edureka!
a Veranda Enterprise | ILLINOIS TECH

POST GRADUATE PROGRAM IN GENERATIVE AI AND ML

Natural Language Processing

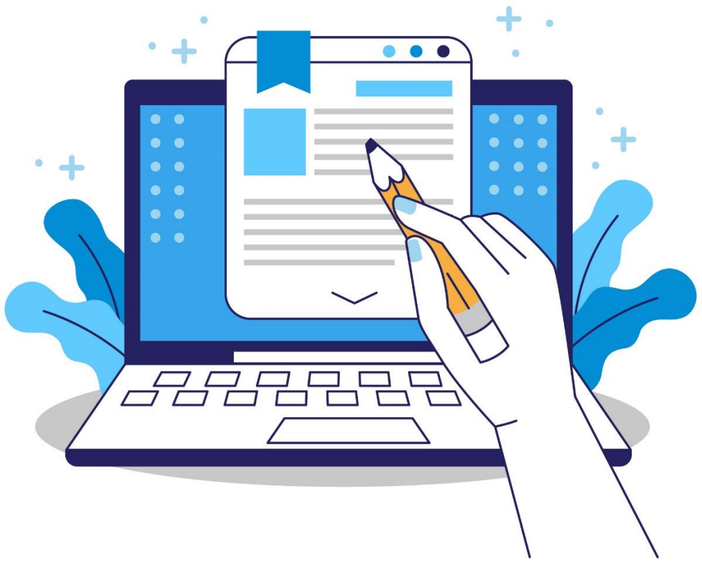
edureka!
a Veranda Enterprise

ILLINOIS TECH

Sentiment Analysis Essentials

Topics

- Introduction to Sentiment Analysis
- Rule-Based Techniques and Sentiment Lexicons
- Text Preprocessing for Sentiment Tasks
- Lexicon Scoring and Heuristics
- Traditional Machine Learning Models
- Dimensionality Reduction and Topic Modeling
- Handling Imbalanced Sentiment Datasets
- Evaluation Metrics for Sentiment Models
- Deep Learning Foundations for Sentiment



Learning Objectives

By the end of this lesson, you will be able to:

- Understand the fundamentals of sentiment analysis and distinguish between various types and applications.
- Utilize rule-based methods and sentiment lexicons (e.g., VADER, SentiWordNet) for basic sentiment detection.
- Perform effective text preprocessing tailored to sentiment tasks, including handling negations, emojis, and slang.
- Build and evaluate sentiment classification models using traditional machine learning techniques and topic modeling approaches.
- Address common challenges such as class imbalance and apply the right evaluation metrics to assess model performance.

Introduction to Sentiment Analysis

What is Sentiment Analysis?

Sentiment analysis classifies text as **negative, neutral, or positive**, aiming to understand opinions to help businesses grow.

Negative

Neutral

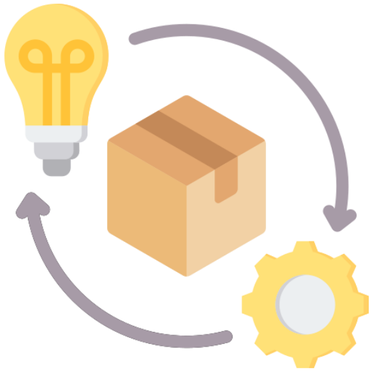
Positive

Why is Sentiment Analysis Important?

Competitor Analysis

Customer Feedback Analysis

Brand Reputation Management



Product Development and Innovation

Types of Sentiment Analysis

Fine-Grained

"Loved the product, but delivery was slow." → ⭐⭐⭐⭐ (4 stars)

Aspect-based

"The screen is vibrant, but the sound quality is poor."→ Screen: Positive, Sound: Negative

Types of Sentiment Analysis (Contd.)

Emotion Detection

"I’m absolutely thrilled with this purchase!" → Emotion: Joy

Intent Detection

"Why hasn't my refund been processed yet?" → Intent: Inquiry

Rule-Based Techniques and Sentiment Lexicons

What Are Rule-Based Techniques?

Use predefined rules and sentiment lexicons (word lists) to determine sentiment.

How it detects sentiment:

- Word lists (e.g., "happy", "bad")
- Punctuation (e.g., "!!!", "?!")
- Capitalization (e.g., "LOVE" versus. "love")
- Negation handling (e.g., "not good“ ≠ "good")

“I LOVE this!”

Strongly Positive

What is a Sentiment Lexicon?

A sentiment lexicon is a list of words, each associated with a sentiment score.

Word	Sentiment Score
good	+1.9
terrible	-2.3
average	0.0

VADER – Valence Aware Dictionary

VADER (Valence Aware Dictionary and sEntiment Reasoner) is designed to handle sentiments in social media text and informal language.

“This is AWESOME!!! 😍” → Compound Score: 0.88 → Strong Positive

SentiWordNet

An extension of WordNet, where each synset (group of synonymous words) is assigned sentiment scores.

- Positivity score
- Negativity score
- Objectivity score (neutrality)

Synset	Pos	Neg
good#a#1	0.75	0.00
bad#a#1	0.00	0.875

When to Use What?

Tool	Best For	Notes
VADER	Social media, reviews	Handles emojis & slang
SentiWordNet	News, documents	Requires WordNet POS tagging

Preprocessing Considerations for Sentiment Analysis Tasks

Why Preprocessing Matters?

Sentiment models require clean, meaningful text; messy input can result in incorrect predictions, even for simple sentences.

Sensitive to:

- Noise (extra characters, symbols)
- Misspellings ("gud" vs. "good")
- Slang ("lit", "meh")
- Emojis (😢 = sadness)
- Negation ("not good" ≠ "good")

"I am NOT happy."

"I am NOT happy."

Common Preprocessing Steps

Negation Handling

Lowercasing

Punctuation

Emoji Handling

Tokenization

Lemmatization

Stopword Removal

Special Considerations for Sentiment

01

Negations

Can reverse the sentiment

"not bad" ≠ "bad"

02

Intensifiers

Show emotional strength

"very good" = stronger positive

03

Emojis

Carry strong emotional signals

😠 = anger, 😍 = love

04

Repeated Letters

Imply emphasis or exaggeration "sooo happy" = more intense emotion

Preprocessing Strategy – Example

"I didn’t like the movie!!! 😡"

- **Lowercase** → "i didn’t like the movie!!! 😡"
- **Tokenize** → ["i", "didn’t", "like", "the", "movie", "!!!", "😡"]
- **Negation Handling** → "didn’t like" → "not_like"
- **Remove Stop words (optional)** → ["not_like", "movie", "😡"]
- **Emoji Mapping** → 😡 → "angry"
- **Remove Extra Punctuation** → ["not_like", "movie", "angry"]

["not_like", "movie", "angry"]

Ready for sentiment scoring!

Lexicon Scoring and Heuristics in Polarity Detection

What is Lexicon Scoring?

Lexicon scoring uses a predefined list of words, each with a sentiment value, to evaluate the overall polarity of a sentence.

"The food was amazing, and the service was horrible."

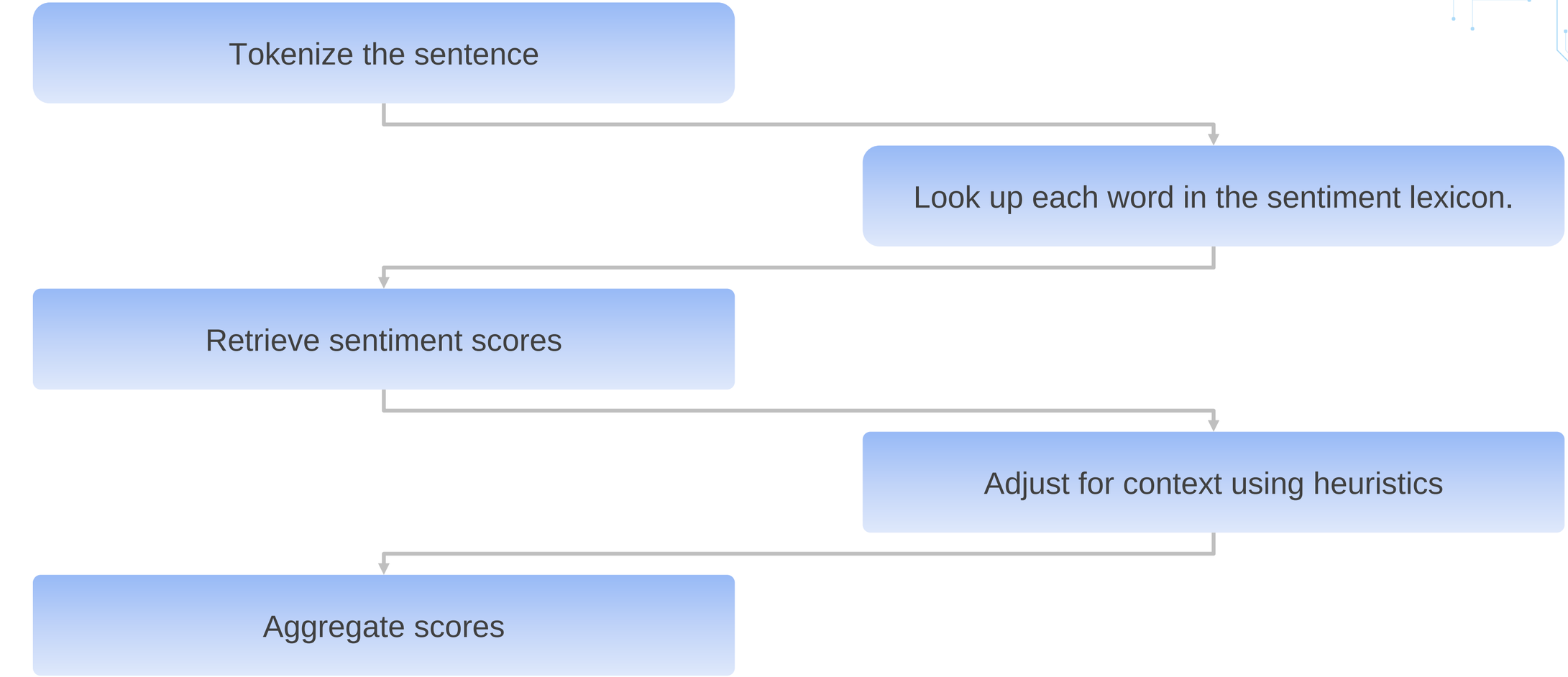
amazing ⟶ **+2.0**

horrible ⟶ **-2.5**

Total Score: -0.5

Overall Sentiment: Slightly Negative

How Are Scores Computed?



Heuristics That Boost Accuracy

Heuristics = Smart rules that refine how sentiment is interpreted.

Heuristic	Example	Effect on Score
Negation	"not happy"	Flip score → happy(+2) → -2
Booster words	"really awesome"	Multiply → awesome(+2) × 1.5
Punctuation	"good!!!"	Boost → +1 becomes +1.5
Capitalization	"AWFUL"	Amplify → -2 becomes -3

VADER’s Scoring Strategy

01 Lexicon Lookup

02 Negation Handling

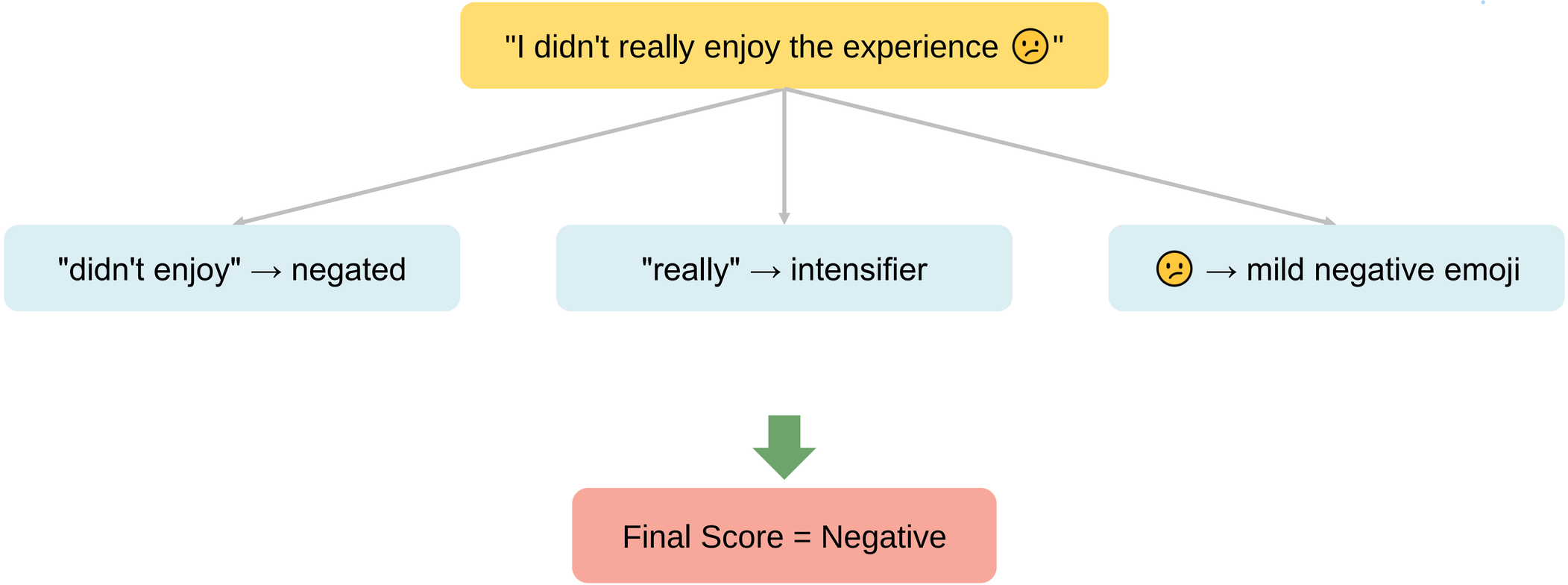
03 Booster/Intensifier Adjustment

04 Punctuation Amplification

05 Capitalization

06 Emoji/Emoticon Analysis

Example – Putting It All Together

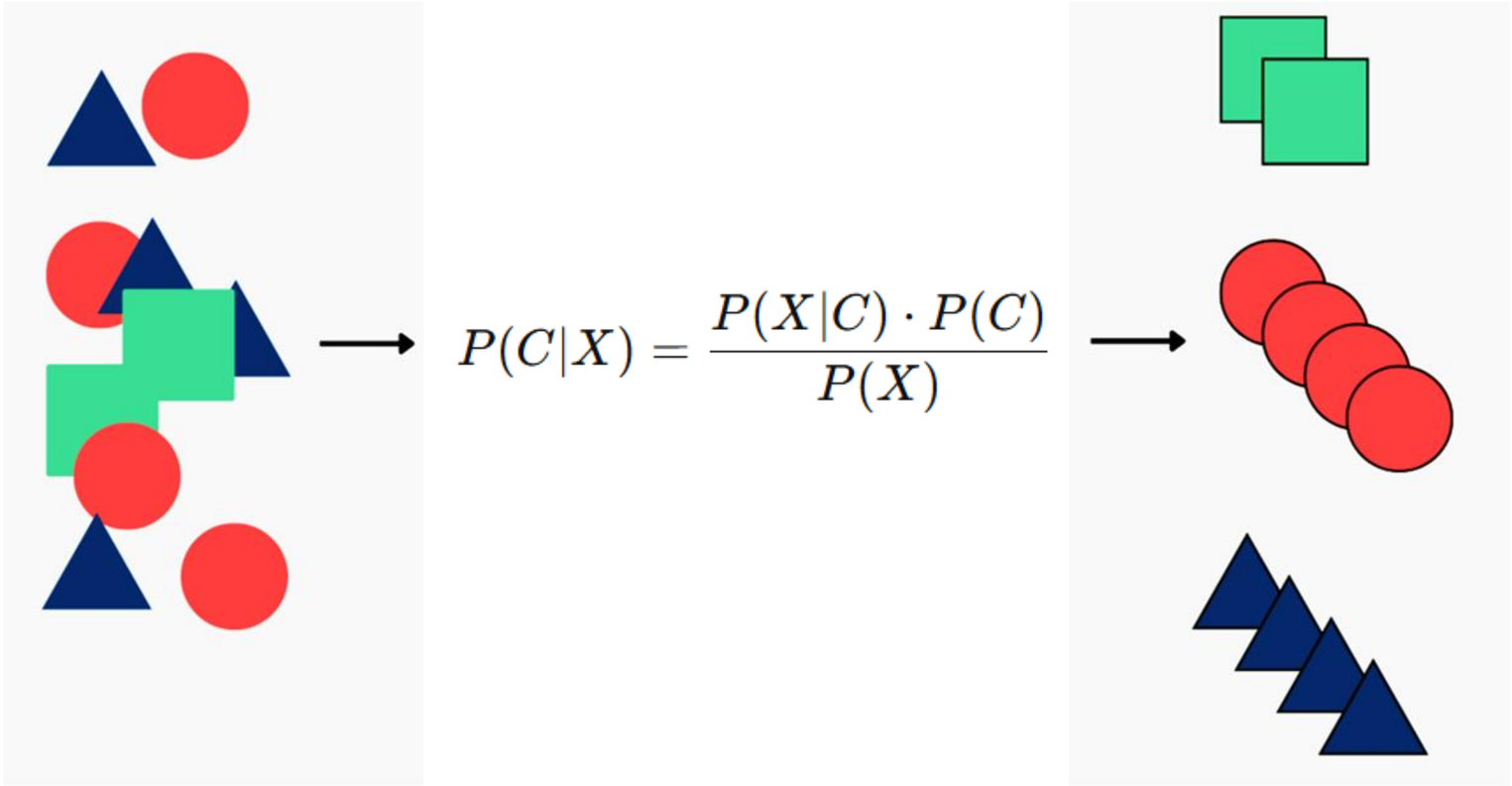


Naïve Bayes and Support Vector Machines for Sentiment Classification

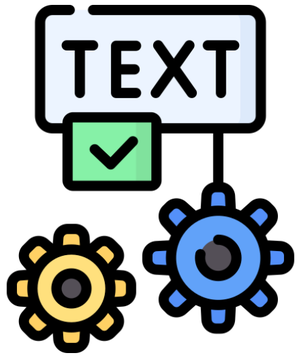
Naïve Bayes

The Bayes theorem describes the probability of an event based on the prior knowledge of the conditions that might be related to the event.

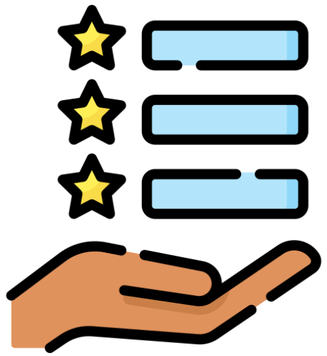
- C is the class label
- X is the feature set
- P(C | X) is the posterior probability
- P(C) is the prior probability
- P(X | C) is the likelihood
- P(X) is the evidence



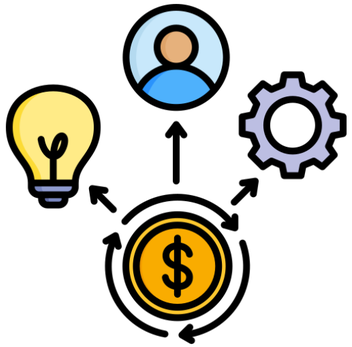
Sentiment Classification (Naïve Bayes)



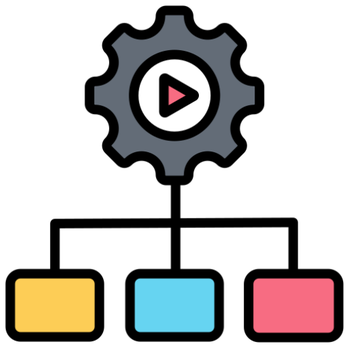
Text Preprocessing



Feature Extraction



Training: Estimate
P(wi | C) and P(C)

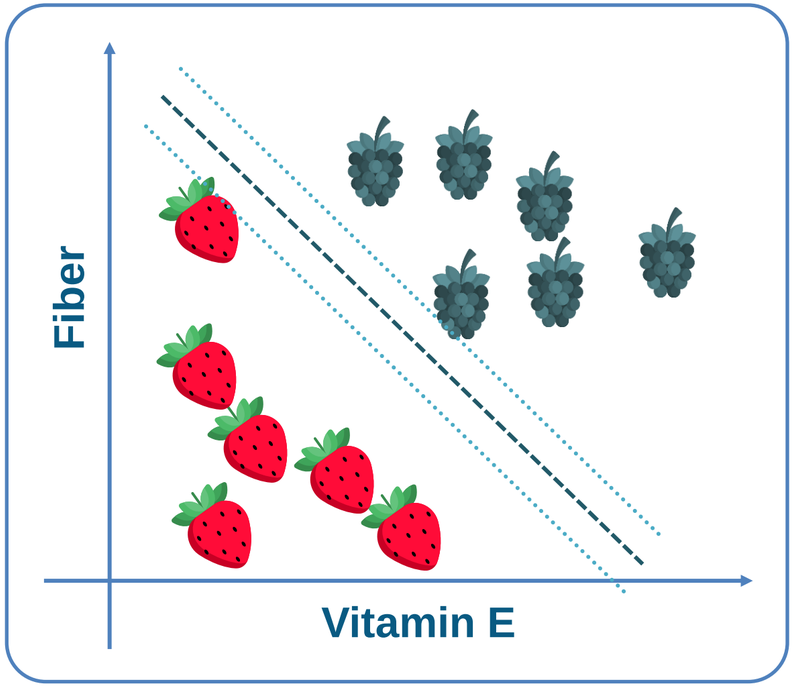


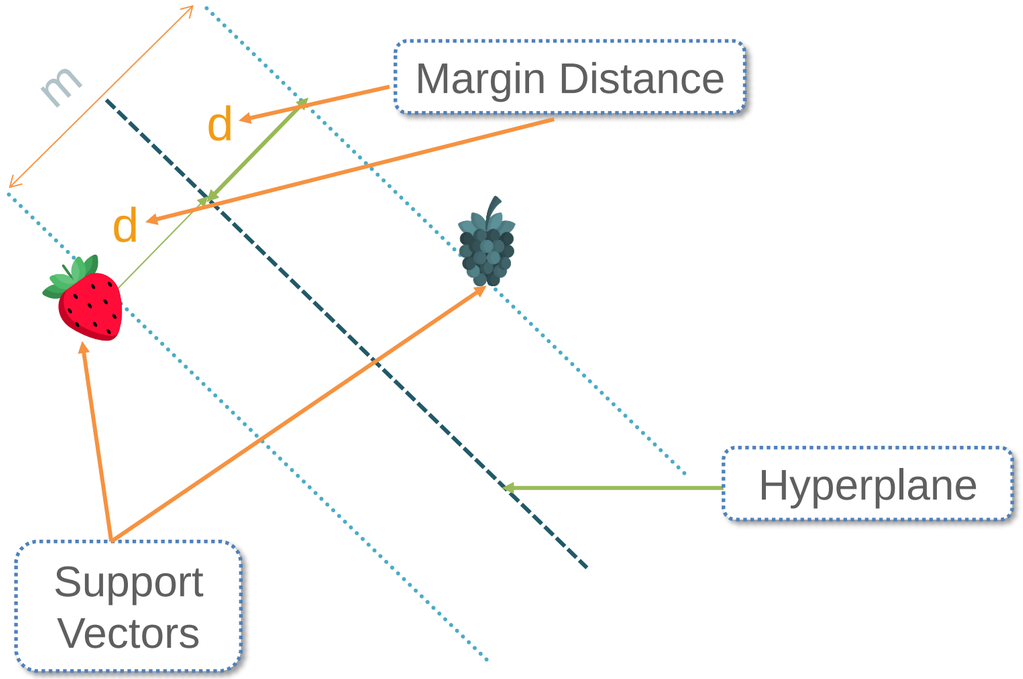
Prediction using

$$\hat{C} = \arg\max_{C} \left(P(C) \prod_{i=1}^{n} P(w_i|C) \right)$$

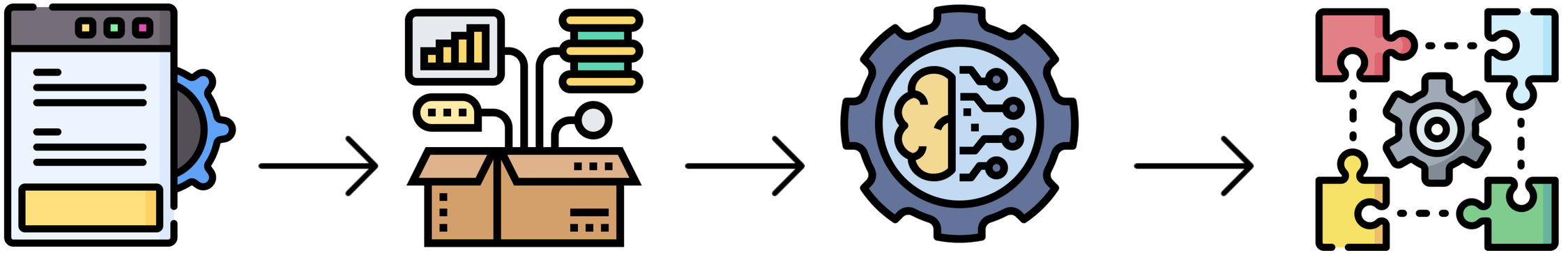
Support Vector Machine (SVM)

SVM is a discriminative classifier that finds the optimal hyperplane that best separates data points from different classes.





Sentiment Classification (SVM)



Preprocessing

Feature Extraction

Training: Learn the hyperplane using labeled data.

Prediction: Classifying based on the side of hyperplane.

Naïve Bayes vs. SVM

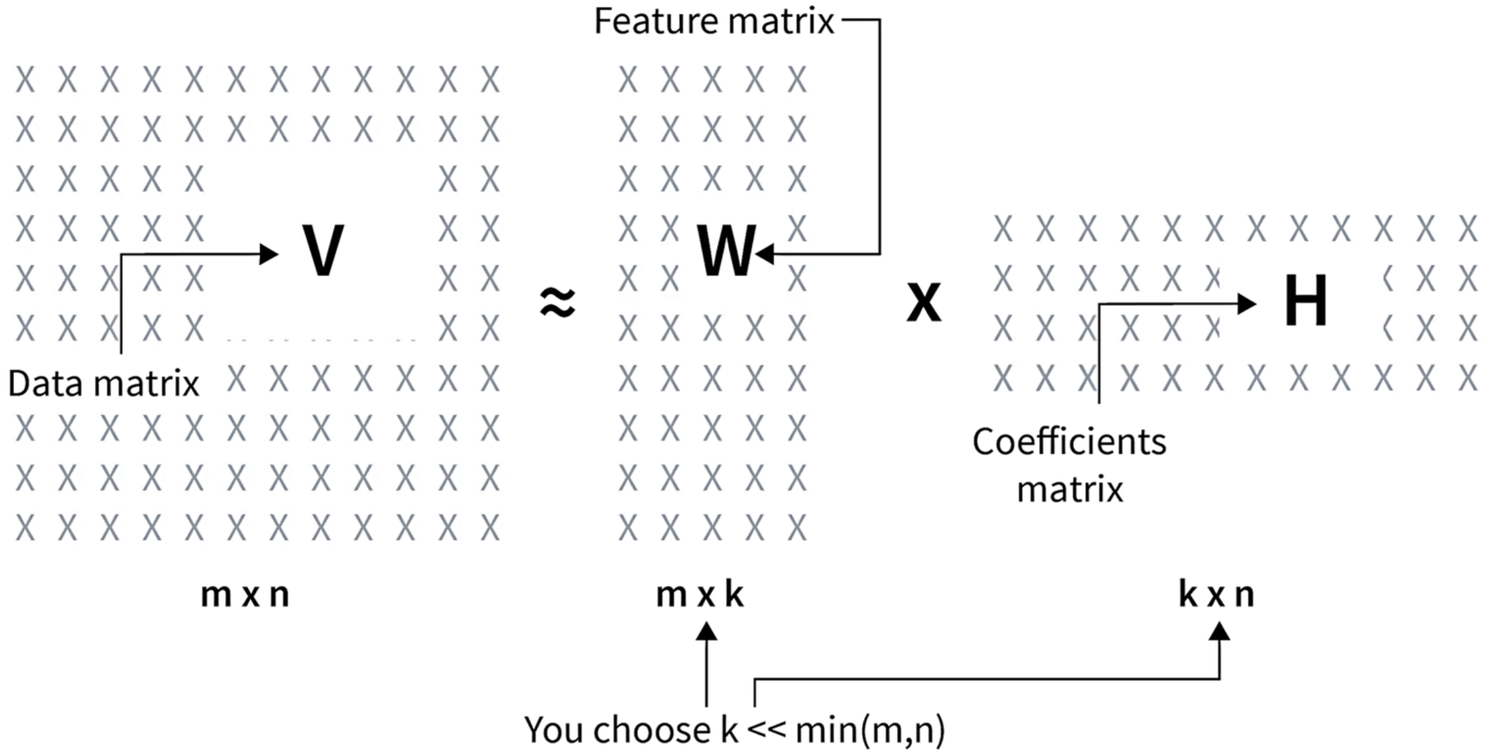
Feature	Naïve Bayes	SVM
Model Type	Probabilistic	Discriminative
Assumptions	Word independence	No specific assumption
Performance	Good baseline	Often better accuracy
Training Time	Very fast	Slower on large datasets
Output	Class probabilities	Hard classification
Handling High Dim.	Excellent	Excellent
Sensitivity to Noise	Less robust	More robust

Non-Negative Matrix Factorization (NMF)

Non-Negative Matrix Factorization (NMF)

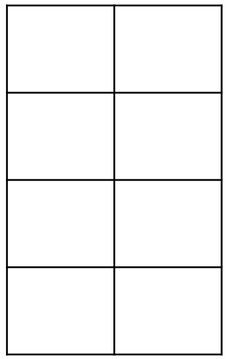
A Technique used to **reduce the number of features** in high-dimensional data, especially when the data is non-negative.

- V = original matrix (size: m × n)
- W = basis matrix (size: m × k)
- H = coefficient matrix (size: k × n)
- k = number of latent features (lower than m or n)

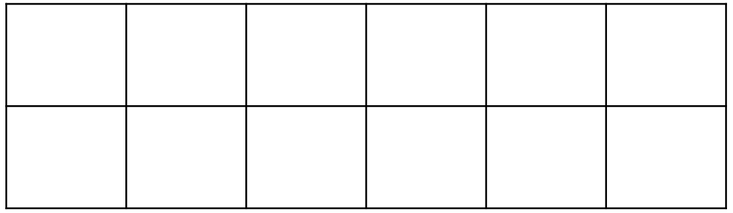


Example (Text-Based)

Let’s say we have a 4-document, 6-word term-frequency matrix V. NMF with k = 2 topics decomposes it into:



W (4×2) → documents represented in 2 topics



H (2×6) → words’ distribution in 2 topics

Now we can:

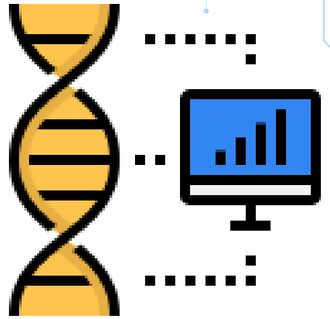
- Interpret topics based on top words in H
- Classify/clustering documents via W

NMF vs. PCA

Feature/Criteria	NMF	PCA
Full Name	Non-Negative Matrix Factorization	Principal Component Analysis
Input Requirements	Non-negative data	Can handle any data
Output Matrices	W (basis), H (coefficients) – both ≥ 0	Components may contain negative values
Interpretability	High (easy to explain parts/topics)	Low (hard to interpret negative weights)
Sparsity	Promotes sparse solutions	Usually dense
Use in NLP	Excellent for topic modeling	Less interpretable topics
Use in Image Processing	Extracts parts-based features	Blurred holistic features
Decomposition Goal	Minimize reconstruction error	Maximize variance captured in projections
Handles Negatives	No (only non-negative)	Yes
Ease of Interpretation	Very easy	Difficult
Best When	Features are non-negative and additive	You want a compact representation of variation

Applications of NMF

Recommender systems



Bioinformatics

Text mining

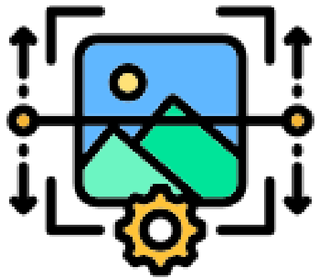


Image processing

Topic Modeling in Sentiment Tasks: Latent Dirichlet Allocation (LDA)

What is LDA (Latent Dirichlet Allocation)?

LDA is a **generative probabilistic model** that assumes:

Each document is a mixture of topics

Each topic is a distribution over words

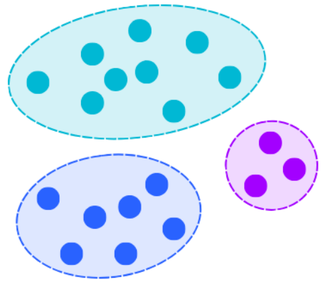
Topic Modeling using LDA

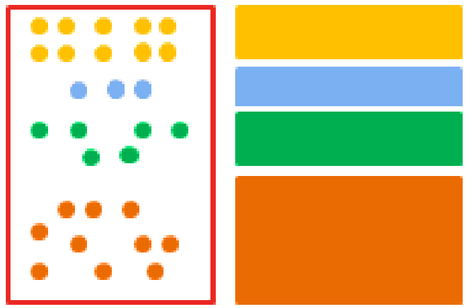
A powerful unsupervised technique that helps in discovering the **hidden thematic structure** (topics) in a large collection of texts.

Collection of Text Documents

Topic Modeling

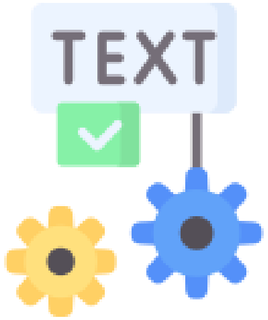
Cluster of words by topic





Cluster of document by topic

How LDA Works?



Text Preprocessing

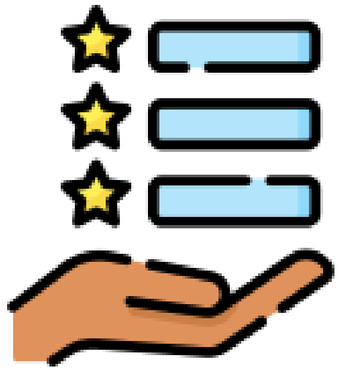
Iterative Update
(Gibbs Sampling)

Input: Collection of documents

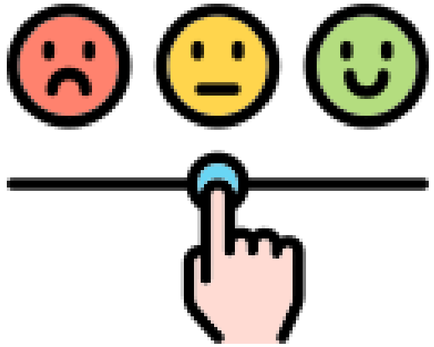
Initialize:
Number of topics **K** to discover
Randomly assigning topics

Output: Top words
Topic distribution

Role of LDA in Sentiment Analysis



Feature Engineering



Topic-Sentiment Pairing

Aspect-Based Sentiment Analysis (ABSA)

Handling Imbalanced Sentiment Datasets

What is an Imbalanced Sentiment Dataset?

An imbalanced dataset has a **disproportionate distribution of sentiment classes.**

For example:

- **90% Positive**
- **8% Neutral**
- **2% Negative**

A classifier trained on this may **always predict "positive"** to get high accuracy, ignoring the minority sentiment classes.

Why Is This a Problem?

Misleading accuracy

Poor recall/precision

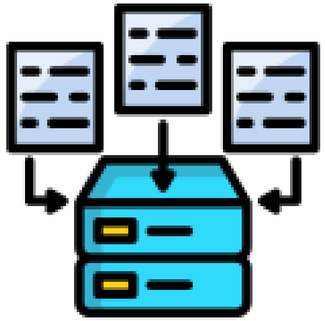
Unfair/biased predictions

Solutions to Handle Imbalanced Datasets

Data-Level Strategies

Data Augmentation (Text)

Filtering Noisy Data



Evaluation Metrics

Precision

Recall

F1- Score

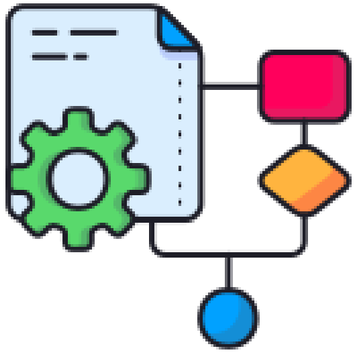
ROC-AUC

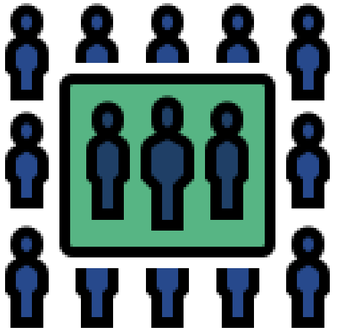
Confusion Matrix

Algorithmic Approaches

Cost-Sensitive Ensemble

Methods for Imbalance





Resampling Techniques

Undersampling the Majority Class

Oversampling the Minority Class

ADASYN

Evaluation Metrics and Semantic Measures

Accuracy

Accuracy is the **overall correctness** of the model and fraction of predictions the model got right.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Predictions}}$$

In Sentiment Analysis: If a transformer like BERT classifies tweets into **positive**, **negative**, or **neutral**, accuracy tells us how many were labeled correctly.

Precision

Precision is the proportion of **correctly predicted positive** observations to the **total predicted positives**.

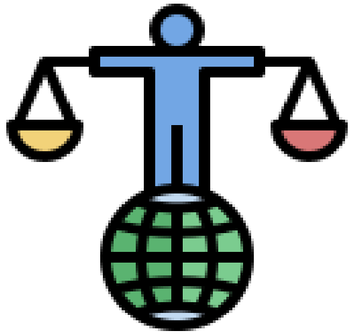
$$\text{Precision (class)} = \frac{TP}{TP + FP}$$

In Sentiment Analysis: High **precision for "negative"** means the model **doesn't wrongly label neutral/positive text as negative**.

Sentiment Polarity

Polarity refers to the direction of sentiment — whether a text is positive, negative, or neutral.

Positive Polarity
Polarity Score: +1



Neutral Polarity
Polarity Score: 0

Negative Polarity
Polarity Score: -1

Sentiment Intensity

Sentiment intensity measures the strength or magnitude of the sentiment expressed.

Intensity Score:

- Indicates how positive or negative the sentiment is.
- Example: “good” vs “excellent” — both are positive, but "excellent" has higher intensity.

LSTMs and GRUs for Sequential Sentiment Modeling

Why Sequential Models in Sentiment Analysis

"I thought the hotel would be terrible, but it wasn't bad — the staff won me over."

Traditional

❌ Negative

Picks up "terrible", "bad"; ignores flow

Sequential

✅ Positive

Understands contrast, negation, sentiment shift

What is LSTM?

Imagine a conveyor belt that runs through multiple LSTM cells — this belt carries the cell state (memory) forward.

Forget Gate

Decides what information to throw away from the cell state.

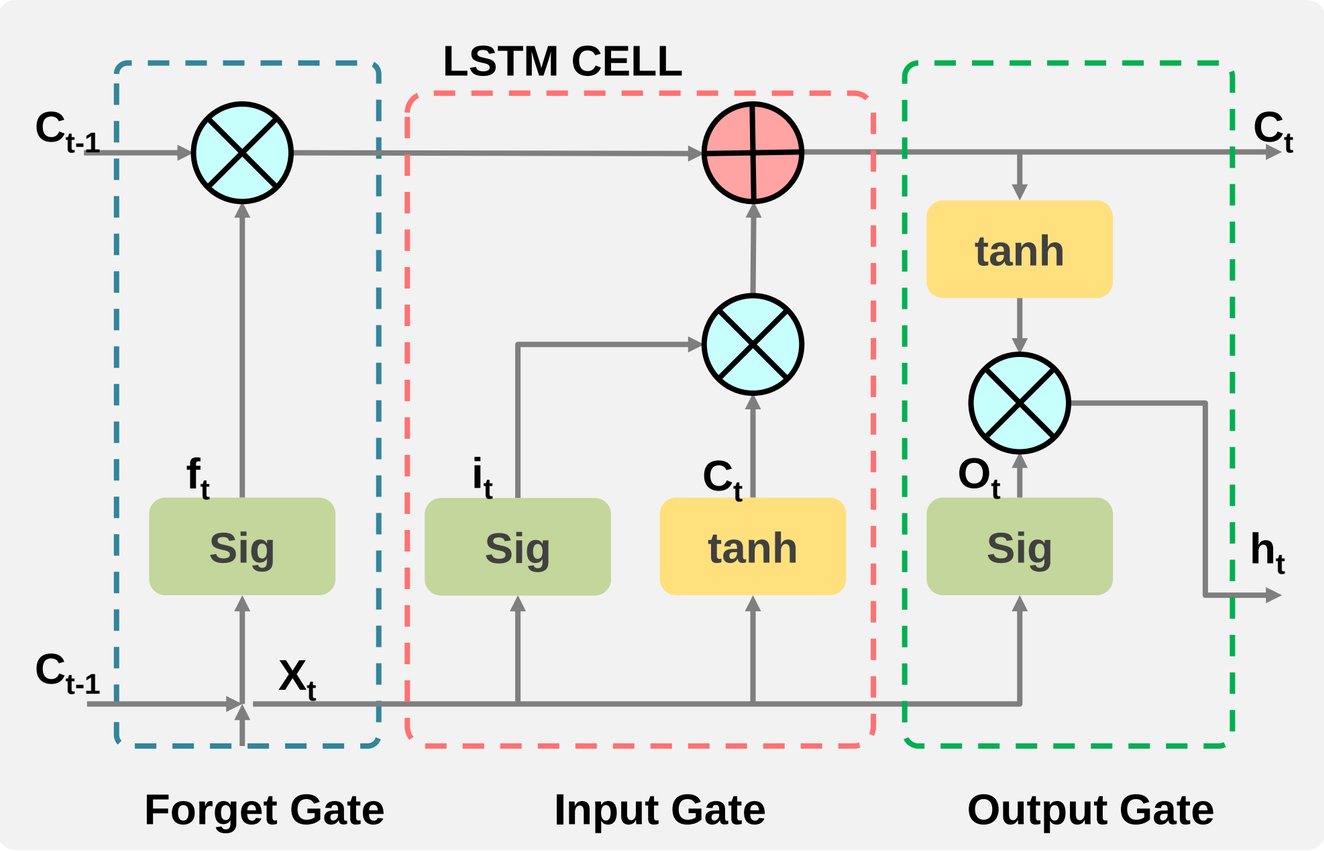
Input Gate

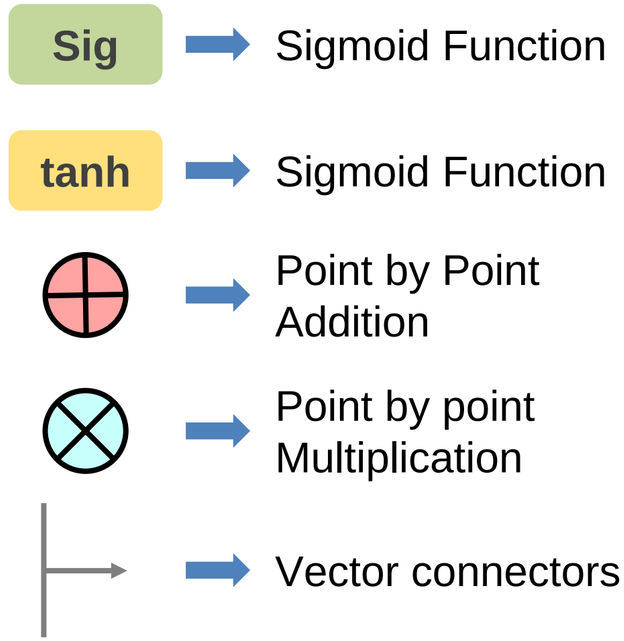
Determines what new information to add to memory.

Output Gate

Decides what to output at this step (based on current memory + input)

LSTM Structure





GRU – Gated Recurrent Unit

Now imagine the GRU as a smart switchboard that makes faster decisions.

Update Gate

controls how much of the past to keep

“Should I update my memory with new info or keep the old one?”

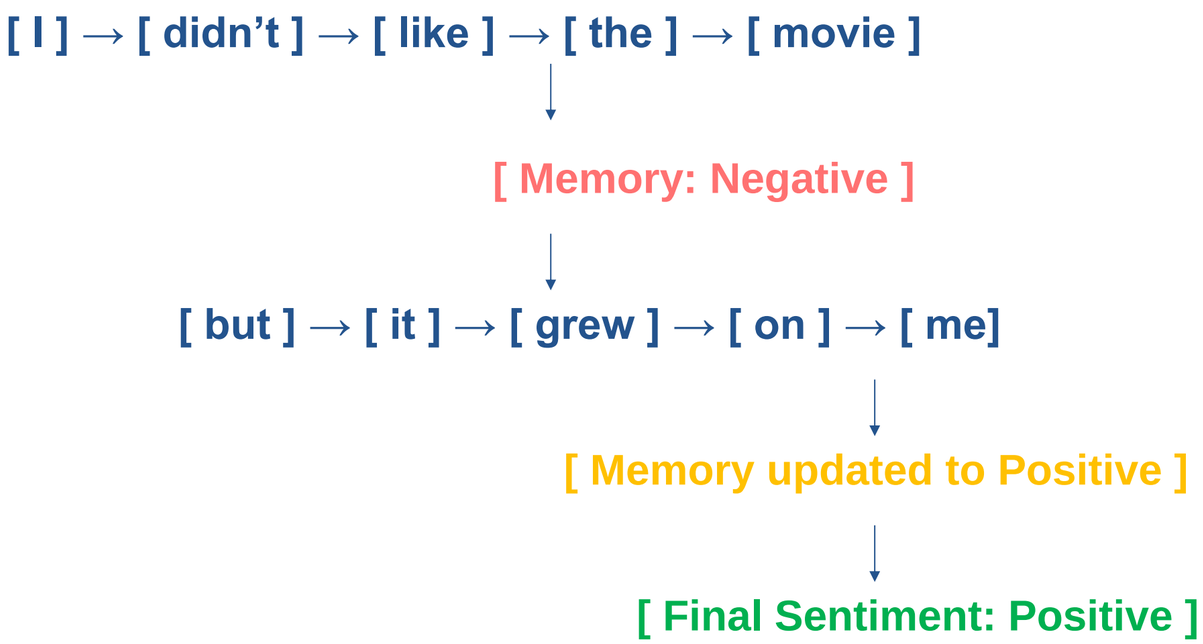
Reset Gate

decides how much of the past to ignore

“Do I want a fresh start for this word, or do I rely on memory?”

How They Work for Sentiment

Sequential models read text like humans — word by word, remembering the past and interpreting the sentiment of the entire sentence based on flow and contrast.



Attention Mechanisms in Deep Sentiment Models

Why Attention?

"The camera quality was amazing, but the battery life is disappointing."

Problem in NLP: Not every word contributes equally to the overall sentiment.

Word	Importance 🔍
The	⚪ Low
camera	⚪ Low
amazing	🔴 High
but	⚪ Low
disappointing	🔴 High

What is Attention?

Text Input → Tokenized → Encoded → Attention Layer

Query × Key → Score

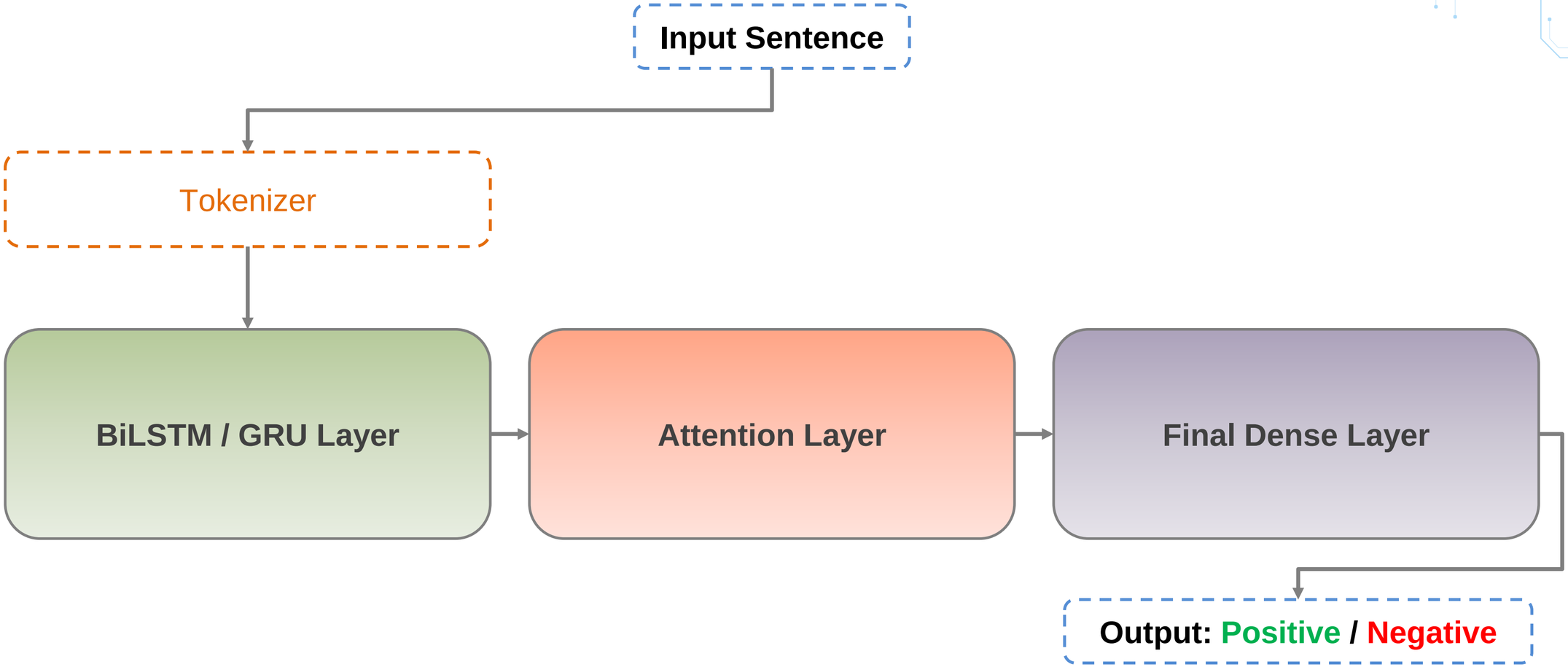
Softmax(Score) → Weight

Weight × Value → Context Vector

"The camera quality was amazing, but the battery life is disappointing."

0.85 **0.91**

Model Architecture



Example – Attention Map

I	enjoyed	the	plot	but	the	ending	was	terrible
0.1	0.7	0.05	0.1	0.05	0.05	0.1	0.1	0.85

High Attention: enjoyed, terrible

Low Attention: I, the, but

Token	Importance
I	0.1
enjoyed	0.7
terrible	0.85

Multi-Model Sentiment Analysis Pipeline for Short Text (Demonstration)

Note: Refer to Module 5: Demo 1 on LMS for detailed steps.

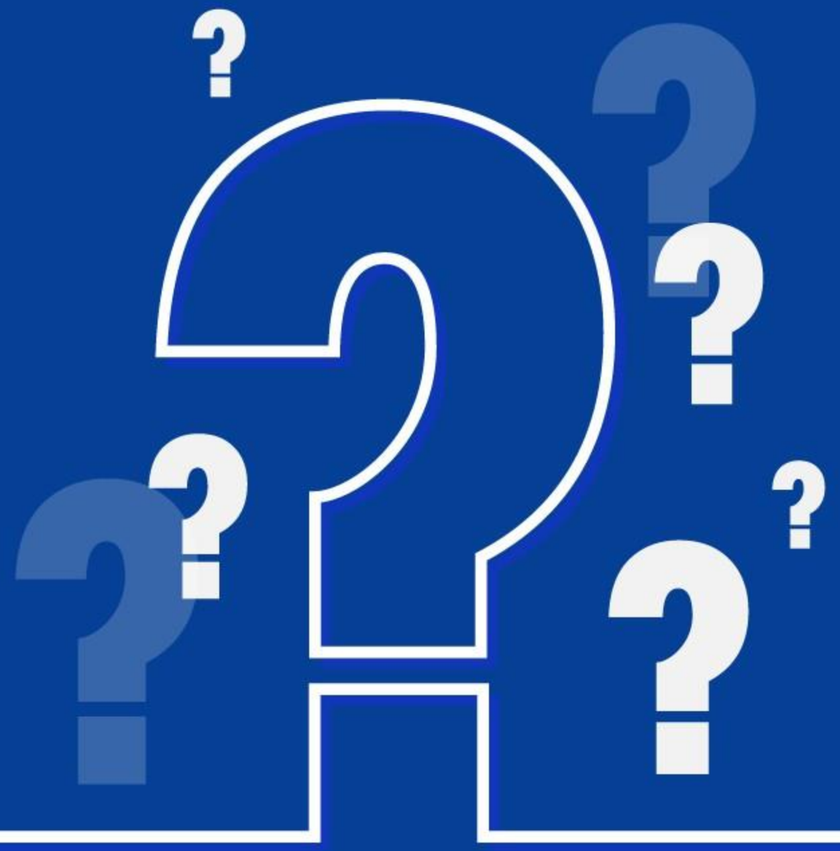
Summary

In this lesson, you have learned to:

- Explore different types and applications of sentiment analysis.
- Use rule-based techniques and lexicons like VADER and SentiWordNet for sentiment detection.
- Apply effective text preprocessing methods tailored for sentiment tasks.
- Build sentiment classification models using traditional machine learning and topic modeling.
- Address challenges like data imbalance and evaluate model performance using appropriate metrics.

edureka! a Veranda ENTERPRISE

Questions



edureka! a Veranda ENTERPRISE

Feedback

edureka! a Veranda ENTERPRISE

Thank You

For information, Please Visit our Website
www.edureka.co