Project Report on

Video Games Data Mining and Analysis

Team:
Under Prof. Jan-Willem can de Meent
Done by: Sumeet Dubey, Harshdeep Singh, Zhiguang Yu

Introduction:

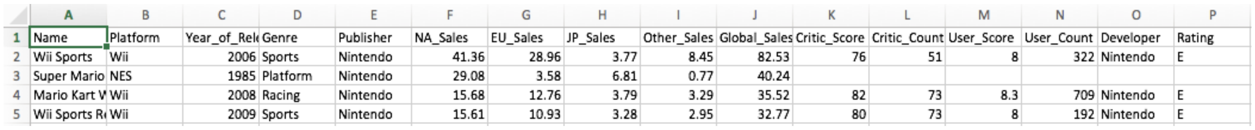
In this project, we have attempted to perform data mining analysis on a Video Games review dataset. For the first part, we do some exploratory analysis on the dataset and make conclusions based on that. Then we run a few data mining algorithms and do a comparative study.

Motivation

We decided to work on this dataset because it allows us to implement some interesting data mining techniques learnt during this course; and also identify some trends in the gaming industry. Video game industry has been around only for a few decades that makes it fairly new compared to many other industries. It is also a vastly growing industry with multiple consoles releasing every year, that in turn provide new games. It is also interesting to analyze the sales of top publishers in this industry and understand a little bit more about their trends.

Dataset

The dataset we are using can be divided into two parts. First we have a csv file consisting of a comprehensive list of all video games released along with certain parameters like global sales, critic score, genre, etc. We do our exploratory analysis on this dataset.



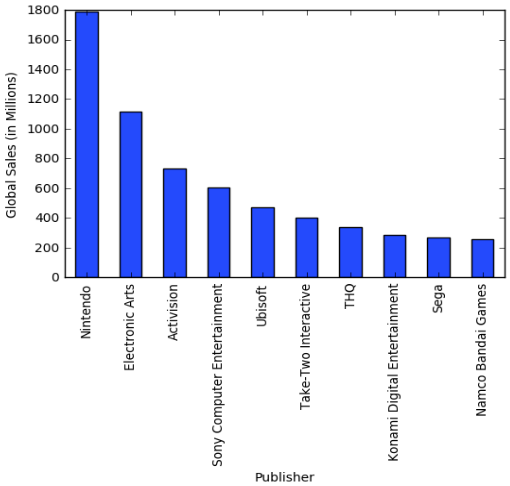
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Name	Platform	Year_of_Rel	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critic_Score	Critic_Count	User_Score	User_Count	Developer	Rating
2	Wii Sports	Wii	2006	Sports	Nintendo	41.36	28.96	3.77	8.45	82.53	76	51	8	322	Nintendo	E
3	Super Mario	NES	1985	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24						
4	Mario Kart V	Wii	2008	Racing	Nintendo	15.68	12.76	3.79	3.29	35.52	82	73	8.3	709	Nintendo	E
5	Wii Sports R	Wii	2009	Sports	Nintendo	15.61	10.93	3.28	2.95	32.77	80	73	8	192	Nintendo	E

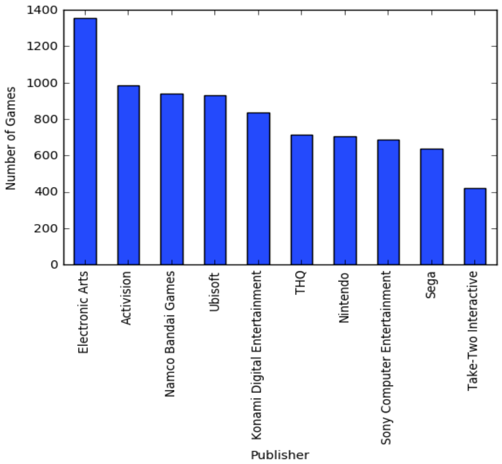
In the second part, we have games along with their written reviews by game critics from Gamespot.com. These written reviews give us a text based dataset and we have tried to implement topic modeling on it. The text reviews also have a score and some other data like publisher of the game, genre, year of release, etc. We perform topic modeling using Latent Semantic Analysis (LSA), Latent Dirichlet allocation (LDA), supervised SLDA and Binary logistic SLDA. We then do a comparative study on the results.

Exploratory Analysis:

Top Gaming Companies:

One of the first things we did was finding companies that contribute most to the industry. We used total number of games released and total global sales as two different evaluation criteria.





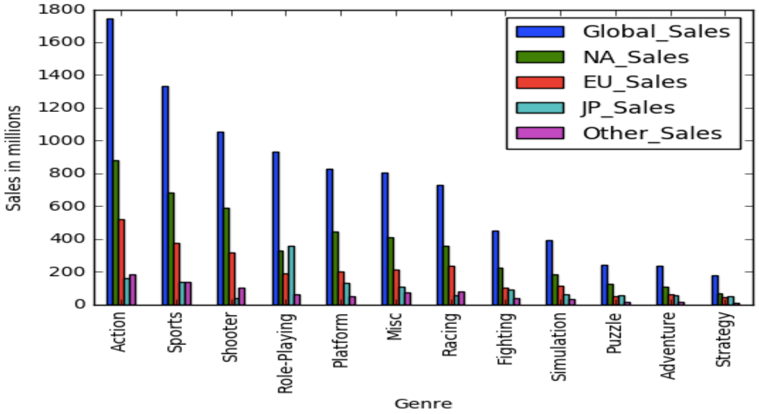
We can observe that the top companies in both our evaluations are the same. Most of these companies have been a part of the industry since a long time. The two biggest companies, Nintendo and Electronic Arts account for around 46% of the total sales.

The second graph looks more balanced that the first one. Game sales seem to skew towards the top companies while production is done in a more balanced sense. This skewness for top publishers might be caused by multiple factors like an already established fan base, quality of games, number of games released, budget, publicity, etc.

Interestingly, Nintendo has produced about half as many games as EA and yet it tops global sales by a good margin. Activision and Namco Bandai have released almost same number of games, but there is a considerable difference in their sales.

Popular Genres:

We have 12 genres in our dataset. To find the most popular ones, we summed them over the total global and regional sales and plotted them as a bar graph.

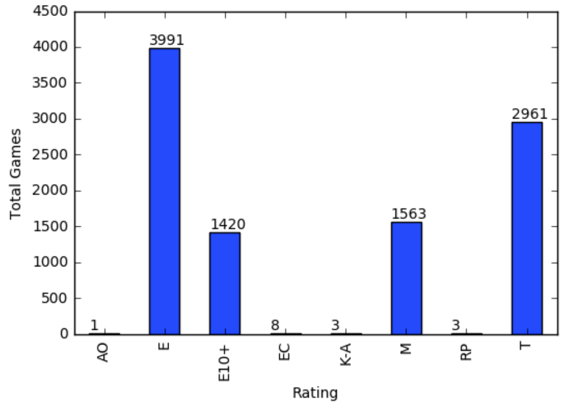


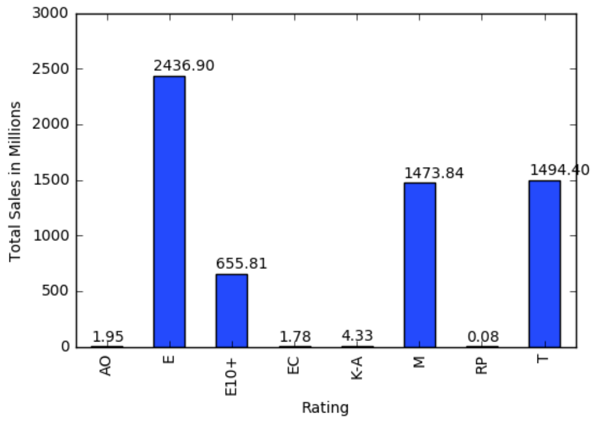
We can see that Action games have dominated the gaming industry for a long time and have the maximum number of sales. This is expected mainly for two reasons. Firstly, many games can be classified under action because it is more like a superset of other genres (basically games that offer physical challenges and multiple levels). Secondly, action games have been around almost since the start of this industry.
Sales in Japan though for action games is the lowest compared to other places. But we can see a nice bump in Japan sales for Role-Playing games (in-fact the highest). We could say that the Japanese gaming industry prefers story-telling and role oriented games than genres like Action and Shooters.

Game Rating Distribution

The two bar chats below show the number of games released grouped by their ESRB ratings, and their total sales. ERSB ratings are given according to the content of the game to filter the users suitable for playing it. The popular ones are:

- E – Everyone
- E10+ - Everyone above 10
- T – Teen
- M - Mature





Games rated E are the most produced and sold games. This is expected as many users of the gaming industry are kids and there are many genres that do not require unnecessary explicit content (sports, strategy, puzzles). Rated-M games have great sales considering their numbers. Many of the Action and Role-playing games are often rated M and hence they are popular. Equally popular are Teen rated games, though their numbers are higher.
Note that only one game in our dataset has the AO rating (Adults Only). Interestinly it can be noted it the second graph that this game generated sales of $1.95 million. It is the popular and critically aclaimed game by Take Two Interactive and Rockstar, Grand Theft Auto: San Andreas.

Sales Distribution

We found the top 5 publishers in the database based on global sales and then the sales distributions for each genre across North America, Europe, Japan and Rest of the world.
North America is the biggest market for platform and strategy games, with Electronic arts and Nintendo being the major publishers. Nintendo has done better sales in Japan as compared to the other regions. Also, it can be observed that Nintendo has somewhat consistent sales for almost all its genres whereas other publishers are a bit inconsistent, having sales in only some specific genres. Though overall North America and Europe are the main contributors towards video games sales.

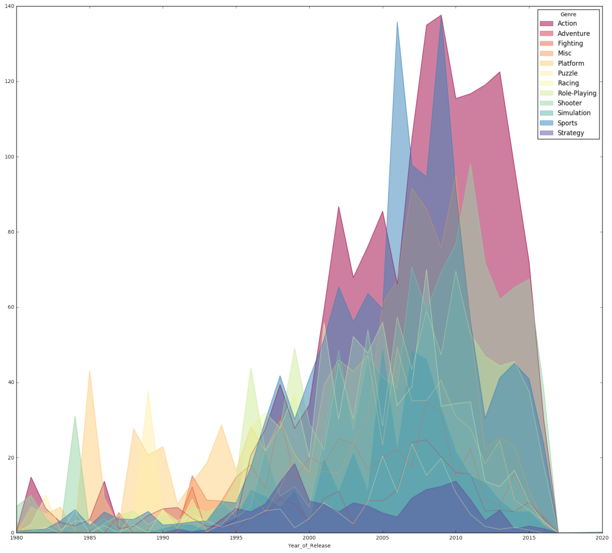
For different regions, we also plotted sale trends over the years for each of the top 5 publishers based on global sales. We have considered buckets of 5 years, and tried to plot total sales for each of those buckets in North America, Europe, Japan and Rest of the world.

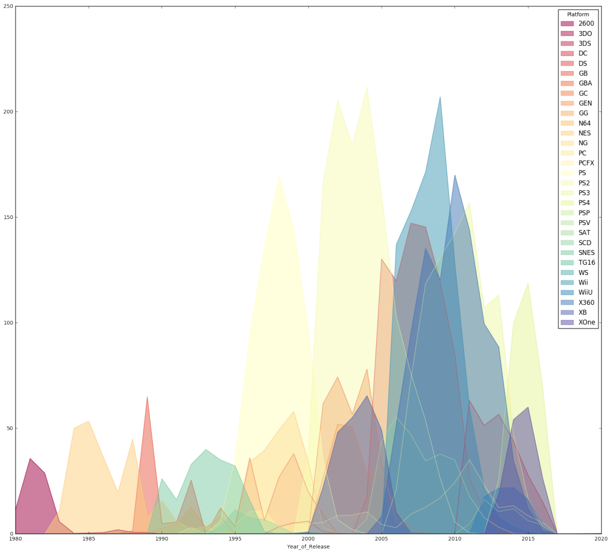
We can make some observations about sale trends from these plots. Note that there has been a sharp rise in sales for Nintendo for all regions in the period 2005-2009. This is mainly due to the release of Wii in 2006, that turned out to be one of the most selling consoles ever. Nintendo also is one of the oldest publishers in the industry, accompanied by Activision, followed by the other three publishers that came later into the market.

Sony Computer Entertainment had good sales in first 10 years (their golden era of PlayStation 1 and 2) but it has seen a continuous decline since. Also note the plot for sales in Japan, showing that Nintendo has performed exceptionally well in Japan compared to other publishers, strengthening our earlier conclusion.

Most Selling Games by Genres and Consoles, over the years:

Trending genres and consoles over the years can be deduced by observing the following two graphs.





We can see that in the early years, the sales are dominated mostly by Action games, shooters and platformers. It should be noted how the sales have increased since the 2000's. This is the time when Sony and Microsoft came up with their flagship consoles, Xbox and the PlayStation.

Data Mining Analysis:

Here we discuss data mining algorithms that we have run on our dataset of written game reviews. They are as follows:

Latent Semantic Analysis (LSA)

Latent Semantic Analysis is a technique to generate concepts from a matrix of documents and terms (in those documents), where the generated concepts help to analyze the relationship between the document and terms. LSA uses mathematical technique called Singular Value Decomposition (SVD) to reduce the dimensionality of the matrix and it returns three matrices U, S & V^T.

Suppose we have a word-document matrix X, then it uses SVD on X to give

$$X = U S V^T \text{ (approx.)}$$

where X is a vxd word-document matrix, U is vxk word-concept matrix, S is kxk concept matrix & V is dxk document-concept matrix. Here v is vocabulary size or number of terms in the given data set and d is the number of documents in the dataset.

Algorithm: This sections explains the steps we performed to do LSA on our dataset. Our dataset consists of Gamespot reviews for PC, DS, PS3, Xbox360 and Wii consoles. Total number of documents are about 4300. We have used **Sci-kit learn** libraries for performing Tfidf and SVD(TruncatedSVD).

1. Read all the text data (Gamespot review only) from each file in the data set, convert to lower case and insert it into a list, name it as "_data".
2. Perform TFIDF on the list prepared in the above step by passing it to *Tfidfvectorizer function*. Stop words removal can be done by passing a list of stop words to *TfidfVectorizer* function. This is the only data preprocessing that we perform to our data.
3. Once we perform step 2, we use fit_*transform* method and pass list of text "_data" to it. This returns a (n_features, n_samples) matrix. In other words, it returns a vxd matrix where v is number of words in dataset and d is the number of documents in the dataset.
4. Now pass the matrix obtained in step 3 to the function that uses *TruncatedSVD* function to perform SVD. We use *TruncatedSVD* because it does not center the data before computing and therefore can be used with sparse matrices. You can pass the number of components (concepts) you want to extract and the number of iteration you want to perform (default is 100) along with the type of algorithm you want to use.
5. We use *TruncatedSVD .fit* method and pass our matrix X (words- documents) to it. It returns a kxv matrix which is nothing but component by term matrix.
6. We then use *TruncatedSVD .transform* method and pass our X matrix to it, and it returns a dxk-matrix , which is a document by component matrix.

Reason for taking document by component matrix: We are taking a document(game) by component matrix in last step of LSA because we use this matrix later to calculate similarity scores between each document (games) based on LSA model and then we show top 5 similar games for a given game. We then later compare LSA and LDA models, based on the top 5 similar documents.

Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation is a generative probabilistic model for discrete data such as text data. LDA basically considers a document to be a mixture of hidden (latent) topics, where each topic is characterized by a distribution over words. LDA is identical to probabilistic latent semantic analysis(pLSA), except that LDA topic distribution is assumed to have sparse Dirichlet prior.

LDA uses following generative process:

1. Choose $\Theta_i \sim \mathrm{Dir}(\alpha)$, where $i \in \{1\ldots, M\}$ and $\mathrm{Dir}(\alpha)$ is a Dirichlet distribution with symmetric parameter α which typically is sparse ($\alpha < 1$).
2. Choose $\phi_k \sim \mathrm{Dir}(\beta)$, where $k \in \{1\ldots, K\}$ and β is typically sparse
3. For each of the words in position i,j where $j \in \{1\ldots, N_i\}$ and $I \in \{1\ldots, M\}$.
    a) Choose topic $z_{i,j} \sim$ Multinomial (Θ_i)
    b) Choose topic $w_{i,j} \sim$ Multinomial $(\phi\ z_{i,j})$

where α is the parameter of the Dirichlet prior on the per-document topic distributions, β is the parameter of the Dirichlet prior on the per-topic word distribution, Θ_i is the topic distribution for document m, ϕ_k is the word distribution for topic k, $z_{i,j}$ is the topic for the j-th word in document i, and $w_{i,\,j}$ is the specific word.

Supervised LDA and Binary Logistic SLDA

SLDA/BLSLDA is a supervised variant of the LDA topic modeling algorithm. The supervised version uses two extra parameters in addition to the LDA parameters. These help to learn about an external variable in relation to topics. The two parameters are a regression coefficient and a response value for each document. For this algorithm, we have considered ~4300 documents.

Preprocessing step: Convert all the reviews into word count vectors and convert game review scores to numeric type.

Algorithm: This algorithm works basically in two steps. We train the model first with data-points (fit) and estimate the model parameters. We then use the trained model to predict scores for new data-points (transform)

1. Divide dataset into train and test parts (we are using 90:10 ratio)
2. Supply the initial parameters and run SLDA/BLSDA. We have observed that the log likelihood converges more or less after about 100 iterations.
3. Use the trained model above to predict scores of new data-points. Pass the new points to the sldaPred() function that will return the predicted scores along with the mean log likelihood of the model.
4. This algorithm does sampling before the prediction. The sampler variable takes an index that is used as a starting index of the samples to be considered. It should be less than the number of iterations.

Error Measure Used: Root Mean Squared Error (RMSE) and ROC Curve(BLSLDA)

Logistic Regression

Logistic regression model is used to estimate the probability of a binary response based on one or more predictor features. It allows one to say that the presence of a factor increases the probability of a given outcome by a specific percentage.

Steps:

1. We chose below features as the input factors: Platform, Genre, Publisher, year of release, critic score, etc.
2. Add a column to the data-frame, and assign 1 when sales is more than one billion dollars, otherwise, assign 0.
3. Convert features to a binary format:
    Platform_3DS:0 Platform3Ds:1 Platform_GC:0 etc.
4. Split the Data into training and testing sets

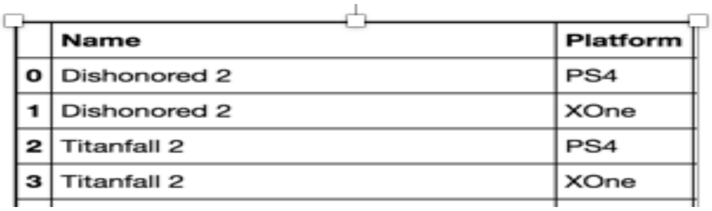
5. Use the training data to train the model, and use the trained model to predict testing data. Then convert the possibility response for the testing data-point to be binary 0 if it is less than 0.5, or 1 otherwise. Evaluate the model by comparing the predicted labels with the original testing data. We observed an 85.86% accuracy

Results / Comparison:

All topic model algorithms that we used did a good job in identifying different topics in the dataset. To perform a comprehensive study, we ran the algorithms in several different scenarios. Will discuss these below:

Logistic Regression

For this algorithm we have considered the set of games released in 2006, and used the trained logistic regression to predict games that could have been one of the most selling ones. We observed that Dishonored 2 on platform PS4 and Dishonored 2 on platform Xbox-One are the top 2 games that have percentages of 77 and 72.



	Name	Platform
0	Dishonored 2	PS4
1	Dishonored 2	XOne
2	Titanfall 2	PS4
3	Titanfall 2	XOne

LDA:

On entire dataset: LDA does a good job in identifying some common topics used in game reviews as we can see from the results. Most of these topics adhere to a particular genre/console, giving an intuition about the documents that contain those topics.

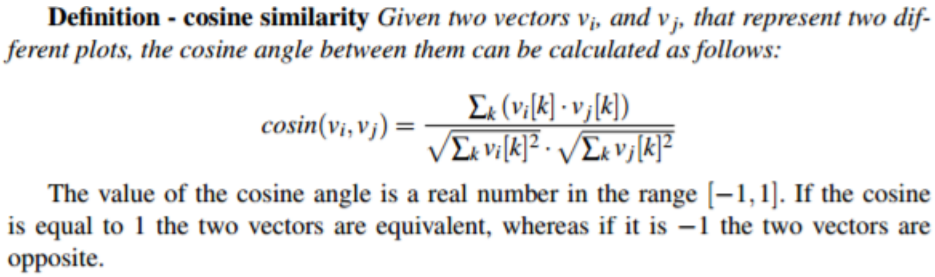
On top publisher titles: Results seen here are similar to above, having dedicated topics to some genres and consoles.

On titles having GameSpot Rating >= 8: Words and topics returned for these games have a rather positive sentiment. This can be noted by observing words such as new, good, great and different. This is expected as these reviews have more positive words in them corresponding to their high rating.

On titles having GameSpot Rating <= 3: As we might have expected, the topics here seem to have a negative kind of sentiments. This is evident by spotting words such as bad, broken and problems.

LDA vs LSA based on document similarity

To do a comparison between LDA and LSA, we compute a cosine similarity score between all documents. Below is the formula for cosine similarity: (Bergamaschi & Po)



Definition - cosine similarity *Given two vectors v_i, and v_j, that represent two different plots, the cosine angle between them can be calculated as follows:*

$$cosin(v_i, v_j) = \frac{\sum_k (v_i[k] \cdot v_j[k])}{\sqrt{\sum_k v_i[k]^2} \cdot \sqrt{\sum_k v_j[k]^2}}$$

The value of the cosine angle is a real number in the range $[-1,1]$. If the cosine is equal to 1 the two vectors are equivalent, whereas if it is -1 the two vectors are opposite.

We then find the most similar games, to a given game, by selecting the highest similarity scores. We have done this for 5 games chosen from varied genres and below are our results:

Game: Grand Theft Auto IV, Genre: Modern Action Adventure
The similar games given by LSA have 3 Modern Action Adventure games, 1 shooter and 1 racing game. This is a relevant result set as all the games returned have similar elements as GTA. GTA includes gameplay elements that can be regarded as Action Adventure (in game missions), Shooter (in game weapons and violence) and a variety of vehicles that might explain the racing game
LDA also seems to do a good job in finding similar titles. Though the results do not look as coherent as in LSA (one of the games returned is a soccer game).

Game: Fifa Soccer 08, Genre: Soccer Simulator
For LSA, we obtain exceptional results as all games returned were of the same genre. In case of LDA, the algorithm does a good job in staying in the sports realm, but gives results from a variety of sports.

Game: Incredible Hulk, Genre: Action
All the games returned by LSA for Incredible Hulk are action games including 2 historic action adventure titles and 2 modern action adventure titles. LDA gives more variety of genres but they seem to still be similar to the input game.

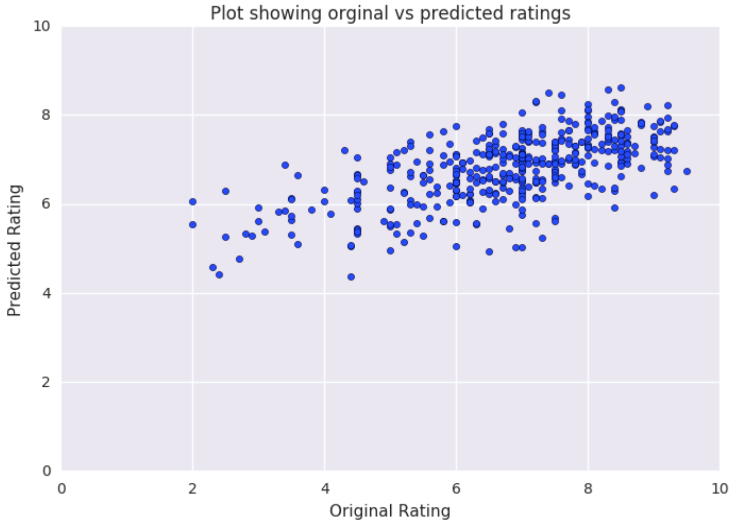
Game: WWE Smackdown Vs Raw 2009, Genre: Wrestling
LSA results for the above game are slightly less perfect compared to previous cases. Though 4/5 games returned are related to fighting/wrestling, there is one Sports game returned called Mario Strikers Charged. LDA results don't look good for this game. None of the games belong to the wrestling genre, though there is one shooter title. Interestingly, Mario Strikers Charged in returned for this too.

Game: Call of Duty 3, Genre: Historic First Person Shooter
LSA performs really well for this title too. All of the games returned are shooters, 4 being historic first person shooter and 1 modern first person shooter. LDA is able to identify other shooters, but it isn't as good as LSA in finding similar shooters.

Conclusion: Both models are able to identify similar games, though the performance of LSA looks better. It does a better job than LDA in identifying the most similar games. Such models can be improved further and used in developing recommender systems, where similarity can be used as a metric to give recommendations.

SLDA/BLSLDA

Our goal with SLDA was to do a prediction for user ratings based on the reviews. SLDA does a good job in identifying topics, but it does an okay job for predicting ratings. We obtained an rmse score of 2.11 after running SLDA with 200 iterations. In particular, the algorithm does not perform well in predicting ratings when actual ratings are too low or too high, but can predict with fairly good accuracy for median ratings. This is evident by observing the observed/predicted ratings graph.



Observe a few straight lines near the center and right side of the graph. These are the ratings for which the algorithm makes good predictions. Note that variance along x-axis (actual rating) is more compared to variance along y-axis. Thus, we can conclude that the model predictions are more concentrated between y=4 and y=8

With BLSLDA, we wish to predict the sentiment of the reviewer. We classified the original data by assigning labels 0 and 1 (0 if rating is less than 0.5, 1 otherwise). The model converged well when we ran 200 iterations. We observed the trained BLSLDA model has an accuracy of 87%.

Improvement: The results could be improved by possibly selecting a bigger corpus. Also some additional preprocessing could be applied to help the model converge better.