Processing Steps

1.) Load the train.tsv file into a pandas dataframe and fill any NA cells

2.) Select feature columns of interest (Name, Long Description) for the x variable and the tag column for the y variable

3.) Clean out the feature columns using regular expressions (mainly to remove HTML tags) and combine the columns into one list of document strings

4.) Fit and transform the cleaned feature strings using the sklearn TfidfVectorizer (combined CountVectorizer and TfidfTransformer)

Custom parameters used in TfidfVectorizer:

ngram_range = (1,2), use_idf=True

Model Used:

Stochastic Gradient Descent Classifier with custom parameters :

n_iter=125, loss='log', alpha (learning rate): 1E-6

Justification for using SGD Classifier:

By performing the 'bag of words'/tf-idf transformation on the documents, each item can be assigned shelf tags according to its respective tf-idf values. And with values between 0 and 1, this classification task can be done using linear methods like SVM or logistic regression. And since the loss functions for these classifiers are differentiable, one can use stochastic gradient descent to minimize the loss function and improve classification.

Justification for the customized parameters:

By manual trial and error, I found that this combination of parameters for the tfidf vectorizer and sgd classifier yielded the highest F1 score for the test set. One could also use grid search to find the optimal set of parameters, but in this case it would be very computationally expensive to search through all the possible combinations.