Rishith Kyatham
CSE353
Professor: Yifan Sun
Date: 9/8/2022

1a) This scenario would be **Unsupervised Training** because this person has just arrived on Middle Earth so he doesn't have any labels on anything. He observes features on them and then groups them into categories. Using these groups he made, he captured feature similarities between these species. These clusters/groups he made are unlabeled datasets that fit Unsupervised Learning well.

1b) This scenario would be **Supervised by Training** because this "entperson" is telling the main character the specific labels and features related to the species. He used labeled inputs to label others as men or women or refines a model using it. These are clearly all pointing towards supervised training.

1c) This scenario would be **Semi-Supervised** Training because he said he had lived on Middle Earth for 150 years so he is pretty familiar with the labels of species but sometimes he meets other creatures that he could not identify clearly. So then he says he uses a "mixture" of the clustering strategy and then uses labels. The clustering strategy takes methods from unsupervised training and using pre-existing labels uses methods from supervised learning so this would be Semi-Supervised Training.

2a) So the formula goes p(A,B) = pA(a) * pB(b) if pA and pB are independent. In this case, if we take suppose pA = 0.25 when a = red and pB = 0.3 when b = hat, you will get 0.25*0.3 which is 0.075. Now if you look at the joint table, you could see when a = red and b = hat, you get 0.075. Now, if you do this same process for the rest of the table and check each value, they should all be equal to the joint table pA,B(a,b). Therefore, pA and pB are **independent** here.

2b) Suppose we say a = -0.2 and b = 0.3. This would fit the uniform distributions, fA(A) = 1 if -1 <= a <= 0 and fB(b) = 1 if 0 <= b <= 1. Now if you take these in correspondence with the fA,B(a,b) = 1 * 1 because the absolute value of a + b is -0.2 + 0.3 = 0.1 <= 0.5. But in the join function, it says fA,B(a,b) = 4/3, and 4/3 is not equal to 1 so, fA and fB are not independent. They are **dependent**.

Cases: If B=1, A=1, ...

2c) $B = A \cdot C$

	A	B	C
0.45	1	1	1
0.05	1	-1	-1
0.05	-1	1	-1
0.45	-1	-1	1

$P(B) = \{$ the four rows above $\}$

$P(B=1|A) = \begin{cases} 0.9 & \text{if } A=1 \\ 0.1 & \text{if } A=-1 \end{cases}$

$P(B=-1|A) = \begin{cases} 0.1 & \text{if } A=1 \\ 0.9 & \text{if } A=-1 \end{cases}$

Now, with these tables, if you take A=1, B=1, C=1, you would get $P(B|A)$ which is 0.9 is not = to $P(B)$ which is = 0.45

So, this random variables are **dependent**

2d) $\rho = E[(A - E[A]) \cdot (B - E[B])]$

Given: $E[A]=0$, $E[B]=1$, $E[A^2]=1$,

$E[(B-1)^2] = \frac{1}{2}$, $E[A(B-1)] = -1$

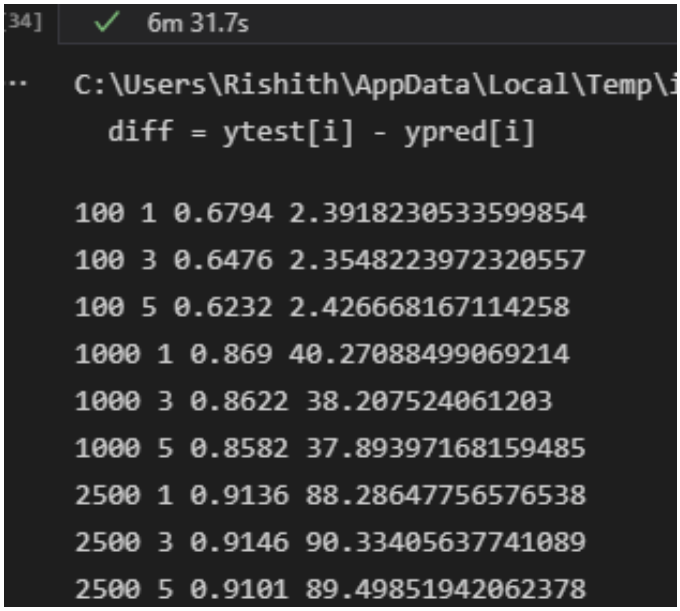
$E[(A)\cdot(B-1)] = E[A(B-1)]$

Since, covariance is not 0 Now, since $E[A(B-1)] = -1$ and $-1 \neq 0$, then these Gaussian distributions are **dependent.**

3.

Is it feasible to run this model for the full m = 60000 training dataset in runtime? Is it advisable?

Feasible means easy to run or do but for m = 60000, **I don't think this is feasible at all.**



My model for m = 2500 took 6m 31.7s on my desktop, now if we think about how long m = 60000 would take, this would take way longer. I would say this is definitely not advisable due to taking so long.

How does the accuracy depend on K for different values of m?

Every k value seems to differ when it comes to the nearest elements such as for m = 100 and m = 1000, the best k value accuracy was k = 1. The best accuracy for m = 2500, the best k value accuracy was k = 3. So, the k would differ for different values of m so the accuracy would not depend on this at all. It seems like for the smaller values of m, we could say that the lower k values would have the highest accuracy, but as m increased, the most accuracy k kept changing after running m = 5000 and m = 10000 too. For higher m values, it seems like k is insignificant, especially when I tried it at m = 10000.