Bradley Erskine’s Big data project

Abstract / Summary

For this project I used Steam Video Game data sets from the UCSD research lab. I used 3 datasets being a user review data set, a item data set and a user data set. I looked into the popularity of different game genres and found the most successful pairs of genres in pc games. I based the success on a mixture of how positive the reviews are and the download counts of each genre pair. I did this using map, group by, fold by reductions and the market-basket analysis algorithm. The intended result was to show what genres are the most successful together. This will be significant as it will give people an idea of a genre that people enjoy for them to create their own games.This is also intended to be used by people to get ideas on what genre of game they would like to play.

Background

Steam has the biggest market for pc games and has a very large variety of games with many different genres.Each game can also incorporate multiple gameplay elements from unique genres. Steam has a simple review system where you can either recommend the game or not recommend the game. This will make it easy to rank success as I can compare the amount of positive reviews versus the amount of views. In this project I will use map and group by algorithms. These are scalable methods in dask that will allow me to process the data and just put the stuff I need together. I will also use Frequent Itemset Analysis involving finding items in data that often occur together and in my case game genres. Market Basket Analysis is an algorithm that allows me to get associations between the genres that appear together and identify the success of them compared to each other.

Motivation

I have played a lot of games and am interested in learning about gaming trends and what type of games do well. I have spent a lot of time before searching for new games and believe that the data I have collected could give insight into popular gaming genres and what genres work well together. Sharing this with others could give them ideas of types of games they want to play or even a good popular genre for them to develop their own game.

Research Question or Hypothesis

The aim is to identify the most popular game genre pairs on the Steam platform. This is relevant to the steam data set as it includes most of the popular games on steam and their genres. I can also find the reviews and the downloads of these games. Implementing the frequent item analysis and the market basket analysis will allow me to find the frequently co-occurring combinations of game genres. I can use these pairs to see which ones have the most success.

Experimental Design and Methods

Step 1

First I have processed a steam review dataset by filtering out all the strange characters in the data while unzipping gzipped json files and converting them into json lines. I extracted the user_ids as well as the item ids of the games they reviewed. I have paired each of these item id's with True or False whether their review was recommending the game or not recommending the game.

I also processed the steam_games dataset in a similar method of converting from gzipped json to Json lines file. This time I used regular expressions as it was easier to choose extra data to include from the data set.

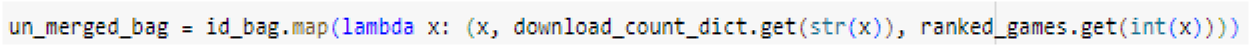
At first I just got the title, genres and game id. This resulted in there only being a very small amount of common genres so I decided to also extract the tag's to get more interesting genres and themes. I had to remove duplicates as most games have their genres also in the tag's. This created my **games_dataframe**

I used the steam games dataframe to find the most common full collection of genres. I did this by converting the 'genre's column to a bag and then using frequencies. Next I found the counts of each individual genre. I did this by mapping each column in the bag created above into 1 bag. I did this using split and flatten. This got me the results of the 15 first results of the most common genres

Step 2

The next step I took was to rank the games. First I put the pairs of (item_id, recommend boolean) into a bag and used frequencies. Using these I created a **ranked_games** dictionary with item_id , and a score that I based on frequency of recommending versus not recommending. This allows me to quickly look up a game score of each game.

I then created a new column in my **games_dataframe** by creating a bagfrom the [id] column and then using map algorithm to convert to Id paired with its game score that I got using the **ranked_games dictionary**

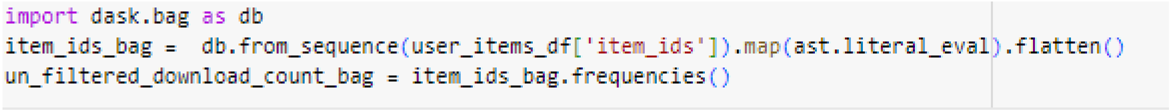


```
un_merged_bag = id_bag.map(lambda x: (x, download_count_dict.get(str(x)), ranked_games.get(int(x))))
```

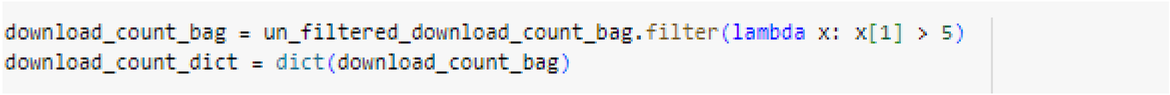
I then put this back into a dataframe and merged with the games_df to update the games_df to include Rank.

Step 3

The games dataset did include download counts so to get around this I used a 3rd dataset user-items. This dataset has users with a list of every game they have played. I processed the data into a dataframe with a similar method as above and extracted user_id and a list of item ids. (games that they played/downloaded.) Next I simply but the item_ids in a bag got frequencies, filtered and converted to a dictionary.



```
import dask.bag as db
item_ids_bag =  db.from_sequence(user_items_df['item_ids']).map(ast.literal_eval).flatten()
un_filtered_download_count_bag = item_ids_bag.frequencies()
```



```
download_count_bag = un_filtered_download_count_bag.filter(lambda x: x[1] > 5)
download_count_dict = dict(download_count_bag)
```

I then extended the method in step 2 to also add a download count column.

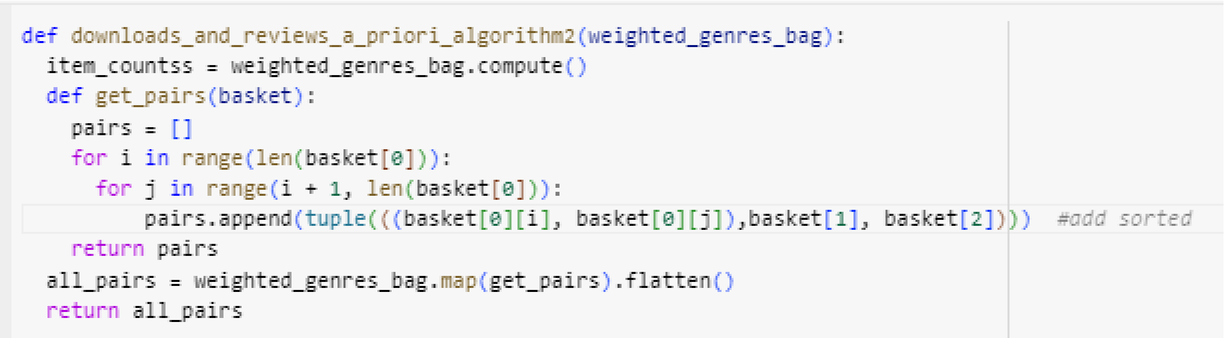
Step 4 (genre scores)

The next step was to use these values to weight the basket analysis that I used in step 1.

First I extracted the relevant information from the **games_df** being the ‘genres’, ‘ranks’ and ‘download’ and put them into **genre_list_bag** that is a list of tuples. Then I got scores for each individual genre. I then used a map algorithm to expand to individual genres. I then used a foldby and a map operation to calculate a score for each genre by averaging the weight score multiplied by the download count.

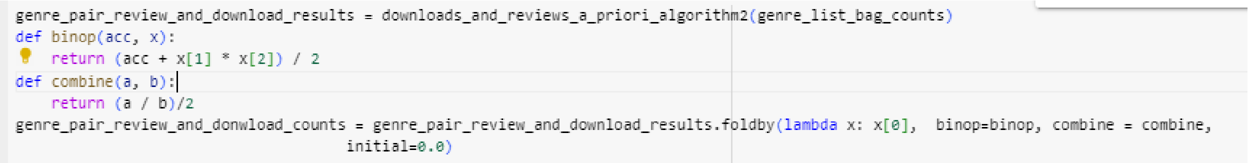
Final Step (weighted basket)

I then reimplemented the weighted basket analysis using the **genre_list_bag** that contains review ranks and download counts.



```
def downloads_and_reviews_a_priori_algorithm2(weighted_genres_bag):
  item_countss = weighted_genres_bag.compute()
  def get_pairs(basket):
    pairs = []
    for i in range(len(basket[0])):
      for j in range(i + 1, len(basket[0])):
          pairs.append(tuple(((basket[0][i], basket[0][j]),basket[1], basket[2])))  #add sorted
    return pairs
  all_pairs = weighted_genres_bag.map(get_pairs).flatten()
  return all_pairs
```

I did this by creating all the pairs without extra filtering as I did this earlier. As frequencies won’t work with the tuple of 3 I used foldby to get the final results and getting the average of a weighted score from download counts and review rank.



```
genre_pair_review_and_download_results = downloads_and_reviews_a_priori_algorithm2(genre_list_bag_counts)
def binop(acc, x):
    return (acc + x[1] * x[2]) / 2
def combine(a, b):
    return (a / b)/2
genre_pair_review_and_donwload_counts = genre_pair_review_and_download_results.foldby(lambda x: x[0],  binop=binop, combine = combine,
                                            initial=0.0)
```

I repeated this project to also have results just based on download counts and one just based on review scores.

Libraries

- **matplotlib.pyplot:** used for graphing results
- **urllib.request:** Urlretreive function to download data from web.
- **ssl:** I use to disable ssl verification to prevent problems when downloading data
- **Re**: I use regular expressions when parsing datasets to json.lines as they are not properly formatted when raw.
- **Ast:** Just used for literal_eval() witch converts to list from string
- **dask.bag:** What I use for most of computation and algorithms
- **Ijson:** Used for when I parsed json files.

- **Json:** Json.dumps to convert from string to json string
- **gzip:** Used to unzip json.zip files as data format of steam data
- **dask.dataframe:** I used to store and show data, for merging diffirent data sets.

Results

Scalability

I ran into an error when trying to use external package ijson with the cloud and couldn't work out how to use it.

All parts of my code except downloading are less then a few seconds except for when I convert from a dataframe to a bag which takes 44 seconds. My final results I have 2700 results and started with 28301 different games in the games dataframe. I would be able to include more games if I had access to a larger review dataset.

- Answer and discuss the hypothesis or research question as best as you can with data. Suggested length: 1-2 paragraphs

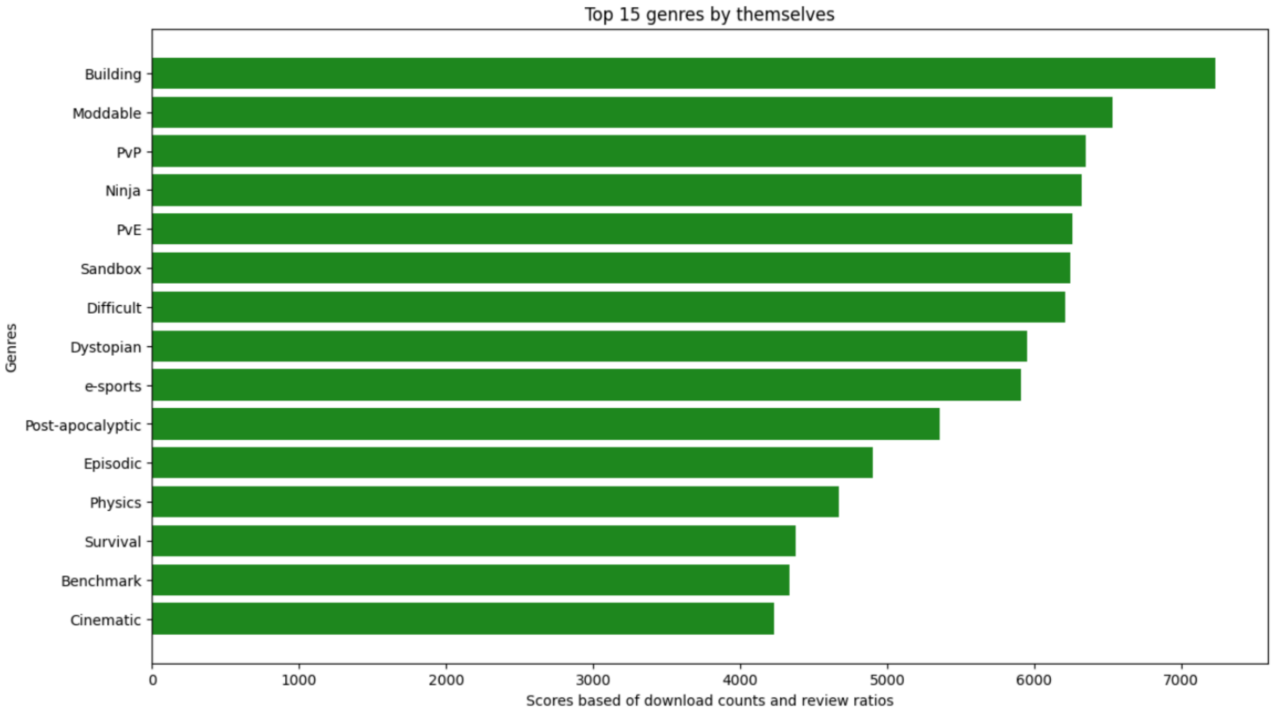


Fig 1

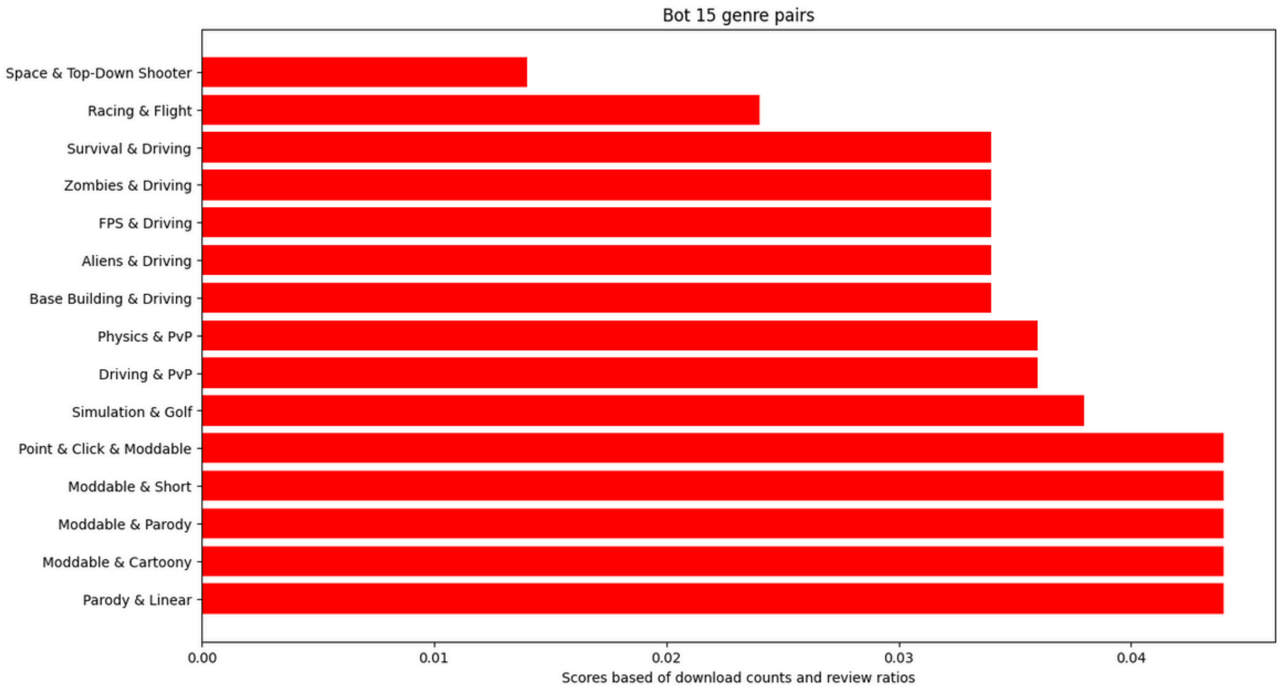


Fig 2

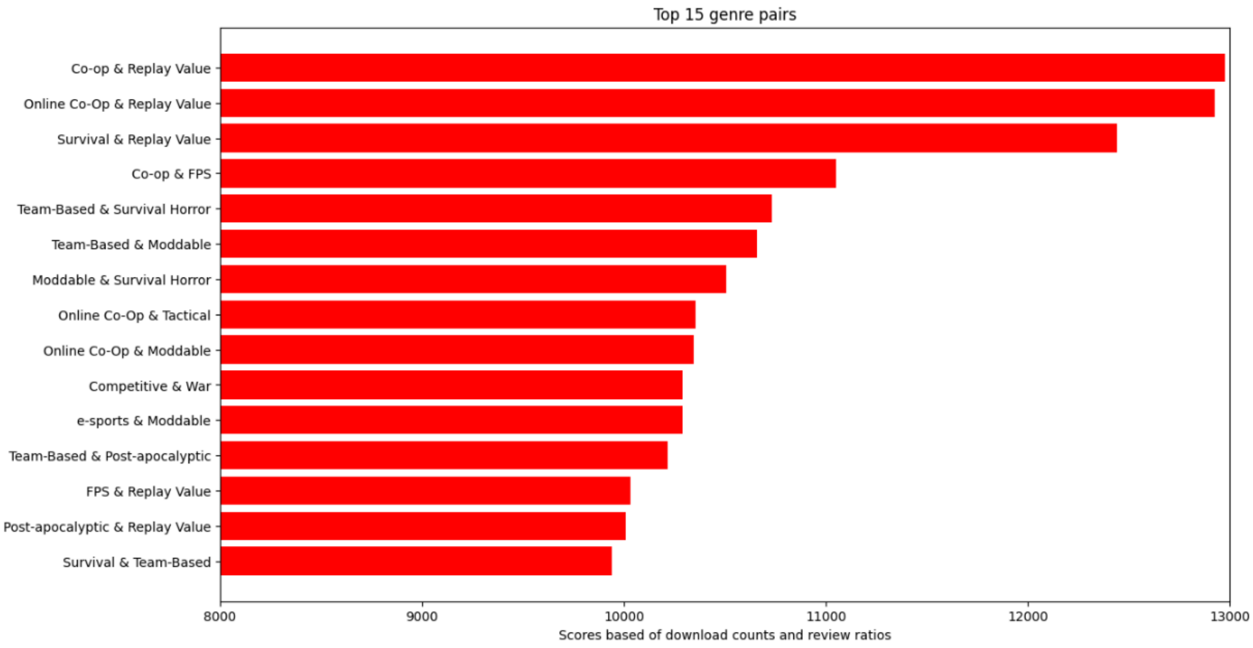


Fig 3

In the results that are shown above can see the most successful genres based on a combination of download counts and review scores. (fig1). I also have shown the top performing genre's under the same scoring system for genre pairs. It is interesting that many of the top performing individual genre's do not appear in the top performing genre pairs. This shows that a combination of genres is more important than just the singular genre's as it will do worse combined with certain genres.

My results have co-op appearing 5 times in the top genre_pairs and not in the bottom genre pairs or the individual genres. This shows that co-op when paired with popular genres does very well. Replay values are similar and it makes sense as is a key aspect of a good game.

Conclusion (suggest 3 paragraphs total, one for each prompt)

I was able to answer my hypothesis of finding the most successful genre pairs for steam games. I did this with success based on both download counts and reviews and have shown what genres are successful and popular together. I did this by combining the game_ids that were shared between the three datasets.

My results provide game developers with insights into what combinations of game genres are the most popular and most successful. This could be helpful for decisions when developers are developing games and what genres work well together but also possibly more importantly what genres do not. Another implication is that players could be drawn to one of the genre pairs that performed well and then go sort in steam by both those genre's to find a perfect game for them.

I got different genres showing up for individual genres and genre's pairs so I think it would be interesting to expand on the basket analysis algorithm and find the interest of 1 genre to another. This would be a bit complex with my set up as they are both weighted. I also wonder how much money each genre pair makes but individual game cost or money spent on the games were not included in datasets that I used.

One part of my design that I feel would work better with a different approach was how I chose to rank success for the games. I used a dataset of reviews to get a score based on the amount of positive reviews versus negative reviews. This didn't work as well as I had hoped as only about 3000 games had enough reviews to give a score in the dataset I had.

I believe a better approach would have been to rank games using the user item dataset. This is the dataset I used to get download counts and had about 28000 games with enough downloads to process. I could have also extracted other stats such as Total time Played. I also could have used the games_downloaded data from each user to weight each download by the amount of games that user had downloaded.

Reflection

Course concepts

Daks bag and dask dataframe were both useful.
Parallelism - use of dask functions such as map reduce and fold by were very useful
Basket Analisis - was a helpful pattern making it easy to get genre pairs.
Data Retrieval: How to get data from the internet into a codebase.

What did I learn

This project should be how useful parallel computing can be to perform large operations in seconds. I learned alot about using the basket analysis algorithm and its limitations when weighting each item. I learned that processing data is harder than I thought it would be and every data set can be formatted a lot differently and have many inconsistencies. I also re learned how to use regular expressions for this.

Overall this project was enjoyable as I got to apply skills gained over the course.

References

Data 301 lecture notes
http://infolab.stanford.edu/~ullman/mmds/ch2a.pdf
http://infolab.stanford.edu/~ullman/mmds/ch6.pdf
https://docs.python.org/3/library/re.html

From Bradley Erskine