Text Classification using Naive Bayes

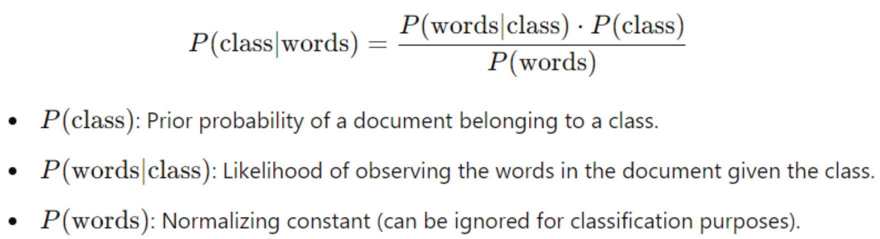
Text classification is a fundamental task in Natural Language Processing (NLP), where we categorize text into predefined classes. One popular algorithm for this task is **Naive Bayes**, a probabilistic classifier based on Bayes' Theorem. It works under the assumption that the features (words in a document) are **conditionally independent** given the class. Despite the strong independence assumption, Naive Bayes performs well in many practical text classification tasks.

Steps in Text Classification using Naive Bayes:

1. **Data Preprocessing**:

   Convert the text data into a numerical format. Typically, this is done by tokenizing the text and constructing a **bag of words** or using **TF-IDF** (Term Frequency-Inverse Document Frequency) representations.

2. **Bayes' Theorem**:



$$P(\text{class}|\text{words}) = \frac{P(\text{words}|\text{class}) \cdot P(\text{class})}{P(\text{words})}$$

- $P(\text{class})$: Prior probability of a document belonging to a class.
- $P(\text{words}|\text{class})$: Likelihood of observing the words in the document given the class.
- $P(\text{words})$: Normalizing constant (can be ignored for classification purposes).

3. **Naive Assumption**:
   The
   naive part of the algorithm assumes that the probability of each word occurring is independent of other words. Therefore, the likelihood P(words∣class) is computed as the product of the probabilities of individual words:

$$P(\text{words}|\text{class}) = P(\text{word}_1|\text{class}) \cdot P(\text{word}_2|\text{class}) \cdot \ldots \cdot P(\text{word}_n|\text{class})$$

4. **Text Classification**:
   After calculating the probabilities for each class, the document is classified into the class with the highest posterior probability.

Laplace Smoothing

One of the challenges in Naive Bayes is dealing with words that may appear in the test data but were not seen in the training data. This results in zero probabilities, which can make the entire product of probabilities zero. To solve this, we use **Laplace Smoothing** (additive smoothing).

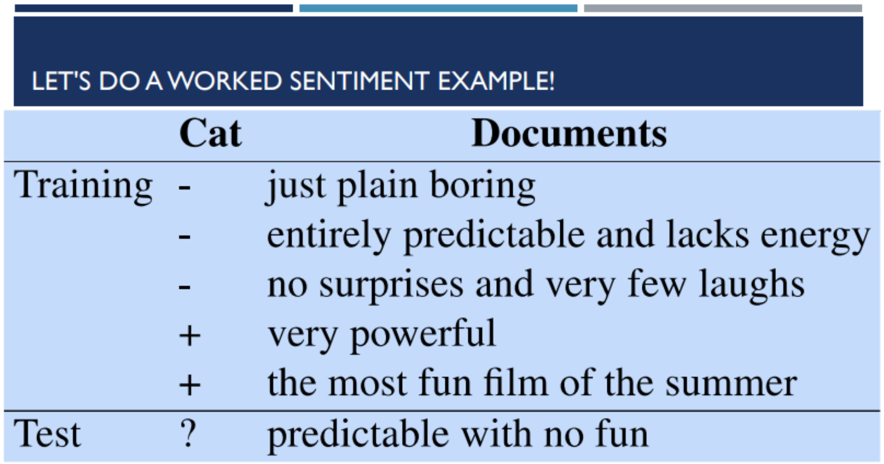
Formula:

Laplace Smoothing adds a small constant (α) to each word count:

$$P(\text{word}|\text{class}) = \frac{\text{count}(\text{word}, \text{class}) + \alpha}{\text{total count of words in class} + \alpha \cdot \text{number of unique words}}$$

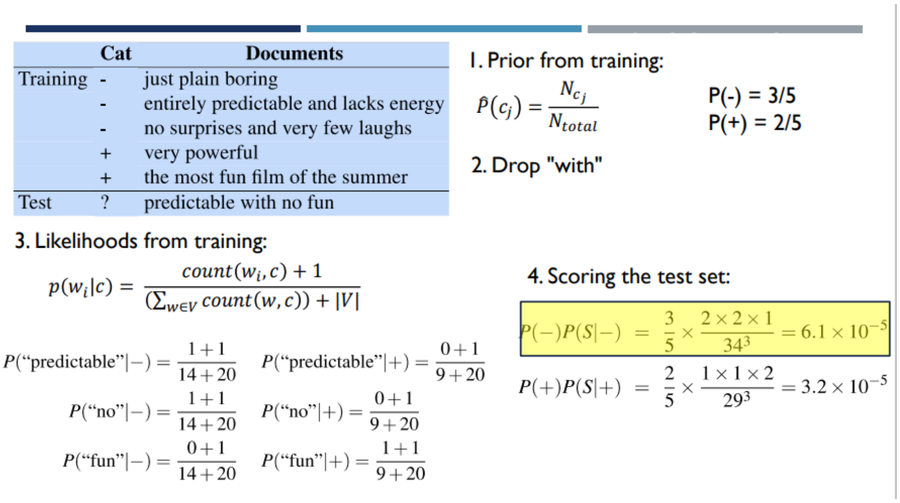
- If $\alpha = 1$, it is called **Laplace smoothing**. For other values of α, it is called **Lidstone smoothing**.

This ensures that no probability becomes zero, even for unseen words, improving the classifier's robustness.



LET'S DO A WORKED SENTIMENT EXAMPLE!

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun



	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

1. Prior from training:

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{total}}$$

P(-) = 3/5
P(+) = 2/5

2. Drop "with"

3. Likelihoods from training:

$$p(w_i|c) = \frac{count(w_i, c) + 1}{(\sum_{w \in V} count(w, c)) + |V|}$$

$$P(\text{“predictable”}|-) = \frac{1+1}{14+20} \qquad P(\text{“predictable”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“no”}|-) = \frac{1+1}{14+20} \qquad P(\text{“no”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“fun”}|-) = \frac{0+1}{14+20} \qquad P(\text{“fun”}|+) = \frac{1+1}{9+20}$$

4. Scoring the test set:

$$P(-)P(S|-) = \frac{3}{5} \times \frac{2 \times 2 \times 1}{34^3} = 6.1 \times 10^{-5}$$

$$P(+)P(S|+) = \frac{2}{5} \times \frac{1 \times 1 \times 2}{29^3} = 3.2 \times 10^{-5}$$

Q1.(Naive Bayes) **(10 Marks)**

You are given a collection of training documents, each labeled as either *Positive* or *Negative*. Using this training data, your task is to apply a Naïve Bayes classifier to predict the sentiment class of a new test document. Smoothing must be used to handle unseen words.

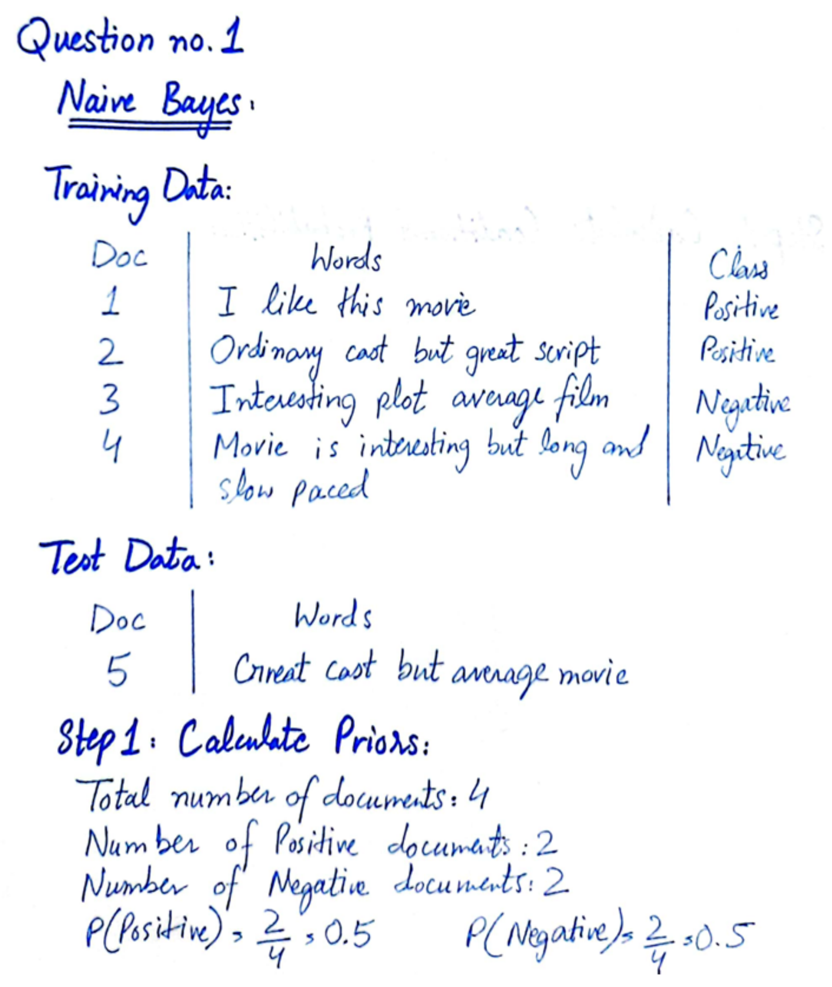
Training Data:

Doc	Words	Class
1	I like this movie	Positive
2	Ordinary cast but great script	Positive
3	Interesting plot average film	Negative
4	Movie is interesting but long and slow paced	Negative

Test Document:

You are given the following test document for classification:

Doc	Words
5	Great cast but average movie



Question no. 1

Naive Bayes.

Training Data:

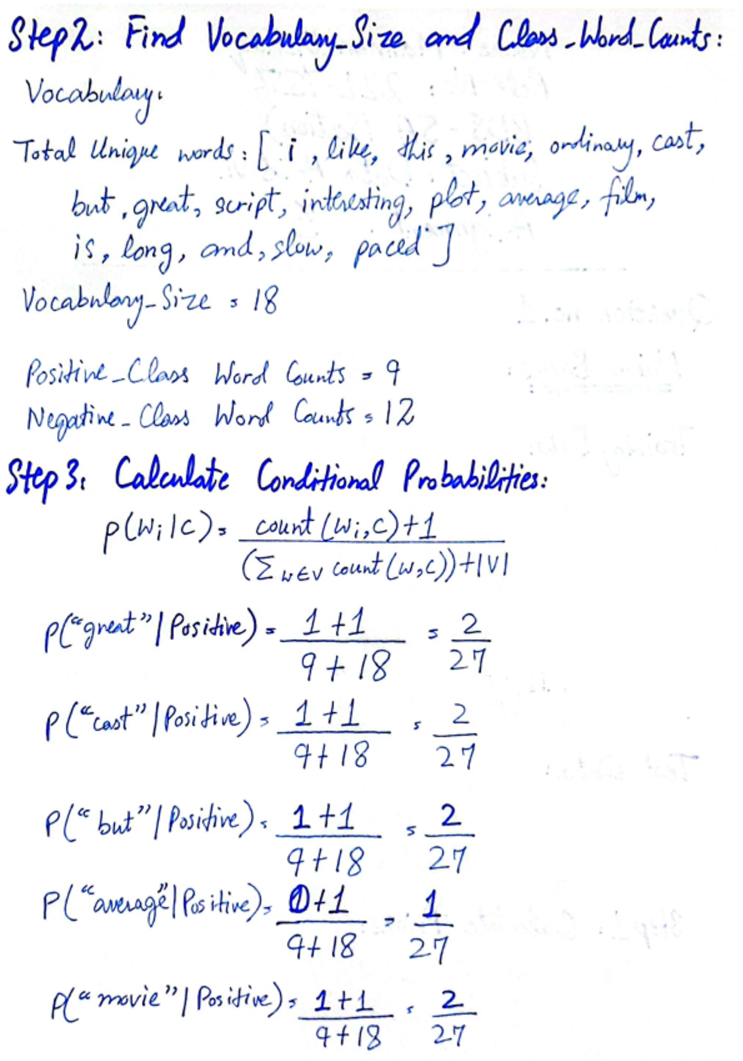
Doc	Words	Class
1	I like this movie	Positive
2	Ordinary cast but great script	Positive
3	Interesting plot average film	Negative
4	Movie is interesting but long and slow paced	Negative

Test Data:

Doc	Words
5	Great cast but average movie

Step 1: Calculate Priors:

Total number of documents: 4
Number of Positive documents: 2
Number of Negative documents: 2
$P(\text{Positive}) = \frac{2}{4} = 0.5$ $P(\text{Negative}) = \frac{2}{4} = 0.5$



Step2: Find Vocabulary_Size and Class_Word_Counts:

Vocabulary.

Total Unique words: [i , like, this , movie, ordinary, cast,
but, great, script, interesting, plot, average, film,
is, long, and, slow, paced]

Vocabulary_Size = 18

Positive_Class Word Counts = 9

Negative_Class Word Counts = 12

Step 3. Calculate Conditional Probabilities:

$$P(W_i|C) = \frac{count(W_i, C) + 1}{(\sum_{w \in V} count(w,c)) + |V|}$$

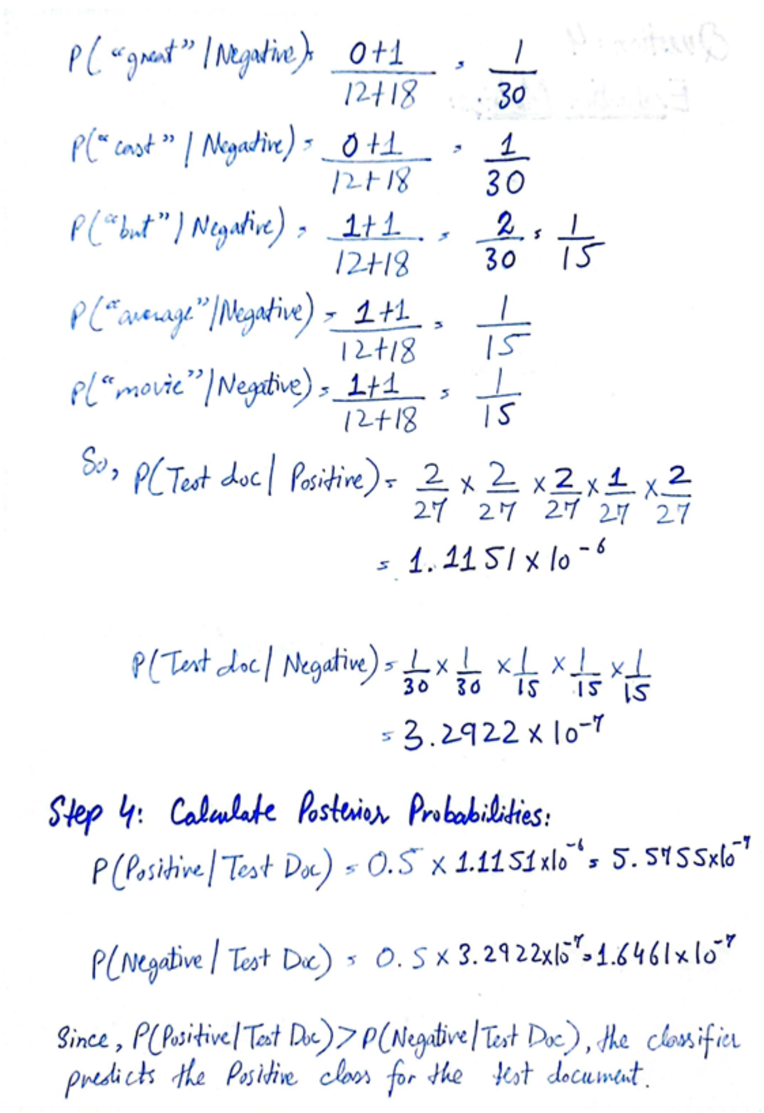
$$P(\text{“great”}|\text{Positive}) = \frac{1+1}{9+18} = \frac{2}{27}$$

$$P(\text{“cast”}|\text{Positive}) = \frac{1+1}{9+18} = \frac{2}{27}$$

$$P(\text{“but”}|\text{Positive}) = \frac{1+1}{9+18} = \frac{2}{27}$$

$$P(\text{“average”}|\text{Positive}) = \frac{0+1}{9+18} = \frac{1}{27}$$

$$P(\text{“movie”}|\text{Positive}) = \frac{1+1}{9+18} = \frac{2}{27}$$



$$P(\text{"great"} \mid \text{Negative}) = \frac{0+1}{12+18} = \frac{1}{30}$$

$$P(\text{"cast"} \mid \text{Negative}) = \frac{0+1}{12+18} = \frac{1}{30}$$

$$P(\text{"but"} \mid \text{Negative}) = \frac{1+1}{12+18} = \frac{2}{30} = \frac{1}{15}$$

$$P(\text{"average"} \mid \text{Negative}) = \frac{1+1}{12+18} = \frac{1}{15}$$

$$P(\text{"movie"} \mid \text{Negative}) = \frac{1+1}{12+18} = \frac{1}{15}$$

So, $P(\text{Test doc} \mid \text{Positive}) = \frac{2}{27} \times \frac{2}{27} \times \frac{2}{27} \times \frac{1}{27} \times \frac{2}{27}$

$= 1.1151 \times 10^{-6}$

$P(\text{Test doc} \mid \text{Negative}) = \frac{1}{30} \times \frac{1}{30} \times \frac{1}{15} \times \frac{1}{15} \times \frac{1}{15}$

$= 3.2922 \times 10^{-7}$

Step 4: Calculate Posterior Probabilities:

$P(\text{Positive} \mid \text{Test Doc}) = 0.5 \times 1.1151 \times 10^{-6} = 5.5755 \times 10^{-7}$

$P(\text{Negative} \mid \text{Test Doc}) = 0.5 \times 3.2922 \times 10^{-7} = 1.6461 \times 10^{-7}$

Since, $P(\text{Positive} \mid \text{Test Doc}) > P(\text{Negative} \mid \text{Test Doc})$, the classifier predicts the Positive class for the test document.

Evaluation Metrics for Classification

After building the Naive Bayes classifier, it is important to evaluate its performance. Several metrics are commonly used for evaluating classifiers: **Confusion Matrix**, **Recall**, **Precision**, **Accuracy**, and **F1 Score**.

1. **Confusion Matrix**:

The confusion matrix is a table that summarizes the performance of a classification algorithm by comparing predicted and actual classes. It contains four values:

- **True Positives (TP)**: Correctly predicted positive instances.
- **True Negatives (TN)**: Correctly predicted negative instances.
- **False Positives (FP)**: Instances incorrectly predicted as positive.
- **False Negatives (FN)**: Instances incorrectly predicted as negative.

Example of a confusion matrix:

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

2. **Accuracy**:

Accuracy measures the percentage of correct predictions (both positive and negative) out of all predictions:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy is useful when the classes are balanced, but it can be misleading when the data is imbalanced.

3. **Precision**:

Precision is the ratio of correctly predicted positive instances to the total predicted positives. It indicates the classifier's ability to avoid false positives:

$$\text{Precision} = \frac{TP}{TP + FP}$$

High precision means that the classifier makes fewer false positive errors.

4. **Recall (Sensitivity)**:

Recall is the ratio of correctly predicted positive instances to the total actual positives. It indicates how well the classifier finds all the positive instances:

$$\text{Recall} = \frac{TP}{TP + FN}$$

High recall means that the classifier captures most of the positive instances, even if it allows some false positives.

5. **F1 Score**:

The **F1 Score** is the harmonic mean of precision and recall. It balances both metrics and is particularly useful when there is an imbalance between precision and recall:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

A higher F1 score means a good balance between precision and recall.

Q4.(Evaluation Metrics) (10 Marks)

In a population of 100, a medical test identifies 55 sick individuals. Out of these, 33 are actually sick. Of the remaining 45 individuals, 37 are also sick. Please write answers for the following parts(mention formulas where required).

a)tp

b)fp

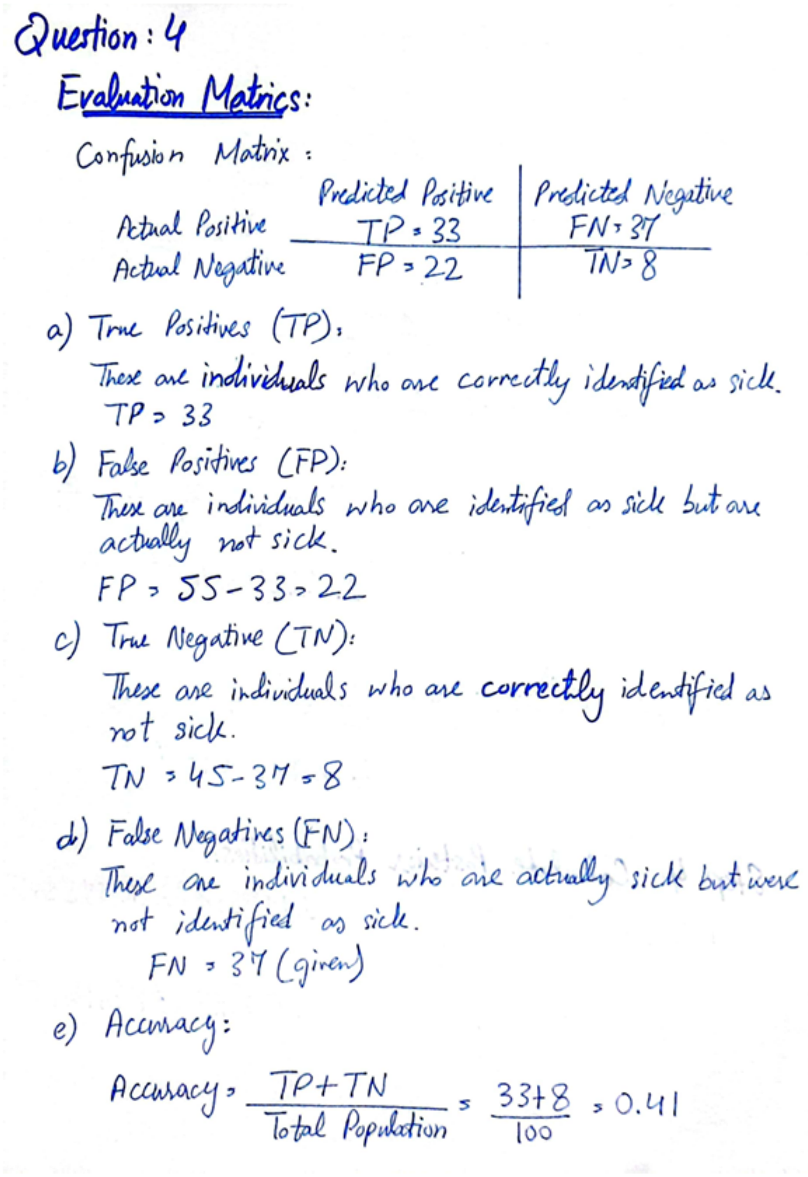
c)tn

d)fn

e)Accuracy

f)Precision

g)Recall



Question : 4

Evaluation Matrics:

Confusion Matrix :

	Predicted Positive	Predicted Negative
Actual Positive	TP = 33	FN = 37
Actual Negative	FP = 22	TN = 8

a) True Positives (TP):

These are individuals who are correctly identified as sick.

TP = 33

b) False Positives (FP):

These are individuals who are identified as sick but are actually not sick.

FP = 55 − 33 = 22

c) True Negative (TN):

These are individuals who are **correctly** identified as not sick.

TN = 45 − 37 = 8

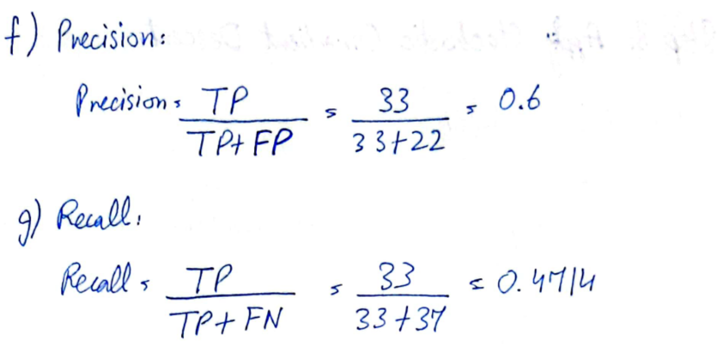
d) False Negatives (FN):

These are individuals who are actually sick but were not identified as sick.

FN = 37 (given)

e) Accuracy:

$$\text{Accuracy} = \frac{TP + TN}{\text{Total Population}} = \frac{33+8}{100} = 0.41$$



f) Precision:

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{33}{33+22} = 0.6$$

g) Recall:

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{33}{33+37} = 0.4714$$