95 | 100

Allister Lim

1. a) $P(e \mid \text{Foobar}) = P(\text{Foo} = \text{True} \mid \text{Foobar} = \text{true}) \cdot P(\text{Hello} = 2 \mid \text{Foobar} = \text{true}) \cdot P(\text{World} = C \mid \text{Foobar} = \text{true}) \cdot P(\text{Bar} = \text{false} \mid \text{Foobar} = \text{true})$

$= (0.8)(0.5)(0.1)(0.4)$

$= 0.016.$

50/50

b) $P(e \mid \overline{\text{Foobar}}) = (0.3)(0.2)(0.5)(0.8) = 0.024.$

c) $P(\overline{\text{Foobar}} \mid e) = 1 - P(\text{Foobar} \mid e)$

$$P(\text{Foobar} \mid e) = \frac{P(e \mid \text{Foobar}) \cdot P(\text{Foobar})}{P(e)}$$

$$= \frac{(0.016)(0.2)}{(0.016)(0.2) + (0.024)(0.8)} = 0.1429$$

$\therefore P(\overline{\text{Foobar}} \mid e) = 0.8571$

45/50

2. a) Feature selection: it is the process of throwing away the less ~~irre~~ relavent features to decrease time and space complexity. ✓ close choosing which features to consider

b) No. ✓ OK; I would have liked more explanation. All values for that feature would already be the same

c) No Free Lunch Theorem: the conclusion of this theorem basically says that there isn't a best machine learning method for all tasks. ✓ OK; specifically, if all models equally likely; no ML method is better than random guessing

d) It takes longer to apply K-nearst neighbor system ~~be~~ because it stores feature vectors & class labels only, during training; while in testing, it assigns the most frequent labels. Although it takes more time at application stage (due to more computation), it is not necessarily a bad thing, because it ~~could adapt to changes without retraining.~~ ✓ cannot generally considered bad. Training only done once; until predictions to be test,

e) ~~Reprens~~

✓ Represented by perception: ii & iv

✓ Represented by neural network: i, ii, iii, iv.