10-01-2023

Assignment # 3

Question # 1: Intro to probability

(a) Basic priors:

- $P(\text{Black} \vee \text{tabby}) = P(\text{Black}) + P(\text{tabby})$

$$= \frac{12}{50} + \frac{11}{50}$$

$$= \frac{23}{50} //$$

- $P(\text{White}) = \frac{5}{50} = \frac{1}{10} //$

- $P(!\text{calico}) = 1 - P(\text{calico})$

$$= 1 - \frac{15}{50}$$

$$= \frac{7}{10} //$$

(b) Inference with the JPD

- $P(\text{calico} \wedge \text{male})$

we know that from the table probability of calico and male is 0.12 //

- $P((\text{black} \wedge \text{male}) \vee (\text{white} \wedge \text{female}))$

we know $P(\text{black} \wedge \text{male}) = 0.1$

$P(\text{white} \wedge \text{female}) = 0.08$

$$P(\text{black} \wedge \text{male}) + P(\text{white} \wedge \text{female}) = 0.1 + 0.08$$

$$= 0.18 //$$

- $P(\text{male} \vee \text{calico})$

$P(\text{male}) = 0.1 + 0.12 + 0.06 + 0.02 + 0.12 = 0.42$

$P(\text{calico}) = 0.12 + 0.18 = 0.3$

$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$ ← 0.12

$= 0.42 + 0.3 - 0.12$

$= 0.6$ //

- $P(\text{female}) = 0.14 + 0.02 + 0.16 + 0.08 + 0.18$

  $= 0.58$ //

- $P((\text{tabby} \vee \text{white}) \wedge \text{female})$

  $P(\text{tabby} \wedge \text{female}) = 0.16$

  $P(\text{white} \wedge \text{female}) = 0.08$

  $P(\text{tabby} \vee \text{white}) \wedge \text{female} = 0.16 + 0.08 = 0.24$ //

- $P(\text{gray} \wedge \text{male}) = 0.12$ //

(c) Conditional Probability

- $P(\text{male} \mid \underbrace{\text{gray} \vee \text{white}})$

  → $P(\text{gray} \vee \text{white}) = P(\text{gray} + P(\text{white}$

  $P(\text{gray}) = 0.12 + 0.02 = 0.14$

  $P(\text{white}) = 0.02 + 0.08 = 0.1$

  → $= 0.14 + 0.1 = 0.24$

So $P(\text{male} \wedge (\text{gray} \vee \text{white})) = P(\text{gray} \wedge \text{male}) + P(\text{white male})$

$= 0.12 + 0.02$

$= 0.14$

$P(\text{male} \mid \text{gray} \vee \text{white}) = \dfrac{P(\text{male} \wedge (\text{gray} \vee \text{white}))}{P(\text{gray} \vee \text{white})} = \dfrac{0.14}{0.24}$

$= 0.58\overline{3}$ //

- P(female | !Black)

  ↳ P(!Black) = 1 − (0.1 + 0.14) = 0.76

P(female ∧ !Black) = P(female) − P(Black ∧ female)

P(female) = 0.14 + 0.02 + 0.16 + 0.08 + 0.18 → = 0.58 − 0.14 = 0.44 //

- P(gray | female)

P(gray ∧ female) = 0.02

P(female) = 0.58

$$P(\text{gray} \mid \text{female}) = \frac{0.02}{0.58} = 0.0345 \text{ //}$$

(d) **Bayes' Rule**

$$P(\text{calico} \mid \text{friendly}) = \frac{P(\text{friendly} \mid \text{calico}) \times P(\text{calico})}{P(\text{friendly})}$$

0.3 → P(calico)

0.2 → P(friendly | calico)

P(friendly | !calico) = 0.4

P(!calico) = 1 − 0.3 = 0.7

P(friendly) = P(friendly | calico) × P(calico) + P(friendly | !calico) × P(!calico)

P(friendly) = 0.2 × 0.3 + 0.4 × 0.7

= 0.34

$$P(\text{calico} \mid \text{friendly}) = \frac{0.2 \times 0.3}{0.34}$$

$$= 0.1765$$

given that you see a friendly cat,
the prob that it's calico is ≈ 0.1765 (17.65%)

Question 3 write up:
Five-fold cross-validation was used to assess the effectiveness of the Naive Bayes classifier on the movie_reviews dataset. Improving the classifier's performance relied heavily on feature selection. The classifier's average accuracy increased from 71.90% to 81.05% with the help of filter selection. The filter space was further polished with the help of filters and transformations. The token length filter omitted words that were too short to convey any meaningful emotion. Tokens containing numbers were filtered out since they are often irrelevant to sentiment analysis in film critics' evaluations thanks to the "no-numbers" filter. Finally, stemming reduces the feature space by transforming words back to their root form, making the model more generalizable across variations of the same word. These preliminary procedures greatly aided the classifier in its classification of review sentiment as positive or negative. The token length and no-numbers filtering assured a more refined feature set, while stemming may have helped the model understand. In conclusion, the specialized collection of filters and transformations greatly helped to obtain an accuracy of 81.05%, particularly the addition of stemming, token length verification, and removal of numbers.

Question 4 write up:
Correct extraction: "iPhone", "model", "headquarters", "yesterday
that were missed from the first sentence: "latest iPhone model", "California headquarters"
it identifies single nouns better compared to the compound noun

#Name entity
Correct: "Apple" (as PERSON), "iPhone" (as ORGANIZATION), "California" (as GPE)
Missed: "iPhone" shouldn't be categorized as ORGANIZATION, and "Apple" is a company, so it might be better classified as ORGANIZATION.
tends to properly recognize single proper nouns. chunker sometimes misclassifies entities, e.g., "Mars" as PERSON or "iPhone" as ORGANIZATION

Noun Phrase Extraction:
We selected noun phrases from those five news-related sentences using the RegexpParser that comes with chunking.py. Simple nouns like "iPhone," "model," and "headquarters" were all easily recognized by the parser. For example, it failed to recognize "latest iPhone model" and "California headquarters" both of which include compound noun phrases or noun phrases with accompanying adjectives or proper nouns. The parser has a clear preference for single nouns over compound or descriptive noun phrases.

Named Entity Extraction:
Using the named entity chunker, I saw that it could extract and identify several named entities, particularly well-known ones like "Apple", "UN", and "NASA". Although "Apple" may be better classed as an ORGANIZATION, the chunker properly identified it as a PERSON. Just as it would be more correct to think of "iPhone" as a PRODUCT rather than an ORGANIZATION, "iPhone" was misidentified. Also noteworthy is the chunker's occasional difficulties with context, as seen by the labeling of "Mars" as PERSON. It's clear that the chunker may misclassify items

based on the minimal context it analyzes, despite its proficiency in detecting single proper nouns.