SENG 474 Project

Predicting Steam Game Sales

Andrew Yung

Jake Rothwell

Zheng Li

Adebayo Ogunmuyiwa

***Abstract* -The problem domain is to identify if a game is going on sale on the Steam game platform based on the available datasets. Our group will web scrape available datasets and arrange/calculate needed data to be put in our final dataset, which relates to each game in order to predict whether a particular game would be on sale or not in a specific month and year. The output will be a classifier representing the sale state of a game.**

1. Problem description

The goal of our project is to draw from a dataset of Steam game sale history to predict if a certain game will go on sale. Given an input game title or appid (Steam's unique identifier), our program would output a classifier which indicates "on sale" vs "not on sale" prediction of a game. Due to the desired output being a discrete value, we have concluded that classification would be a requirement in the models (such as logistic regression) that we train. Something important to note is that most Steam games will eventually go on sale in their lifespan, so we have discussed ways to narrow our problem description down to a more specific one. To elaborate, another input parameter would be a certain month and year - the output being the classification of the input title going on sale within the given month and year. This would significantly narrow our problem down and make is more intuitive for users - users will want to know specifically when the game they are interested in will go on sale.

In terms of the format for our data, games will be organized as a time series by month and year - a given row will have a month and year tuple, for instance December 2017. We could then plot the individual features against time, resulting in a decision boundary problem for a binary classification where the labels are either 0 or 1, meaning that the label for a given game, month and year is 1 if it's on sale and 0 for no sale during anytime within the specified month and year tuple. As mentioned before, this will be a classification problem, so the labels for our dataset is 1 or 0 for every game and for every month and year since the game's release date.

2. Related work

A project [6] that attempts to forecast product sales volume doesn't involve the historical pricing data in it's training but rather with online data such as reviews (volume and rating) and promotional marketing (ie. free delivery options and discounts). The article mentions that previous "forecasting techniques include methods such as historical sales data" [6] which is what our data mostly consists of. How this relates to our project is that the volume of sales is often influenced during the time of product discount as "one of the most established marketing strategies implemented by vendors are price discounts offerings" [6]. Therefore the goals of both projects have some correlation (predicting product sales volume vs time of product discounts). It may be useful in being able to predict when a product is on sale, as it may be used to help predict the volume of sales; the reverse is also true.

In contrast to our problem description where we proposed a logistic regression approach, the article discusses the benefits of a neural network solution to the product sales prediction problem. Specifically, a multilayer perceptron (MLP) was used for training on the data. The article discusses the benefits of MLP, notably its ability build a model around both linear and nonlinear functions, making use of multiple hidden layers within the network. An input layer of 15 features was used, along with 4 hidden layers and one output layer. However, the output of the neural network used differs from our trained models in that the model discussed in the article which would output the volume of products sold, versus our models which would be a binary decision of whether or not the game will go on sale.

From their results, they had noticed there was a larger correlation to the accuracy of predictions from "the interplay effects" [6] of multiple variables rather than independently. This could be useful to know in our project as we may find that combinations of features might perform better rather than each being their own independent feature. The results of using online reviews and promotional strategies proved successful as such features contributed to the final prediction (within a certain degree of accuracy) for their project.

3. Data description

The data currently exists as two separate datasets, Steamsales.Rhekua [4] containing the pricing history and SteamSpy [3] containing various information on each game. At the time of this

proposal, there are around 20,000 products available on steam (not all of which are games, but can be filtered by their genre).

The attributes provided or that can be determined from the datasets are:

- Developer
- Publisher
- Score ranking of the game based on user reviews
- Ratio of positive to negative reviews
- Total number of owners
- Player playtime and activity (total and past 2 weeks)
- Days since last sale
- Average number of days between each sale (if there has been more than 1 sale)
- Variance of the days between sales (if there has been more than 1 sale)
- If there's a sale for a specific month and year (label for each row of features)
- Months since release

SteamSpy data can be accessed their own api [5], responding in the json format. In SteamSpys json response, there is a steam appid which can be used to request the sale history data from steamsales.rhekua.com through an http request. This data is easily pulled from the html response as each start and end date of the sale is under the same html tag class. Our script to retrieve this data is on average 3 games per second.

4. Proposition

A. Data

Data can be found on Steamsales.Rhekua [4] and SteamSpy [3] where it shows previous sales start and end dates in a sorted order. In addition, SteamSpy provides a publicly accessible csv dataset of all games that are currently on sale.

One problem that arose during the discussion of our project is how the model will be evaluated - more specifically, how the accuracy of predictions will be determined. We discovered that we must split the data into two distinct sets - previous sale data and current game sales. With the current month (whether or not it's on sale) being omitted in our data as it's not complete in regards to the possibility of game's going on sale later in the current month. With the previous sale data, the labels are represented as the game going on sale or not as the whole month has already elapsed.

B. Algorithm

Our aim is to provide a binary prediction from our various trained models and determining the best model suited for our data. We used Python along with the sci-kit learn library for creating our models. The models trained and tested for our project are logistic regression, SVM, and bagging classifier. We would also attempt to tune the models hyper parameters. To analyze each model, sci-kitlearn offers tools to evaluate the performance and accuracy of our model. Alternatively from related works, SMOreg provides an implementation for vector machine regression which categories data into several discrete classes. "The method attempts to find a boundary that correctly divides the training data into classes, and then uses this boundary to classify unseen data" [1].

C. Evaluation

Evaluation would be done using the error and accuracy of models. The dataset would be divided into testing, training and validation datasets. Our validation dataset would be used to tune hyperparameters of our model to improve accuracy rates if the model is suitable with our data. Due to the output being binary, we have concluded that the models would be binary capable classifiers.

5. Final Results

With our dataset finalized, we were able to see trends and the varying importance of certain features. We've found that many features have little to no impact in our accuracy due to the lack of relation between 2 or more games. This was most obvious with our publisher feature as a majority of publishers only have one game released on Steam and those that do have multiple games often don't have a consistent sales pattern for all their games. Towards the end of the project timeline, we were stuck with two models that achieved accuracy values less than ~70% which was suboptimal - SVM and logistic regression. Our third model was the Bagging classifier and had the accuracy we were looking for.

Further fine tuning the SVM model, shuffling our data before with a random state coefficient value of 300, and training our model with datasets > 1000 produces significant level of accuracy of about ~70%. With over 400,000 rows in our dataset, testing and training with anything <1000 sets of data reduces the accuracy and produces basise in the test model. From our result, a major factor that could be taken into account is whether or not to include the "average days between sales", "days between sales variance" and the "months since release" features when training the model.

Logistic regression gives us around 65% accuracy and by looking at the predictions it makes, its equivalent to prediction all 0's (not on sale) which may be due to the large number of new games that don't go on sale at all. The training data and testing data gets splitted into 0.9 and 0.1 correspondingly. We observe that there is a large increase in the number of new game releases since 2014 [8], to the point where 2017 itself has nearly half of the games released from all games. We have tried to tune the hyperparameters and size of the testing set, and have found that the increasing number of training data can slightly improve the performance of SVM, which slightly improves past 70% at the cost of exponentially increasing our training time. Our group has tried using different kernels to obtain the optimized accuracy.

Our best model accuracy-wise ended up being the bagging classifier from scikit learn. The way the bagging classifier works is by aggregating the predictions of each individual randomized model

to form a final prediction, where the base estimators are decision trees. This model makes it harder to overfit than if it were just a single decision tree. Each model is a subsample of the dataset and may contain bootstrap samples. Bootstrap samples are duplicate samples from the dataset so that the desired number of samples in each model can be reached. This could be useful when there are a lack of samples for a particular game or time period.

The accuracy for the bagging classifier is consistently around 85% (after hyperparameter tuning) when testing on random months and year pairs. The hyper parameter changes made have been by increasing in the number of base estimators from the default 10 to a value where there is very little improvement in accuracy (around 40 base estimators). The accuracy when forecasting is slightly lower as new games that are released are difficult to predict, as our current data has no way of know whether a game will have a launch sale as our current data is insufficient to predict such outcomes. This model works well as games that do go on sale are often based on a specific time period in each year (Christmas and summer sales).

7. Unexpected Challenges

As previously stated, the current prediction accuracies of our logistic regression and SVM models are less than ideal. For our SVM implementation, we found that when increasing the size of our training set, the accuracy can be marginally improved. For instance, if we have a training size of 300 samples , we achieve an accuracy of ~65%. On the other hand if we increase the training size to 3000 we get an accuracy of ~70%. Although this may seem like an improvement, it comes with a cost - the training time is significantly increased with large training sets to the point where it takes an unreasonable amount of time.

The large amount of data we have makes it a challenge for us to train the whole dataset. SVM model takes unreasonable training time when it runs on the whole dataset no matter what hyperparameter we chose. The training time has significantly increased with when the training dataset gets large. Our SVM model only takes small amount of random dataset, which may results huge bias in the prediction. Thus, SVM model does not show the expected optimal results.

After researching the capabilities of pytorch, we may have found a potential solution to the problem of large training time. Pytorch uses the GPU to significantly increase training time across model implementations. This could be to our advantage - however it is important to note that Pytorch is specifically meant for neural network implementations. So, a goal for us is to attempt to find a way to implement a classifier using a neural network in Pytorch.

We encountered a problem when using the bagging classifier - strictly to do with noise in the data, where it's difficult to account for flash and random sales that occurred throughout the year, rather than the more predictable sale periods such as summer and the holiday season. When this classifier is trained and tested on a subset of data (where the rows are only of a particular game), it's likely to perform more poorly depending on how new the game is. The problem with new games is that they're less than a year old which doesn't cover the whole year to bootstrap samples unless we may expect it to have similar behaviour to our other samples.

After implementation of logistic regression, we have found the classifier that predicts the model corresponds to game not on sale is roughly the same as normal model performance. This essentially means that the model is not accurate regardless of label assignment. We have compared different results when we change labels or drop label assignment, which ends up close accuracy. Therefore, our logistic regression model proved to be unsuccessful.

6. Ethical Implications

The ethical issues of our project is that it's not in the best interest of game publishers as people would be more inclined to wait until a game is on sale if they knew when it may be on sale. The volume of sales could have larger spikes as its logical for consumers to buy during a discounted period so there would be less buyers at the regular price. This could affect the revenue of businesses (especially smaller studios) and influence publishers decisions

Professor of film and media studies at Dartmouth University, Flanagan proposed that games can be a way to achieve social intervention because games are something most people are used to. [7]. The active game market can create an active social intervention for people. They feel more natural and unforced when they talking to the people in the game. People can respond positively in real-life situation when they play games that help you develop social communication.

Although some consumers may not care whether a game is discounted, we do not know the significance it may have on a particular game as there are different groups of target audiences for games. However, it would impact the buying decisions of some consumers and propagate its effects on to the market. From this standpoint we believe that potentially disrupting the game sale market in favor of consumers could be considered unethical because there is a potential detriment to the success and profits of game developers who currently struggle in an oversaturated market.

Games on sale also promote the sense of community. Players have shared understanding and values when they play the same game.A long-term interaction lays on the foundation for a feeling of community. Social interaction between members of multiplayer communities share the similarities between face-to-face communication. Thus, people become more active in real-life, and enable to a form lasting relationship with friends.

7. Future Goals

After completing this project we believe that we were successful in reaching the goals that we set out to achieve at the beginning of the term. Nevertheless, we believe there are numerous

potential improvements to project that we could implement in the future.

First and foremost, we believe that the size and variety of our dataset is currently limiting our models' ability to predict frequent outliers. To elaborate, our model is currently biased towards sales that occur in the most popular times - summer, fall and holiday seasons. This is because most of our sales occur at these times, and as a result the model is biased toward these data points. Furthermore, it is rare and difficult for our model to predict outliers outside of these time periods due to the limited variety of our dataset. We believe that if we diversified our data to include more sales that occur during odd times, our model would be better able to predict such outliers and therefore perform more accurately overall.

Of course another improvement for our project could be to improve the accuracy of all our attempted models to above 80% (for example, getting pytorch working with SVM so that training can be done with more samples within a reasonable amount of time), while also trying out some other classification methods to see if there is an improvement in accuracy. Ideally we would want several different model implementations to compare with to see which is the most effective for this problem.

A final improvement to our project could be to provide more visual metrics along with the prediction results.
Being that our goal is provide the user with as much information as possible so that they are confident when a game will go on sale, the more visual representations of the data that we can provide the user the better.

8. Timeline

Task	**Date**
Acquire initial dataset and supplementary dataset	February 14
Integrate data set structure	February 24
Validate the total dataset for completeness	February 27
Finalize and submit mid-term report	March 6
Train Algorithm and evaluate prediction performance	March 10
Explore potential interest and investigate potential regression algorithms	March 14
Tune features and parameters for tested algorithms	March 20
Perform tweaks on final algorithm and parameters	March 26
Finalize and submit report	April 9

9. Task distribution

Member	**Distribution**
Andrew Yung	Model Development (Bagging Classifier) Dataset Preparation Presentation Script
Jake Rothwell	Model Development (Logistic Regression) Documentation Presentation Video
Zheng Li	Model Development(SVM) API Development Presentation Script
Adebayo Ogunmuyiwa	Model Development(SVM) Predictions/Evaluation

References

[1] S. Ehrenfeld, “PREDICTING VIDEO GAME SALES USING AN ANALYSIS OF INTERNET MESSAGE BOARD DISCUSSIONS”, 2011. [Online]. Available: http://sdsu-dspace.calstate.edu/bitstream/handle/10211.10/1073/Ehrenfeld_Steven.pdf. [Accessed: 03- Feb- 2018].

[2] "sklearn.linear_model.LinearRegression — scikit-learn 0.19.1 documentation", Scikit-learn.org. [Online]. Available:http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html. [Accessed: 03- Feb- 2018].

[3] “Game sales”, SteamSpy - All the data about steam games. [Online]. Available: https://steamspy.com. [Accessed: 03- Feb- 2018].

[4] “Steam Sales”, steamsales.rhekua.com, 2010. [Online]. Available: http://steamsales.rhekua.com. [Accessed: 03- Feb- 2018].

[5] S. Galyonkin, “SteamSpy API”, steamspy.com/api.php. [Online]. Available: https://steamspy.com/api.php. [Accessed: 03- Feb- 2018].

[6] A. Chong, B. Li, E. Ch’ng, F. Lee. “Predicting online product sales via online reviews, sentiments, and promotion strategies: A big data architecture and neural network approach”, 2011. [Online]. Available: http://www.emeraldinsight.com/doi/full/10.1108/IJOPM-03-2015-0151 [Accessed: 24- Feb- 2018]

[7] Paul Rautner. “Video Games can have a meaningful social impact, says Davos expert”, 2018. [Online]. Available: http://bigthink.com/paul-ratner/video-games-can-have-a-meaningful-social-impact [Accessed: 6- Apr- 2018]

[8] Luke .“Did I just waste 3 years?”, 2018. [Online]. Available: https://infinitroid.com/blog/posts/did_i_just_waste_3_years [Accessed: 06-Apr-2018]