Guncha Babajanova
ECON900_ps1
MACHINE LEARNING AND BIG DATA

Data Description

In the first portion of the exercise we collected up to 7,978 non-null objects from the www.boardgamegee.com website. The data include information on name (or title) of the game and the year, board game rank, geek rating, average rating, number of voters, and prices (i.e. list price, lowest amazon price, new amazon price, and iOS app price). Among these, all variables except prices have 7,978 observation (see summary statistics table for key variables in the appendix).

Machine Learning Exercise Summary

For the machine learning exercise, we will try to predict or identify which category the game belongs to, i.e. identify "good" games vs. "bad" games using geek rating as the outcome variable and given some characteristics (i.e. independent variables, average rating, number of voters, and board game rank). To achieve this goal, we will be using a classification model to classify the games into "good" vs. "bad" games given some characteristics. Specifically, we will use support vector machines or SVM, which is available in Scikit-learn software machine learning library. We will use SVM's supervised learning method by classification.

It is natural to wonder why one would want to classify things. In general, classifying objects into groups could be helpful in decision making process, especially when the number of objects are large. This exercise will attempt to predict "good" games given some characteristics in a very simple and crude way using machine learning tools.

Before we begin, it is important to note that data cleaning (or pre-processing) might be needed. This is especially true if there are missing observations (i.e. null objects). In such cases, one would have to decide how to handle these missing values. In our case, variables of interest all have 7,978 observations therefore we can begin the exercise.

We will begin by classifying our outcome variable, geek rating, into two groups: bad games with ratings below 6.5 and good games with ratings 6.5 or above. It is assumed that geek rating is a variable that could range from minimum of 0 to maximum of 10. We will also label our two newly created groups into "bad" and "good", such that "bad" is less than "good" (this is essential for sensible results). In our case, this step resulted in 7,069 "bad" games and 909 "good" games (also see figure 1 in the appendix).

Next, to predict the "good" games we need to define our dependent and independent variables. We will also split the data into the "train" and "test" parts to check how well the model performs (i.e. how well the predictions are). For this exercise, we will designate 20% of the data as test data and the rest will be the train data. More specifically, we will have two X vectors (X_train and X_test) and two y vectors (y_train and y_test).

Moreover, we will scale our independent variables to ensure that magnitudes of these variables do not bias the impact of these variables on the model. In other words, for more accurate predictions we want to make sure that that the scale of the variables are not causing them to be less or more influential than they should be. Therefore, we will transform both, train and test data.

We are now ready to use SVM classifier to predict "good" games. We will first use the train data to fit the data. Once the "training" process is done, we will then use the test data to predict "good" games and "bad" games given the characteristics in the test data.

To see how well the model performs (i.e. how well it predicts "good" game or "bad" game in the test data) we will use classification report (see appendix). According to the test, 99% of games in the test data that were labeled "bad" were actually "bad" and 98% of the games in the test data that were labeled "good" were actually "good." On average, the model predicted "good" and "bad" games with 99% precision. So, we can conclude that model is slightly better at predicting "bad" games than "good" games, but overall it performs well at classifying games.

To further investigate the performance of our model, we will also use confusion matrix test/report. The results from the confusion matrix show that the model correctly predicted 1,401 "bad" games and misclassified 3 of them. Moreover, the model correctly predicted 182 "good" games and misclassified 10 of them. These results are consistent with the classification report.

As a final step of the exercise, we will check whether classifying games into 2 groups is sensible. We will use Gaussian Mixture Model (GMM) and k-Means methods along with their appropriate silhouette scores to determine whether classifying (or clustering) games into two groups was sensible. According to the k-Means silhouette score results (see table 3 in the appendix) there are 3 or 4 predetermined clusters in this multidimensional dataset. On the other hand, GMM silhouette score results suggest that 2 components or groups are more appropriate for the given dataset. However, given the objective of this exercise 2 groups (i.e. "bad" games vs. "good" games) is more appealing and is more sensible in terms of interpretation. That being said, as a further step one could reclassify geek rating variable into 3 or 4 groups and test the performance of the model with these groups.

Appendix:

Table 1: Summary Statistics for some Key variables

	Geek Rating	Average Rating	Board Game Rank	Number of Voters
Count	7978	7978	7978	7978
Mean	5.89	6.68	5400.73	1525.28
Standard Deviation	0.4835	0.8183	4462.37	4312.59
Minimum Value	3.47	1.05	1	30
Maximum Value	8.61	9.58	16967	84203

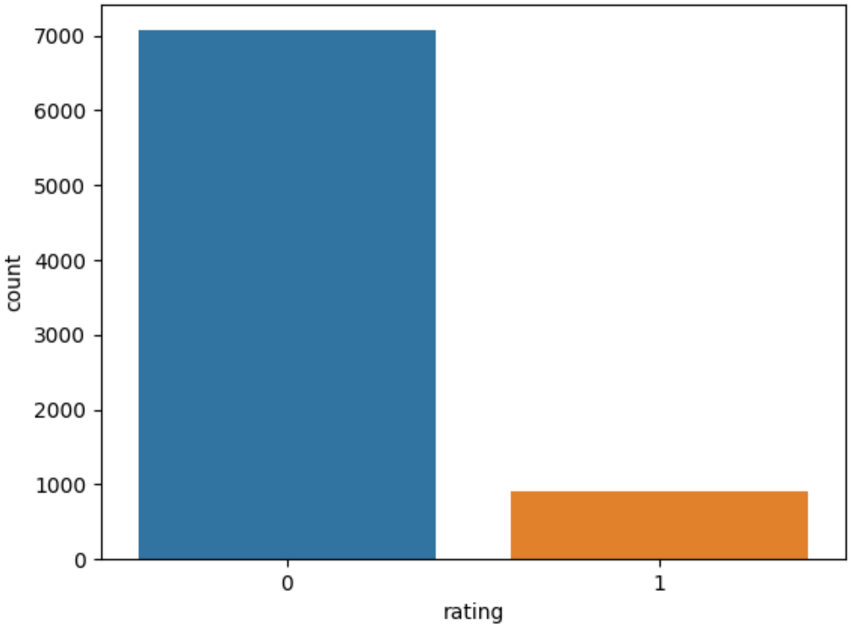
Table 2: Classification Report Results

	Precision	Recall	F1-score	Support
Bad (0)	0.99	1.00	1.00	1404
Good (1)	0.98	0.95	0.97	192
Micro average	0.99	0.99	0.99	1596
Macro average	0.99	0.97	0.98	1596
Weighted average	0.99	0.99	0.99	1596

Table 3: Silhouette Scores: k-Means vs. GMM

	2 groups	3 groups	4 groups	5 groups	6 groups
k-Means	0.5609	0.6152	0.6213	0.5172	0.5306
GMM	0.4874	0.3258	0.1867	0.1223	0.1891

Figure 1: Count of “bad” games (bad=0) vs. “good” games (good=1)


count
rating