Machine Learning Engineer Nanodegree

Video Games Recommendation System Proposal
Mahmoud Ahmed
October 2nd, 2018

Proposal

Domain Background

Till recently, people generally tended to buy products recommended to them by their friends or the people they trust. This used to be the primary method of purchase when there was any doubt about the product. But with the advent of the digital age, that circle has expanded to include online sites that utilize some sort of recommendation system. **A recommendation system filters the data using different algorithms and recommends the most relevant items to users. It first captures the past behavior of a customer and based on that, recommends products which the users might be likely to buy.**

If a completely new user visits a site, that site will not have any past history of that user. So how does the site go about recommending products to the user in such a scenario? One possible solution could be to recommend the best selling products, i.e. the products which are high in demand. Another possible solution could be to recommend the products which would bring the maximum profit to the business.

If we can recommend a few items to a customer based on their needs and interests, it will create a positive impact on the user experience and lead to frequent visits. Hence, businesses nowadays are building smart and intelligent recommendation systems by studying the past behavior of their users.

Problem Statement

Video-game players generate huge amounts of data, as everything they do within a game is recorded. In particular, among all the stored actions and behaviors, there is information on number of hours user played some games. Such information is of critical importance in gaming world, where gamers tend to spend more time playing games

they like most, so we want to recommend similar games to games he spend more time playing them.

Datasets and Inputs

The data used in our analysis comes from Steam offered from Kaggle
Steam is the world's most popular PC Gaming hub. With a massive collection that includes everything from AAA blockbusters to small indie titles.

This dataset is a list of user behaviors,(200K rating) with columns:

1. User-id
2. Game-title
3. Behavior-name : included :-
    a. ‘Purchase'
    b. 'Play'
4. Value : The value indicates the degree to which the behavior was performed - in the case of 'purchase' the value is always 1.0, and in the case of 'play' the value represents the number of hours the user has played the game.

Solution Statement

Such systems can better achieve their goal by employing machine learning algorithms that are able to predict the rating of an item or product by a particular user like Matrix Factorization (using SVD) or using deep neural networks In this project we build video-games collaborative recommendation system : using deep neural network, which promising candidates for operational video-game recommender system. it should be possible to integrate them into the game stores, so that users automatically get personalized recommendations while exploring games. The presented models are able to meet relevant predictions of the games that a particular player will find attractive and related to his taste.

Benchmark Model

One of contributors of the dataset used Matrix Factorization approach by:

- Using basic SVD:
- Using SVD via Gradient Descent

And get 2.933 RMSE score for basic SVD , and 1.369 RMSE score for Gradient Descent SVD.

Evaluation Metrics

Student proposes at least one evaluation metric that can be used to quantify the performance of both the benchmark model and the solution model presented. The evaluation metric(s) proposed are appropriate given the context of the data, the problem statement, and the intended solution.

The performance of each model is evaluated using **Root-mean-square error (RMSE)** :

Root-mean-square error is one of the most popular, frequently used, simple measures to find the accuracy of a model. In a general sense, it is the difference between the actual and predicted values. By definition, it is the square root of mean square error, as given by the following equation:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (x_{act} - x_{pred})^2}{n}}$$

Here, Xact refers to the observed values, and Xpred refers to the predicted values.

Project Design

The workflow of solving this problem will be in the following order:

- Exploring the Data
  - ○ Loading Libraries and data
  - ○ Dimensions of data
  - ○ Statistical summary

- Data preprocessing/cleaning
  - ○ Preprocess dataset columns
  - ○ Data cleaning
  - ○ Training and Validation data split
  - ○ Feature Scaling - Standardization/Normalizing data

- Evaluate Algorithms
  - ○ Build model
  - ○ Make predictions on the validation set

- Model Tuning to Improve Result
  - ○ Set Number of epochs
  - ○ Set number of latent factors
- Final conclusion

Visualizations will be provided in some sections as needed.