Machine Learning

CS-583 DMTM Project 2: Sentiment Analysis

# Phase 1

## Stage I

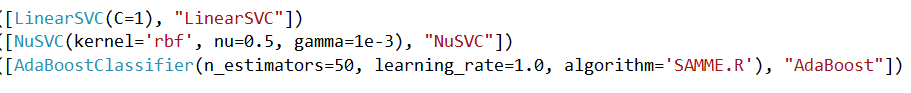
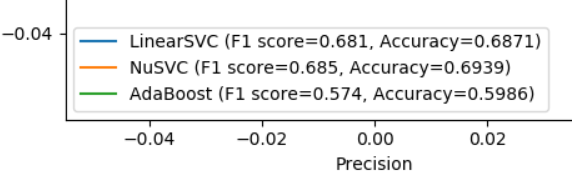
1. Loaded data using Pandas.read\_data()
2. In Preprocessing
   1. Throw away punctuations.
   2. Use SnowBall stemmer with stopwords: English.
   3. TODO: consider pickle dumping.
   4. TF-IDF Vectorization
      1. TODO: Use 2-grams
3. Resolved issue with multiclass F1. Used f1-score(average=’weighted’)
4. Issue with precision\_recall\_curve() not supporting multiclass.
5. Used only ‘document’ and ‘label’ columns for now.
6. Used pyplot to write plotting library.
7. Initial run o/p

Processing 2203 samples with 5 attributes  
Evaluating classifiers

Plotting the results

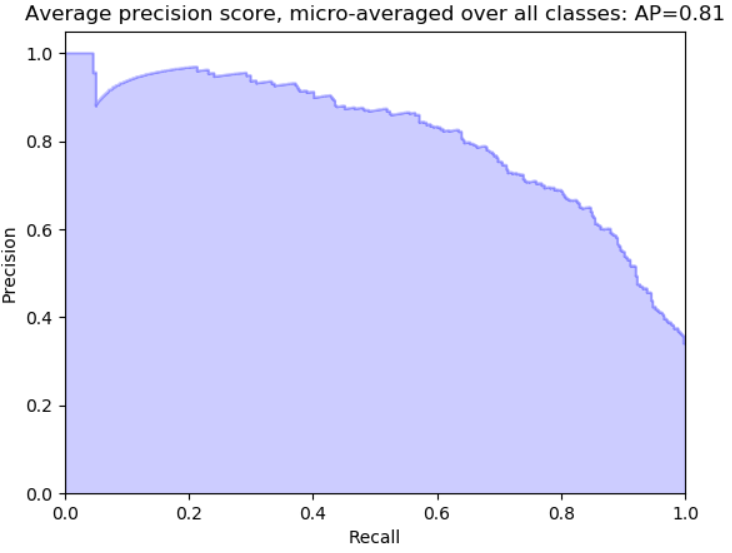
Accuracy of clf: Linear SVC (F1 score=0.721) = 0.7188208616780045

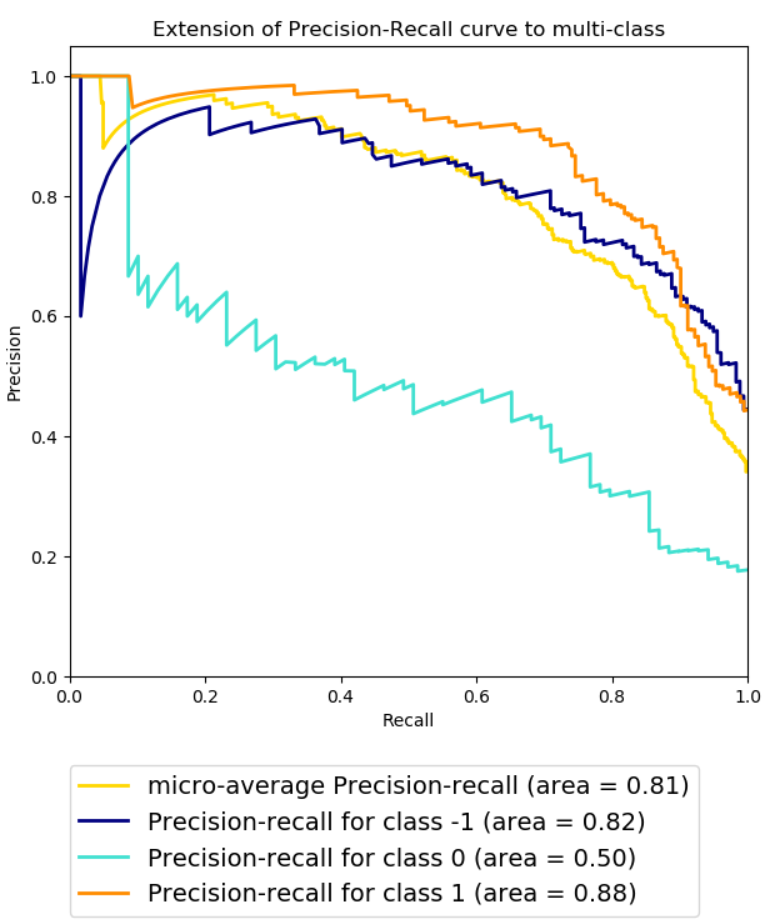
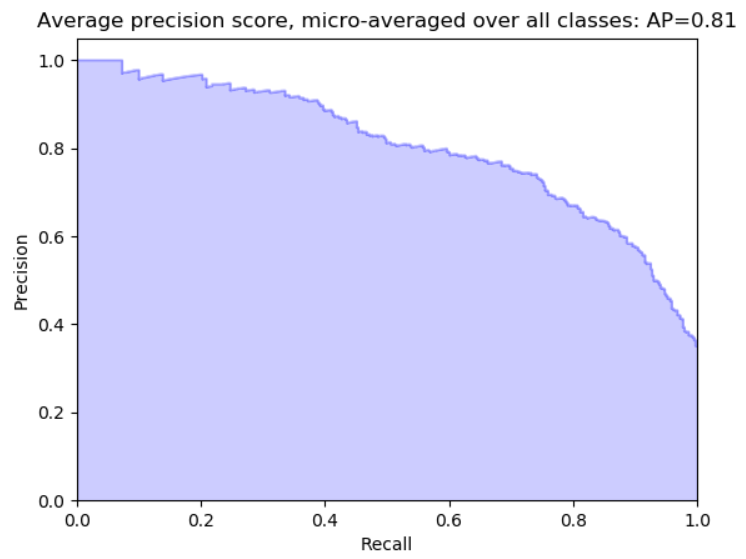
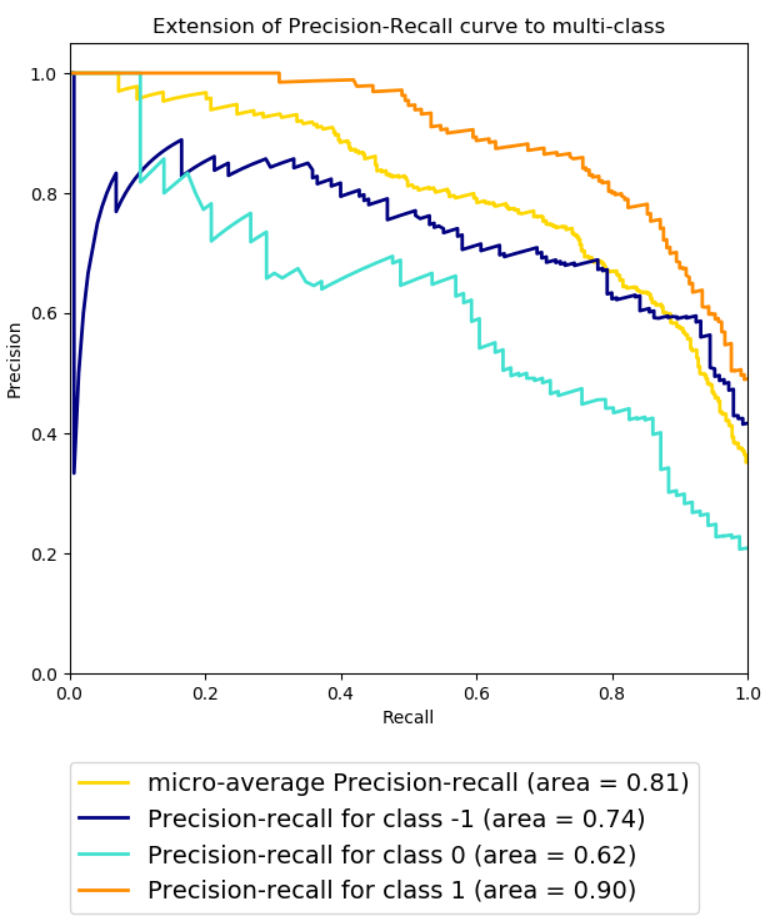
Accuracy of clf: NuSVC (F1 score=0.729) = 0.7301587301587301

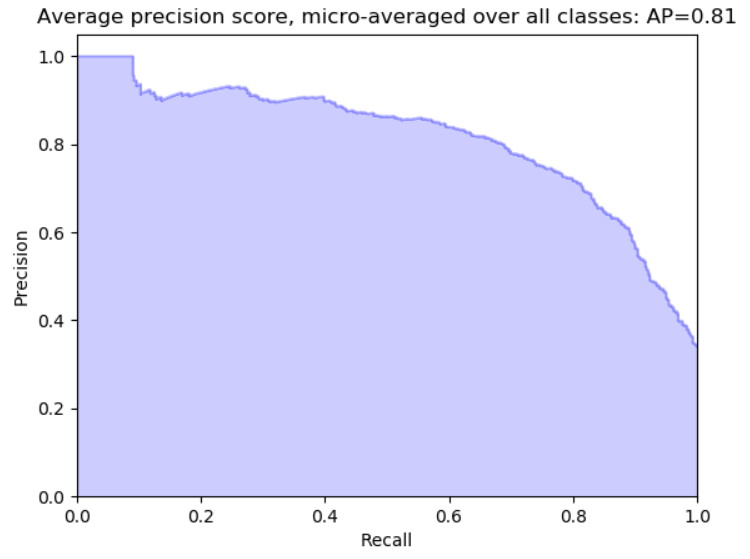
Accuracy of clf: Ada Boost (F1 score=0.589) = 0.6009070294784581  
  
  
  


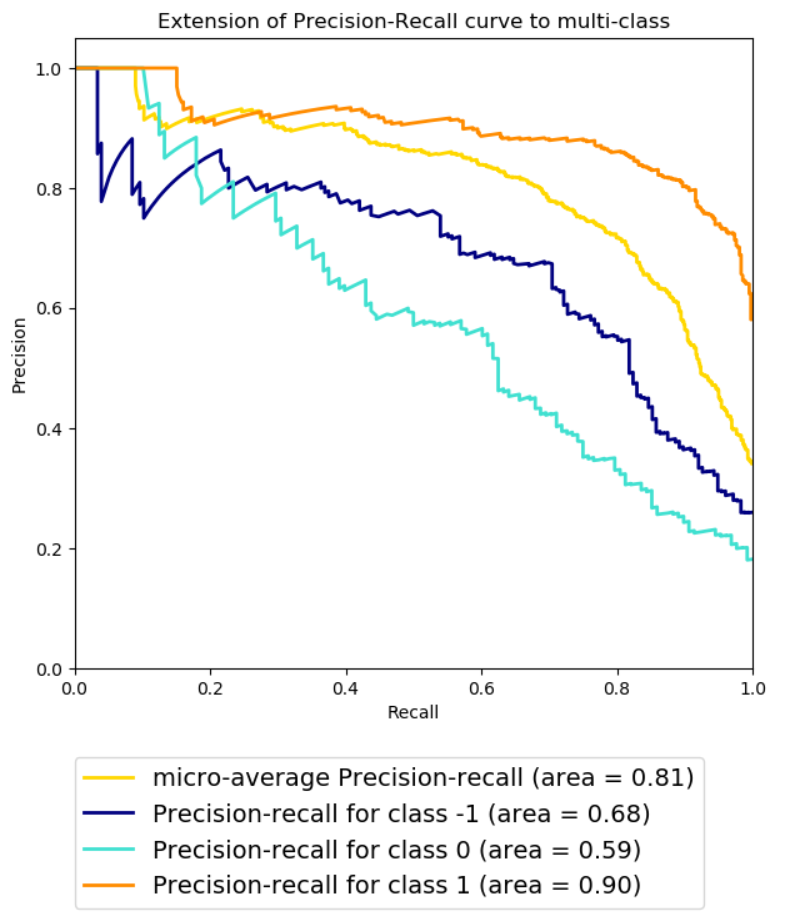
## Stage II

* 1. Implemented multiclass Pre/Re
     1. Used label binarizer
     2. Used OVR with LinearSVC (default params)
     3. Used Micro-Averaged PR
  2. CV
     1. Single (80-20) fold for now.
     2. TODO: 10-fold CV
  3. Uses average weighted F1, and overall accuracy.
  4. o/p  
     **OVR LinearSVC (F1 score=0.725, Accuracy=0.6644)**

**Average precision score, micro-averaged over all classes: 0.81**  
  


* 1. 
  2. A comparison w/o Stemmer
     1. **OVR LinearSVC (F1 score=0.719, Accuracy=0.6667)**
     2. **Average precision score, micro-averaged over all classes: 0.81**
  3.   
       
     
  4. o/p from data 2 (with stemming)  
     **OVR LinearSVC (F1 score=0.725, Accuracy=0.6907)**

**Average precision score, micro-averaged over all classes: 0.81**  
  




* 1. TODO:
     1. Sample data visualization
     2. Feature engineering
        1. Subject-predicate identification (Target-aspect).
        2. Context-sensitive weighting vectorizer.
     3. Outlier removal
     4. Scaling (? Is this really relevant in text classification?)
        1. Centering
        2. Min-Max
        3. PCA (reduction)
     5. More classifiers (KNN, NB, SVC (poly/rbf), RandomForest)
     6. Hyper-parameter tuning
        1. GridSearch
        2. Manual
     7. NN / Deep Learning