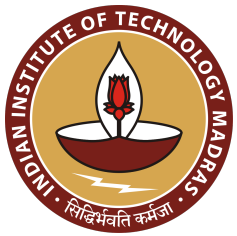
# BSCS5002: Introduction to Natural Language Processing

## Classification Models: Naïve Bayes, Logistic Regression, Clustering

Parameswari Krishnamurthy




Language Technologies Research Centre
IIIT-Hyderabad

*param.krishna@iiit.ac.in*

# POS and NER

**Natural Language Processing (NLP):**

- POS Tagging and NER are fundamental tasks in NLP pipelines.
  - POS: Assigning parts of speech to each word in a sentence.
  - NER: Identifying entities like names, locations, and organizations in text.
- Provides linguistic insights that can improve model performance.
- Aids in feature extraction for downstream tasks like text classification.

# Importance of Classification Models

**Classification Models:**

- Essential for predicting categories like POS tags or named entities.
- Handle a variety of features like:
  - Word frequency
  - Contextual information
  - Orthographic features
- Provide probabilistic outputs that can be interpreted and analyzed.
- Offer flexibility in feature engineering to capture relevant patterns.
- Can be used in both supervised and semi-supervised learning settings.
- Important for building robust NLP models that generalize well across domains.

# Classification Models for POS and NER Tasks

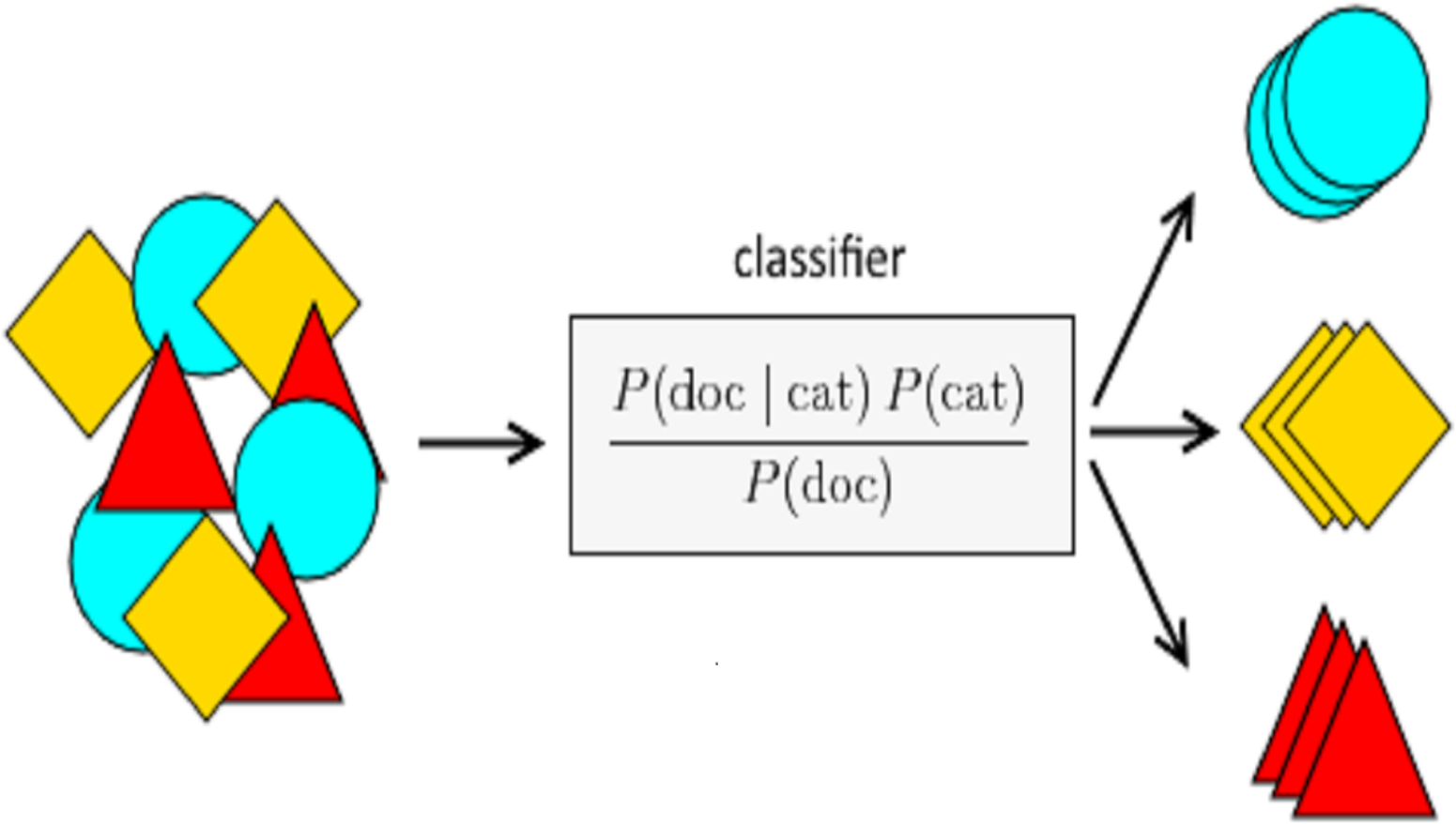
- **Naive Bayes**
  - Simple probabilistic model
  - Assumes independence between features
  - Often used for POS tagging and NER
- **Logistic Regression**
  - Discriminative model
  - Predicts the probability of POS tags or named entities
  - Can be extended for multi-class classification
- **Clustering**
  - Groups words into clusters based on feature similarity
  - Can be used for unsupervised NER
  - Examples: K-means

# 1. Naive Bayes

- A simple yet powerful probabilistic classification algorithm.
- Based on Bayes' Theorem, which describes the probability of a class given a set of features.
- Assumes that each feature contributes independently to the probability of the class — this is called the **naive assumption**.

**Key Characteristics:**

- **Probabilistic Model:** Uses probability theory to predict the class of new, unseen data.
- **Independence Assumption:** Assumes that the presence of one feature does not affect the presence of another, given the class.
- **Simple and Fast:** Particularly effective for large datasets with high dimensionality, such as text data.
- Performs well with limited computational resources and data.




source: https://insightimi.wordpress.com/2020/04/04/naive-bayes-classifier-from-scratch-with-hands-on-examples-in-r/

# Naive Bayes Classifier - Bayes' Theorem

**Bayes' Theorem:**

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

- $Y$: Target class (POS tag or NER category).
- $X$: Observed features (e.g., words, context).
- Assumption: Features are conditionally independent given the class.
- Simplicity and efficiency make it suitable for large-scale NLP tasks.
- Works well with sparse data, common in text classification.
  - Often used as a baseline model due to its simplicity.
  - Provides a good balance between performance and interpretability.

# Naive Bayes for POS Tagging

**How it works:**

- Given a word, predict its POS tag by calculating $P(\text{Tag}|\text{Word})$.
- Use prior probabilities $P(\text{Tag})$ and likelihood $P(\text{Word}|\text{Tag})$.
- Assumes that the occurrence of each word is independent of the others.
- Suitable for applications where speed is crucial, like real-time text processing.
- Works well with a small amount of training data due to its probabilistic nature.
  - Often combined with smoothing techniques to handle unseen words.

**Example:**

$$P(\text{VB}|\text{run}) = \frac{P(\text{run}|\text{VB}) \cdot P(\text{VB})}{P(\text{run})}$$

# Naive Bayes for NER

**Application to NER:**

- Predict entity type (Person, Location) based on features.
- Example: Predicting "London" as a location.
- Useful for classifying entities in domains with limited labeled data.
- Adaptable for multi-class classification with multiple entity types.
- Often used in conjunction with other models in an ensemble.
- Handles noisy and imbalanced data effectively.

**Formula:**

$$P(\text{Location}|\text{London}) = \frac{P(\text{London}|\text{Location}) \cdot P(\text{Location})}{P(\text{London})}$$

# Feature Selection for POS and NER

**What is Feature Selection?**

- Process of identifying and selecting the most relevant features (e.g., words, word shapes, suffixes) from the dataset.
- Reduces dimensionality by eliminating irrelevant or redundant features.

**Why is Feature Selection Important?**

- **Improves Accuracy:** Helps the Naive Bayes model focus on the most informative features.
- **Reduces Noise:** Removes irrelevant words that don't contribute to POS or NER predictions.
- **Prevents Overfitting:** Reduces the risk of the model memorizing the training data rather than generalizing.
- **Increases Efficiency:** Less data to process means faster training and testing.

**Applications in POS Tagging and NER:**

- Selects relevant features such as word prefixes, suffixes, capitalization, and neighboring words.

# Multinomial Naive Bayes

**What is Multinomial Naive Bayes?**

- A variant of Naive Bayes used for discrete, count-based data.
- Particularly effective for text classification tasks, where the features are word frequencies or occurrences.
- Assumes that feature vectors represent counts of occurrences (e.g., word counts in a document).

**Key Characteristics:**

- **Discrete Features:** Suitable for categorical data like word frequencies, not continuous features.
- **Feature Independence:** Assumes each word's occurrence is independent of others given the class.
- **Commonly Used in NLP:** Highly effective in applications like spam filtering, text categorization, and sentiment analysis.

**Formula:**

$$P(X|Y) = \prod_{i=1}^{n} P(x_i|Y)^{x_i}$$

Where $x_i$ is the count of feature $i$, and $Y$ is the class label.

# Logistic Regression - Introduction

**Logistic Regression:**

- Predicts probabilities for binary or multi-class classification.
- Uses the logistic function:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n)}}$$

- Provides a linear decision boundary for classification tasks.
- Can handle both binary and multi-class problems.
- Regularization techniques:
  - L1 regularization: Encourages sparsity.
  - L2 regularization: Prevents overfitting by penalizing large coefficients.
- Interpretable model: Coefficients $\beta_i$ provide insights into feature importance.

# Logistic Regression for POS Tagging

**How it works:**

- Model learns weights $\beta_i$ for each feature $X_i$.
- Predicts the tag with the highest probability.
- Captures linear relationships between features and the target variable.
- Suitable for tasks where interpretability is important.
- Can be extended to include interaction terms between features.
- Effective in domains with a large number of features, like text processing.

**Example:**

- Word: "book"
- Features: Context words, POS tags of surrounding words.
- Model output: Predicts the POS tag (e.g., noun, verb) with the highest probability.

# Logistic Regression for NER

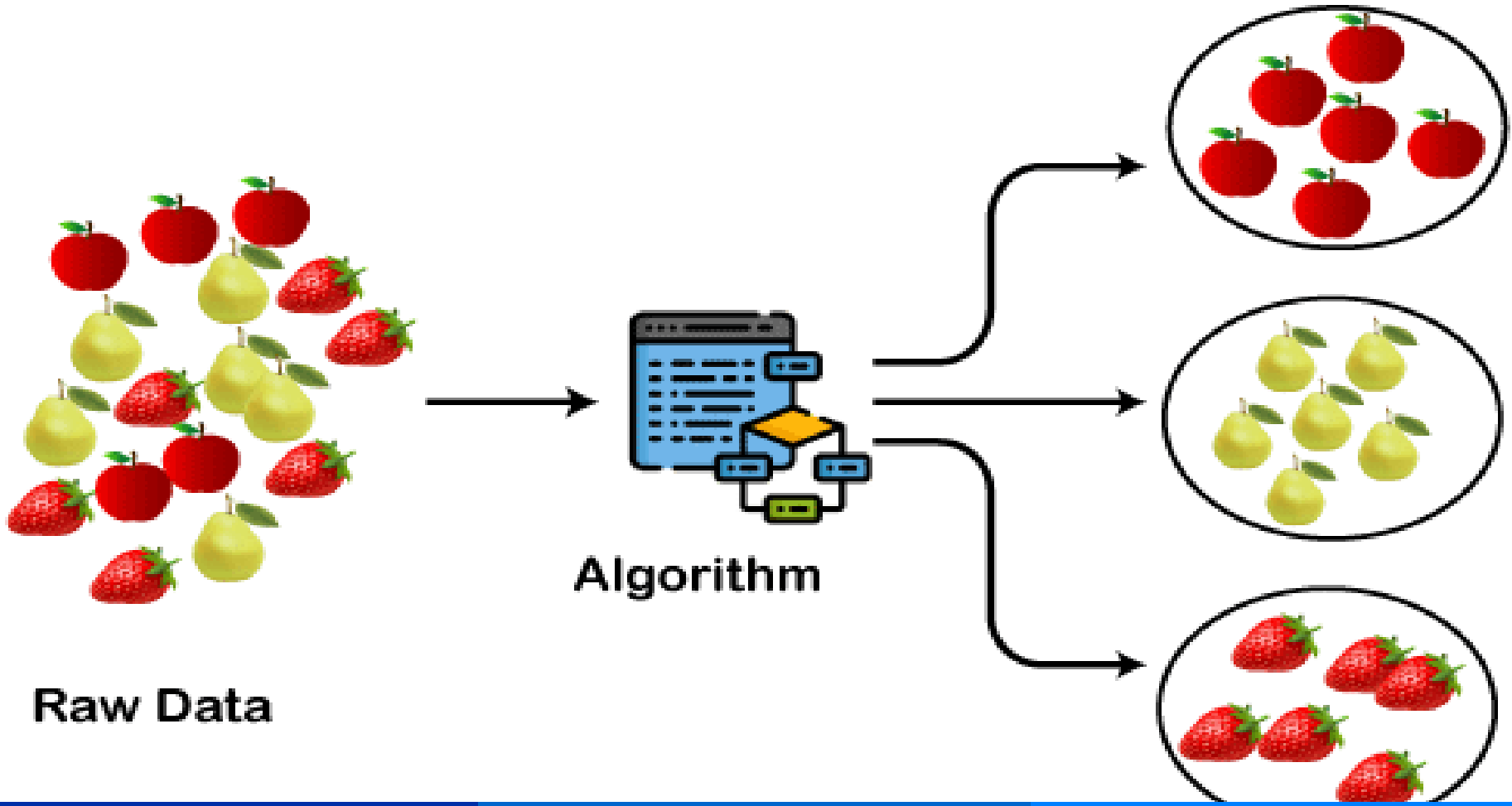
**Application to NER:**

- Predicts the probability of a word being an entity (e.g., person, organization).
- Leverages features like:
  - Capitalization.
  - Word shape (e.g., camel case).
  - Position in the sentence.
- Can be used for multi-class classification:
  - Different entity types: Person, Location, Organization.
- Handles large feature spaces effectively with regularization.
- Allows for the inclusion of domain-specific features.

# Clustering - Introduction

**Clustering:**

- Unsupervised learning technique for grouping similar data points.
- Clustering is the process of arranging a group of objects in such a manner that the objects in the same group (which is referred to as a cluster) are more similar to each other than to the objects in any other group.



# K-Means Clustering

**K-Means Algorithm:** It is a centroid-based algorithm where the user must define the required number of clusters it wants to create.

- Step 1: Initialize $K$ cluster centroids.
- Step 2: Assign points to the nearest centroid.
- Step 3: Recompute centroids based on the current assignment.
- Step 4: Iterate until centroids no longer change.
- Sensitivity to the initial choice of $K$ and centroids.
- Commonly used for clustering tasks in NLP like:
  - Word clustering into POS categories.
  - Entity clustering in NER tasks.

# K-Means Clustering

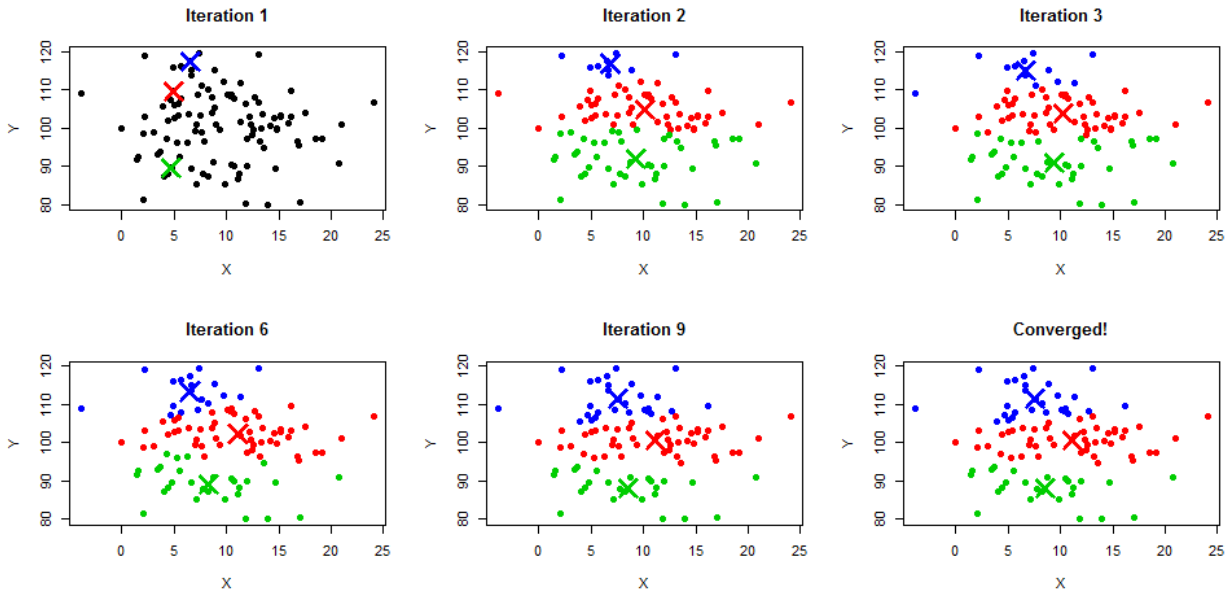



Figure: K-Means Clustering

# Clustering for POS Tagging

**Application:**

- Group words with similar contexts into clusters.
- Helps in unsupervised POS tagging in low-resource settings.
- Can reveal hidden syntactic structures in a corpus.
- Useful as a preprocessing step before applying supervised models.
- Helps in identifying new POS tags in evolving languages or domains.
- Can be combined with other models for semi-supervised training.

# Clustering for NER

**Application:**

- Group similar entities based on features like embeddings.
- Example: Clustering "London", "Paris", "New York" as Locations.
- Useful for domain-specific entity discovery.
- Enables unsupervised NER in the absence of labeled data.
- Can assist in expanding NER systems by identifying new entity types.
- Provides insights into entity relationships and hierarchies.

# Evaluation Metrics

**Accuracy:**

- Measures the proportion of correct predictions.
- Simple but can be misleading for imbalanced datasets.
- Often used as a baseline metric.

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

**Precision:**

- Measures the proportion of true positives out of all positive predictions.
- Important in tasks where false positives are costly.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

**Recall:**

- Measures the proportion of true positives out of all actual positives.
- Crucial in tasks where missing positives is costly.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

**F1-Score:**

- Harmonic mean of Precision and Recall.
- Balances both metrics, useful in imbalanced datasets.

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

# Classification Models in Other NLP Tasks

**Text Classification:**

- Sentiment analysis.
- Topic categorization.
- Spam detection.

**Speech Recognition:**

- Predicting phonemes.
- Mapping speech to text.

**Machine Translation:**

- Language model predictions.
- Word alignment.

**Information Retrieval:**

- Document ranking.
- Query classification.

# Conclusion

**Summary:**

- Classification models are crucial for NLP tasks like POS and NER.
- Naïve Bayes and Logistic Regression offer a balance between simplicity and performance.
- Clustering helps in discovering patterns in unlabeled data.
- Evaluation metrics provide insights into model effectiveness.
- These models are adaptable to other NLP tasks, demonstrating their versatility.

**Final Thoughts:**

- Understanding these models is foundational for advanced NLP applications.
- Feature engineering and model selection are key to success in NLP projects.