# CSE258 Assignment 2

Martin Borge Heir
Henrik Larsson Hestnes

November 2021

# Contents

# Introduction

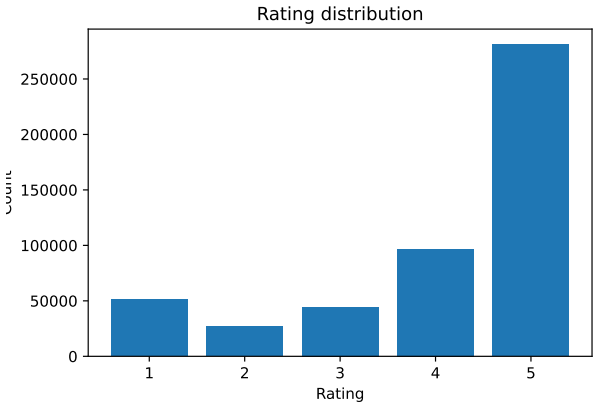
The video game industry is one of the fastest growing industries, with new games being released every day. Games being bought online versus physical has increased from 20% in 2009 to 83% in 2018[1]. Thus, recommending games online has become an increasingly important problem in recent years. In this assignment we will study the problem of predicting ratings in the context of video game reviews. First, we will dive deeper into the statistics of the selected dataset. Secondly we'll have a look at the state-of-the-art papers on the matter, before we introduce our models and their results.

# 1 Dataset

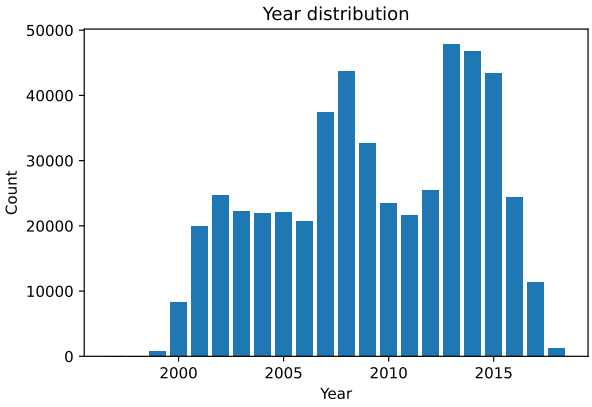
The dataset chosen for this assignment is an Amazon dataset containing 2.565.349 reviews of video games[2]. The dataset can be downloaded from this link. Since this is a huge dataset, the dataset were narrowed down to the 500.000 first entries. These entries were again split up in a 300.000 reviews train set, 100.000 test set and also 100.000 validation set, and will be used throughout the rest of the assignment.

To start out with the most basic exploration, this subset of the dataset has a total of 344.165 users, where each user on average has a total of 1,45 reviews. The fact that each user has such a low amount of reviews will undoubtedly have an impact of how the predictive model has to be made. On the other hand, we only have a total of 9.597 different video games in the subset, where each game has been reviewed a total of 52,1 times on average, which means that we have a lot of data per video game.

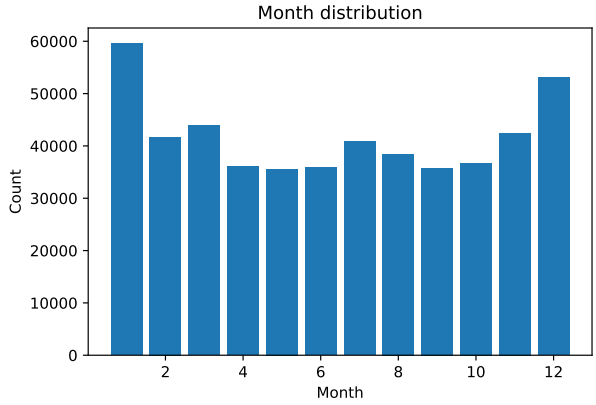
The second explorations performed on the dataset was to analyze very basic distributions; how the review ratings are distributed and how the reviews are distributed across years, months and weekdays.



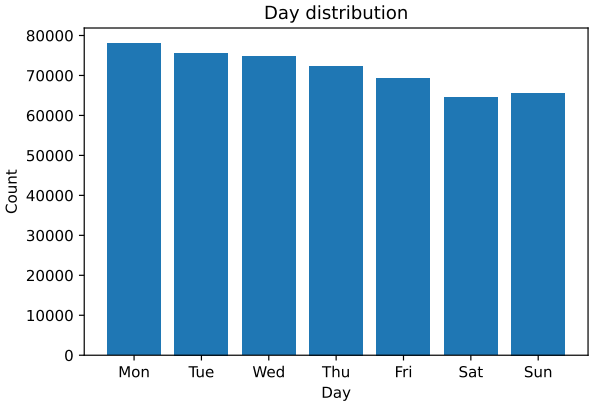

(a) Rating distribution




(b) Year distribution




(c) Month distribution

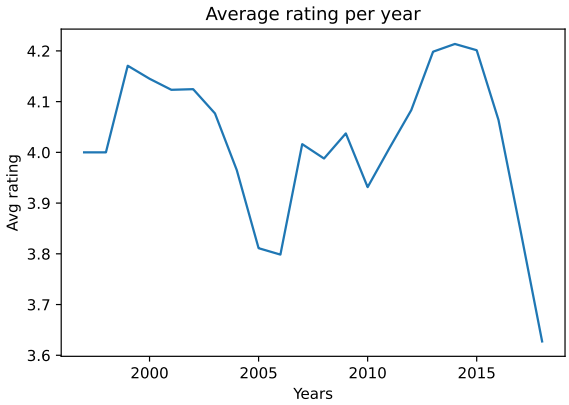



(d) Day distribution

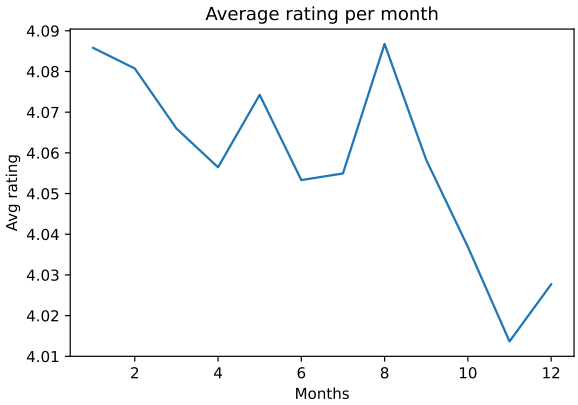
Figure 1: Distribution of the dataset

As can be seen from fig. 1a, there is a clear majority of positive reviews; over 50% of the reviews has given a 5-star rating to the game, while only about 10% of the reviews are 1-star. The majority of the reviews are given between year 2000 and 2017, with December and January as the months with the most reviews, perhaps due to Christmas holiday[Ed.]. Monday is marginally the day with the most reviews, with a slightly declining trend throughout the week.

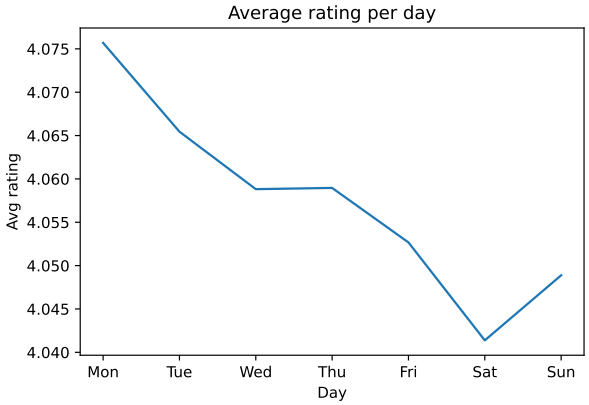
The third exploration done was to see how these metrics influence the rating trend.




(a) Average rating per year




(b) Average rating per month

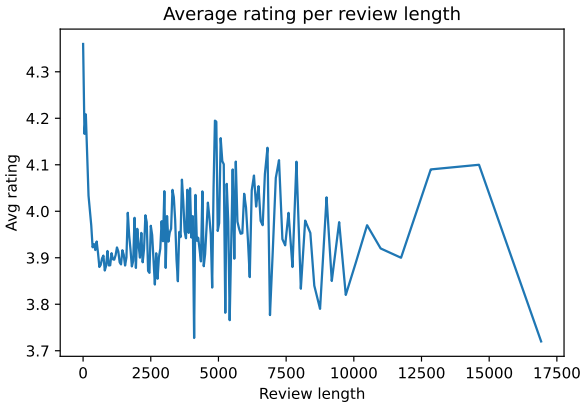



(c) Average rating per day

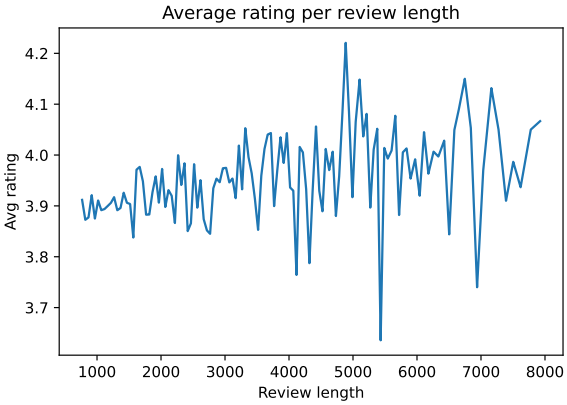
Figure 2: Average ratings

As can be seen from fig. 2, there are some clear trends regarding the rating. In fig. 2a it can be observed that the year with the highest ratings were 2014, with 2103 and 2015 coming right behind. In fig. 2b it can be observed that reviews from August and January in general has the best ratings, while November has the worst. The weekday of the review does not seem to matter much, but it is still a trend that reviews on Mondays tend to be better, while the further out in the week you get, the worse the reviews get, until Sunday when it starts getting better again.

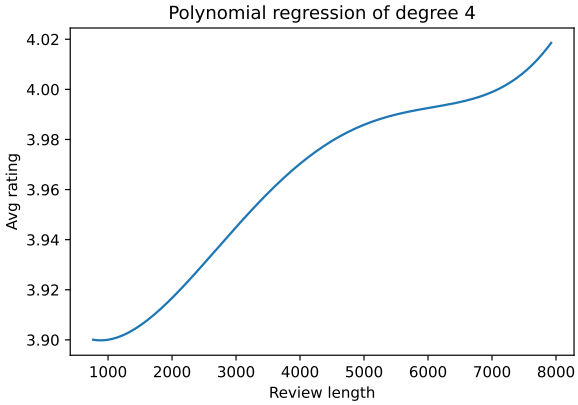
Another interesting trend to look at is how the length of the review text impacts the rating.




(a) Average rating per review length




(b) Average rating per review length without outliers




(c) Regression of average rating per review length without outliers

Figure 3: Average ratings per review length in buckets of 50

In fig. 3a the average rating per review length is collected in buckets of size of 50 characters. This depiction is however quite misleading, as we have some outliers in the dataset when it comes to review length. In fig. 3b the reviews with the 10% longest and shortest review texts have been removed. Here you can see a slight trend that the longer the review length, the better the review rating. To make this trend even clearer, fig. 3c depicts a polynomial regression of 4th degree of the data depicted in fig. 3b. Here it becomes quite clear that an increasing review length in general contributes to a better rating.

A last, but very interesting metric to look at is what words are used in the review texts. This can be used as features in a sentiment model.


Figure 4: Wordcloud of the most used words

Figure 4 depicts this metric in a wordcloud after all the words from the review texts have been filtered by removing stopwords, and also gone through stemming such that e.g. the word play, plays and played falls under the same word play. The stemming is also the reason for some strange words in the wordcloud, like wa and thi.

## 2 Predictive task

The task for this assignment is to predict the rating of a review based on user id, item id, temporal information and review text. In this section we will explain how we will proceed and what measures we will take to ensure the validity of our results.

For evaluation, it's important to split the dataset into training and test sets, such that the performance can be evaluated on an "unseen" set. This is an important measure against overfitting. A problem with only using two sets is that when tuning the hyperparameters, we can still overfit to the test set. It is therefore wise to also have a validation set, which can be used for hyperparameter tuning, such that the test set remains unseen.

To evaluate the results of our model we need to have some sort of baseline comparisons. For instance, a very simple baseline is to always predict the mean. Thus if a model predicts worse than this baseline, the model is not very good. To have a sense of measure for how well our final model performs we will model several baselines and compare their performance to the final model.

# 3 Literature

There have been a lot of studies in the past of similar datasets, with the The Netflix Prize from 2006 as the highest profiled which lead to a huge amount of research on rating prediction. The dataset from The Netflix Prize did however only consist of the features user, item, rating, timestamp, while the dataset used in this assignment contains significantly more metadata[3].

Another study of a more similar dataset have used reviews of restaurants from Yelp to predict how a user will rate a given restaurant[4]. This study tries to predict rating by looking at the problem as both a classification problem predicting integer ratings, and by looking at it as a regression problem predicting decimal ratings. To predict the rating the study takes advantage of the review text, and tries to predict the rating based on this text. In short words, the result from this study was that the linear regression performs best in terms of the MSE, which is the type of model that has been utilized in this assignment.

Another study of the same dataset, but with a slightly different approach were also done in 2015. This paper tests the following models for achieving the rating prediction; support vector machine, latent factor, collaborative filtering and random forest[5]. Each of these models utilized different features to do the prediction. The conclusion of the paper is that the random forest model is the best performing, and thus the preferred model.

The state-of-the-art method currently employed to this type of data are different types of deep-learning-based rating predictors[6][3]. The real advantage of these models is that they can capture complex, non-linear relationships among users and items, beyond what is possible with the simple aggregation functions used in a basic latent factor model. There is however an ongoing debate whether deep-learning models are really worth it, with regards to that we get a lot of added complexity to our model for just the modest improvement in performance.

# 4 Model and Results

For the sake of pedagogy and learning, we've chosen not to dive deep into the state-of-the-art deep learning models for solving this task, but rather start from the basic models taught in this course, and try to reason behind these to build a better model ourselves.

Based on the information we have gathered from the dataset in section 1, we can make some assumptions about what models will perform well on this dataset for this task. First of all, the fact that we have few reviews per user will make user based information less valuable. On the other hand, the fact that we have many reviews per game will make the item information more valuable. Furthermore, the temporal information for different years shows great promise.

In this section we will describe the models we have developed, their performance as well as their strenght and weaknesses.

## 4.1 Baseline 1 - predicting mean

$$\hat{y} = \alpha \tag{1}$$

where $\alpha$ is the mean of the ratings in the training set. This is the first and simplest baseline. The performance on the test set was measured to $mse = 1.7866$. This model is far too simple to be considered a good model and the downsides of predicting the same for every user should be quite obvious.

### 4.2 Baseline 2 - Regression

In the second baseline model we are fitting a vector $\vec{\theta}$ such that

$$\vec{\hat{y}} = X\vec{\theta} \tag{2}$$

where $X$ is a matrix where each row can be represented as a vector representing each review on the form

$$\vec{x_i} = [1 \underbrace{000...010}_{\text{year}} 0 \underbrace{001...000}_{\text{month}} \underbrace{x}_{\text{review length}}] \tag{3}$$

where the year between 2002-2015, and month from jan-dec have been one-hot encoded. This model performed with an $mse = 1.7784$ on the test set and is performing slightly better than baseline 1, which is expected. However the increase is not substantial, which motivates for even more sophisticated models.

### 4.3 Baseline 3 - User, item biases

In this model we are considering a unique bias term for each user and item, which is a model on the form

$$\hat{y} = \alpha + \beta_{\text{user}} + \beta_{\text{item}} \tag{4}$$

where each $\beta_{\text{user}}$ , $\beta_{\text{item}}$ is a personal bias for each user and item in the training set. However, this means that if the user or item is not in the test set, the model reduces to $y = \alpha$ (not equivalent to the mean). From section 1, we found that there are about 1.45 reviews per user, which means in very few cases the $\beta_{\text{user}}$ term will apply. This is an argument against relying on such a parameter for this dataset. To further optimize the model, and to avoid overfitting, a regularization parameter was introduced. The objective function can be written as

$$\arg\min_{\alpha,\beta} \sum_{u,i} (\alpha + \beta_u + \beta_i - R_{u,i})^2 + \lambda \left[ \sum_u \beta_u^2 + \sum_i \beta_i^2 \right] \tag{5}$$

After tuning the hyperparameter $\lambda$, the model ended up performing with an $mse = 1.6957$, which beats the preceding baselines by a lot. This is to be expected, since this model is popular in recommender systems and often form the basis of more advanced models[3].

### 4.4 Baseline 4 - LFM

Latent Factor Model (LFM) is a model that gained it's popularity during the "Netflix Price" movie-ranking competition [7]. In this model we are also capturing underlying relations between user preferences and item features through the variables $\gamma_{\text{user}}$ and $\gamma_{\text{item}}$ , so the model becomes

$$\hat{y} = \alpha + \beta_u + \beta_i + \gamma_u \cdot \gamma_i \tag{6}$$

Using the same principle for tuning the regularization parameter $\lambda$ as in the previous section we achieved performance on the test set of $mse = 1.6902$. This result is slightly better than Baseline 3, but also somewhat disappointing as we believed the latent variables would up the performance by even more. To it's defence however, since we are having few user reviews in the dataset, the latent variables reduces to zero for most instances in the test set, which can explain why we're not experiencing a greater increase in performance.

### 4.5 Baseline 5 - Text based sentiment model

Another strategy than what we have previously looked at, is to model the sentiment of the words in the text body of the reviews. Such models have been previously known to achieve good performance on applications such as predicting rating from review text[3]. To be able to model these features, we need to do some preprocessing of the data. First of all we remove the punctuations and stopwords in the text, since this information doesn't provide anything useful for our model. Furthermore we reduce the number words in our dictionary by a process called stemming. In short a stemming algorithm reduces words like *drinks* and *drinking* to simply *drink*, to be able to model similar words only once [3][8]. Lastly, because the dictionary of words is too large to train efficiently, we select only the 1000 most popular words in the dictionary to model. The model looks like this

$$\hat{y} = \alpha + \sum_{w \in \text{text}} \text{count}(w) \cdot \theta_w \tag{7}$$

where $count(w)$ is the number of times a word $w$ occurs in the text body, and $\theta_w$ is the corresponding weight to be fitted.
This model achieved an astonishing $mse = 1.2923$ which makes it far better than the previously considered models. This is the last baseline we need before introducing our final model.

### 4.6 Proposed model - a combination of several elements

After making all the previous baseline models, we had a theory that if we could combine the best elements of these models, we could make a better model that could outperform them all. This is the general assumption behind the following proposed model.
Since the sentiment model worked so well, we wanted to use this as a base, and try to add to this. Combining the sentiment model with the LFM model proved not to work better than the sentiment model alone, however when including only the $\beta_i$ parameter we were able to land a solid model. The proposed model is on the form

$$\hat{y} = \alpha + \lambda\beta_i + \sum_{\omega \in W} \text{count}(w)\theta_\omega \tag{8}$$

where the weights $\beta_i$ is the same as in the baseline 3 model. The parameter $\lambda$ is a hyperparameter to experiment with. This model achieved $mse = 1.2147$ which is 0.0776 better than the sentiment model. The reasoning behind including the $\beta_i$ instead of the including the $\beta_u$ parameter is due to the small number of data per user, and large number of data per game. This model is very robust for new instances of games, since we can rely on the sentiment model. It also won't affect the model that we have not seen a user before, which happens almost every time. In cases where the user hasn't written any review text however, we are left with $\alpha + \beta_i$ which we have seen from baseline 3 performs worse than the sentiment model in baseline 5. However, this is still better than the sentiment model who is only left with $\alpha$, which could be one of the reasons this model performs better. The results are summarized in table 1, and as we can see the proposed model performs a better than the other baselines and a lot better than the first baseline.

| **Model** | **Performance (mse)** |
|---|---|
| Predicting mean | 1.7866 |
| Regression | 1.7784 |
| User, item biases | 1.6957 |
| LFM | 1.6902 |
| Text based sentiment | 1.2923 |
| Proposed model | 1.2147 |

Table 1: A comparison of the model results on test set

## 5 Conclusion

In this assignment we implemented different models taught in this course for predicting the rating of video game reviews from Amazon. By inspecting the dataset we found that there are very few reviews per user and many reviews per game. Using this knowledge, we found that the best performing models included a bias term for each game and sentiment analysis of the review text. We were then able to combine these results to build a model that outperformed the baselines. Comparing the performance of our model to the models in the papers mentioned in section 3 is tricky because the datasets are different. However, our model is still pretty simple and could be improved by incorporating more features and lateral variables.

## References


[1] Statistics on video game retail purchases. https://www.statista.com/statistics/190225/digital-and-physical-game-sales-in-the-us-since-2009/, 2021.

[2] Jianmo Ni, Jiacheng Li, and Julian McAuley. Empirical Methods in Natural Language Processing (EMNLP), 2019.

[3] Julian McAuley. *Personalized Machine Learning*. Cambridge University Press, in press.

[4] Sasank Channapragada and Ruchika Shivaswamy. Prediction of rating based on review text of Yelp reviews, 2015.

[5] Kun Luo, Meng Li, Shuaiqi Xia, and Zhenjie Lin. Prediction of yelp star rating, 2015.

[6] Zahid Younas, Zhendong Niu, Sulis Sandiwarno, and Rukundo Prince. Deep learning techniques for rating prediction: a survey of the state-of-the-art. *Artificial Intelligence Review*, 54, 01 2021.

[7] Robert M. Bell and Yehuda Koren. Lessons from the netflix prize challenge, 2007.

[8] Kaisong Song, Wei Gao, Shi Feng, Daling Wang, Kam-Fai Wong, and Chengqi Zhang. Recommendation vs sentiment analysis: A text-driven latent factor model for rating prediction with cold-start awareness. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2744–2750, 2017.