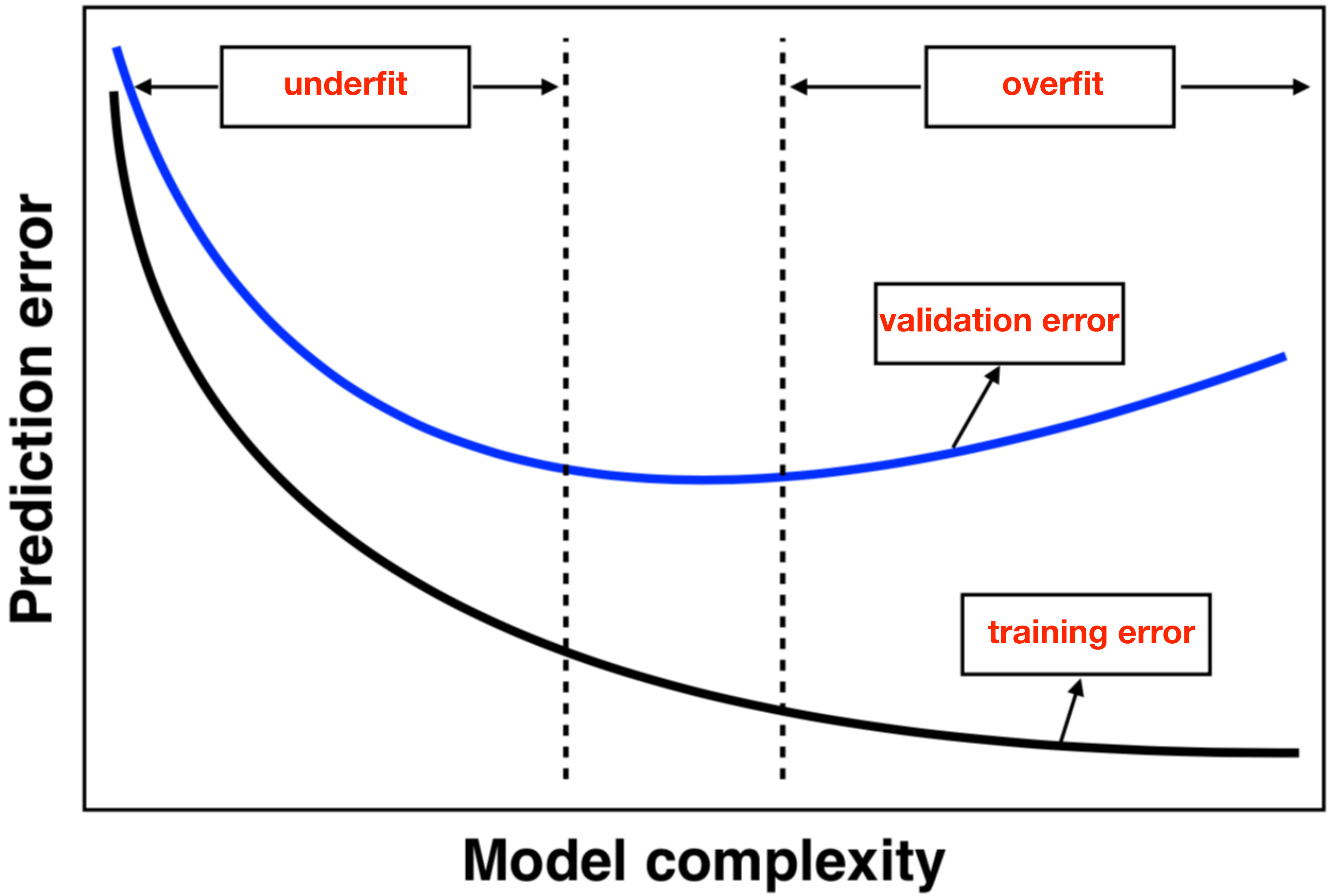
CS5785 Practice Prelim Solutions

Yichun Hu

Q1 Short Answer

Question	Answer
A	TRUE
B	FALSE (should be Var[X] + 4Var[Y])
C	TRUE
D	Lots of submissions led to overfitting to the public set
E	Problematic: model could be overfitting to noise, need to validate
F	Problematic: could have very poor precision (i.e. low accuracy on positive examples)
G	Problematic: did parameter selection on full data not just train, so going to be biased

Q2 Training and Validation



Q3 Regularized Regression-(a/b)

(a) Larger lambda = simpler model.

(b) 0.3. From Lecture note 6,

The AML approach (AKA "one-std-err" rule of thumb)

$$\text{Std Err}(\hat{R}^{cv}(\lambda)) = \frac{1}{K}\sqrt{\sum_j (\hat{R}^{cv}(\lambda) - cv^{(j)}(\lambda))^2}$$

Pick the "simplest" algo w/ $\hat{R}^{cv}$ within one std err of the minimal one

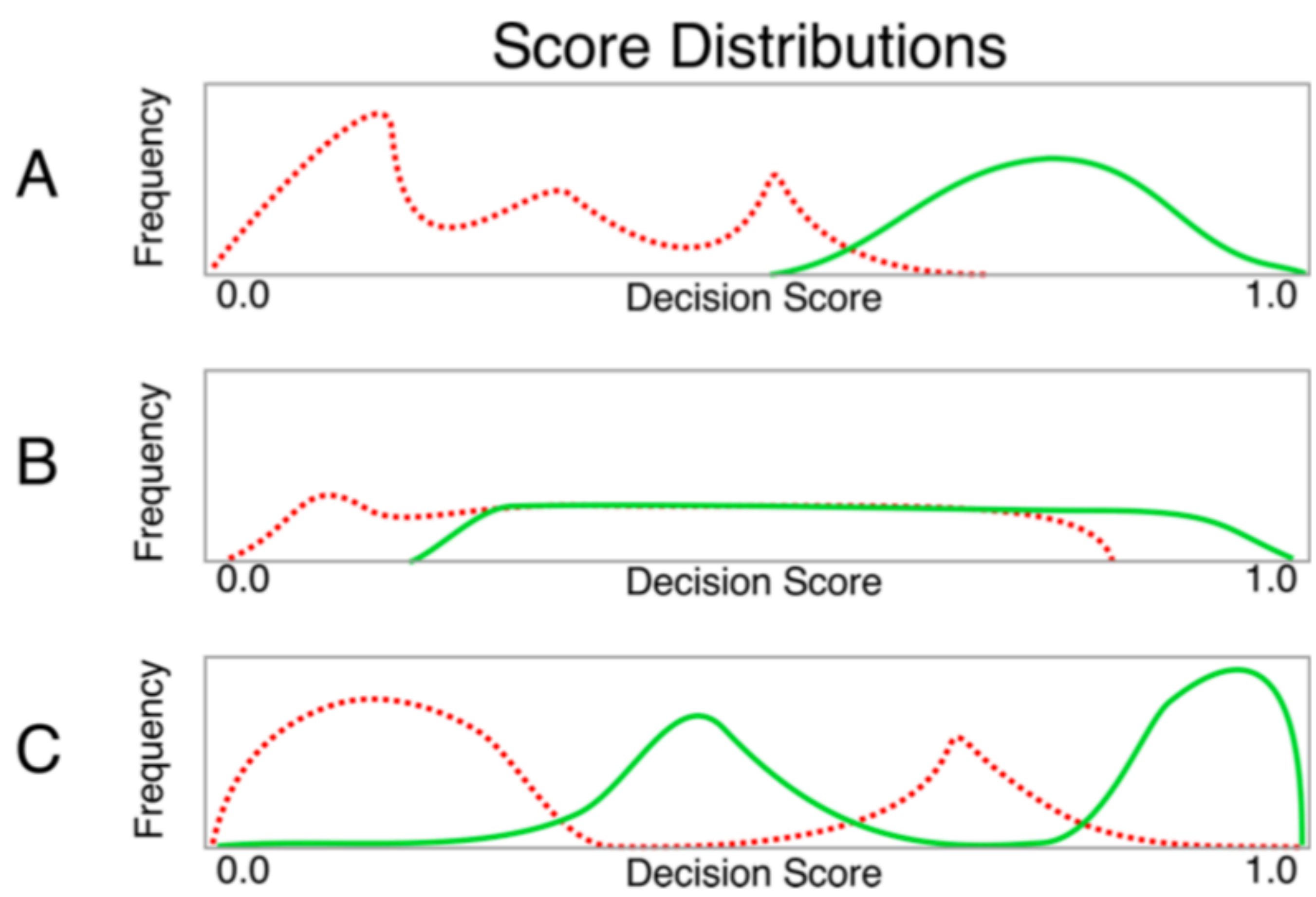
Q3 Regularized Regression-(c/d/e)

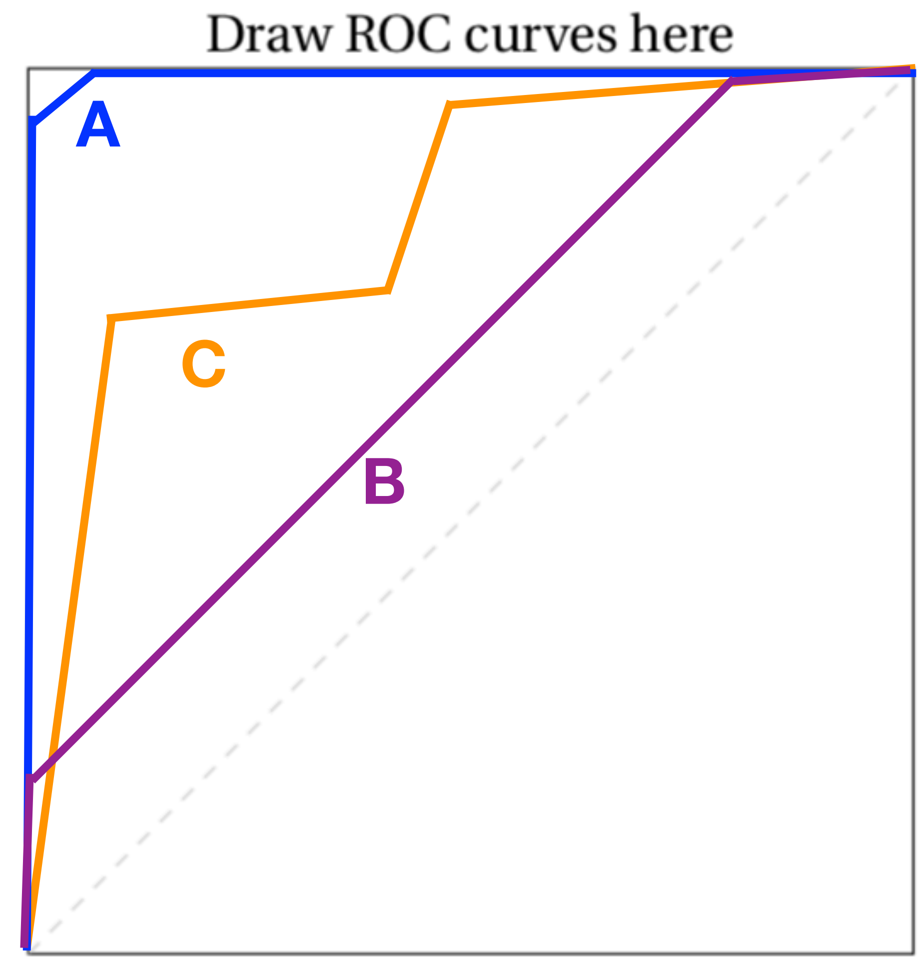
(c) 0.

(d) less.

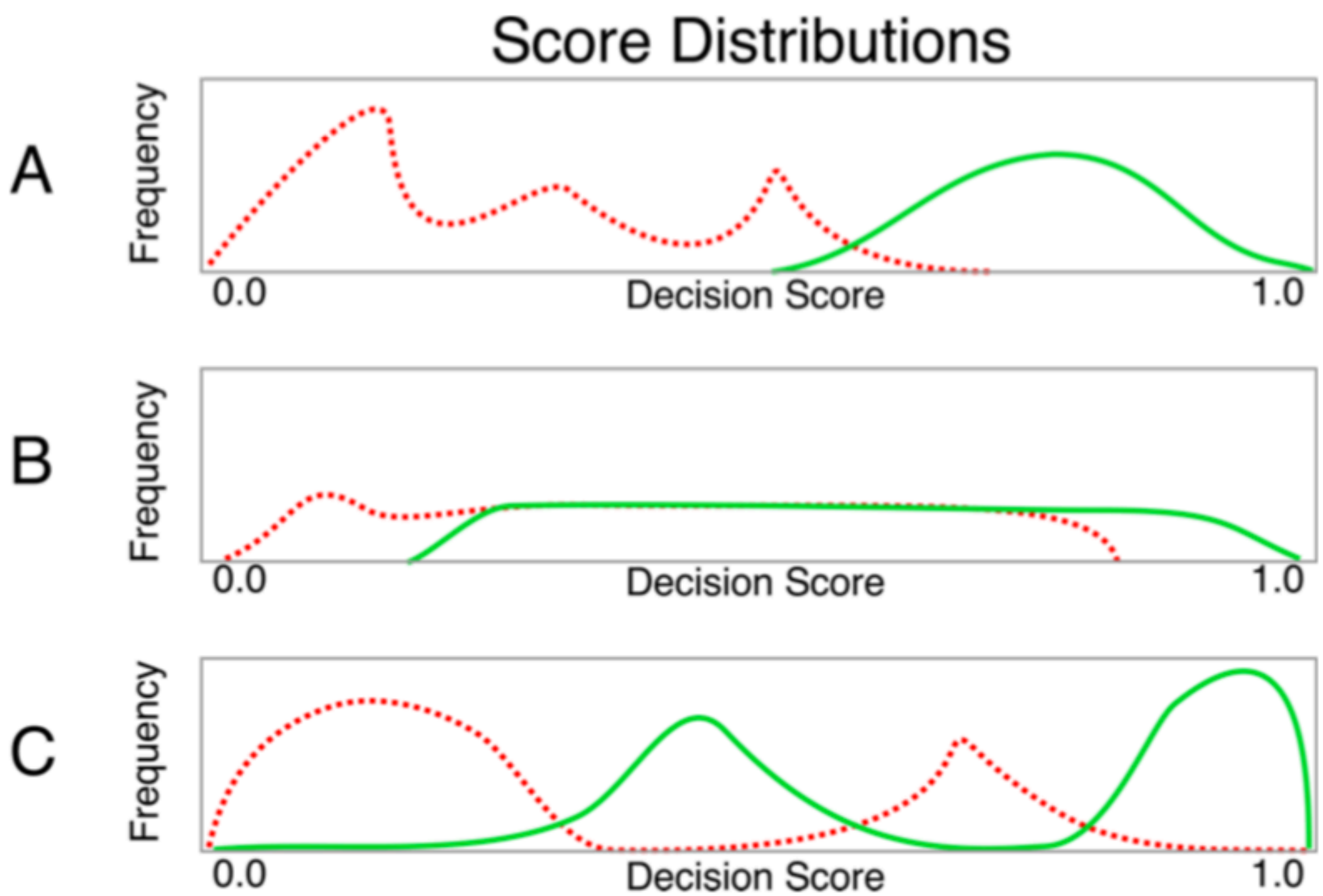
(e) 0.5 underfit, 0 overfit.

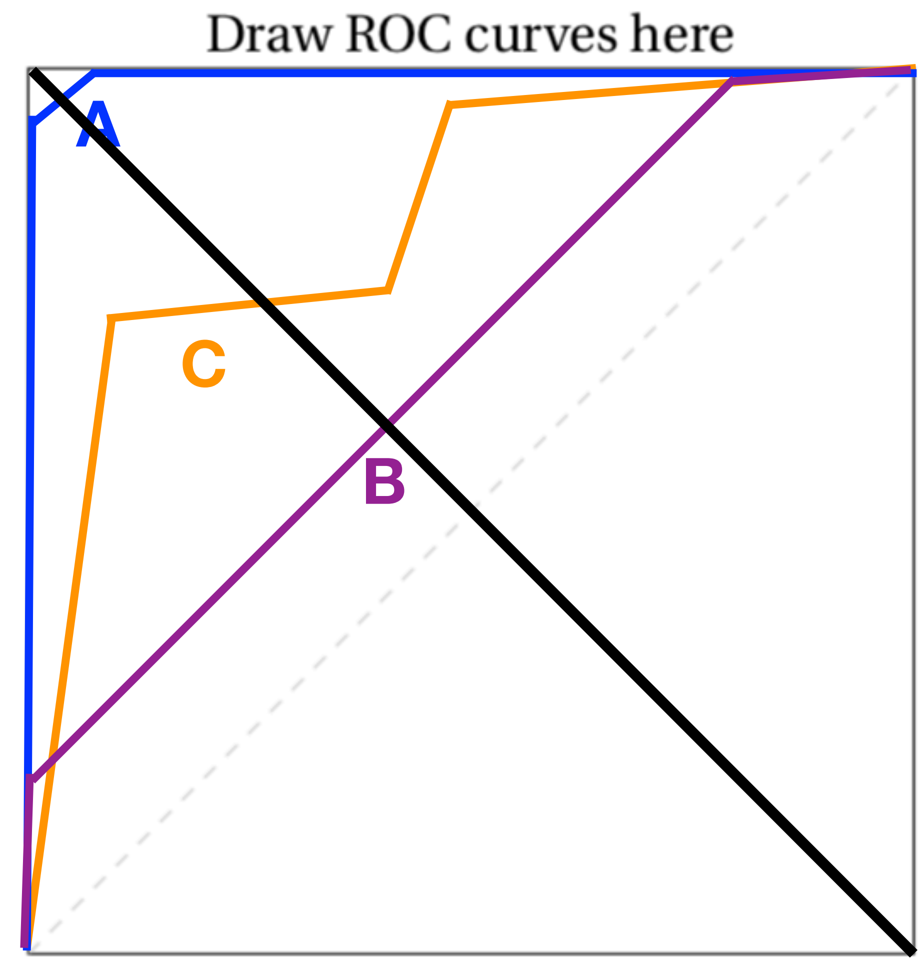
Q4 ROC Curves and Score Distributions - (a)





Q4 ROC Curves and Score Distributions - (b)





A) Misclassification rate ~ 0.02
B) Misclassification rate ~ 0.40
C) Misclassification rate ~ 0.25

Q4 ROC Curves and Score Distributions - (c/d)

C: optimize for lowest *false positive rate* (don’t want break ins to vault)

D: optimize for highest *recall = TPR* (don’t want any infected food to pass)

Q5 Bayes Law - (a)

	Predicted Rotten	Predicted Good	Total
Is Rotten			1,000
Is Good			999,000
Total			1,000,000

“One in one thousand bananas is infested”

Q5 Bayes Law - (a)

	Predicted Rotten	Predicted Good	Total
Is Rotten	990	10	1,000
Is Good			999,000
Total			1,000,000

“99% of rotten bananas are detected as rotten”

Q5 Bayes Law - (a)

	Predicted Rotten	Predicted Good	Total
Is Rotten	990	10	1,000
Is Good	49,950	949,050	999,000
Total			1,000,000

"95% of good bananas are detected as good"

Q5 Bayes Law - (a)

	Predicted Rotten	Predicted Good	Total
Is Rotten	990	10	1,000
Is Good	49,950	949,050	999,000
Total	50,940	949,060	1,000,000

check totals

Q5 Bayes Law - (b)

$$P(\text{bad} \mid \text{marked bad}) = \frac{P(\text{marked bad} \mid \text{bad})P(\text{bad})}{P(\text{marked bad})}$$

Q5 Bayes Law - (b)

$$P(\text{bad} \mid \text{marked bad}) = \frac{P(\text{marked bad} \mid \text{bad})P(\text{bad})}{P(\text{marked bad})}$$

$$= \frac{0.99 * 0.001}{0.99 * 0.001 + 0.05 * 0.999}$$

$$\approx 0.019$$

Q5 Bayes Law - (c)

$$\mathbb{E}[\text{Profit}] = \mathbb{E}[\text{Profit} \mid \text{Bad}]P(\text{Bad}) + \mathbb{E}[\text{Profit} \mid \text{Good}]P(\text{Good})$$

Q5 Bayes Law - (c)

$$\mathbb{E}[\text{Profit}] = \mathbb{E}[\text{Profit} \mid \text{Bad}]P(\text{Bad}) + \mathbb{E}[\text{Profit} \mid \text{Good}]P(\text{Good})$$

$$= -499 * 0.001 + 1 * 0.999$$

$$= 0.5$$

Q5 Bayes Law - (d)

$$\begin{aligned}\mathbb{E}[\text{Profit}] = {} & P(\text{Bad, Predicted Good})\mathbb{E}[\text{Profit} \mid \text{Bad, Predicted Good}] \\ & + P(\text{Good, Predicted Good})\mathbb{E}[\text{Profit} \mid \text{Good, Predicted Good}] \\ & + P(\text{Predicted Bad})\mathbb{E}[\text{Profit} \mid \text{Predicted Bad}]\end{aligned}$$

Q5 Bayes Law - (d)

$$\begin{aligned}\mathbb{E}[\text{Profit}] = {} & P(\text{Bad, Predicted Good})\mathbb{E}[\text{Profit} \mid \text{Bad, Predicted Good}] \\ & + P(\text{Good, Predicted Good})\mathbb{E}[\text{Profit} \mid \text{Good, Predicted Good}] \\ & + P(\text{Predicted Bad})\mathbb{E}[\text{Profit} \mid \text{Predicted Bad}]\end{aligned}$$

$$= \frac{10}{1,000,000} * -499.2 + \frac{949,050}{1,000,000} * 0.8 + \frac{50,940}{1,000,000} * -0.2$$

$$\approx 0.744$$

Q6 Naive Bayes with Bag of Words - (a)

	win	score	learning	deep	loss
Sports 1	1	1	0	0	0
Sports 2	0	0	1	1	1
Sports 3	1	1	0	0	1
ML 1	0	1	1	1	1
ML 2	1	0	0	1	0
ML 3	0	0	1	0	0

Q6 Naive Bayes with Bag of Words - (b)

P(Feature | Class)

	win	score	learning	deep	loss
Sports	2/3	2/3	1/3	1/3	2/3
ML	1/3	1/3	2/3	2/3	1/3

	win	score	learning	deep	loss
Tweet	1	0	1	1	1

$$P(\text{class}|x) \propto P(x|\text{class})P(\text{class})$$

Q6 Naive Bayes with Bag of Words - (b)

P(Feature | Class)

	win	score	learning	deep	loss
Sports	2/3	2/3	1/3	1/3	2/3
ML	1/3	1/3	2/3	2/3	1/3

	win	score	learning	deep	loss
Tweet	1	0	1	1	1

$$P(\text{class}|x) \propto P(x|\text{class})P(\text{class})$$

$$P(\text{class}_{\text{sport}}|x) \propto 2/3 * (1 - 2/3) * 1/3 * 1/3 * 2/3 * 1/2 = 4/486$$

$$P(\text{class}_{\text{ML}}|x) \propto 1/3 * (1 - 1/3) * 2/3 * 2/3 * 1/3 * 1/2 = 8/486$$

Q6 Naive Bayes with Bag of Words - (b)

P(Feature | Class)

	win	score	learning	deep	loss
Sports	2/3	2/3	1/3	1/3	2/3
ML	1/3	1/3	2/3	2/3	1/3

	win	score	learning	deep	loss
Tweet	1	0	1	1	1

$$P(\text{class}|x) \propto P(x|\text{class})P(\text{class})$$

$$P(\text{class}_{\text{sport}}|x) = 1/3$$

$$P(\text{class}_{\text{ML}}|x) = 2/3$$

Q6 Naive Bayes with Bag of Words - (c)

$$X = UDV^T = \sum_k d_k u_k v_k^T$$

In above sum each **u** is an n-dimensional vector, each **v** is a p-dimensional vector

Approximate sum by just keeping first 300 entries in sum (those with highest **d** values)

The first 300 **u** vectors give the n 300-dimensional document vectors, and the first 300 **v** vectors give the p 300-dimensional word vectors

This is mathematically equivalent to taking the first 300 columns of **U**, and the first 300 columns of **V**

Q6 Naive Bayes with Bag of Words - (d)

$$X \approx U[:, :300]D[:300, :300]V[:, :300]^T$$

Approximate by taking just the first 300 columns of **U** and **V**, and the first 300 entries in **D**

Q6 Naive Bayes with Bag of Words - (e)

$$Loss = ||X - X^{\text{reconstructed}}||_F$$

$$= \sum_{i,j} (X_{ij} - X_{ij}^{\text{reconstructed}})^2$$

What We’ve Learned
So Far…

- Bayes rate - best possible risk
- Confusion matrix, accuracy, precision, recall, ROC curve - measure the performance of our classifier
- Linear regression - OLS
- Logistic regression - log odds, maximum likelihood
- Subset selection, cross validation - choose parameters/ algorithms
- Shrinkage - ridge, lasso
- Naive Bayes - independent <u>given Y,</u> bag of words

- Kernel density estimation
- SVD, PCA
- K-means
- Gaussian mixture model, EM algorithm
- Similarity, multidimensional scaling