2-(a)

Factored joint distribution P (X_1, X_2, … X_p, C) = P(C) * P(X_1 | C) * P(X_2 | C) … P(X_p | C) .

2-(b)

Among the words in data, the words which are contained in most frequent 3,000 words of randomly chosen dataset which is composed of 16,000 reviews with 3-10 "votes" gonna be the set of parameters that need to be estimated.

For a single classification task named "funny", specify maximum likelihood estimate.

Ex) 5 training set 1. {fun, couple, love, love} → funny

2. {fast, furious, shoot} → not funny

3. {couple, fly, fast, fun, fun} → funny

4. {furious, shoot, shoot, fun} → not funny

5. {fly, fast, shoot, love} → not funny

If we want to estimate a test data {fast, furious, fun, apple}, at first, remove words that do not exist in any training set. Because they are not most frequent 3,000 words.

Then, get P(fast, furious, fun, "funny") , and P(fast, furious, fun, "not funny") .

P("funny") * P(fast | "funny") * P(furious | "funny") * P(fun | "funny") = 0

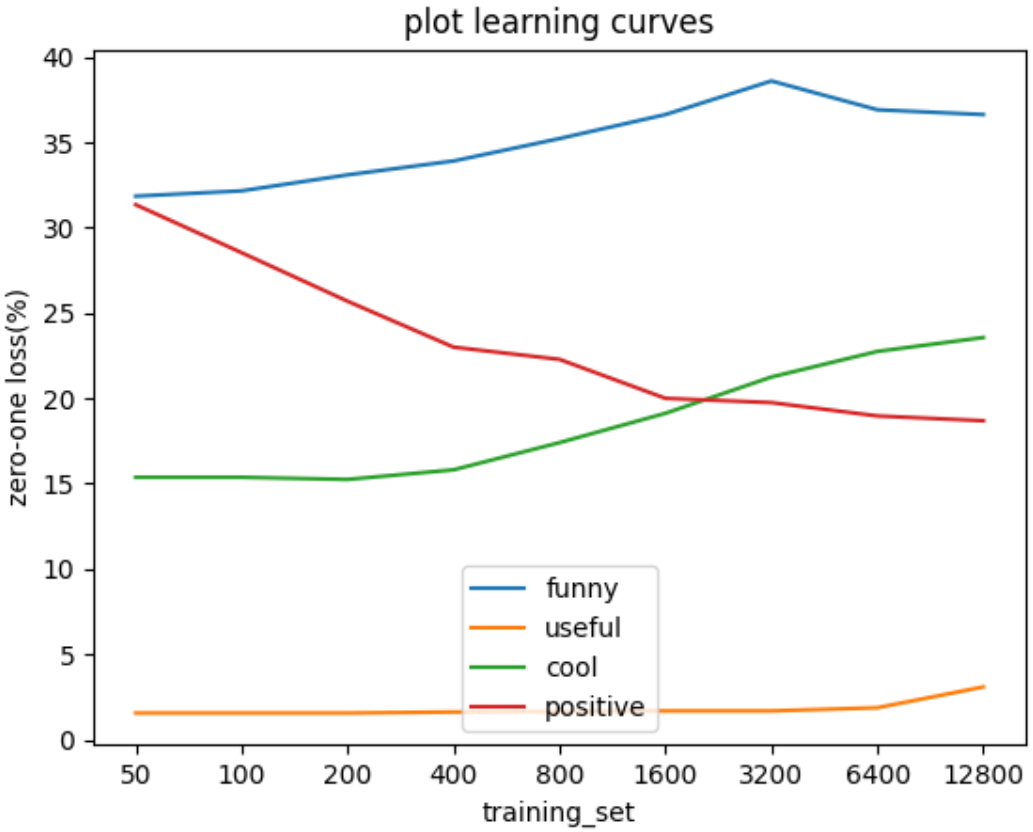
P("not funny") * P(fast | "not funny") * P(furious | "not funny") * P(fun | "not funny") = 0.018 (just assumption)

0 < 0.018 , so it can be classified to "not funny".

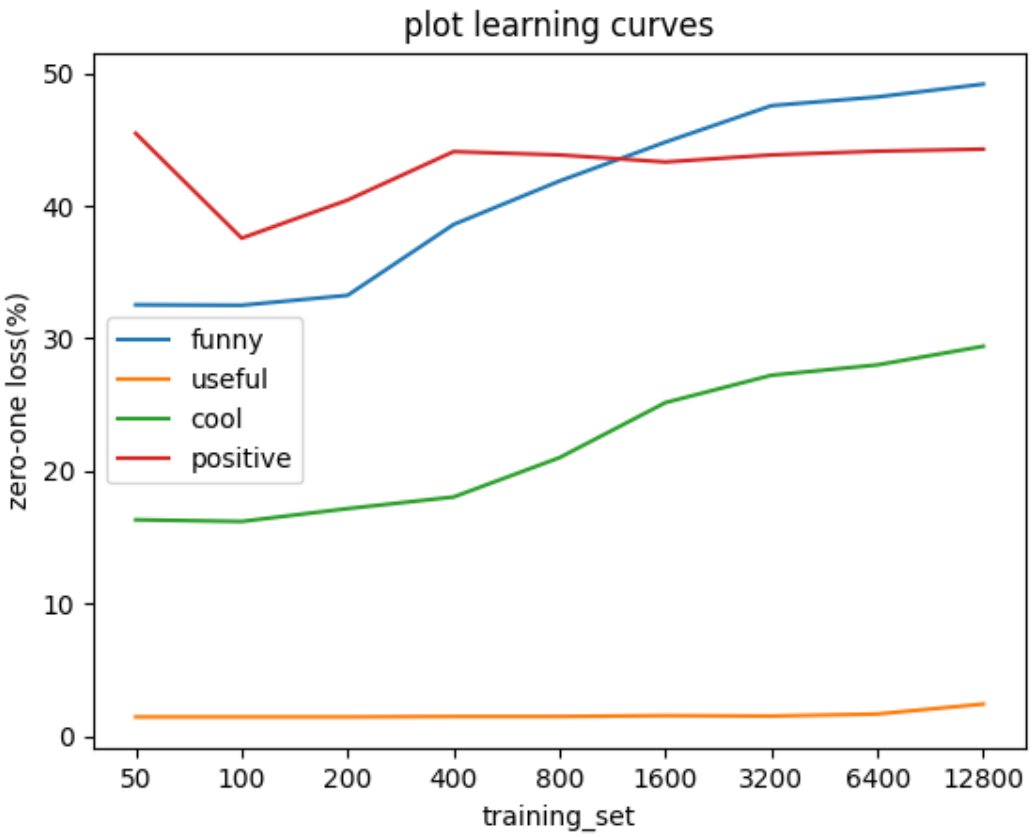
Probability of first one is zero, because of P(furious | "funny"), so although other words indicate to class "funny", the ultimate probability goes zero. Therefore, we need Laplace smoothing.

If P(furious | "funny") = 0 / 9 = 0, add the number of unique words in training set to denominator(분모), and add 1 to numerator(분자) . (0+1) / (9+7) = 1 / 16 . We can avoid the situation that some probability goes to zero.

3-(c)



This is plot curves of my learned classifier. In case of class 'useful', my classifier shows more than 97% of prediction accuracy regardless of the training set size. Also in case of class 'positive', it shows great effect of learning by decreasing zero-one loss. However, in class 'funny' and 'cool', the bigger size of training set, getting worse accuracy of prediction.



This is plot graph of my baseline default model. I just predict all of reviews to most frequent class label. That is, if 30 reviews of 50 are predicted to 'funny' and 20 are predicted to 'not funny', just deal with 'funny' of all 50 reviews. In case of class 'useful', result is very similar to my learned classifier. Even when training set size is 12800, baseline model shows little bit higher accuracy than my classifier. I think the reason is that almost of reviews in dataset have class 'useful', so foolishly predict to just 'useful' may show perfect accuracy. In case of class 'funny', zero-one loss goes up to 50% while my classifier's zero one loss goes stable in 35% zero-one loss. In case of class 'positive', it shows consistently higher zero-one loss than my classifier. In case of class 'cool', it seems similar zero-one loss to my classifier in first training set, size 50. However, like class 'funny', zero-one loss goes up to 30% as training set size increases when my classifier have 24% of zero-one loss.