Rishith Kyatham
CSE353
9/25/2022

1. a) This would **not be a reasonable hypothesis when it comes to model generalization** because the set of variables of houses in Texas, in general, has not been tested yet. For this scenario to have a possible hypothesis, he should observe for many years over Texas homes and the relationship between house sizes and their correlation to prices. Even after that, there should be a good balance for linear regression to be possible.

b) This could possibly be a good job in model generalization as the model would use historical data from the late 2000s to use how the factors affected rent prices back then, to predict how these factors affect rent prices now even though they may have increased due to inflation. The linear regression would exist in this case. But, external factors such as COVID would cause this regression to break as the whole economy would change. The factors that applied in the late 2000s would cease to exist here as they may be completely different now. Therefore, I would go with this would **not be a reasonable hypothesis when it comes to model generalization.**

c) The historical data that was tested and collected for the past 5 years of classifying images of dogs and cats have not much correlation with applying neural networks to remote sensing satellite images to detect underground marijuana farms. Therefore, this would **not be a reasonable hypothesis when it comes to model generalization.** It's not reasonable to expect the same level of accuracy when there is no historical data on detecting underground marijuana farms with neural networks. They're both completely different models.

2.

2a)

fur \ tail	furry	rope-like
blue	$\frac{5}{50}$	0
gray	$\frac{5}{50}$	0
brown	0	40/50

b) The features are dependent because we know $P(A,B) = \frac{5}{50}$ from the chart so

$p(A,B) = P(A) \cdot P(B)$ ~~Now~~

Now, $P(A) = \frac{5}{50}$ when A is = Blue

$P(B)$ = Furry tail prob $= \frac{10}{50}$

$\frac{5}{50} = \frac{10}{50} \cdot \frac{5}{50}$ $1 \neq \frac{5}{50}$ so, it is $\boxed{\text{dependent}}$

c)

fur\tail	furry	rope-like
blue	$\frac{5}{10}$	0
gray	$\frac{5}{10}$	0
brown	0	0

d) $P(A,B) = P(A) \cdot P(B)$

When A = ~~furry~~ Blue B = Furry tail

$\frac{5}{10} = \frac{1}{2} \cdot 1 \quad \frac{1}{2} = \frac{1}{2}$ ✓

$P(A,B) = P(A) \cdot P(B)$

When A = gray B = furry tail

$\frac{5}{10} = \frac{5}{10} \cdot 1$ ✓

$P(A,B) = P(A) \cdot P(B)$

when A = brown B = furry tail

$0 = 0 \cdot 0$ ✓

$P(A,B) = P(A) \cdot P(B)$

when A = blue B = rope-like

$0 = \frac{1}{2} \cdot 0$ ✓

$P(A,B) = P(A) \cdot P(B)$

when A = gray B = rope-like

$0 = \frac{1}{2} \cdot 0 = 0$ ✓

$P(A,B) = P(A) \cdot P(B)$

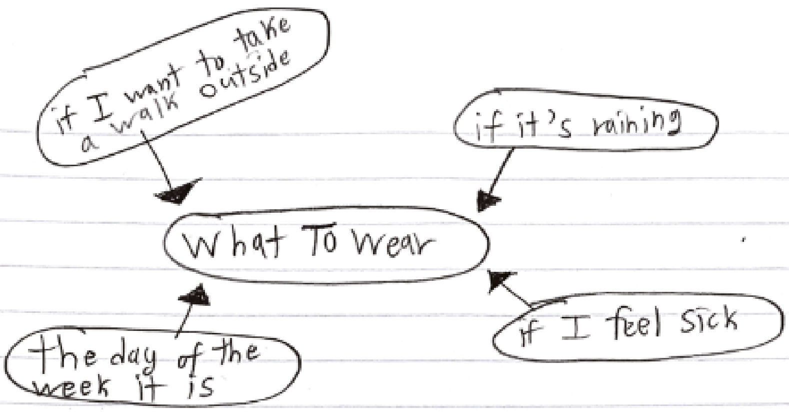
when A = brown B = rope-like

$0 = 0 \cdot 0$ ✓

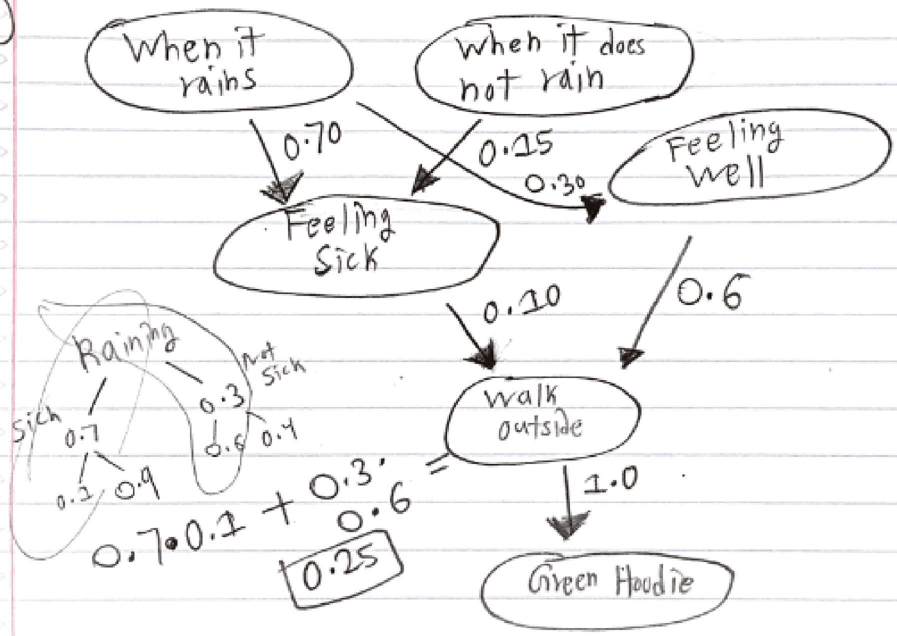
This is independent because all of ~~theses~~ these cases are equal.

3.

3. a)

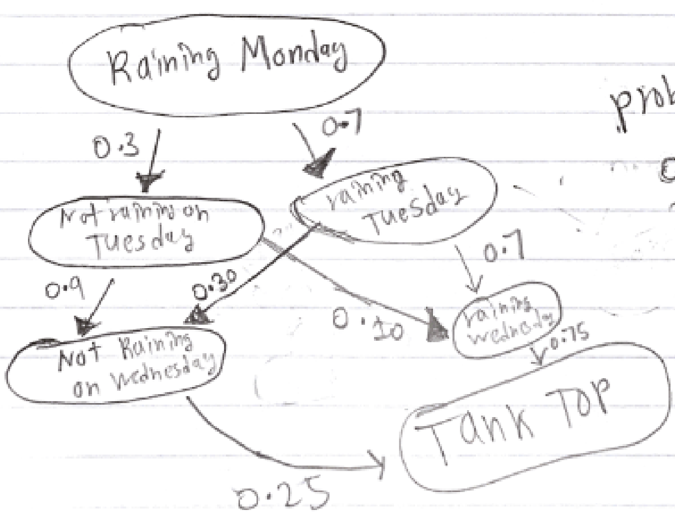


b)



$0.7 \cdot 0.1 + 0.3 \cdot 0.6 = \boxed{0.25}$

c)



probability is

$0.3 \cdot 0.9 \cdot 0.25 +$
$0.3 \cdot 0.10 \cdot 0.75 +$
$0.7 \cdot 0.30 \cdot 0.25 +$
$0.7 \cdot 0.7 \cdot 0.75 =$
$0.0675 + 0.0225 +$
$0.0525 + 0.3675 =$
0.51

4. a)

This is the classification accuracy below, and it's also in the code

```
    print(pred_2gram('cheshire'))
    print(pred_2gram('mock'))

5]  ✓ 2.3s

  0.2500098740076622
  ('was', 0.0007109004739336493)
  ('queen', 0.002764612954186414)
  ('cat', 0.00019747235387045816)
  ('turtle', 0.0022511848341232226)
```

b) Called countgram, find the # of words that occured, divide etc. Then I multiplied prior and posterior for every word and returned the likelihood at the end. This code took more than 6 hours to code so I tried finding a more efficient way of doing it. I tried using a 3d array and then each index would basically be the posterior2_gram. Such that, the 0th index would be posterior1_gram, 1st index would be posterior2_gram, etc until posterier_ngram. The prior would remain the same. The formula would basically be posterior(n) * posterior(n-1) * … posterior(0) would be = to the likelihood which I would return.

c) I would call the prediction function 25 times using the last 3 words. This would require an efficient working part b and right now mine is not working or not efficient enough so I am writing an explanation on what I would do instead for c and d.

d) I would call the prediction function 25 times using the last 3 words with random.choices(). I would use random.choices and use weights and plug in the probability list you get from the probability function.