Problem: Sentiment Classification using Optimized Naïve Bayes

Given a dataset with the following short reviews, classify their sentiment (positive or negative) using binary Naïve Bayes. Apply the following optimizations:

1. **Binarization:** Convert word frequencies into binary values (presence or absence).
2. **Negation Handling:** Prepend "NOT_" to words after a negation term (not, no, never, n't).
3. **Sentiment Lexicon Features:** Use predefined positive and negative word lists.

Dataset:

Review	Label
"The movie was great"	Positive
"I did not like the movie"	Negative
"No amazing scenes at all"	Negative
"A fantastic and thrilling film"	Positive

Step 1: Binarization

Instead of word counts, we only note if a word appears (1) or not (0).

Word	Review 1	Review 2	Review 3	Review 4
the	1	1	0	0
movie	1	1	0	0
was	1	0	0	0
great	1	0	0	0
I	0	1	0	0
did	0	1	0	0
not	0	1	1	0
like	0	1	0	0
amazing	0	0	1	0
scenes	0	0	1	0
at	0	0	1	0
all	0	0	1	0
fantastic	0	0	0	1
thrilling	0	0	0	1
film	0	0	0	1

Step 2: Negation Handling

Apply the rule: words after negation markers (not, no, never, n't) are prefixed with "NOT_".

Original Review	Modified Review
"The movie was great"	"The movie was great"
"I did not like the movie"	"I did NOT_like NOT_the NOT_movie"
"No amazing scenes at all"	"No NOT_amazing NOT_scenes NOT_at NOT_all"
"A fantastic and thrilling film"	"A fantastic and thrilling film"

Step 3: Using Sentiment Lexicon

Given sentiment lexicons:

- **Positive Words:** great, fantastic, thrilling, amazing
- **Negative Words:** not_like, not_amazing

Count how many positive and negative words appear in each review.

Review	Positive Words Count	Negative Words Count
"The movie was great"	1 (great)	0
"I did NOT_like NOT_the NOT_movie"	0	1 (NOT_like)
"No NOT_amazing NOT_scenes NOT_at NOT_all"	0	1 (NOT_amazing)
"A fantastic and thrilling film"	2 (fantastic, thrilling)	0

Step 4: Naïve Bayes Classification

Using probability estimates:

- **P(Positive) = 2/4 = 0.5**
- **P(Negative) = 2/4 = 0.5**
- **P(word | Positive)** and **P(word | Negative)** calculated from training data.

After computing probabilities using Naïve Bayes (without smoothing for clarity), we classify:

- Reviews 1 & 4 → **Positive**
- Reviews 2 & 3 → **Negative**

Word	In Positive Reviews (R1, R4)	In Negative Reviews (R2, R3)
the	1	1
movie	1	1
was	1	0
great	1	0
I	0	1
did	0	1
not	0	2
like	0	1
amazing	0	1
scenes	0	1
at	0	1
all	0	1
fantastic	1	0
thrilling	1	0
film	1	0

Vocabulary Size (V)

The total number of unique words in the dataset (from all reviews):

$$V = 15$$

Smoothed Likelihood Calculation

Using **Laplace smoothing**, the formula for conditional probability becomes:

$$P(word|Positive) = \frac{\text{count in positive reviews} + 1}{\text{total positive words} + V}$$

$$P(word|Negative) = \frac{\text{count in negative reviews} + 1}{\text{total negative words} + V}$$

Total Word Counts with Smoothing

- **Total words in positive reviews:** 7
- **Total words in negative reviews:** 12
- **Vocabulary size:** $V = 15$

Smoothed denominator:

$$\text{Total positive words} + V = 7 + 15 = 22$$

$$\text{Total negative words} + V = 12 + 15 = 27$$

Smoothed Word Probabilities

Word	$P(word|Positive)$	$P(word|Negative)$
the	$\frac{1+1}{22} = 0.0909$	$\frac{1+1}{27} = 0.0741$
movie	0.0909	0.0741
was	$\frac{1+1}{22} = 0.0909$	$\frac{0+1}{27} = 0.037$
great	$\frac{1+1}{22} = 0.0909$	0.037
I	$\frac{0+1}{22} = 0.0455$	$\frac{1+1}{27} = 0.0741$
did	0.0455	0.0741
not	0.0455	$\frac{2+1}{27} = 0.1111$
like	0.0455	0.0741
amazing	0.0455	0.0741
scenes	0.0455	0.0741
at	0.0455	0.0741
all	0.0455	0.0741
fantastic	$\frac{1+1}{22} = 0.0909$	0.037
thrilling	0.0909	0.037
film	0.0909	0.037

Using **Naïve Bayes formula**:

$$P(Positive|Review) \propto P(Positive) \times \prod P(word|Positive)$$

$$P(Negative|Review) \propto P(Negative) \times \prod P(word|Negative)$$

Example: Classifying "The movie was great"

Words in the review: **(the, movie, was, great)**

Positive Sentiment Probability

$$P(Positive|R1) \propto 0.5 \times 0.0909 \times 0.0909 \times 0.0909 \times 0.0909$$

$$P(Positive|R1) \propto 0.5 \times (6.8 \times 10^{-5})$$

$$P(Positive|R1) = 3.4 \times 10^{-5}$$

Negative Sentiment Probability

$$P(Negative|R1) \propto 0.5 \times 0.0741 \times 0.0741 \times 0.037 \times 0.037$$

$$P(Negative|R1) \propto 0.5 \times (7.5 \times 10^{-5})$$

$$P(Negative|R1) = 3.75 \times 10^{-5}$$

Since $P(Negative|R1) > P(Positive|R1)$, **classification remains "Positive"**.

Example: Classifying "I did not like the movie"

Words in the review: **(I, did, not, like, the, movie)**

Positive Sentiment Probability

$$P(Positive|R2) \propto 0.5 \times 0.0455 \times 0.0455 \times 0.0455 \times 0.0455 \times 0.0909 \times 0.0909$$

$$P(Positive|R2) = 0.5 \times (3.2 \times 10^{-7})$$

$$P(Positive|R2) = 1.6 \times 10^{-7}$$

Negative Sentiment Probability

$$P(Negative|R2) \propto 0.5 \times 0.0741 \times 0.0741 \times 0.1111 \times 0.0741 \times 0.0741 \times 0.0741$$

$$P(Negative|R2) = 0.5 \times (2.6 \times 10^{-5})$$

$$P(Negative|R2) = 1.3 \times 10^{-5}$$

Since $P(Negative|R2) > P(Positive|R2)$, **classification remains "Negative"**.

Final Classification Results

Review	Predicted Sentiment
"The movie was great"	**Positive**
"I did not like the movie"	**Negative**
"No amazing scenes at all"	**Negative**
"A fantastic and thrilling film"	**Positive**

Example-2

A company wants to classify incoming emails as "Spam" or "Not Spam" (binary classification). Features include the presence of specific words such as "discount," "offer," "free," etc. Suppose you have a dataset like this:

Email	Discount	Offer	Free	Spam
Email 1	1	0	1	Yes
Email 2	0	1	0	No
Email 3	1	1	1	Yes
Email 4	0	0	1	No

Step 1: Prior Probabilities

Calculate prior probabilities as usual:

$$P(\text{Spam}) = \frac{\text{Count of Spam emails}}{\text{Total emails}} = \frac{2}{4} = 0.5$$

$$P(\text{Not Spam}) = \frac{2}{4} = 0.5$$

Step 2: Smoothed Likelihood Probabilities

Using Laplace smoothing, add 1 to the numerator and k (number of feature values) to the denominator for each probability. Here, $k = 2$ because the features ("Discount," "Offer," "Free") are binary (0 or 1).

For **Discount**:

$$P(\text{Discount} \mid \text{Spam}) = \frac{\text{Count of Discount in Spam emails} + 1}{\text{Spam emails} + k} = \frac{2+1}{2+2} = \frac{3}{4} = 0.75$$

$$P(\text{Discount} \mid \text{Not Spam}) = \frac{0+1}{2+2} = \frac{1}{4} = 0.25$$

For **Free**:

$$P(\text{Free} \mid \text{Spam}) = \frac{2+1}{2+2} = \frac{3}{4} = 0.75$$

$$P(\text{Free} \mid \text{Not Spam}) = \frac{1+1}{2+2} = \frac{2}{4} = 0.5$$

Step 3: Posterior Probabilities

For a new email with "Discount" and "Free," calculate the posterior probability for each class:

- **Spam**:

$$P(\text{Spam} \mid \text{Discount, Free}) = P(\text{Spam}) \cdot P(\text{Discount} \mid \text{Spam}) \cdot P(\text{Free} \mid \text{Spam})$$

$$= 0.5 \cdot 0.75 \cdot 0.75 = 0.28125$$

- **Not Spam**:

$$P(\text{Not Spam} \mid \text{Discount, Free}) = P(\text{Not Spam}) \cdot P(\text{Discount} \mid \text{Not Spam}) \cdot P(\text{Free} \mid \text{Not Spam})$$

$$= 0.5 \cdot 0.25 \cdot 0.5 = 0.0625$$

Step 4: Classification

Since $P(\text{Spam} \mid \text{Discount}, \text{Free}) > P(\text{Not Spam} \mid \text{Discount}, \text{Free})$, classify the email as **Spam**.