SENG 474 Project

Using Regression Models to Predict 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 this will be a logistic regression problem. 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 logistic regression 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. The activation function used was a sigmoid - which relates to our problem where we would be using a sigmoid with our logistic regression. However, the output of the neural network used differs from our model in that the model discussed in the article which would output the volume of products sold, versus our model 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 model using various forms of logistic regression.
We plan on using Python along with the sci-kitlearn library for our model. Logistic Regression seems to be the best fit for our problem at this point, however we will test all the various regression models that sci-kitlearn has to offer 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 regression algorithms and the dataset would be divided into testing , training and validation datasets. Our validation dataset would be used to tune hyperparameters of our model in achieving desirable and improving accuracy rates. Due to the output being a binary probability value, we have concluded that the best model to use would be Logistic Regression [2].

5. Progress Report

A. Completed Tasks

The needed dataset has been finalized and represents a time series for each game. With our data, we have setup a logistic regression model for training as well as a simple SVM. The logit model from Sklearn library has been used to fit our data. Our group obtained the mean accuracy on the given test data and labels. The logistic regression model also give us the probability of correct prediction for each feature.

A simple SVM model has been used to fit our dataset. The purpose of using SVM is to perform multi-class classification on a dataset because the desired decision boundary needs to be found. Our group has tried using different kernels to obtain the optimized accuracy.

B. Results

The accuracy for logistic regression is around 64% for training whole datasets. The training data and testing data gets splitted into 0.9 and 0.1 correspondingly. The data includes months, year, average days per sale, number of months since release. The data and target labels gets fitted into the logistic regression model, and use score() for getting output accuracy.

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.

C. Roadblocks

As previously stated, the current prediction accuracy of our model is less than ideal. We have 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. We have been trying to find a balance between an adequate accuracy value and a realistic training time.

D. Future Goals

Our plan for the remainder of the project term mainly consists of improving the prediction accuracy of our model - which currently sits at around ~67.5% for both the logistic regression and SVM models. Clearly, this accuracy value is not ideal for an adequate classifier, so our group’s current priority for the remainder of the project is to improve the prediction accuracy to at least 85%.

We are currently using scikit-learn for both our logistic regression and models, however we plan to implement s different models in pytorch to see if there is an accuracy improvement. Pytorch has a built in optimizer function which might prove beneficial to our model’s accuracy.

As stated in the previous section, we ran into an issue of an unreasonable training time. 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 Logistic Regression model using a neural network in Pytorch.

6. 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 6

7. Task distribution

Member	Distribution
Andrew Yung	Model Development Dataset Preparation
Jake Rothwell	Model Development Documentation
Zheng Li	Model Development API Development
Adebayo Ogunmuyiwa	Model Development 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]