From the equations presented below, express the probability of a tweet being positive given that it contains the word happy in terms of the probability of a tweet containing the word happy given that it is positive

$$P(\text{Positive}|\,\text{``happy''}) = \frac{P(\text{Positive} \cap \,\text{``happy''})}{P(\,\text{``happy''})} \qquad P(\,\text{``happy''}|\,\text{Positive}) = \frac{P(\,\text{``happy''} \cap \,\text{Positive})}{P(\text{Positive})}$$

$$P(\text{Positive}|\text{``happy''}) = P(\text{``happy''}|\text{Positive}) \times \frac{P(\text{``happy''})}{P(\text{Positive})}$$

$$P(\text{Positive}|\,\text{``happy''}) = P(\,\text{``happy''} \cap \text{Positive}) \times \frac{P(\text{Positive})}{P(\,\text{``happy''})}$$

$$P(\text{Positive}|\,\text{``happy''}) = P(\,\text{``happy''} \cap \text{Positive}) \times \frac{P(\,\text{``happy''})}{P(\text{Positive})}$$

®
$$P(\text{Positive}|\text{"happy"}) = P(\text{"happy"}|\text{Positive}) \times \frac{P(\text{Positive})}{P(\text{"happy"})}$$

✓ Correct

That's right. You just derived Bayes' rule.

Question

Here, again, is Bayes' rule:

$$P(X|Y) = P(Y|X) \times \frac{P(X)}{P(Y)}$$

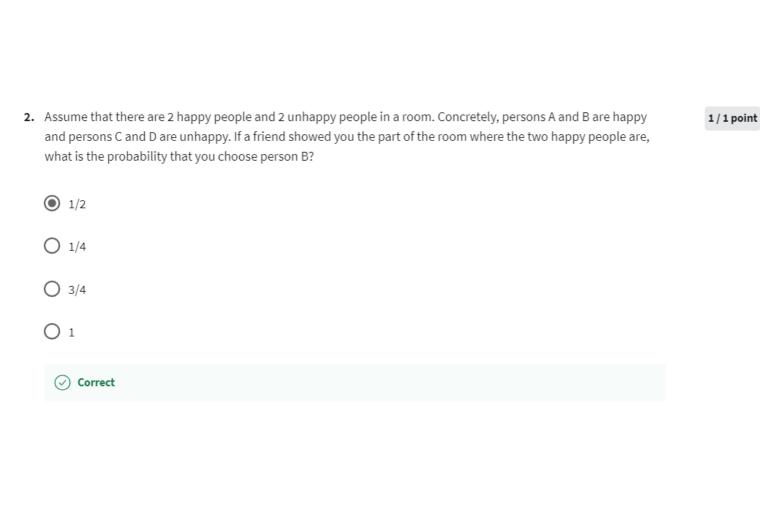
Suppose that in your dataset, 25% of the positive tweets contain the word 'happy'. You also know that a total of 13% of the tweets in your dataset contain the word 'happy', and that 40% of the total number of tweets are positive. You observe the tweet: 'happy to learn NLP'. What is the probability that this tweet is positive?

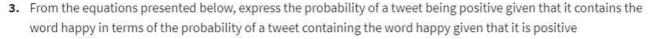
- P(Positive | "happy") = 0.77
- O P(Positive | "happy") = 0.08
- O P(Positive | "happy") = 0.10
- P(Positive | "happy") = 1.92



That's right. You just applied Bayes' rule.







1/1 point

$$P(\text{Positive} \mid \text{"happy"}) = \frac{P(\text{Positive} \cap \text{"happy"})}{P(\text{"happy"})}$$

$$P("happy" Positive) = \frac{P("happy" \cap Positive)}{P(Positive)}$$

(a)
$$P(\text{Positive} \mid \text{happy}) = P(\text{happy} \mid \text{Positive}) \times \frac{P(\text{Positive})}{P(\text{happy})}$$

O
$$P(\text{Positive} \mid \text{happy}) = P(\text{"happy"} \mid \text{Positive}) \times \frac{P(\text{happy})}{P(\text{Positive})}$$

O
$$P(\text{Positive} \cap \text{happy}) = P(\text{happy} \mid \text{Positive}) \times \frac{P(\text{Positive})}{P(\text{happy})}$$

O
$$P(\text{Positive } \cap \text{happy}) = P(\text{"happy"} \mid \text{Positive}) \times \frac{P(\text{happy})}{P(\text{Positive})}$$

Yes, that is the correct answer.

$$P(X \mid Y) = P(Y \mid X) \times \frac{P(X)}{P(Y)}$$

$$\bigcirc P(X \mid Y) = P(Y \mid X) \times \frac{P(Y)}{P(X)}$$

$$\bigcirc P(X \mid Y) = P(X \mid Y) \times \frac{P(X)}{P(Y)}$$

$$\bigcirc P(X \mid Y) = P(Y \mid X) \times \frac{P(X)}{P(Y \mid X)}$$

Yes.

5. Suppose that in your dataset, 25% of the positive tweets contain the word 'happy'. You also know that a total of 13% of the tweets in your dataset contain the word 'happy', and that 40% of the total number of tweets are positive. You observe the tweet: "happy to learn NLP". What is the probability that this tweet is positive?
0.77
Correct
That's right. You just applied Bayes' rule.

1 / 1 point

6.	The log likelihood for a certain word w_i is defined as: $\log(\frac{P(w_i pos)}{P(w_i neg)}).$	1 / 1 point
	Positive numbers imply that the word is positive.	
	⊘ Correct	
	Positive numbers imply that the word is negative.	
	✓ Negative numbers imply that the word is negative.	
	⊘ Correct	
	☐ Negative numbers imply that the word is positive.	

7.	The log likelihood mentioned in lecture, which is the log of the ratio between two probabilities is bounded between	1 / 1 point
	O -1 and 1	
	$left(left) = -\infty$ and ∞	
	\bigcirc 0 and ∞	
	O and 1	

(1)	Set or annotate a detaset with positive and negative tweets
	2. Preprocess the tweets: process_tweet(tweet)
	5. Compute freq(x), class)
	4. Get P(w.: pas), P(w.: neg)
	5. GetA[m]
	5. Compute logorior = log(P(poe) / P(neg))
0	Get or annotable a detaset with positive and negative tweets
	2. Preprocess the tweets: process_tweet(tweet)
	s. Compute Peq(n, class)
	4. G=£ \(\lambda(\pi)\)
	s. Get P(w) post, P(w) negl
	6. Compute logarior = log(P(pos) / P(neg))
0	Get or annotate a detaset with positive and negative tweets
	2. Compute freig(vi, class)
	5. Preprocess the bweets process_tweet/bweet/
	4. Get P(w pox), P(w neg)
	s. GetA[w]
	e. Compute logorior = log(P(pos) / P(neg))
0	Get or annotate a detaset with positive and negative tweets
	2. Compute freq(in, class)
	5. Preprocess the tweets: process_tweet(tweet)
	4. Compute logorior = log(P(poe) / P(neg)
	5. GetP(w pos),P(w neg)
	e. GetA[m]
6) Corpact
	Yes that is cornect

- 9. To test naive bayes model, which of the following are required?
 - \bigcirc $X_{val}, Y_{val}, \lambda, logprior$
 - $\bigcirc X_{val}, Y_{val}, logprior$
 - $\bigcirc X_{val}, \lambda, logprior$
 - $\bigcirc Y_{val}, \lambda, logprior$
 - **⊘** Correct

This is correct.

10. Which of the following is NOT an application of naive Bayes?	1 / 1 point
O Sentiment Analysis	
O Author identification	
O Information retrieval	
O Word disambiguation	
Numerical predictions	