CIS 530 - HW2

Team Members:

1. **Srinivas Suri**
2. **Ipek Onur**

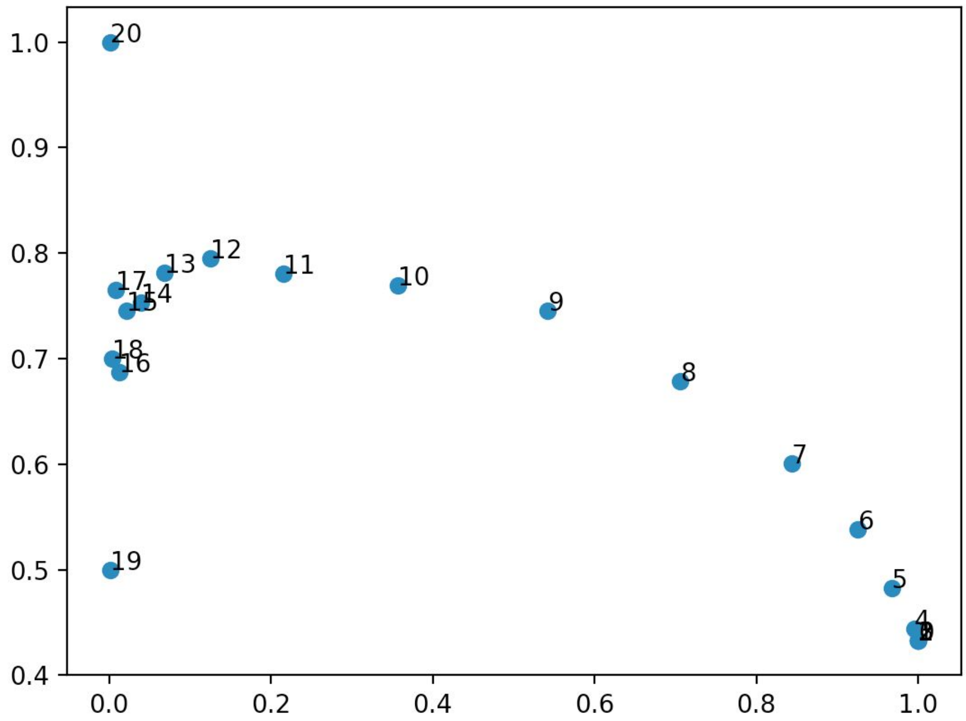
PART 2: Baselines

For all the baselines, we tuned the parameter on the training data

1. **Majority Class Baseline**
    - **Train**
        - Recall: 1
        - Precision: 0.43275
        - F Score: 0.60408
    - **Dev**
        - Recall: 1
        - Precision: 0.418
        - F Score: 0.5895

2. **Word Length Baseline**
   Tried threshold values in the range [0,20] and threshold = 7 gave the highest F Score
    a. **Train**
        - Recall: 0.84402
        - Precision: 0.6007
        - F Score: 0.70189

    b. **Dev**
        - Recall: 0.86602
        - Precision: 0.60535
        - F Score: 0.71259



3. Word Frequency Baseline

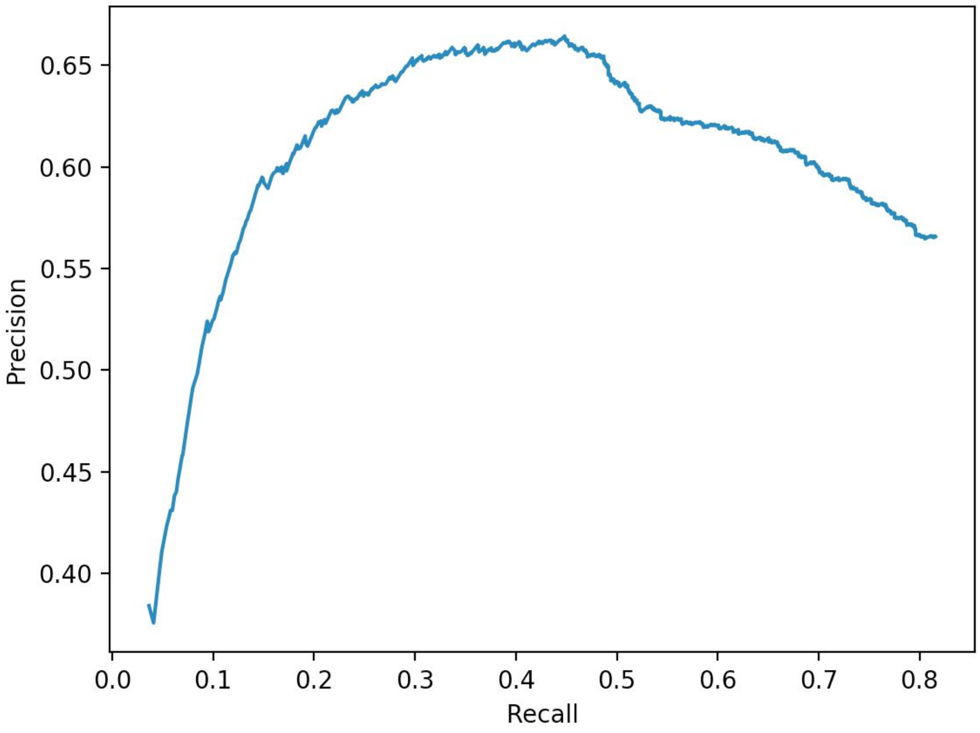
We realized that max frequency is 47,376,829,651 and we looked at the range [0,47376829651] with high skip value 10000000. Best threshold was 20000000 so we looked at the range [0,40000000] with skip value 10000. Best threshold was 19840000. Finally we looked at [19840000, 19900000] range with skip value 5000. We decided that best threshold is **19840000**

For the words that didn't occur in the counts dictionary, we assumed them to be simple

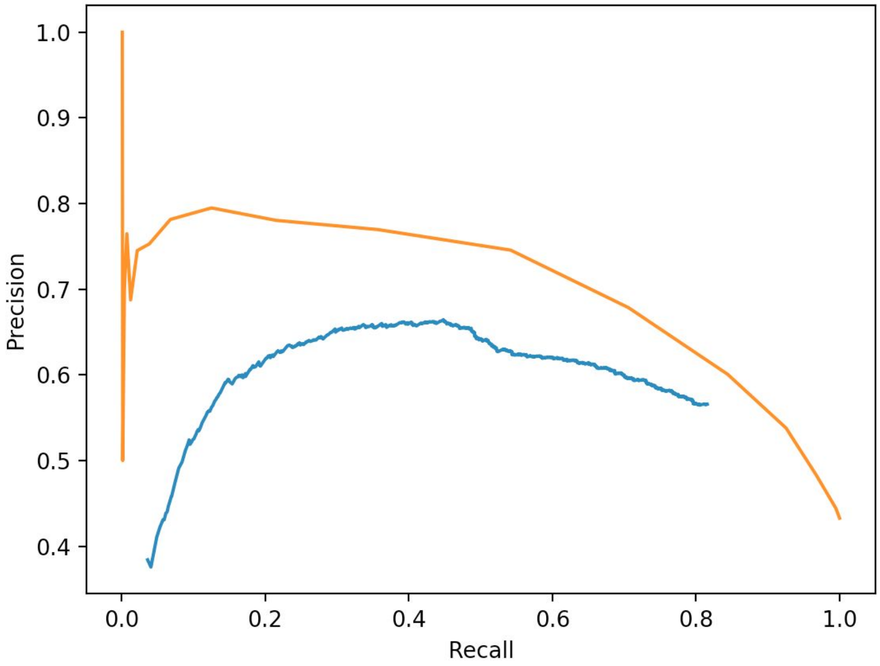
a. **Train**
   - Recall: 0.81571
   - Precision: 0.5657
   - F Score: 0.66808

b. **Dev**
   - Recall: 0.84449
   - Precision: 0.5567
   - F Score: 0.67110

This graph is for the range [0, 19900000] with skip value 5000



For the combined graph:



PART 3: Classifiers

1. **Naive Bayes**
    a. **Train**
        - Recall: 0.97978
        - Precision: 0.49503
        - F Score: 0.657746
    b. **Dev**
        - Recall: 0.968899
        - Precision: 0.46929
        - F Score: 0.6323

2. **Logistic Regression**
    a. **Train**
        - Recall: 0.6580
        - Precision: 0.7250
        - F Score: 0.68988
    b. **Dev**
        - Recall: 0.6937
        - Precision: 0.726817
        - F Score: 0.709914

NAIVE BAYES PERFORMANCE VS LOGISTIC REGRESSION PERFORMANCE

We observe that on both the training and test data, Logistic Regression performs better as it has higher F Scores. We can explain the performance difference w.r.t. The algorithms' natures. Naive Bayes expects features to be conditionally independent however we can't claim that certainly. There might be a correlation as shorter words are preferred over longer words so they have a higher frequency. Logistic Regression on the other hand, can handle some of this correlation because it doesn't make independence assumption.

PART 4: Build your own Models

We have tried the following features and trained the below mentioned classifiers. The results are reported below.

#Feature description:

#1. word length
#2. word frequency
#3. sentence length
#4. avg. word length in sentence
#5. avg. word frequence
#6. length of the biggest word in the sentence
#7. no. of syllables in the sentence
#8. avg. no. of syllables in the sentence
#9. max syllable count for a word in the sentence

Our Intuition for choosing these features:

- Our intuition is that words and the sentences where these words occur are correlated with each other. So, we have built a feature set that's based on both the Word complexity and built several features related to the corresponding sentence.
- We have also built several features around the syllables of the words occuring in the sentence.

Table: Classifier Results

Classifier	Train Precision	Train Recall	Train FScore	Dev Precision	Dev Recall	Dev FScore	General Comments
Naive Bayes	0.500	0.973	0.661	0.478	0.964	0.639	Default Parameters
Logistic Regression	0.725	0.657	0.689	0.725	0.688	0.706	Default Parameters
Random Forests	0.724	0.665	0.693	0.712	0.686	0.699	Estimators - 50 Max depth - 3
Decision Trees	1.0	0.99	0.997	0.653	0.691	0.672	Default Parameters
Ada Boost	0.747	0.778	**0.762**	0.721	0.803	**0.760**	Estimators - 50
MLP	0.710	0.712	0.711	0.707	0.765	0.735	Alpha = 0.01 (10,5) hidden layers
SVM Linear	0.721	0.666	0.692	0.718	0.696	0.707	C = 1

SVM RBF	0.735	0.692	0.713	0.696	0.681	0.689	C = 5

Error Analysis:

We have presented few words that our best classifier (AdaBoost) has correctly/ incorrectly classified below:

Truly Simple	bay,personal,fled,math,water,fitted, given,rule,opened,fathers
Truly Complex	painfully,enthralled,biosphere,premature abandoned,protester-erected,worded curriculums,cyberbullying,determined Fashioned,dexterity,economist, participates
Simple but predicted as complex	evocative, university eighth-grader, cheating, decisions, editor-in-chief, ballots, near-vertical, beautiful, advances, fourth-round
Complex but predicted as simple	tenants,cottage,analyst,relics, materials,humanity,requiring

1. Our Classifier misclassifies a combination of multiple words with hyphen ‘-’ as complex although a lot of words in this category belong to simple category. For example, out classifier made errors to identify eighth-grader, near-vertical, fourth-round and editor-in-chief although these words belong to the Simple Category. As our feature set has around 9 features and 50 boosted trees make the classification choice,it becomes a bit difficult and non-intuitive to find an exact reason for this behavior. However, our intuition is as follows, a lot of complex words have hyphen ‘-’ between words in them and thus our best classifier identifies simple words with hyphen between them as complex.

2. Our classifier misclassifies complex words tenants, cottage, analyst , expand... as simple, one reason why this might be happening is because these words have higher syllable count. Since, we trained the model with syllable features as well, we suspect that this is the reason why the classifier is considering these complex words as simple. However, that being said, we are not completely sure as it is difficult to pinpoint the exact error as the Adaboost takes the average vote from 50 different simple classifiers.