2022

Sentiment without Sentiment Analysis: Using the Recommendation Outcome of Steam Game Reviews as Sentiment Predictor

Anqi Zhang

SENTIMENT WITHOUT SENTIMENT ANALYSIS:

USING THE RECOMMENDATION OUTCOME OF STEAM GAME REVIEWS AS SENTIMENT PREDICTOR

BY

ANQI ZHANG

A thesis submitted in partial fulfillment of the requirements for the

Master of Science

Major in Computer Science

South Dakota State University

2022

THESIS ACCEPTANCE PAGE

Anqi Zhang

This thesis is approved as a creditable and independent investigation by a candidate for the master's degree and is acceptable for meeting the thesis requirements for this degree. Acceptance of this does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department.

Kaiqun Fu
Advisor Date

George Hamer
Department Head Date

Nicole Lounsbery, PhD
Director, Graduate School Date

This thesis is dedicated to any of you out there who have been told that you could not accomplish something. Stay undeterred.

ACKNOWLEDGEMENTS

Dr. Fu, you have been an amazing thesis advisor and provided endless helpful insights regarding this thesis and beyond. I appreciate you taking the time out of your busy schedule for our weekly thesis meetings; I have learned so much from each one. I cannot express my gratitude enough for your guidance and patience; I thank you sincerely.

Dr. Won, thank you for explaining and reviewing numerous Computer Science concepts to me when I first took your classes, especially the theoretically tricky ones, and even afterwards. The one phrase I will never forget you saying is “it’s easy right?” of which I will carry with me into the future.

Dr. Hamer, you were one of the most passionate and compassionate professors I have had in a long time; both traits that really elevated the energy of the whole class and made learning fun. I appreciate your witty humor and Batman is forever awesome.

I would also like to thank my two friends from school who made things less insufferable, Julie Leidholt and Mengling Ding. They are both wonderful people and I wish the best for them.

Lastly, I want to thank my significant other, Kevin Zywicki. If it was not for his unconditional love and kindness, I would not have survived until today.

TABLE OF CONTENTS

ABBREVIATIONS

AI	Artificial Intelligence
BoW	Bag of Words
CSV	Comma Separated values
DF	DataFrame
LDA	Latent Dirichlet Allocation
LR	Linear Regression
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MLPR	Multi-layer Perceptron Regression
MLR	Multilinear Regression
NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
RMSE	Root Mean Squared Error
RR	Ridge Regression
TF-IDF	Term Frequency-Inverse Document Frequency

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

SENTIMENT WITHOUT SENTIMENT ANALYSIS:

USING THE RECOMMENDATION OUTCOME OF STEAM GAME REVIEWS AS SENTIMENT PREDICTOR

ANQI ZHANG

2022

This paper presents and explores a novel way to determine the sentiment of a Steam game review based on the predicted recommendation of the review, testing different regression models on a combination of Term Frequency-Inverse Document Frequency (TF-IDF) and Latent Dirichlet Allocation (LDA) features. A dataset of Steam game reviews extracted from the Programming games genre consisting of 21 games along with other significant features such as the number of helpful likes on the recommendation, number of hours played, and others. Based on the features, they are grouped into three datasets: 1) either having keyword features only, 2) keyword features with the numerical features, and 3) numerical features only. The three datasets were trained using five different regression models: Multilinear Regression, Lasso Regression, Ridge Regression, Support Vector Regression, and Multi-layer Perceptron Regression, which were then evaluated using RMSE, MAE, and MAPE. The review recommendation was predicted from each model, and the accuracy of the predictions were measured using the different error rates. The results of this research may prove helpful in the convergence of Machine Learning and Educational Games.

CHAPTER 1. INTRODUCTION

1.1 OVERVIEW

Machine learning for NLP has been applied extensively on online text reviews for predicting factors such as the helpfulness of a review [1], success of a product based on reviews [2], popularity of a product [3], next to other factors that may influence a user's behavior and increase profitability of the product [4].

While there has been a breadth of research and analysis done on NLP aspects pertaining to online text reviews of products, there have only been a handful that focus specifically on Steam game reviews and the influence of its user recommendation system. Furthermore, there has been sparse to none research on how positively recommended Steam games based on those text reviews can be applied to game-approach learning in higher education. This research aims to predict and compare the recommendation outcomes of Steam games within the programming genre using different features of the reviews through a combination of unsupervised and supervised machine learning models.

1.2 BACKGROUND

Video games have garnered much attention and popularity over the years, across multiple platforms, accruing a massive user base. One of these platforms is called Steam, which started as a software client for distributing digital games, but has since expanded into a digital game market with social aspects, streaming services, a community hub for PC gamers, and much more. Steam has an extensive library of games ranging from a multitude of genres, also referred to as "tags". Users who have purchased games on Steam can write reviews on the platform that can also be rated by other users in number

of upvotes of “helpful” or “funny”, and may even be awarded certain titles. Additionally, the reviews are classified as either “recommended” or “not recommended” by the reviewer before the whole post goes live. An example review is as follows:

Fig 1. Screenshot of a user review of Shenzhen IO game

User reviews can be filtered by “positive” or “negative”, language, date range, and playtime. Aside from filtering the user reviews, one can also filter Steam games in general through various means. In this research, we will be using game review texts of users from top-rated “Programming” games in the Steam library.

1.3 MOTIVATION AND OBJECTIVES

Based on some studies done on learning games in academia, it seems beneficial to incorporate game-based learning in certain aspects of education [18]. In addition, for someone like me, coming from a social sciences background, specifically psychology, learning Computer Science has required me to shift my creative and analytical way of

thinking quite a bit into a more mathematical and theoretical way of problem-solving that took me a while to adapt to, and had trouble grasping in the beginning. Not only that, but many times I would not be able to apply what I learned in class to real-world problems because of the gap that exists between the theoretical knowledge and the actual implementation of that knowledge on something like creating and deploying a web application from scratch, as an example.

I have always clung onto the notion that learning should be fun, just as most games should be. That notion gave rise to an interesting thought: what if computer games could be used in higher education in tandem to classes to motivate and engage students in learning Computer Science concepts that were perceived as challenging by the student? There have been an abundant number of studies done on how using learning games/educational games or gamification in grade school through high school can be beneficial towards students' success in academics, but sparse studies have focused on using learning/educational games for higher education for academic success, specifically computer games in the field of Computer Science.

There are such computer games which can be found on the Steam platform, specifically Computer Science or Programming games. Some of those programming games provided a clear visual representation of certain Computer Science concepts, as well as immersive simulations of real-world engineering applications, which enabled me to understand what was being taught in class more thoroughly. Of course, the construct of fun can be debated, but that is not within the scope of our research. Given the time constraint and scope of our research, however, what can be discussed is whether or not the accuracy of predicting the user recommendation of programming games can be an

indicator of levels of different sentiments, which in turn may be a gateway for future research on the use of positively recommended Programming games in higher education.

Some ways we might be able to measure the accuracy is by applying and comparing different ML models to predict if a game is recommended or not on Steam based on different features of user reviews, as well as by exploring potential correlations between different groupings of features and their prediction outcomes. More explicitly, our main objectives will be to explore:

1. How accurate will the prediction of a recommended game be from Steam Programming game review text with different input features using different ML models?

2. How will different combinations of input features affect the accuracy of predicted models?

From those objectives, we may be able to explore if user recommendations of Programming games on Steam have an impact on the sentiment of the review or not.

1.4 THESIS STRUCTURE

This thesis contains a total of 6 chapters, with Chapter 1 as the introduction. In Chapter 2, some basic terminologies of this thesis are explained and related works regarding NLP and learning games are analyzed. Chapter 3 discusses the preliminary methods used and reviews some vocabulary. Chapter 4 explores data collection and processing steps, as well as feature generation and discusses models and methods utilized on the processed dataset. Chapter 5 details the experiments and results. Lastly, Chapter 6 concludes with the conclusion and future works.

CHAPTER 2. LITERATURE REVIEW

2.1 CONCEPT REVIEW

Natural Language Processing expands across many disciplines and has been an important tool in the field of AI. AI contains both ML and NLP at an intersect, but ML is divided deeper by DL, while NLP is overarching. Even though NLP technically began sometime in the 1940s, it only started gaining traction towards the 1980s and evolved in popularity at a fast pace partially due to the availability of larger amounts data, otherwise known as “big data” [5]. At the core of it all, NLP is a field of study bridging together AI and Linguistics in which machines try to understand all spheres of human language to the best of its ability and develop learning models to mimic that in order to predict speech, text, or other computations.

Moreover, NLP can be broken down into either Natural Language Understanding (NLU) or Natural Language Generation (NLG). NLU includes understanding the natural language through finding meaning or emotion through some corpus, whereas NLG creates or outputs new text, speech, or other capacities of language [6].

2.2 RELATED WORK

Many works regarding NLP have been useful in the real world in a wide range of areas. Current advances in NLP can use machines to detect spam emails, extract information from multiple sources, bring about medical advances, and create sentient-like chatbots, just to name a few [6]. A big part of NLP includes the use of sentiment analysis across diverse contexts [29].

In the context of online reviews, sentiment analysis has been routinely utilized. More precisely, there have been a several studies regarding sentiment analysis in the area of online game reviews from the Steam platform. One such study was conducted by Zuo [4] where he showed the complete process of using sentiment analysis with Gaussian Naive Bayes and Decision Tree algorithms to classify whether or not Steam game reviews resulted as negative or positive based on the distribution of analyzed words per review. From his work, we are able to compare between the accuracy results from the two classifiers. Another paper from Charkrabory et al. [10] uses other algorithms in addition to Gaussian Naive Bayes, such as Support Vector Machine, Logistic Regression, and Stochastic Gradient Descent, to evaluate the accuracies from the models. One paper that explored a variety of machine learning algorithms is Jie Ying Tan's [10] where they compared the performance of the aforementioned ML models, in addition to Multi-layer Perceptron Classifier and Extreme Gradient Boosting Classifier. It was found that their SVC model produced the best results. Even with a wide selection of classifiers and algorithms to choose from, there still exists identifiers such as sarcasm or terminology that is negative by nature, that can prevent the correct sentiment to be predicted [11]. Still, Markos et all proposes that these types of errors can be adjusted for and corrected accordingly, in their case through testing models on each genre of game.

Aside from only using the review text as an input, we were also interested in other features such as the helpfulness or funniness of a review, the recommendation criteria, and eventually the success of a game based on all those features. One measure of success may be from user text reviews, but it can also come in forms such as the number of searches, rate of being clicked on from those searches, price, genre, developer of the

game, or even video reviews of the game [2]. A game's success could also be measured in terms of popularity, which can be determined through the influence of features such as the release date, supported languages, size, in addition to the previously stated genre and price [3].

While predicting the helpfulness of online reviews of products in general is essential for roles in e-commerce, there are always concerns on how to eliminate low-quality reviews [1]. Predicting the helpfulness of a Steam game review can assist in filtering through low-quality reviews by getting rid of essentially bad reviews that users help assess [12]. This is useful because it can be difficult to differentiate between a good review that is actually helpful or a bad review that provides ineffectual information [13]. Nonetheless, Eberhard et al presented specific features that differentiate the two. A similar approach using data mining aspects also analyzed features of helpfulness to further indicate its importance was used in the research of Ha-Na et al, incorporating Classification and Regression Tree as well as Artificial Neural Network [14].

In the realm of learning games or educational games for educational purposes, there tends to be some discrepancies with respect to the total effectiveness of using educational games in the classroom setting or higher, due to user acceptance [15]. Even though students seem to be keen on using educational games for learning, with the notion that "learning with games can be fun," and that games can enhance their learning abilities, other parties such as parents or teachers may have other opinions [15]. What is interesting about this study is that the participants are information technology university students, and they are being tested on acceptance factors of specifically online educational games, which coincides with our area of research as well. Sometimes the lack

of acceptance from teachers can come from a lack of resources for the teacher to search for learning games of specific topics or grade levels [16]. Wielfrid et al proposes a solution to help teachers with selecting suitable learning games by extracting the metadata of learning games from the web and creating a catalogue that is easily accessible [17]. One tangible example of gamification in the real world, Maria's study implements educational games in job training of managing logistic projects with not only undergraduate or graduate students, but also project managers. This demonstrates that people of all ages may benefit from some form of gamification in their job or education. However, it is also crucial to note that academic institutions should not rely solely on educational games a means of overall education, but that learning games should be used as a companion in learning practices for the best outcome [18].

Looking at everything as a whole, the convergence of technology, specifically the field of ML, and education can lead to quicker advancement in online gamification techniques that can be highly beneficial for future learners of all branches of knowledge [19]. Especially in these times of the COVID-19 pandemic, many physical processes have been moved into the online environment. Based on feedback from many online users of Steam in the midst of the pandemic, Pedro reports that within 2 months, the amount of positive Steam user reviews increased by 25% [20] which marks a trend in more online users. If there are already so many active users engaged on the Steam platform that only seems to be increasing at this time, it would be wise to study what kind of an impact online learning games would have in the future.

CHAPTER 3. PRELIMINARY METHODS

3.1 Sentiment Analysis

Sentiment analysis falls under the category of NLU and is used to predict the sentiment (positive, negative, or neutral) of some text, and attempts to scale the rating from -5 to +5. Even though sentiment analysis has been used profusely in English language corpora, recent work has incorporated other languages as well, such as Hindi or Arabic [6].

3.2 LDA in Topic Modeling

Topic modeling is an unsupervised machine learning method of NLP and is important for providing an overview of what a corpus may contain very quickly, and discover any correlations between each document in a given corpus in terms of "topics". Topic modeling results in a list of words from some corpus that corresponds to some type of undetermined topic, but the words are grouped together in such a way that they are assumed to have some sort of collective qualities based on some probabilistic calculation. There can be as many lists of words as there are topics, and usually the user will set that limit [7]. In other words, topic modeling relays what topics (numerically represented) there are in a corpus after parsing through all the words and constructing a topic distribution. It is important to note that topic modeling cannot actually comprehend the meaning of each word in the corpus, which makes it different from sentiment analysis.

One of the most popular ways currently to implement topic modeling is through the Latent Dirichlet Allocation (LDA) method. LDA is an unsupervised statistical model, more specifically, a "generative probabilistic model" used frequently on texts of documents [7]. LDA first assumes that all words in a document are related in some

manner before assigning them into different topics. As an example, let us assume that we have a triangle filled with different topics (denoted as dots or points). We want the machine to differentiate between them through gauging the position of the points relative to each topic through a calculated percentage, or probability. Based on those probable topics, the amount of times a certain topic appears in each text is counted. Finally, we can compute the total number of words in each document that correlates to each topic and produce a word list per topic. Even though LDA is regularly used on text data, it can actually be applied to any type of discrete data [8], and is not only restricted to NLP. The generative steps of LDA using our chosen hyperparameters are displayed below:

1. Select θ (topic distribution for document) with some symmetric parameter of $\alpha < 1$, in our case we set our number of topics = 3

2. Select ϕ (word distribution for topic) with some sparse value of β, in our case we set our number of words = 4

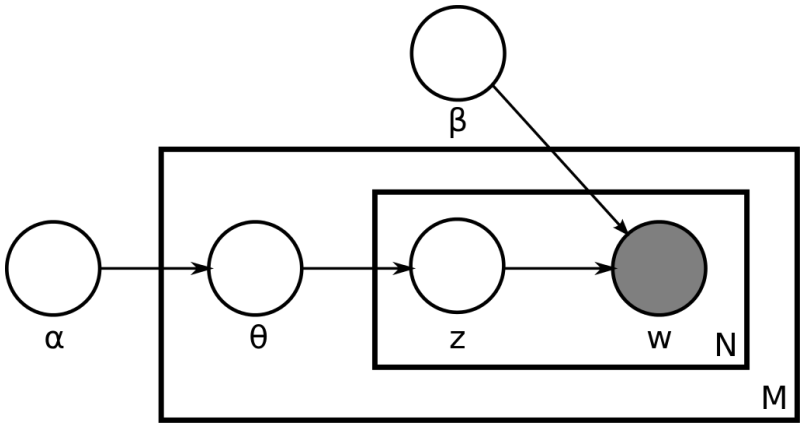

Fig 2. LDA model representation [8]

3.3 TF-IDF and BoW

Term frequency-inverse document frequency is used in NLP for not only calculating the frequency of words in a corpus, but also the importance of those words. TF-IDF can be broken down into two parts:

1. Term frequency - the frequency of a specific word relative to the text which can be measured by:
    a) The raw count (# of times)
    b) The raw count with adjustment (raw count / #words in document)
    c) The raw count scaled logarithmically
    d) Boolean count (0 for non-occurrence, 1 for occurrence)
2. Inverse document frequency - the commonness or uncommonness of a specific word relative to the text which can be measured by:
    a) log (# of documents / # of documents in which the word appears in)

Compared to TF-IDF, the Bag of Words (BoW) model does not take into consideration the importance of the words nor where they appear in a document. The purpose of the BoW model is to count only the frequency that a word occurs in a document. Informally, BoW would be considered to be just the TF part of TF-IDF [9].

CHAPTER 4. METHODOLOGIES

4.1 DATA ACQUISITION

This dataset of Steam game reviews was collected using Aesuli's open-source steam-crawler [21]. The steam-crawler scraped all games on steam during runtime, so games published past the scraped date were not included. All game data were collected from October 30, 2021 to November 4, 2021, totaling to 24,973 games. From the 24,973 games that were scraped from the Steam website, 21 games of interest were selected for the final data set. The games of interest came from the top-rated "Programming" tag games on Steam, which had a majority of positive reviews. Programming games with less than 100 reviews were not considered for this research, and it just so happened that those games had more neutral to negative reviews. Even so, it would be interesting to see differences in the use of negative reviews in future research.

The steam-crawler contains five different scripts, executed in order, although only the first four were used here. Individual game IDs were extracted from the downloaded data and saved into a CSV file from the second script. The steam-crawler's third script saved each game's data into its own folder with a unique ID, which should have corresponded to the actual game ID on the Steam website, in this format: "app-370369". However, after checking a few game data folders to ensure the correctness of the game IDs, it was found that some game IDs were actually incorrect, resulting in manual changes for each game ID of each folder to the given game ID on the Steam website. Because the game data was in HTML format, all the HTML files were merged and converted into a single CSV file using the fourth script in the steam-crawler in preparation for data preprocessing and analysis.

The game data CSV files consist of ten different features in order from leftmost column to rightmost column as follows: game ID; number of people that found the review to be useful; number of people that found the review to be funny; username of the reviewer; number of games owned by the reviewer; number of reviews written by the reviewer; 1=recommended, -1=not recommended; hours played by the reviewer on the game; date of creation of the review; text of the review.

4.2 DATA PRE-PROCESSING

In order to fully prepare the dataset for the ML pipeline, the CSV files were imported into a Jupyter Lab Notebook first for data preprocessing. Next, all CSV files of the 21 games were joined into one DataFrame (DF) to be processed using Python and the Pandas library. After that, columns that were unnecessary were dropped and the resulting columns were organized as: “Date_of_Review,” “Review_Text,” “Num_Helpful,” “Num_Funny,” “Hours_Played,” and “Recommend.”

	Date_of_Review	Review_Text	Num_Helpful	Num_Funny	Hours_Played	Recommend
0	November 1, 2021	baba is youbaba is deadbaba is funeral	0	0	16.6	1
1	November 1, 2021	Fun challenging puzzle game. cute graphics and...	0	0	7.2	1
2	November 1, 2021	baba is Good :)	0	0	10.5	1
3	November 1, 2021	this game is hard. it stresses me out and make...	0	0	1.9	1
4	October 31, 2021	super clever puzzle game! talk about thinking ...	0	0	5.2	1
5	October 31, 2021	Big fan of this game. Super fun and challengin...	0	0	32.5	1

Fig 3. First 5 rows of the DF

These features were chosen based on a variety of reasons. The date of a review may be useful in determining the popularity of a game over a certain period of time, which can show how well a game is thriving in the digital market [3]. Identifying the number of funny votes, a review received may gather the sentiment of a particular

review, and may even detect levels of sarcasm [22]. The number of hours a game was played would most likely be an indicator of how much the user enjoyed the game, unless the game was left running by accident (but then the likelihood of that user leaving a review would most likely be predictably low). The review text contains a wealth of data for tasks such as sentiment analysis or other NLP related techniques. The helpfulness of a review may be valuable for future insights regarding the likelihood of players who are about to purchase the game [13]. Of course, we would like to predict if the user will recommend a game or not based on a combination of features.

The “Review_Text” column was dropped to be processed in another step. Because we also wanted to retain the date as a usable feature in case it would be viable, we converted the date-time format into epoch time, which is just a numerical value. From there, all the numerical data was aggregated by total number of occurrences per column and grouped by the epoch date in order from earliest to latest week, and lastly, normalized. This DF was saved for use in testing ML models.

	Date_Epoch_Week	Num_Helpful	Num_Funny	Hours_Played	Recommend
0	0.000000	0.001590	0.000000	0.160009	0.065913
1	0.001852	0.000530	0.000000	0.049318	0.026365
2	0.003704	0.000000	0.000000	0.040596	0.006591
3	0.005556	0.002120	0.000000	0.002601	0.003766
4	0.007407	0.004240	0.000620	0.011025	0.001883
...	...	...	...	...	...
536	0.992593	0.114997	0.018610	0.131616	0.089454
537	0.994444	0.073132	0.050868	0.102505	0.080979
538	0.996296	0.023317	0.003722	0.106501	0.066855
539	0.998148	0.008479	0.000620	0.067836	0.056497
540	1.000000	0.000000	0.000000	0.000646	0.001883

Fig 4. Prepared DF of dataset

Before the data was normalized, however, we found a total of 20,293 text reviews starting from July 3, 2011 and ending on November 4, 2021 for the 21 games. This gives us roughly a time frame of 11 years of our Steam game review data. Presented below are some more aggregation metrics that give a broader conception of the prepared dataset:

	Games	Avg Reviews per day	Avg Reviews per week	Total Num Reviews	Number of Weeks	Start Epoch	End Epoch
2	Baba	7.73	51.86	7208	138	2566	2704
3	Exapunks	1.93	5.43	820	168	2536	2704
4	Gladiabots	1.85	3.97	464	166	2536	2702
5	HRMachine	1.68	4.14	1094	314	2389	2703
6	Infinifact	1.55	3.28	660	257	2447	2704
7	Logicbots	1.36	1.67	122	355	2333	2688
8	Mechanica	1.69	4.28	304	87	2617	2704
9	MHRD	1.37	2.1	242	251	2453	2704
10	Molek	2.21	4.8	312	104	2600	2704
11	OpusMag	1.9	7.9	600	75	2630	2705
12	PrimeMover	1.27	1.84	123	177	2524	2701
13	QuadCowboy	1.75	3.38	615	272	2429	2701
14	Screeps	1.41	3.66	934	281	2424	2705
15	SevenBil	1.73	4.06	544	166	2538	2704
16	Shenzhen	1.61	4.81	640	149	2555	2704
17	SiliconZeroes	1.5	2.54	188	202	2489	2691
18	SpaceChem	1.54	3.28	1440	539	2165	2704
19	StoneStory	1.63	2.6	179	116	2588	2704
20	TIS100	2.17	6.46	2060	335	2369	2704
21	TuringCom	7.87	48.8	244	4	2700	2704
22	WhileTrue	2.58	12.61	1500	120	2584	2704
23	**Total Count**	48.33	183.47	20293			
24	**Total Count Avg**	2.301428571	8.736666667	966.3333333			

Fig 5. Aggregated metrics in details about the 21 games

The previously dropped “Review_Text” column was then used for text cleaning and formatting in this step. All duplicate rows and n/a rows were dropped, as well as rows with text that was in another language or text containing special characters. The review text was saved as a JSON file for further processing.

```
Date_Epoch_Week
2165    I played this game enough that it invaded my d...
2166    A unique puzzler that allows for creativity. T...
2167    This is the greatest puzzle game... of all tim...
2168    A quite difficult but satisfying production li...
2169    Easily the best Puzzle game I've ever played.O...
```

Fig 6. List of text reviews

4.3 FEATURE GENERATION

The previous JSON file containing the text reviews was loaded into a new Jupyter Lab Notebook for advanced text cleaning and feature engineering. Lemmatization and

tokenization steps were applied to the full corpus and parsed through using the spaCy library along with Part of speech tagging to include nouns, adjectives, verbs, and adverbs. In addition, text was also converted into all lower-case letters and digits and punctuation were removed. Using Genism's TF-IDF model, frequently used words were removed from the corpus that were not descriptive of the game itself to account for overfitting the data, followed by building our own Bag of Words (BoW) model to process the rest of the text. The removal of stopwords was taken into consideration, but deemed unnecessary due to previous text cleaning procedures which included some sort of removal of commonly used non-descriptive words.

Topic modeling was used to group the most frequently used words from all game reviews in each epoch week. This was accomplished through utilizing the LDA model, also from the Genism library, to the processed text list. After experimenting with different sets of parameters, we settled upon 3 number of topics, chunksize to be 100, and passes to be 10. From there, we limited the number of words per game to be the 4 most frequently occurring words, for a total of 88 total keywords as features, after duplicate keywords were removed. The list of keywords was further vectorized and stored into a NumPy array in preparation for testing supervised ML models. We were careful to calculate the correct number of rows to match the rows of the DF from our data preparation step. From our current dataset, we ended up with 541 rows, 88 columns for the keyword features, and 5 columns for the numerical DF features.

Besides the feature words list that we generated from all 21 games, we also wanted to see how each game individually differed from each other by seeing what types of words they would produce for various topics and check for distinct nuances. The same

parameters were used to run the LDA model for each game separately, and visualized with pyLDAvis and WordClouds. Below is a set of WordClouds of 3 topics from the game Moleksyntez:

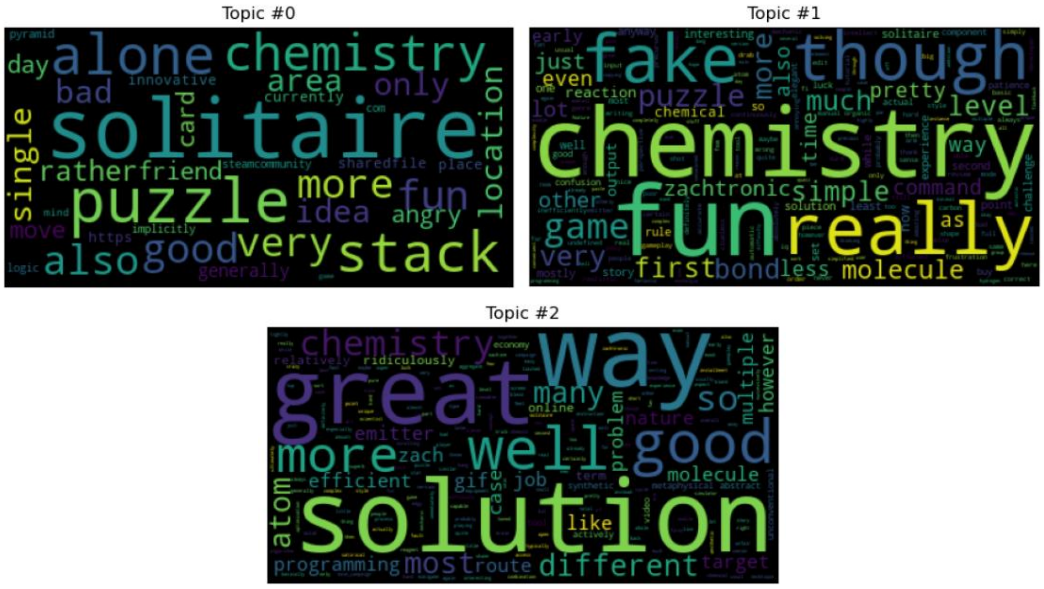

Fig 7. WordClouds of Moleksyntez game

We also experimented with using bigrams and trigrams per the game for comparison purposes. Below is a set of WordClouds of 3 topics from the game Mechanica characterizing some bigrams:

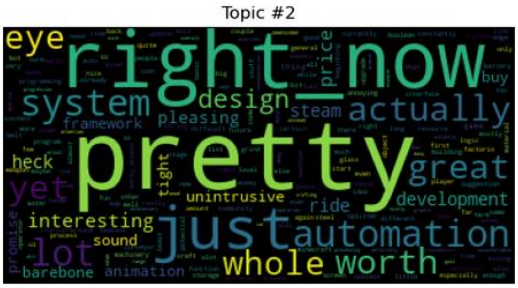

Fig 8. WordClouds of Mechanica game

While the outcome was interesting, we continued with unigrams for more coherence and less complexity.

4.4 MODELS

In a new Jupyter Lab Notebook, we set the keywords as the titles of each column of the NumPy array, and then merged the array with the numerical features from the first DF. The combined table of final features came out to be 541 rows by 93 columns, as shown below:

	Date_Epoch_Week	Num_Helpful	Num_Funny	Hours_Played	Recommend	node	expectation	circuit	optional	level	...	baba	gate	robot	tutorial	more	computer	message	great	many	also
0	1.000000	0.000000	0.000000	0.000646	0.001883	0.0	0.0	0.0	0.0	10.0	...	0.0	0.0	0.0	1.0	5.0	1.0	0.0	4.0	3.0	5.0
1	0.998148	0.008479	0.000620	0.067836	0.056497	0.0	0.0	0.0	0.0	3.0	...	0.0	0.0	0.0	0.0	6.0	0.0	0.0	3.0	2.0	2.0
2	0.996296	0.023317	0.003722	0.106501	0.066855	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	2.0
3	0.994444	0.073132	0.050868	0.102505	0.080979	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	1.0	3.0	0.0	0.0	0.0	2.0	0.0
4	0.992593	0.114997	0.018610	0.131616	0.089454	0.0	0.0	0.0	0.0	2.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
536	0.007407	0.004240	0.000620	0.011025	0.001883	2.0	0.0	18.0	0.0	44.0	...	0.0	17.0	2.0	4.0	22.0	49.0	0.0	12.0	3.0	18.0
537	0.005556	0.002120	0.000000	0.002601	0.003766	0.0	0.0	7.0	1.0	21.0	...	0.0	15.0	0.0	1.0	21.0	24.0	0.0	13.0	7.0	12.0
538	0.003704	0.000000	0.000000	0.040596	0.006591	0.0	0.0	2.0	0.0	11.0	...	1.0	5.0	0.0	5.0	17.0	19.0	0.0	12.0	2.0	10.0
539	0.001852	0.000530	0.000000	0.049318	0.026365	0.0	0.0	2.0	0.0	7.0	...	1.0	3.0	0.0	0.0	5.0	12.0	0.0	11.0	1.0	8.0
540	0.000000	0.001590	0.000000	0.160009	0.065913	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0

541 rows × 93 columns

Fig 9. Combined final features

From there, we categorized the input features into 3 different groups for modeling:

1. Combined features = 92 features

2. Keyword-only features = 88 features

3. Non-keyword features = 4 features

Even though “Date_Epoch_Week” was initially included, after some testing we decided to omit “Date_Epoch_Week” as a feature for this study.

We decided upon the following 5 supervised learning algorithms to compare the accuracies of the recommendation predictions:

1. **Multilinear Regression**, which can be compacted to this equation [23]:

   Y = XB + U, where X is a matrix filled with independent variables, Y is a matrix with the dependent variable(s), and U is a matrix of errors

2. **LASSO Regression**, which wants to minimize the sum of squares with the constraint of $\Sigma |B_j \leq s$ [24]:

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Fig 10. LASSO regression minimization equation

3. **Ridge Regression**, which can be compacted into this base equation:

   Y = XB + e, where X is a matrix filled with independent variables, Y is a matrix with the dependent variable(s), and e is the residual error

4. **Support Vector Regression**, which can be defined with the parameters of $C > 0$ and $\varepsilon > 0$, with the base form of [25]:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^*} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{l}\xi_i + C\sum_{i=1}^{l}\xi_i^* \\ \text{subject to} \quad & \boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b - z_i \leq \epsilon + \xi_i, \\ & z_i - \boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b \leq \epsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \geq 0, i = 1, \ldots, l. \end{aligned}$$

Fig 11. SVR base equation

5. **MLP Regression**, which uses a multi-layer perceptron that contains a set of nodes on the left as the input features, followed by one or more hidden layers in order that will output values to optimize errors [26].

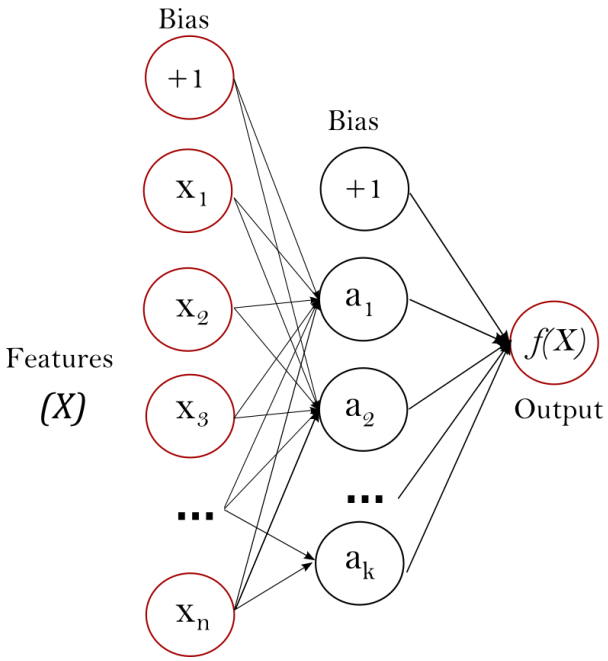

Fig 12. Example of hidden layer [26]

The accuracies will be based on three different types of error rate metrics:

1. RMSE - root mean squared error

2. MAE - mean absolute error

3. MAPE - mean absolute percentage error

4.4.1 MULTILINEAR REGRESSION

As with any regression model, it can be assumed that there is a relationship between some dependent and independent variable, and the primary purpose should be to minimize some type of error measurement. Linear Regression (LR) is a commonly used regression model to show a relationship between two variables, where the more data, the better and more accurate the results. Then, a line of best fit with the least variance, which is a line that results in the minimum sum of squared residuals, is created to visualize the approximation of values. In our case, because we have more than just one variable, we required a more complex version of LR, notably Multilinear Regression (MLR) which makes use of multiple dependent variables.

4.4.2 LASSO REGRESSION

The next method we chose was LASSO regression, which is an extension of LR. LASSO regression is beneficial to use when overfitting occurs, which can be mitigated through regularization. In regularization, the smaller the coefficient, the better for minimizing the loss function. In the case of LASSO regression, the coefficients can ultimately be decreased to zero for improving the performance and reducing variance of the model.

4.4.3 RIDGE REGRESSION

Another extension of LR is Ridge Regression (RR), which is useful for smaller sample sizes. This was a good fit for our dataset, as our dataset of user reviews were

combined for all games instead of separately manipulated, resulting in a more compact dataset. RR also puts to use shrinking the coefficient values for regularization purposes to circumvent overfitting; however, it defers from LASSO regression in that the coefficients will only ever get close to zero at some point, but not necessarily reach zero explicitly.

4.4.4 SUPPORT VECTOR REGRESSION

While the previous types of regression algorithms are all about reducing error rates, support vector regression allows us to actually customize the error range that would be acceptable for our data and then fit the most optimal line within that range [27].

4.4.5 MULTI-LAYER PERCEPTRON REGRESSION

A more complex regression algorithm is the Multi-layer Perceptron Regression (MLPR) model which utilizes artificial neural networks. What separates MLPR from a normal regression model such as LR or MLR, is the addition of any number of hidden layers between the input and output layer [28]. We tried different amounts of hidden layers from 10 to 100 in increments of 10.

CHAPTER 5. EXPERIMENTS AND RESULTS

5.1 SETUP AND PROCEDURES

In the same Jupyter Notebook in which we loaded the DF, we also tested the different models there. A Jupyter Notebook is an interactive development environment that can be accessed via any web browser and allows for modularity of running code.

5.1.1 ENVIRONMENT, LIBRARIES, AND DEPENDENCIES

The libraries and versions used throughout the whole experiment include:

1. Python 3.9.12
2. Pandas 1.4.2 – data analysis tool with simple to use data structures
3. NumPy 1.22.3 – processing of arrays for objects, numbers, etc.
4. Gensim 4.2.0 – LDA and TF-IDF tools
5. Wordcloud 1.8.1 – wordcloud generation
6. Scikit-learn (sklearn) 0.24.2 – machine learning library using Python
    a) MinMaxScaler
    b) mean_squared_error
    c) mean_absolute_error
    d) mean_absolute_percentage
    e) LinearRegression
    f) Lasso
    g) Ridge
    h) SVR
    i) MLPRegressor

5.1.2 HARDWARE SPECIFICATIONS

The four most important factors that may influence the performance of the models include the processor (CPU), video card (GPU), memory (RAM), and storage (Drives). These are the hardware specifications of the machine that I used, but much of the code can still be run with some lower specifications without any negative implications:

1. CPU: Processor 11th Gen Intel(R) Core(TM) i7-11700K @ 3.60GHz, 3600 Mhz, 8 Core(s), 16 Logical Processor(s)
2. GPU: NVIDIA GeForce RTX 3080
3. RAM: 16.0 GB
4. Storage: 1 TB

5.2 TRAINING PARAMETERS

The Scikit-learn library allows us to tune some hyperparameters per regression model, although some parameters may vary. Overall, the weights or coefficients are usually the changeable values in each model. The most critical hyperparameter that can be tuned for LASSO regression on sklearn is the alpha value, or the coefficient that controls regularization. In our experiment, we tried alpha values of 0.3, 0.2, and 0.1. RR in sklearn also uses an alpha value. We tried the same alpha values of 0.3, 0.2, and 0.1. For SVR, we chose ‘rbf’ as the kernel, tested the C values (regularization parameter) from 0 to 10, and chose different epsilon values from 0.0001 to 0.1 in increments of base 10. For MLPR, we tried different amounts of hidden layers from 10 to 100 in increments of 10.

5.3 RESULTS AND ANALYSIS

Table 1. Error Comparisons of Recommendation Prediction Based on Different Feature Groups

Method	Combined Features			Keywords Only			Non-Keywords		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
MLR	0.0334	0.0179	3.3087	3.1213e-16	2.3504e-16	5.6705e-14	0.0326	0.0149	1.9385
LASSO	0.0765	0.0364	7.2924	0.0765	0.0364	7.2924	0.0765	0.0364	7.2924
RIDGE	0.0334	0.0179	3.3087	4.1584e-16	2.8446e-16	9.8450e-14	0.0326	0.0149	1.9385
SVR	0.0381	0.0206	3.8895	0.0065	0.0039	1.1203	0.0577	0.0151	0.5431
MLPR	0.0545	0.0260	5.9208	0.0585	0.0399	9.6416	0.0455	0.0140	0.8115

The table above shows the comparisons between the 3 different error rates measurements of the 5 regression models depending on the feature groups. Based on the above results, we can break down the comparisons for each group in more detail. In general, however, the smaller the error values are, the more accurate the predictions will be. Out of all the hyperparameters we tested, the predicted results are the most accurate based on optimal hyperparameter combinations.

From the results, we find that the Keywords only feature produced the smallest rates of errors across all 3 error metrics, with MAE being the lowest at 2.3504e-16 from MLR. This outcome could be attributed to MLR being the simplest regression model without added complexities. It seems that as complexity increases, the performance of the model decreases, especially with MLPR. This may be due to MLPR, being a neural network design, is not a linear regression model and perhaps contains too many unnecessary features that induces more error. RR using the Keywords only feature seemed to be the next best predictor, while LASSO regression demonstrated to be the worst on the average across all errors. This may be explained because of our small dataset size, which RR excels at. Predominately, the regression models we applied were able to correctly predict recommendations of Steam game reviews to a great degree, which could indicate a positive sentiment for recommended reviews, and negative sentiment for non-recommended reviews.

CHAPTER 6. CONCLUSION AND FUTURE WORK

Although we were able to predict the recommendations of Steam Programming game reviews to a high degree of accuracy, the feasibility of using Steam games in higher education can endlessly discussed. Nonetheless, there are many ways of incorporating gamification into learning without an added cost to the academic institution. Some developers of the Programming games on Steam are willing to provide digital copies of their games for free for educational purposes if the school requests them. One such developer is Zachtronics, who created the game Shenzhen IO, which is one of the Programming games from our dataset. It would be interesting to see future research on measuring how effective Steam Programming games for learning topics in Computer Science can be, based on highly recommended games.

Another interesting factor future studies can delve into is to use sentiment analysis together with the features we have provided and discussed to compare the results of both methods. In doing so, the differences or similarities in sentiment prediction may be analyzed, for if a text review is predicted to be recommended, then the sentiment should also be positive. One other facet we could explore is using the game images as input features instead of only text or numerical data, and mix in image classification methods. On a more general level, we could also compare game review text recommendations from different type of platforms, not just limited to the Steam platform.

Some limitations of our study included the scope of our research as well as time and resources. The Steam game reviews data we gathered were only from a certain period of time, and there will always be an influx of new games uploaded to the Steam store with new reviews added weekly or even daily. It would be interesting to be able to map

those changes in real time, along with further analysis and predictions of Steam game recommendations with the ebb and flow of time. Another limitation includes the size of our dataset. We could always increase the dataset in the future to include more than just Programming games on Steam and even apply more complex deep learning algorithms on them.

If we wanted to increase accuracy and reduce the error rate even more in future experiments, we could fine-tune the regression models through using other methods such as sklearn's GridSearchCV module, which would find the most optimal coefficients and values to use for the hyperparameters automatically. This could potentially save time as well. All in all, this is only an initial stride in gathering sentiment through successful Steam game recommendation predictions, and there are plenty of questions still to be considered and answered.

REFERENCES

[1] Park, Y. J. (2018). Predicting the helpfulness of online customer reviews across different product types. *Sustainability (Switzerland)*, *10*(6). https://doi.org/10.3390/su10061735

[2] Trněný, M. (2017). *Machine Learning for Predicting Success of Video Games*. 1–68.

[3] de Luisa, A., Hartman, J., Nabergoj, D., Pahor, S., Rus, M., Stevanoski, B., Demšar, J., & Štrumbelj, E. (2021). *Predicting the Popularity of Games on Steam*. http://arxiv.org/abs/2110.02896

[4] Zuo, Z. (2018). Sentiment Analysis of Steam Review Datasets using Naive Bayes and Decision Tree Classifier. *Student Publications and Research - Information Sciences*. https://analytics.twitter.com

model size," *arXiv preprint arXiv:1602.07360*, 2016.

[5] Hirschberg, J., & Manning, C. D. (2015). *Advances in natural language processing*. http://science.sciencemag.org/

64270-64277, 2018.

[6] Khurana, D., Koli, A., Khatter, K., Singh, S., & Tools, M. (2022). *Natural language processing: state of the art, current trends and challenges*. https://doi.org/10.1007/s11042-022-13428-4

[7] Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., & Zhao, L. (2017). *Latent Dirichlet Allocation (LDA) and Topic modeling: models, applications, a survey*. http://arxiv.org/abs/1711.04305

[8] Blei, D. M., Ng, A. Y., & Edu, J. B. (2003). Latent Dirichlet Allocation Michael I. Jordan. *Journal of Machine Learning Research*, *3*, 993–1022.

[9] Stephen, A., Lubem, T., & Adom, I. T. (2022). Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets Implicit Feedback System for the Recommendation of Relevant Web Documents View project Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets View project. *Article in International Journal of Information Technology*. https://doi.org/10.1007/s41870-022-01096-4

[10] Tan, J. Y., Sai, A., Chow, K., & Tan, C. W. (2021). *SENTIMENT ANALYSIS ON GAME REVIEWS: A COMPARATIVE STUDY OF MACHINE LEARNING APPROACHES*.

[11] Viggiato, M., Lin, D., Hindle, A., & Bezemer, P. (2020). *What Causes Wrong Sentiment Classifications of Game Reviews?* https://github.com/asgaardlab/sentiment-analysis-Steam

[12] Wang, Z., Jiaotong, an, Chang, V., & Horvath, G. (2021). Explaining and Predicting Helpfulness and Funniness of Online Reviews on the Steam Platform. *Journal of Global Information Management*, *29*(6). https://doi.org/10.4018/JGIM.20211101.oa16

[13] Eberhard, L., Kasper, P., Koncar, P., & Gutl, C. (2018). Investigating Helpfulness of Video Game Reviews on the Steam Platform. *2018 5th International Conference on Social Networks Analysis, Management and Security, SNAMS 2018*, 43–50. https://doi.org/10.1109/SNAMS.2018.8554542

[14] Kang, H.-N. (2017). A Study of Analyzing on Online Game Reviews Using a Data Mining Approach: STEAM Community Data. *International Journal of Innovation, Management and Technology*, *8*(2), 90–94. https://doi.org/10.18178/ijimt.2017.8.2.709

[15] Ibrahim, R., Masrom, S., Yusoff, R. C. M., Zainuddin, N. M. M., & Rizman, Z. I. (2018). Student acceptance of educational games in higher education. *Journal of Fundamental and Applied Sciences*, *9*(3S), 809. https://doi.org/10.4314/jfas.v9i3s.62

[16] Morie, M. W., Marfisi-Schottman, I., & Goore, B. T. (2020). LGMD: Optimal Lightweight Metadata Model for Indexing Learning Games. *Communications in Computer and Information Science*, *1207 CCIS*, 3–16. https://doi.org/10.1007/978-3-030-45183-7_1

[17] Wielfrid, M. M., Iza, M. S., & Tra, G. B. (2020). Information extraction model to improve learning game metadata indexing. *Ingenierie Des Systemes d'Information*, *25*(1), 11–19. https://doi.org/10.18280/isi.250102

[18] Vodenicharova, M. (2022). Gamed-based Learning in Higher Education. *TEM Journal*, *11*(2), 779–790. https://doi.org/10.18421/TEM112-35

[19] Khakpour, A., & Colomo-Palacios, R. (2021). Convergence of Gamification and Machine Learning: A Systematic Literature Review. *Technology, Knowledge and Learning*, *26*(3), 597–636. https://doi.org/10.1007/s10758-020-09456-4

[20] Nuno, P., & Mota, Â. (2021). *ASSESSING COVID-19 IMPACT ON USER OPINION TOWARDS VIDEOGAMES SENTIMENT ANALYSIS AND STRUCTURAL BREAK DETECTION ON STEAM DATA*.

[21] Esuli, A. (2020). *aesuli/steam-crawler: A set of scripts that crawls STEAM website to download game reviews.* https://github.com/aesuli/steam-crawler

[22] Bais, R., Odek, P., & Ou, S. (2017). *Sentiment Classification on Steam Reviews*. 1–6.

[23] Gao, X., Rangarajan, A., Banerjee, A., Su, Y., Member, S., Li, X., & Dacheng Tao, and. (2012). Multivariate Multilinear Regression Related papers A Met hod for Compact Image Represent at ion Using Sparse Mat rix and Tensor Project ions O… Multivariate Multilinear Regression. *CYBERNETICS*, *42*(6). https://doi.org/10.1109/TSMCB.2012.2195171

[24] Ranstam, J., & Cook, J. A. (2018). *Statistical nugget LASSO regression*. https://doi.org/10.1002/bjs.10895

[25] Chang, C.-C., & Lin, C.-J. (2001). *LIBSVM: A Library for Support Vector Machines*. www.csie.ntu.edu.tw/

[26] Pedregosa FABIANPEDREGOSA, F., Michel, V., Grisel OLIVIERGRISEL, O., Blondel, M., Prettenhofer, P., Weiss, R., Vanderplas, J., Cournapeau, D., Pedregosa, F., Varoquaux, G., Gramfort, A., Thirion, B., Grisel, O., Dubourg, V., Passos, A., Brucher, M., Perrot andÉdouardand, M., Duchesnay, andÉdouard, & Duchesnay EDOUARDDUCHESNAY, Fré. (2011). Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot. *Journal of Machine Learning Research*, *12*, 2825–2830. http://scikit-learn.sourceforge.net.

[27] Smola, A. J., Sch¨olkopf, B., & Sch¨olkopf, S. (2004). A tutorial on support vector regression *. *Statistics and Computing*, *14*, 199–222.

[28] Gaudart, J., Giusiano, B., & Huiart, L. (2004). Comparison of the performance of multi-layer perceptron and linear regression for epidemiological data. *Computational Statistics & Data Analysis*, *44*(4), 547–570. https://doi.org/10.1016/S0167-9473(02)00257-8

[29] Chowdhury, G. G. (2003). Natural language processing. *Annual Review of Information Science and Technology*, *37*, 51–89. http://eprints.cdlr.strath.ac.uk/2611/