com for this project, it doesn't seem to be a bias regarding the rating of these 50 games.

Statistical exploratory data analysis



```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:                   rate   R-squared:                       0.636
Model:                            OLS   Adj. R-squared:                  0.575
Method:                 Least Squares   F-statistic:                     10.48
Date:                Sun, 22 Sep 2019   Prob (F-statistic):           1.56e-07
Time:                        11:25:49   Log-Likelihood:                -21.989
No. Observations:                  50   AIC:                             59.98
Df Residuals:                      42   BIC:                             75.27
Df Model:                           7                                         
Covariance Type:            nonrobust                                         
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    -79.2237     14.211     -5.575      0.000    -107.904     -50.544
year           0.0428      0.007      6.055      0.000       0.029       0.057
min_play      -0.0935      0.102     -0.914      0.366      -0.300       0.113
max_play      -0.0989      0.038     -2.587      0.013      -0.176      -0.022
min_time       0.0043      0.003      1.347      0.185      -0.002       0.011
max_time      -0.0003      0.002     -0.167      0.868      -0.004       0.003
min_age        0.0947      0.032      2.920      0.006       0.029       0.160
nb_rate     8.415e-06   3.98e-06      2.116      0.040     3.9e-07    1.64e-05
==============================================================================
Omnibus:                       21.055   Durbin-Watson:                   2.033
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               39.160
Skew:                          -1.223   Prob(JB):                     3.14e-09
Kurtosis:                       6.579   Cond. No.                     1.11e+07
==============================================================================
```

We can see that the features min_play, min_time and max_time have both a very high p_value. They are not good predictors for the mean rate of boardgames. The other features have a significant p_value (for an α=0.05). The year, the min_age and the nb_rate have positive coefficients, the mean rate of boardgames increases when the value of these three features also increase. The max_play has negative coefficient, the mean rates of boardgames increases when the value of this feature decreases. The year and the min_age have the lower p_values, less than

0.01. If the year increases by one unit (1 year), the mean rates of boardgames increases by 0.04. If the min_age increases by one unit (1 year), the mean rates of boardgames increases by 0.095.

Machine Learning

To predict the rate of boardgames given by players, in a first place, we pre-processed the text in the ‘review’ column. Then, we tried different options to build the best model which will allow us to predict the rate of the boardgames. First, we tried to use CountVectorizer, TfidfVectorizer or Doc2Vectorizer () on the ‘review’ column after the pre-processing of the text. Then we used two different machine learning algorithms, Random Forest Regressor or Support Vector Regressor to also choose the best model. To combine all these functions, we used a pipeline.

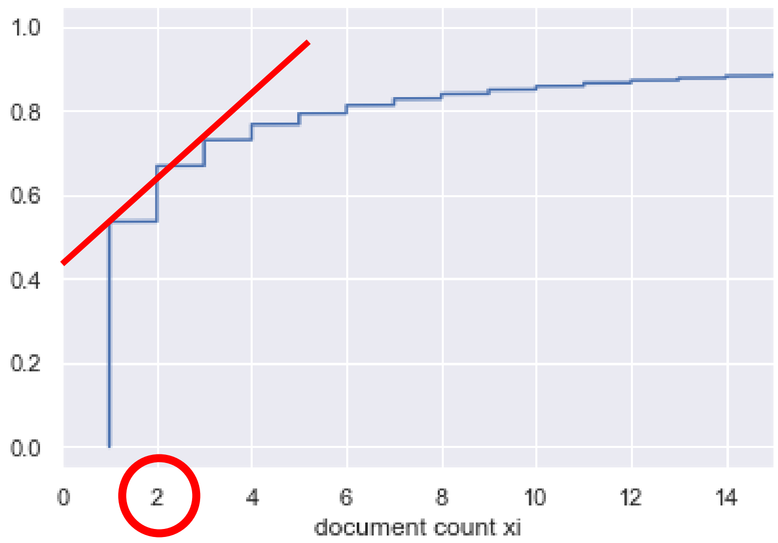
➢ Natural Language processing

First, we **pre-processed** the text in the ‘review’ column to prepare it for the next step:

- We put all the text in **lower cases**
- We used **tokenize** to select each words and not the punctuation
- We then used **lemmatizer** to group together the similar words
- Finally, we eliminated the **stop words**

Then, we tried to use 3 different models to process the review column to choose the best way to process our text. These are **CountVectorizer ()**, **TfidfVectoriwer ()** and **Doc2Vectorizer ()**.

First of all, we have to tune the hyperparameter min_df, this is the minimum number of documents a word must appear in for it to be included in the vocabulary. To select it, we constructed the cumulative distribution of document frequencies using CountVectoriser () and we choose a **min_df = 2**.



After using Countvectorizer () or TfidfVectorizer (), we used **TruncatedSVD ()** to reduce the dimensionality with a hyperparameter **n_components = 100**. Doc2Vectorizer () is used with spacy to integrate it in the pipeline.

- **Machine Learning Algorithms**

Then, with the different 'review' processed column and the other columns of the dataframe ('year', 'min_play', 'max_play', 'min_time', 'max_time', 'min_age', 'nb_rate'), we tried two different machine learning algorithms to choose the model that best predicts the rate of the boardgames. These are **Random Forest Regressor** and **Support Vector Regressor**.

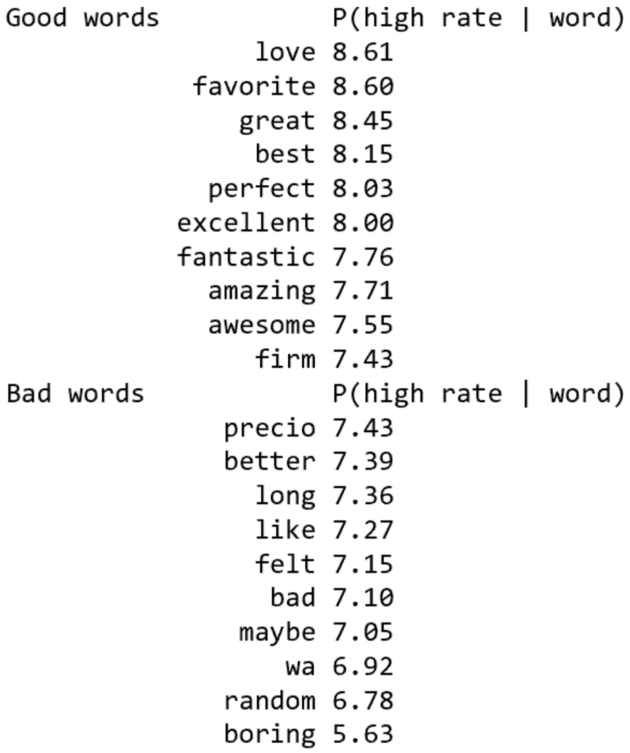
For Random Forest Regressor, we tuned two hyperparameters **max_depth** (The maximum depth of the tree) and **min_samples_leaf** (The minimum number of samples required to be at a leaf node). We choose them by using **GridSearchCV ()** with cv = 5. We obtained a max_depth = 8 or 9 depending on the natural language processing choose for the 'review' column and a min_samples_leaf = 50.

Finally, we used a **pipeline** with **FeatureUnion** to combine all the different processes by selecting only the 'review' column for the natural language processing part and selecting all the other columns for the machine learning part. When Doc2Vectorizer and Support Vector Regressor were a part of the pipeline, we used a sample of the dataframe to be able to complete the pipeline in reasonable amount of time. We used the pipeline in combination with

cross_val_score with **cv = 5**. We get the final accuracy by calculating the mean of the five accuracies obtained by the cross_val_score. All the accuracies obtained with the different combinations of natural language processing for the 'review' column and the machine learning algorithms were a bit low.

The better combination was a pipeline with TfidfVectorizer (), TruncatedSVD () and Random Forest Regressor. The **accuracy of the training data was 0.21** and **the accuracy of the test data was 0.21** also. So, we don't have an overfitted model.

Finally, we were looking for the words that are the best predictors for a good rate and the best predictors for a poor rate. For doing that we had first, to create a new model using: a sample of the dataframe and only the 'review' column for the X. And we did not use the dimensionality reduction after using TfidfVectorizer and we did not use a pipeline. Then, we created a data set such that each row has exactly one feature and we used the trained classifier to make predictions on this matrix. Finally, we sorted the rows by predicted probabilities, and pick the top and bottom *K* rows.



```
Good words          P(high rate | word)
                love 8.61
            favorite 8.60
               great 8.45
                best 8.15
             perfect 8.03
           excellent 8.00
           fantastic 7.76
             amazing 7.71
             awesome 7.55
                firm 7.43
Bad words           P(high rate | word)
              precio 7.43
              better 7.39
                long 7.36
                like 7.27
                felt 7.15
                 bad 7.10
               maybe 7.05
                  wa 6.92
              random 6.78
              boring 5.63
```

To conclude this project, we built a model to predict the rates given by players to boardgames by using several features like the number of players, the average time of a game, the number of rates and more importantly the reviews of the players. We try to use several combinations of natural language processes for the ‘review’ column and machine learning algorithms to predict the rates of boardgames, to find the most efficient combination. We finally used TfidfVectorizer to process the ‘review’ column and Random Forest Regressor for the machine learning algorithm. We found similar accuracies for the data training and the data set although the value is weak. Based on the results of the linear regression model, we could recommend to managers of gaming shops or department stores to focus on more recently created games, games with a minimum age not too low and a maximum number of players not too high. To finish, we built another model with only the reviews of boardgames as features to identify the words that have the higher predictive power to match a good rating (‘love’, ‘favorite’, ’great’, ’best’….) and the words that have the lower predictive power to match a good rating (‘boring’, ‘random’, ’bad’, ’maybe’….).