Walmart CodeSprint (Machine Learning)

This document serves as documentation for the Hackerrank ML competition. It will provide a brief summary of how the dataset was processed and how machine learning algorithms were chosen.

1. **"Write a few lines about training dataset quality and any errors found in the training dataset"**
    - The training set that was given provided information about each of the products. Some of the features that were used to describe the products were the seller name, product name, item class id, publisher, etc. Given the diverse nature of the dataset, it is natural that some of the features will only have values for a small percentage of the products (For example, the MPAA Rating feature will only be filled in for those products that are movies or TV shows). Other than a couple lines that didn't look right in Excel (might have been a problem with formatting), the training dataset was able to convey a lot of information about the different products. Something that was frustrating to see was that the test and train datasets had different features (For example, Test had "Product Long Description" and "Product Short Description" while Train did not).

2. **"Explain the data preprocessing steps"**
    - The basic idea behind my data preprocessing pipeline is taking all of the string values and representing them numerically through categories. First, I took a look at all the features that each product had and made decisions on which features to keep and which to remove. I decided to remove Item_id, Color, Recommended Room, Synopsis, and Actual_color. I chose these because they were either too hard to parse through (future work would definitely include these) or they were extremely similar to another category. Then, for the categories of Artist ID, Genre ID, ISBN, Literary Genre, Recommended Location, Publisher, and Recommended Use, I changed the string values to 1 and 0, depending on whether or not the value for that feature was null or not. The reasoning behind this is that if, for example, the value for Artist ID was a not null value, then a 1 representation would convey information about that feature (convey the info that it might be a music related product). If it was a null value, then the 0 representation shows that it is *not* a music related product. Then, for MPAA Rating, Actual Color, Item Class ID, Seller, and Product Name, I introduced multi class categories for the most frequent values in each of the columns (For example, assigning the

value 2 for Walmart.com in Sellers). After all the preprocessing, Xtrain is a 10320 x 12 array. For Ytrain, I just assigned each product with the first tag that it was associated with (future work would include trying to incorporate information about the other tags), so Ytrain is a 10320 x 1 array. Since there are 10,374 (readable) test examples, Xtest is a 10374 x 12 array.

3. **"Explain and justify the model you've chosen for the prediction"**
    - I used a K Nearest Neighbors classifier to make the tag prediction for each example in the test set. I used KNN because the training dataset had a lot of examples (N > 10,000), but it was still a relatively low dimensional dataset where there were less than 15 features per example. For these 2 reasons, I decided to use a simple ML approach rather than trying to implement anything with deep learning or SVMs (which work better with high dimensional spaces).