WEEK 7 LAB

Naives Bayes Tutorial

Task 1: Accident Prediction For the table above, find the probability of the scenario being in an accident (YES or NO).

In this lab session, I will be calculating probability of the given scenario ,

1. Objective: Calculate P(C1| X = (Rain, Good, Normal, No)). Determine how the Bayes classifier would classify the data instance X = (Rain, Good, Normal, No).

   Total instance = 10
   YES = 5
   NO = 5
   P(accident = yes) = 5/10 = 0.5
   P(accident = no) = 5/10 = 0.5

   Here, I will be calculating probability of the given condition of X where given that the chances of accident is yes:
   Calculating probabilities when accident = yes where X = (rain , good, normal, no)
   For rain; P(rain/accident = yes) = 1/5 = 0.2
   For good; p(good/accident=yes) = 1/5 =0.2
   For normal; ,p(normal/accident=yes) = 1/5 = 0.2
   For No; p(no/accident=yes) = 2/5 = 0.4

   Here, I will be calculating probability of the given condition of X where given that the chances of accident is no:
   Calculating probabilities when accident = No where X = (rain , good, normal, no)
   For rain; P(rain/accident = no) = 2/5 = 0.4
   For good; p(good/accident=yes) = 3/5 =0.6
   For normal; ,p(normal/accident=yes) = 2/5 = 0.4
   For No; p(no/accident=yes) = 4/5 = 0.8

   Calculating probability of X given that accident =yes,
   P(X/ accident=yes) = 0.2*0.2*0.2*0.4 = 0.0032

   Calculating probability of X given that accident =No,
   P(X/accident = no) = 0.4*0.6*0.4*0.8 = 0.0768

   Now,

P(accident=yes/X) = 0.0032*0.5 = 0.0016

P(accident=no/X) = 0.0768*0.5 = 0.0384

Reflection:

I used the Naive Bayes method in this challenge to assess the likelihood of an accident based on four category features. Understanding how priors and likelihoods combine to generate the posterior probability was the primary takeaway. Additionally, I observed that the likelihoods had a significant impact on the final categorisation even though both classes had equal priors. This exercise made it easier for me to understand why Naive Bayes is regarded as a straightforward yet effective classifier, particularly when there are several categorical variables. Additionally, I became more conscious of the limits, such as the presumption of attribute independence, which would not always apply in actual accident situations.

Task 2: Weather-Based Game Prediction In this dataset, there are five categorical attributes: outlook, temperature, humidity, windy, and play. We are interested in building a system to classify whether to play based on weather conditions.

1. Question 1: Calculate P(C1 | X = (sunny, hot, high, false)). How would the Bayes classifier classify the data instance X = (sunny, hot, high, false)?

   Total instance = 14,
   Yes = 9
   No = 5
   P(yes) = 9/14 = 0.64
   P(no) = 5/14 = 0.36
   Calculating probabilities for X (sunny, hot, high, false) given that play = yes
   P(sunny/play = yes)= 2/9 = 0.2
   P(hot/ play=yes) = 2/9 =0.2
   P(high / play=yes) = 3/9 = 0.3
   P(false/ play=yes) = 6/9 = 0.67

   Calculating probabilities for X (sunny, hot, high, false) given that play = No
   P(sunny/play = no)= 3/5 = 0.6
   P(hot/ play=no) = 2/5 = 0.4
   P(high / play=no) = 4/5 = 0.8
   P(false/ play=no) = 2/5 = 0.4

Calculating probability of X given that play =yes and play = no
P(X/play=yes) = 0.2*0.2*0.3*0.67 = 0.00804
P(X/play=no) = 0.6*0.4*0.8*0.4 = 0.0768

Calculating probabilities of play=yes and play=no given that X
P(play=yes/X) = 0.00804*0.64 = 0.0051
P(play=no/X) 0.0768*0.36 = 0.028

2. Question 2: Does this agree with the classification in Table 1 for X = (sunny, hot, high, false)?
   => NO

3. Question 3: Consider a new data instance X'= (overcast, cool, high, true). How would the Bayes classifier classify X'?
   Total instance = 14,
   Yes = 9
   No = 5
   P(yes) = 9/14 = 0.64
   P(no) = 5/14 = 0.36

   Calculating probabilities for X (overcast, cool, high, True) given that play = yes
   P(overcast/play = yes)= 4/9 = 0.44
   P(cool/ play=yes) = 3/9 =0.0.33
   P(high / play=yes) = 3/9 = 0.33
   P(true/ play=yes) = 3/9 = 0.33

   Calculating probabilities for X (overcast, cool, high, True) given that play = No
   P(overcast/play = no)= 0/5 = 0
   P(cool/ play=no) = 1/5 = 0.2
   P(high / play=no) = 4/5 = 0.8
   P(true/ play=no) = 3/5 = 0.6

   Calculating probability of X given that play =yes and play = no
   P(X/play=yes) =0.44*0.33*0.33*0.33 = 0.0158
   P(X/play=no) = 0*0.2*0.8*0.6 = 0

   Calculating probabilities of play=yes and play=no given that X
   P(play=yes/X) = 0.0158*0.64 = 0.0051

P(play=no/X) = 0*0.36 = 0

Reflection:

My comprehension of how Naive Bayes functions on bigger datasets with more feature-value pairs has improved as a result of this task. I was able to see how sensitive the classifier is to frequency counts by calculating probabilities for two distinct circumstances. One important realisation was that, as seen in the second case, zero-frequency problems can totally eliminate the probability for an entire class. This demonstrated how crucial methods like Laplace smoothing are for preventing classification errors. Additionally, I was able to see how a probabilistic model can disagree with datasets provided by humans by comparing the classifier's output with the original table, illustrating how models generalise differently from raw data.

Task 3:

Loan Approval Prediction Use Naive Bayes to classify whether a new applicant's loan will be approved based on employment status, credit history, and income level.

1. A new applicant is Employed with a Good credit history and a Medium income level. Objective: Calculate P(LoanApproved = Yes EmploymentStatus = Employed, CreditHistory = Good, IncomeLevel = Medium) using Naive Bayes.

Total instance = 5

Yes = 3,

No= 2

P(LoanApproved=Yes) =3/5= 0.6

P(LoanApproved=No) = 2/5= 0.4

Calculating probabilities for X(employed, good, medium) given that loanapproved = yes

P(employed / yes) = 2/3 = 0.67

P(good / yes) = 3/3 = 1

P(medium/ yes) = 1/3 = 0.33

Calculating probabilities for X(employed, good, medium) given that loanapproved = no

P(employed / no) = ½ = 0.5

P(good / no) = 0/2 = 0

P(medium/ no) = ½ = 0.5

P(X/ yes) = 0.67*1*0.33 = 0.22

P(X/ no) = 0.5*0.5*0 = 0

2. Another applicant is Unemployed with a Bad credit history and a Low income level.
   Objective: Calculate P(LoanApproved = No || EmploymentStatus = Unemployed, CreditHistory = Bad, IncomeLevel = Low)
   For loan approved = no
   P(unemployes/no) = ½ = 0.5
   P(bad / no) = 2/2 = 1
   P(low / no) = ½ = 0.5
   For loan approved = yes
   P(unemployes/yes) = ½ = 0.5
   P(bad / yes) = 0/2 = 0
   P(low / yes) = ½ = 0.5

   P(X/ no) = 0.5*0.5*1 = 0.25
   P(X /YES)= 0.5*0*0.5 = 0

   Here, probability of getting loanapproved= yes for (unemployment, bad, low) is less that of loanapproved = no so the loean is not approved.

3. (Advanced) The bank introduces a scoring system for applicants, assigning scores to each feature (e.g., "Employed" has a score of 3, "Unemployed" has 1). Discuss how this might impact probability calculations.

Objective: Consider the impact of a scoring system on Naive Bayes probability calculations.

Reflection:

I used Naive Bayes in this job to make a financial choice. I discovered that the classifier can use basic probability multiplications to distinguish between positive and negative outcomes.

Classification outcomes were once more impacted by the existence of zero likelihood values, highlighting the significance of managing sparse or unbalanced data. The "advanced" section, which presented scoring systems, forced me to consider encoding methods more carefully. I came to see that switching from categorical values to numerical scores can significantly alter how Naive Bayes analyses the data, sometimes necessitating the use of various variations like Gaussian Naive Bayes. In general, this job made a direct connection between machine learning and practical uses such as credit rating.

Task 4

Total instance = 5
Positive = 3
Negative = 2
P(positive) = 3/5 = 0.6
P(negative) = 2/5 = 0.4

FOR question 1:
For X (fever=yes, cough=no, fatigue=yes, travelhistory=no) given that diseasesdiagnosis = positive;
P(fever / positive) = 3/3 = 1
P(cough / positive) = 2/3 = 0.67
P(fatigue / positive) = 3/3 = 1
P (travelhistory / positive) = 2/3 = 0.67

Calculating probability of X when diseasesdiagnosis is positive
P(x/positive) = 1*0.67*1*0.67 = 0.45

Calculating probability of diseasesdiagnosis is positive given that of X
P(positive/ X) = 0.45*0.6 = 0.27

For question 2:
For X (fever=no, cough=yes, fatigue=no, travelhistory=no) given that diseasesdiagnosis = negative;
P(fever / negative) = 2/2 = 1
P(cough / negative) = 2/2 = 1
P(fatigue / negative) = 2/2 = 1
P (travelhistory / negative) = 1/2 = 0.5

Calculating probability of X when diseasesdiagnosis is negative
P(x/positive) = 1*1*1*0.5 = 0.5

Calculating probability of diseasesdiagnosis is negative given that of X
P(positive/ X) = 0.5*0.4 = 0.2

Reflection:

This exercise illustrated the use of Naive Bayes in medical situations where symptoms are used to make diagnostic predictions. Here, I learnt more about how likelihoods and priors affect categorisation in medical datasets. My capacity to methodically calculate probabilities and decipher their meaning was strengthened by the computations. I also thought about how Naive Bayes can still function remarkably well even when the independence assumption might not apply to medical symptoms. My understanding of Naive Bayes as a useful diagnostic tool has grown as a result of this exercise, particularly in situations with sparse data or basic features.