Machine Learning Engineer Nanodegree

Capstone Proposal

Oliver Tacke
January 20, 2017

Proposal

It was more difficult than I thought to come up with a proposal for a capstone project. There are tons of interesting datasets out there and dozens of questions that you could possibly ask, but I'd love to create something that someone actually needed.

I contacted a friend of mine who co-created OpenSNP, a platform hosting raw genotype data. In fact, he could need someone to use labeled data from 1000Genomes in order to create a classifier that would tell you what's the origin of genetic data. Unfortunately, my knowledge of biology is too poor. Anyway, what I learned form this inquiry is how important domain knowledge and interdisciplinary teams are for data science and machine learning.

I chose to have a closer look at the field of education where I know a thing or two. I found a paper that dealt with undergraduate student generic problem-solving skills. It was based on an empirical study that had produced some data. It could have been used for creating a predictive model for problem solving skills performance, for detecting/classifying sub-groups of students, etc. Unfortunately, the data was not freely available. We need more Open Science! I contacted the author, but I guess he was more afraid of "losing" his data than excited about getting new tools he could use. He wanted to have my resume - that he received - but I never heard of him again.

Well, I browsed the web for some more datasets and found something that would be interesting for me, that might have an appropriate level of difficulty, and that (hopefully) doesn't require domain knowledge that I don't possess.

Domain Background

One of the first video games that I have ever played was Munchkin. It was released in 1981 when the video game industry was still in its infancy. Today, it is a multi-billion dollar business. In 2014 in the U.S. alone, 155 million people played video games (cmp. Entertainment Software Association, 2015, p. 2). In total, they spent 15.4 billion US dollars (cmp. Entertainment Software Association, 2015, p. 12).

In an industry, success is not only measured in good critics, but the amount of money earned or unit sold. In order to make decisions about future productions,

publishers may want to predict the sales figures which they can expect after a new game has been released. Those decisions could possibly be based on historical data and a suitable regression model. For instance, one could hypothesize that good scores in reviews correlate positively with high sales figures. Also I could assume, that those reviews can help us to project future sales. In fact, the general plausibility of this approach has been investigated and proven for the movie picture industry a decade ago: "Online movie reviews are available in large numbers within hours of a new movie's theatrical release. Their use, thus, allows the generation of reliable forecasts much sooner than before." (Dellarocas, Zhang & Awad, 2007, p. 39). For the video game industry, Beaujon (2012) developed a third-degree polynomial formula for predicting sales from historical data - basically manually using a spreadsheet. I would like to search for a better solution using a suitable machine learning algorithm, more features and more data.

Problem Statement

The earlier a video game company knows how many copies it can expect to sell of a game, the earlier it can estimate the revenue and the earlier it knows whether the game will be a financial success or not. For example, a decision about starting the production of a sequel or the production of a port to a different platform might depend on this information. In consequence, early knowledge about the sales figures can speed up the decision process.

The problem can clearly be "measured", because the success of a video game can be expressed in sales volume or revenue if the retail price is known. Also, it should be possible to create a regression model that takes input data and transforms those to a prediction of units sold. Since there are no random parameters involved, the results will also be reproducible given the same model and the same input parameters.

Datasets and Inputs

There are at least two relevant datasets that I'd like to use for creating a regression model for predicting the sales figures of a video game.

- Video Game Sales with Ratings: There's a dataset about video game sales based on data scraped from VGChartz. It contains information about games, including name, platform, year of release, genre, and sales figures for several regions. This dataset has been extended with several features from Metacritic, adding e.g. quality ratings from metacritic's staff and from users, the amount of reviews, and also adding age/content ratings from the entertainment software rating board. In total, there are more that 5.500 complete cases.

- IGN scores: There's also a dataset that contains ratings from Imagine Games Network (IGN).

Both datasets might be merged, and there may be similar sets that could be obtained by scraping some other sources. Also, some more features might be added, e.g. a flag indicating whether a game is part of a franchise such as "Super Mario". The goal of all the effort is to train a model that uses information about the games' genre, platform, year of release, the different ratings, etc. to predict the sales volume.

Merging several datasets will require to cleanse them. Also, some fuzzy string comparison might be beneficial in order to identify matching entries across the datasets. As a side project, using machine learning for tuning string distance metrics such as Damerau-Levenshtein distance or Jaro-Winkler distance might be interesting and useful (I recently used both in a completely different project).

Solution Statement

The solution to the problem will be a regression model that will output a prediction of sales figures for a video game. As input it will use those variables that have turned out to have significant influence on the outcome - possibly also a "best guess" if not all data can be acquired. There are several possible algorithms that I might use such as support vector machines or even neural network models. However, for this proposal, there has not yet been an investigation about what approach might be most beneficial.

Benchmark Model

The dataset seems to be quite large compared to the number of features that I intend to use, so I should not have trouble splitting it into a training set and a test set. This way, I can train the model and test its performance on the test set. Of course, I can also use k-fold cross validation in order to check for overfitting.

I can also use new data from the data sources (VGChartz, Metacritics, IGN) that are not available yet. This would basically be a real world test.

Finally, I can compare my results with those of others. Some people at Kaggle seem to be experimenting with prediction models, too. For example, Jonathan Bouchet built a polynomial regression model in R and reports an R^2 score of 0.098404. Since the best possible score is 1.0, there might still be some room for improvement.

Also, I could check how the model compares to SimExchange. This platform tries to predict video game sales based on the concept of "wisdom of crowds".

Evaluation Metrics

My goal is to predict the sales volume of video games, which simply is an integer. I can apply common statistic approaches to compare and visualize the deviation of the predictions from the correct results. The explained variance score or the R^2 score could be used to quantify the performance of my model.

Project Design

The first step of every machine learning project should be to **define the problem**. This proposal reflects the first part of this task, but there's more to be done. If this proposal is accepted, I am going to investigate the video game business some more in order to find out if there are some more data that might help me on my way.

Step two will be to **analyze and prepare the data**. First, I will have to check the data quality on a syntactical level. For example, columns that are expected to contain numbers such as the predicted sales volume should not contains strings. Secondly, I will have to merge my two datasets which will definitely require some adjustments. In order to connect data from one set to the other correctly, there must be some features that can be used to clearly identify a game, so at least name and platform should match in both sets. After that, it is crucial to understand the data before experimenting with algorithms. I will have to check the data for plausibility, because some feature values might be off the chart for whatever reason, etc. Also, by visualizing the data in different ways, I hope to get some more insight into the problem. This will probably also involve a principal component analysis to filter for features that might not be relevant and thus speed down the algorithms.

Step three is to **select algorithms**. I will consider different methods and approaches that might be appropriate for building a regression model. Althought quite a lot algorithms might suit the problem, the right choice will depend on different criteria, such as efficiency and scalability. Since I am dealing with more than 50 but less than 100.000 samples for predicting a quantity, the scikit-learn cheat-sheet suggest to consider Lasso, ElasticNet, Ridge Regression, Support Vector machines or even Ensemble Regressors. Out of curiosity, I might also tinker with Neural Networks.

Step four will be to **run and evaluate the algorithms** that I have chosen in step three. By comparing each algorithm's performance using e.g. k-fold cross validation, I want to identify the algorithms that are best at "understanding" the data and at attacking the prediction of video game sales.

As step five I am going to **improve and finalize the results with focused experiments and fine tuning**. Each algorithm has certain parameters that can be tweaked in order to get better results. Approaches such as a grid search

can be helpful to support this task systematically. Possibly, even ensemble methods could be tried out for fine tuning the results.

I intend to use a Jupyter Notebook to document all these steps including meanders and dead ends.

Literature

- Beaujon, Walter S. (2012). *Predicting Video Game Sales in the European Market.* Retrieved from https://www.few.vu.nl/nl/Images/werkstuk-beaujon_tcm243-264134.pdf (January 12, 2016).
- Dellarocas, Chrysanthos, Zhang, Xiaoquan (Michael) & Awad, Neveen F. (2007). Exploring the value of online product reviews in forecasting sales: The case of motion pictures. *Journal of Interactive Marketing, 21(4)*, 23-45.
- Entertainment Software Association (2015). *Essential Facts About The Computer And Video Game Industry. 2015 Sales, Demographic And Usage Data.* Retrieved from http://www.theesa.com/wp-content/uploads/2015/04/ESA-Essential-Facts-2015.pdf (January 12, 2016).