Project 3: Naive Bayes

Sam Beckmann

April 21, 2017

Project 3 is create a naive Bayes classifier and test it on a dataset. I choose to use a dataset I found online that contained the platform, genres, and review score of 17,534 video games reviewed by IGN. For the purposes of my classifier, I used the platform as a multi-state variable and all of the genres as independent boolean variables, with the classifications being scores. Since the naive Bayes algorithm classifies discretely, I choose to divide the the results in categories by taking the floor of the actual score. This resulted in 11 possible classifications, as IGN uses a 0-10 scale.

Using a training set made of 80% of the data, the classifier performed with an accuracy of 28%, and an average absolute error of 1.36 from the predicted score to the actual score (For the sake of this calculation, I used the floor of the actual score, so it was consistent with the predictions being made). Considering both that random choice would yield an accuracy of 9%, and that platform and genre are not incredibly informative evidence as to the rating a game will get, I consider this result mildly impressive. In particular, the the classifier often had >20% accuracy with a training set of only 20% of the data (around 3,500 examples). I believe this accuracy mostly came from the understanding that most games get scores in the 6-9 range, and thus using the priors to semi-accurately classify.

For comparison purposes, I ran the test set through a completely random classifier. This random algorithm classified with an accuracy of 3% and an average error of 3.33. The significant improvement of naive Bayes can be seen in this comparison, especially in the reduction of average error by around 60%. Seeing as the classifications are along a scale, this statistic is arguably more important than just the accuracy, as getting close to a desired result can be considered better than missing the mark by a large margin. In this, the benefits of using the naive Bayes classifier, even with evidence that is not wholly predictive, can be seen

It would be interesting to see how much the classifier could improve if the dataset contained more predictive data, such as a game's developer or the number of copies it sold. Unfortunately, in the time constraints of this project I could not find a dataset that had all the information I wanted, and did not have the time to source the data from multiple sources and verify the correctness of merging it all together.

Both the massaged dataset (as a JSON file) and the Python 3 code that I wrote to implement naive Bayes are attached to this submission.