ARTIFICIAL INTELLIGENCE

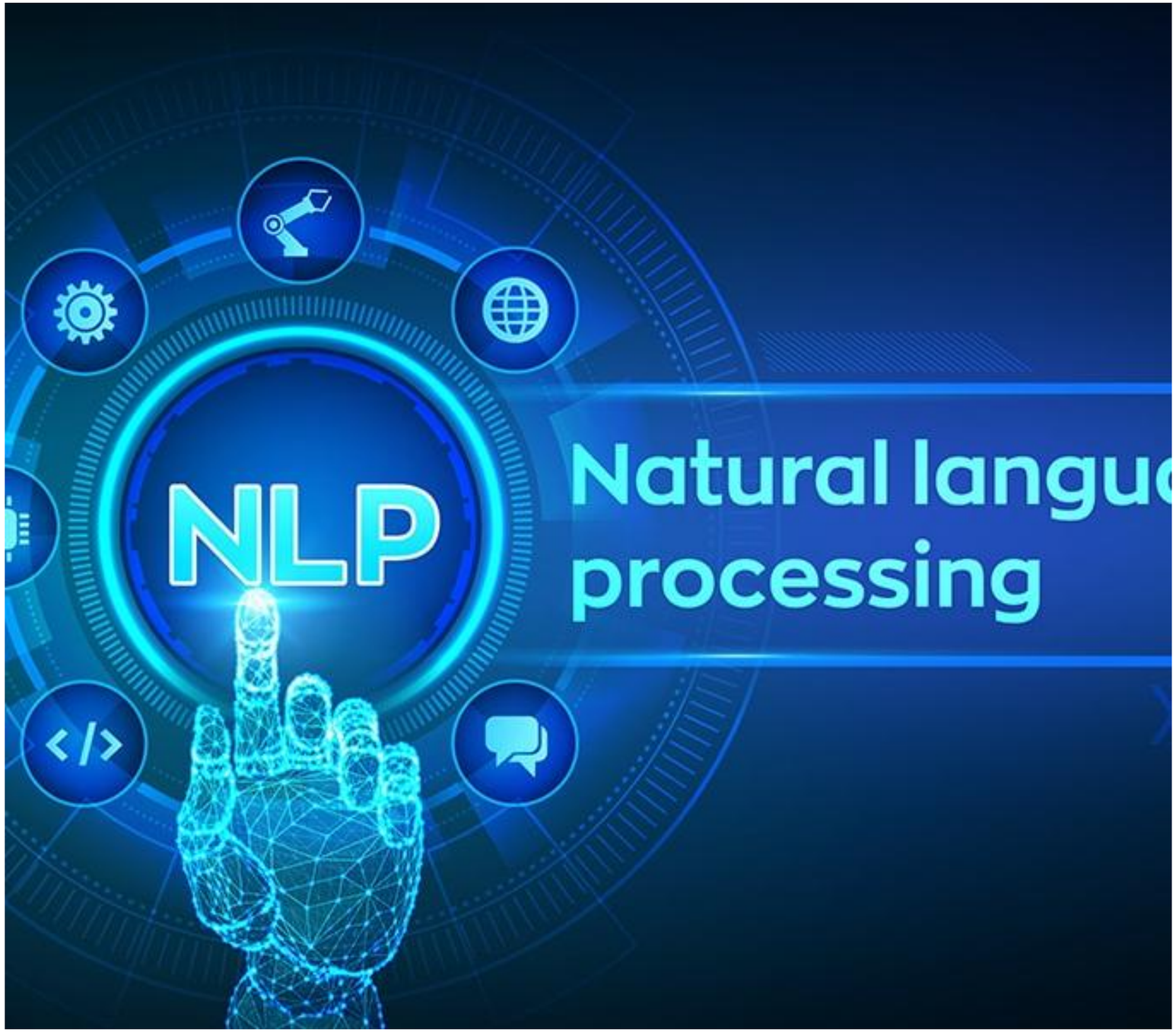
SUPERVISED AND UNSUPERVISED NLP

CHRISTIAN SY

SPECIFIC LEARNING OUTCOMES

By the end of this topic, students will be able to:

1. Explain the fundamental differences between supervised and unsupervised machine learning
2. Identify and classify real-world problems as supervised or unsupervised learning tasks
3. Apply basic supervised algorithms and unsupervised algorithms
4. Interpret and evaluate the results of supervised and unsupervised models using appropriate performance metrics



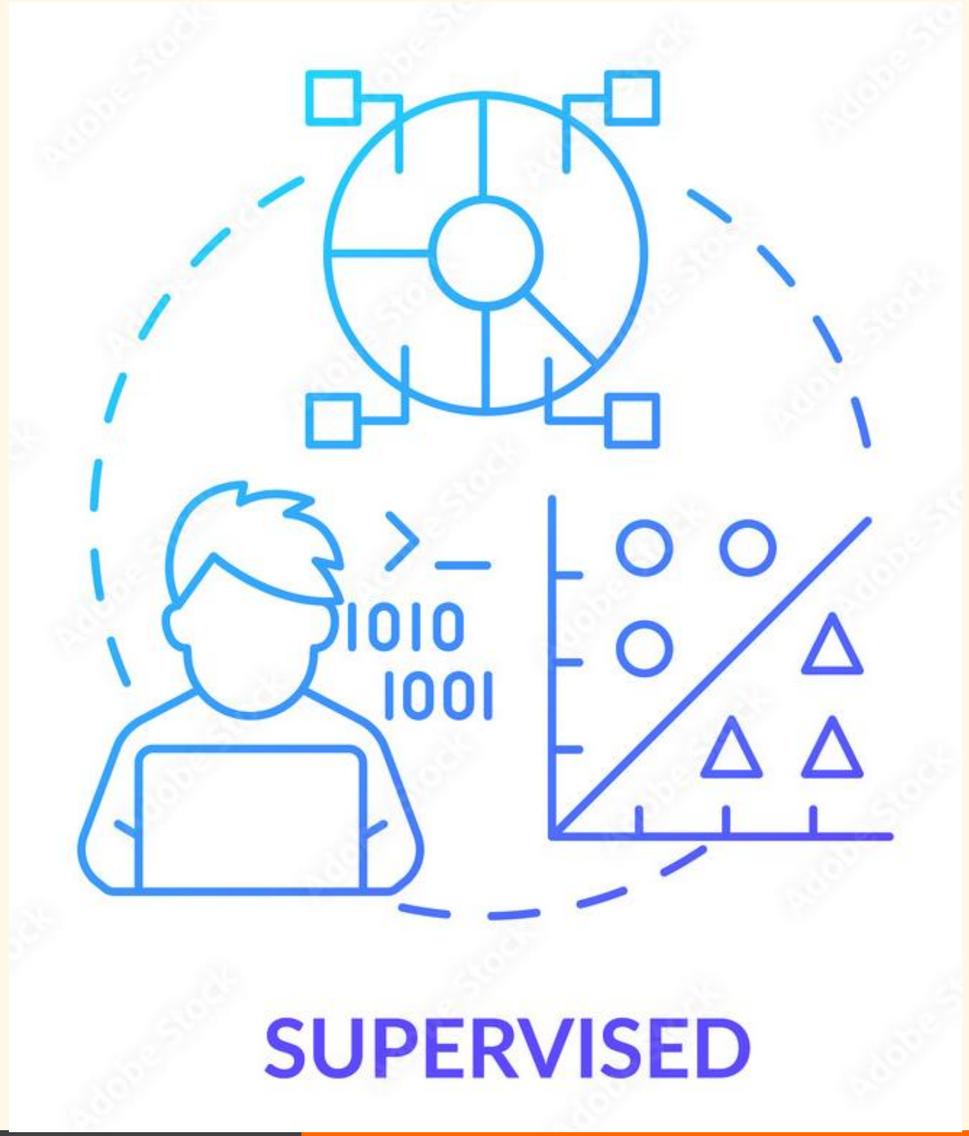
OUTLINE

1. SUPERVISED ML
2. UNSUPERVISED ML
3. TEXT CLASSIFICATIONS
4. NLP PROCESS
   - DATA COLLECTION
   - PRE-PROCESSING
   - FEATURE EXTRACTION
   - MODEL TRAINING
   - EVALUATION

SUPERVISED MACHINE LEARNING

SUPERVISED MACHINE LEARNING

It is a concept that involves teaching computers how to understand and process language by using labeled examples.



Supervised Machine Learning Pipeline

DATA GATHERING/TEXT PREPROCESSING

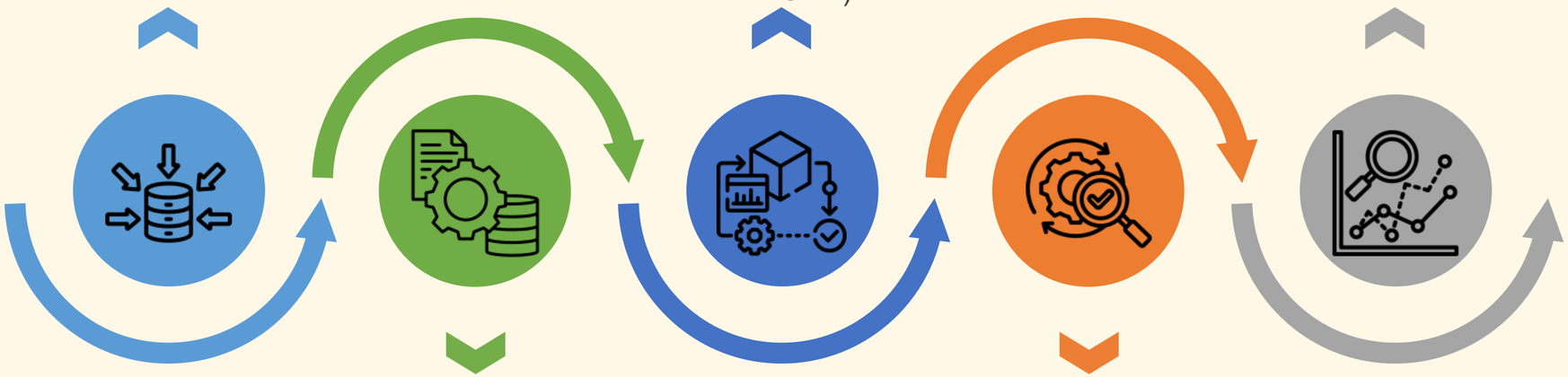
A systematic process of collecting, compiling, and acquiring information from various sources. Cleaning the data, tokenization, lowercasing

MODEL TRAINING

Choose a suitable ML model to learn the relationship between the input text and the corresponding labels. (logistic regression, linear SVM, decision trees, random forests, RNN, CNN).

PREDICTION

Use trained model to make predictions on new, unseen text data



FEATURE EXTRACTION

Convert the text data into numerical representations (vectors) that machine learning algorithms can process.(BoW, TF-IDF, Word2Vec, GloVe)

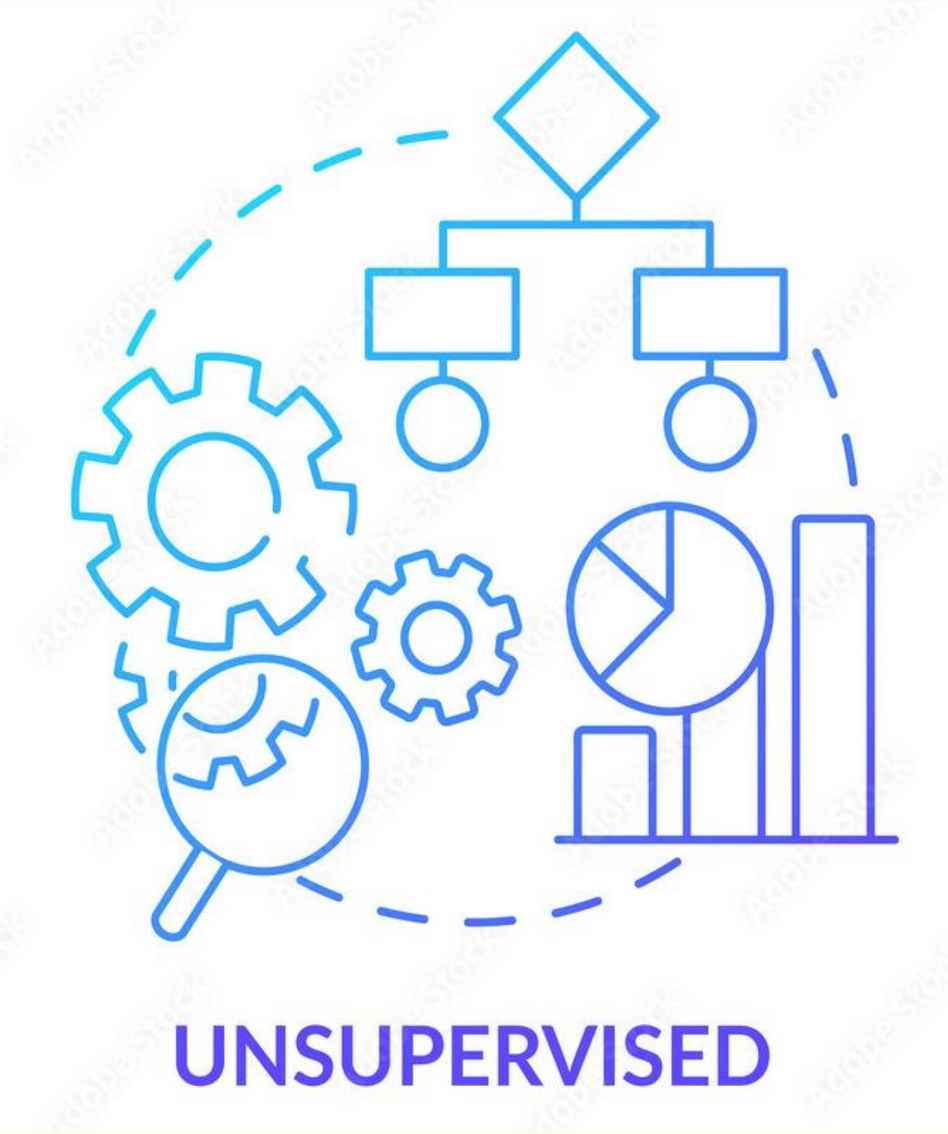
MODEL EVALUATION

Use a separate validation or test dataset to assess the performance. (accuracy, precision, recall, F1 score, confusion matrix).

UNSUPERVISED MACHINE LEARNING

UNSUPERVISED MACHINE LEARNING

It is a concept that involves teaching computers how to discover patterns and relationships in language without explicit guidance or labeled examples.



Unsupervised Machine Learning Pipeline

DATA GATHERING/ PREPROCESSING

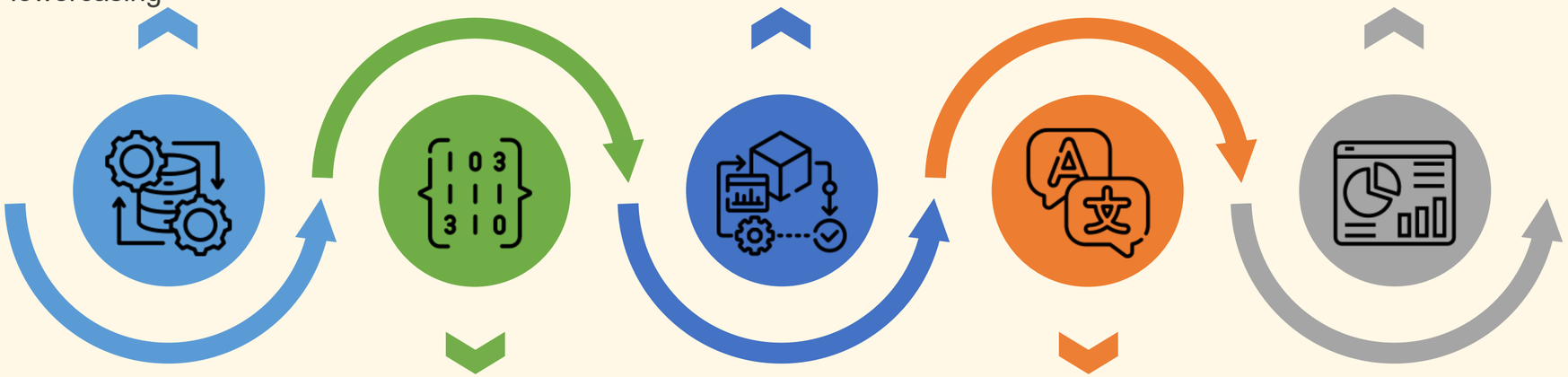
Build large corpus of text data from various sources, clean the text data by removing irrelevant information,tokenize the text into words or subwords, lowercasing

MODEL TRAINING

Clustering (K-means, hierarchical clustering, or DBSCAN)
Topic Modeling(LDA or Non-Negative Matrix Factorization (NMF).

VISUALIZATION

Techniques like word clouds, bar charts, or interactive visualizations can be used to visualize the clustered data or topics to gain insights and interpret the results.



FEATURE EXTRACTION

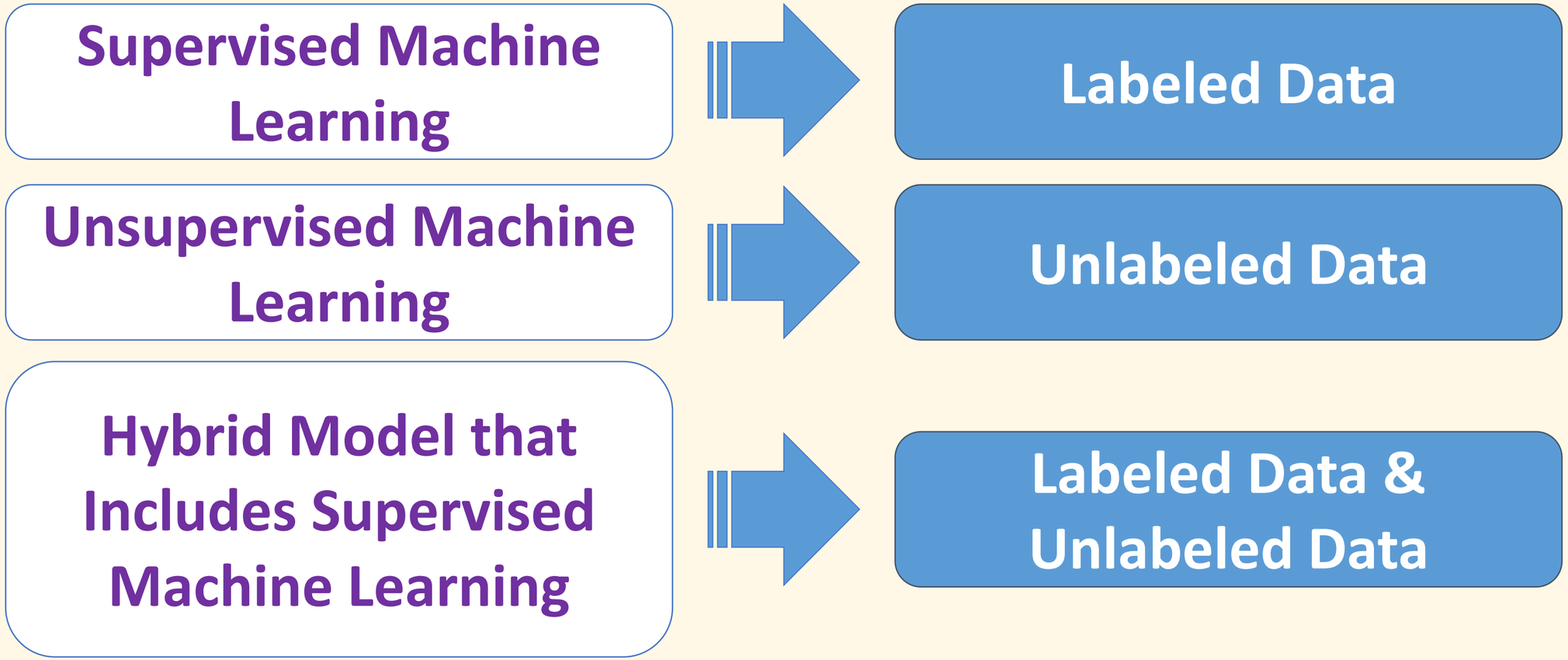
Convert the text data into numerical representations (vectors) that machine learning algorithms can process.(BoW, TF-IDF, Word2Vec, GloVe)

EVALUATION

Evaluate the quality of the clustering or topic modeling results using metrics like silhouette score for clustering or coherence score for topic modeling.

SUPERVISED & UNSUPERVISED MACHINE LEARNING

Data in Supervised Machine Learning vs Unsupervised Machine Learning



SUPERVISED & UNSUPERVISED MACHINE LEARNING

Supervised Learning

- text classification, sentiment analysis, named entity recognition, and machine translation

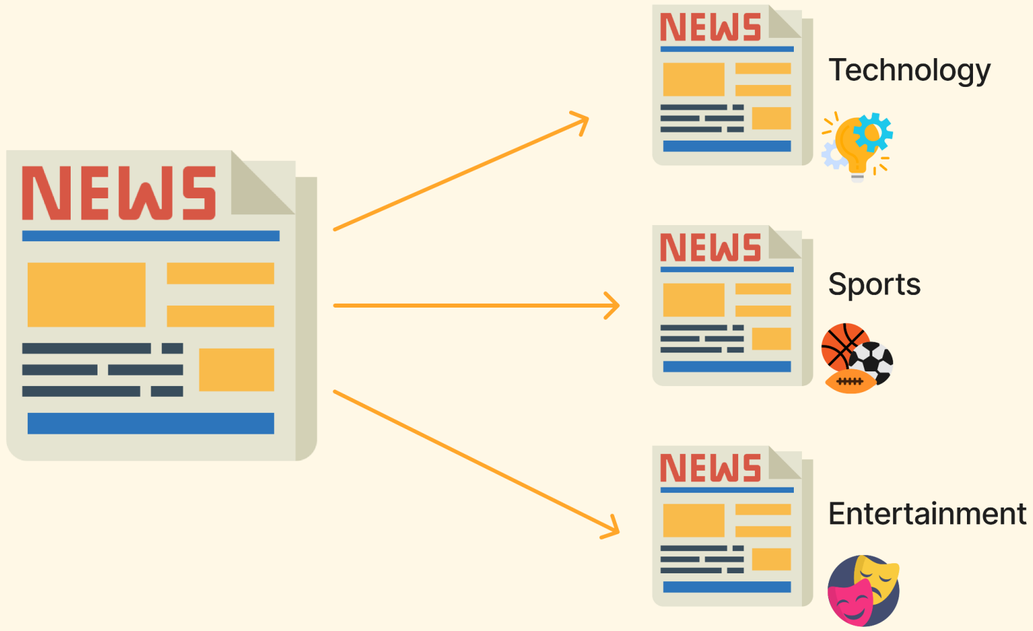
Unsupervised Learning

- text summarization, topic modeling/discovery, document clustering

Text Classification

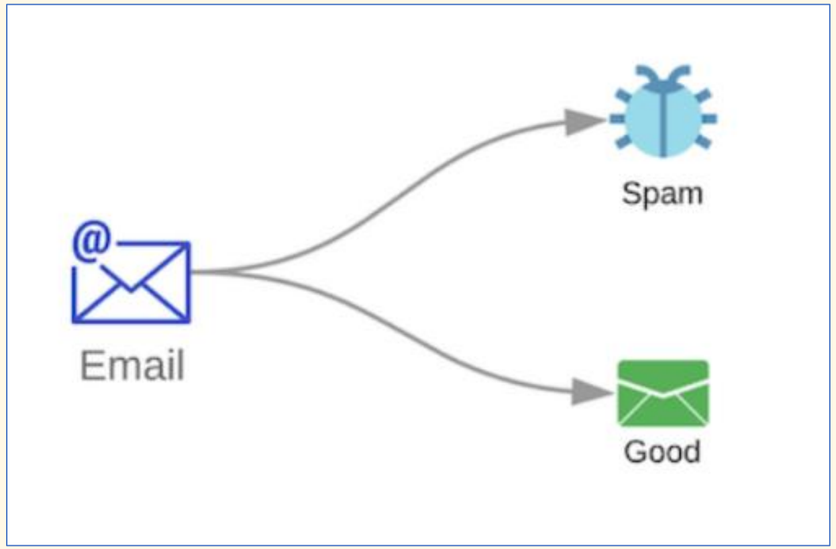
- **Text classification or Text Categorization** is the activity of **labeling natural language texts with relevant** *categories* from a predefined set.

- The algorithm is trained on the **labeled dataset** and gives the desired output(the pre-defined categories).

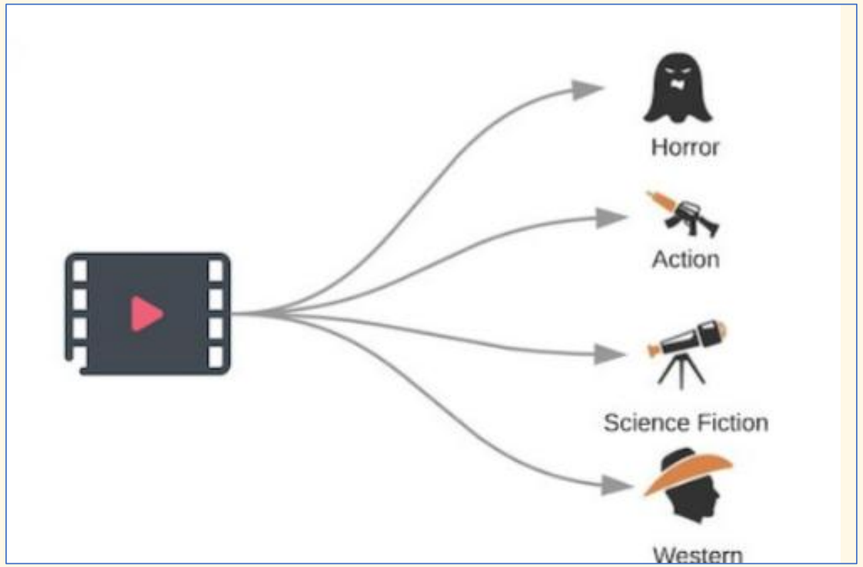


Types of Classification

Binary - classifying data into **two** mutually exclusive groups or categories



Multiclass and Multi-label - reviewing textual data and assigning one (single label) **or more** (multi) labels to the textual data.



https://abeyon.com/textclassification/

Text Classification

Text Classification can be applied in:

1. **Language Detection**
2. **Sentiment Analysis**
3. **Spam Filtering**
4. **Email Routing**

Text Classification - Language Detection

The task of classifying a text according to the **language** it is written in.

Example Scenario:

A social media platform like Twitter wants to automatically identify the language of every tweet so it can offer real-time translation or route it to appropriate moderators.

Example Data:

Text	Label
"Bonjour, comment ça va?"	French
"Hello, how are you?"	English
"Hola, ¿cómo estás?"	Spanish
"Kamusta ka?"	Filipino

Goal: Train a model to predict the correct **language label** given a new piece of text.

Text Classification – Sentiment Analysis

Classifying text according to the **emotional tone** or **attitude** expressed — typically *positive*, *negative*, or *neutral*.

Example Scenario:

An e-commerce site wants to analyze product reviews to understand customer satisfaction.

Example Data:

Review Text	Sentiment Label
"The laptop works perfectly and the battery lasts all day!"	Positive
"This phone is terrible — it crashes constantly."	Negative
"The delivery was okay, nothing special."	Neutral

Goal: Train a model to predict whether a given review expresses positive, negative, or neutral sentiment.

Text Classification – Spam Filtering

Classifying messages or emails as either **spam (unwanted)** or **ham (legitimate)**.

Example Scenario:

An email service provider wants to automatically move unwanted promotional or phishing emails to the Spam folder.

Example Data:

Email Text	Label
"Congratulations! You've won a free iPhone. Click here to claim!"	Spam
"Your Amazon order has been shipped."	Ham
"Earn $5000 a week from home with no experience!"	Spam
"Meeting rescheduled to 3 PM tomorrow."	Ham

Goal: Train a model to identify and filter out spam messages automatically.

Text Classification – Email Routing (Topic Classification)

Classifying emails into **predefined categories** or **departments** based on their content.

Example Scenario:

A company uses NLP to automatically route incoming customer emails to the appropriate department.

Example Data:

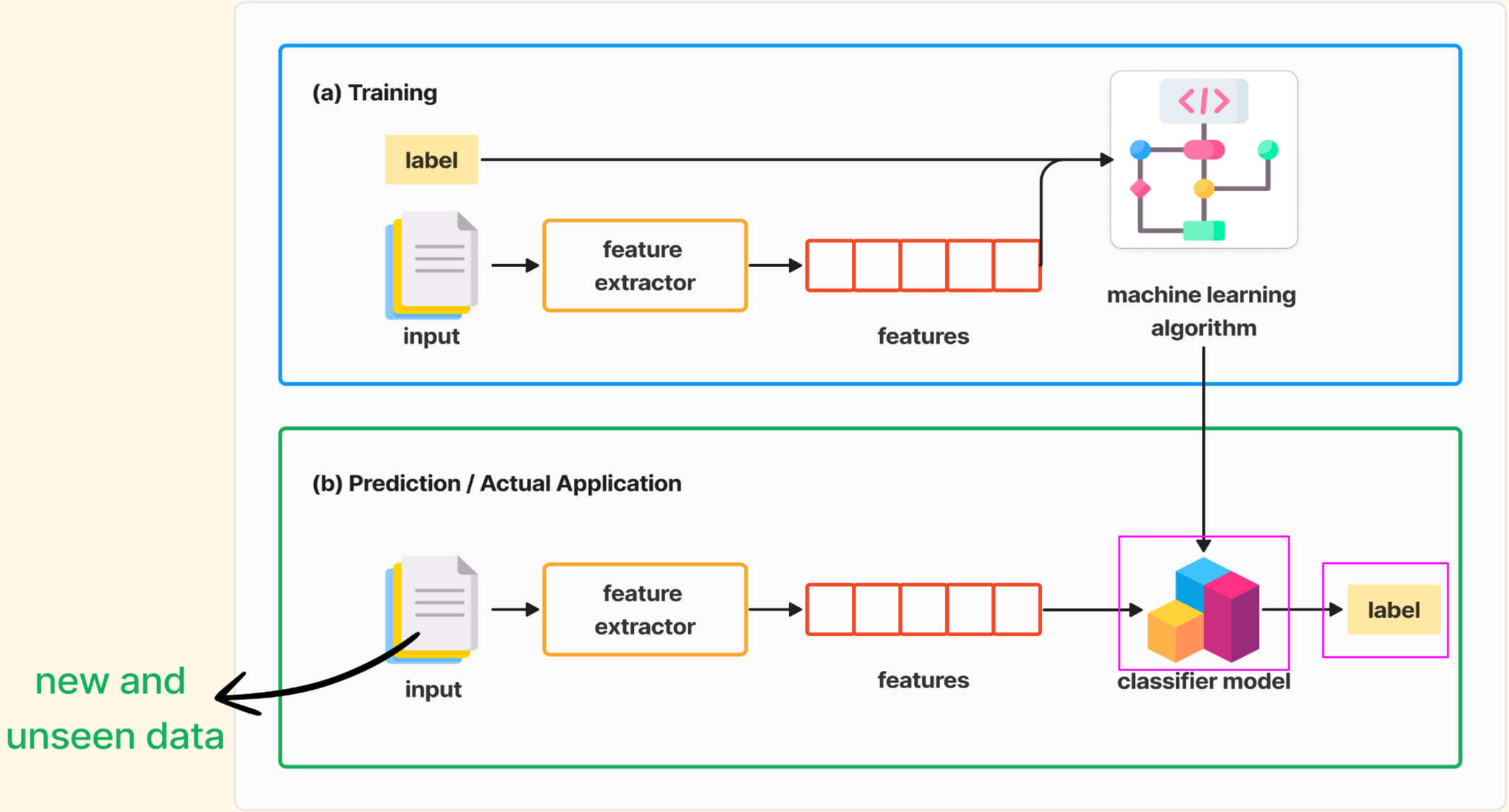
Email Text	Category
"I need to reset my password."	Technical Support
"I want to return a product I purchased."	Customer Service
"Can you provide a quote for 100 units?"	Sales
"I'd like to collaborate on marketing."	Business Development

Goal: Train a model to predict which department (class label) should receive the email.

How and where to start?

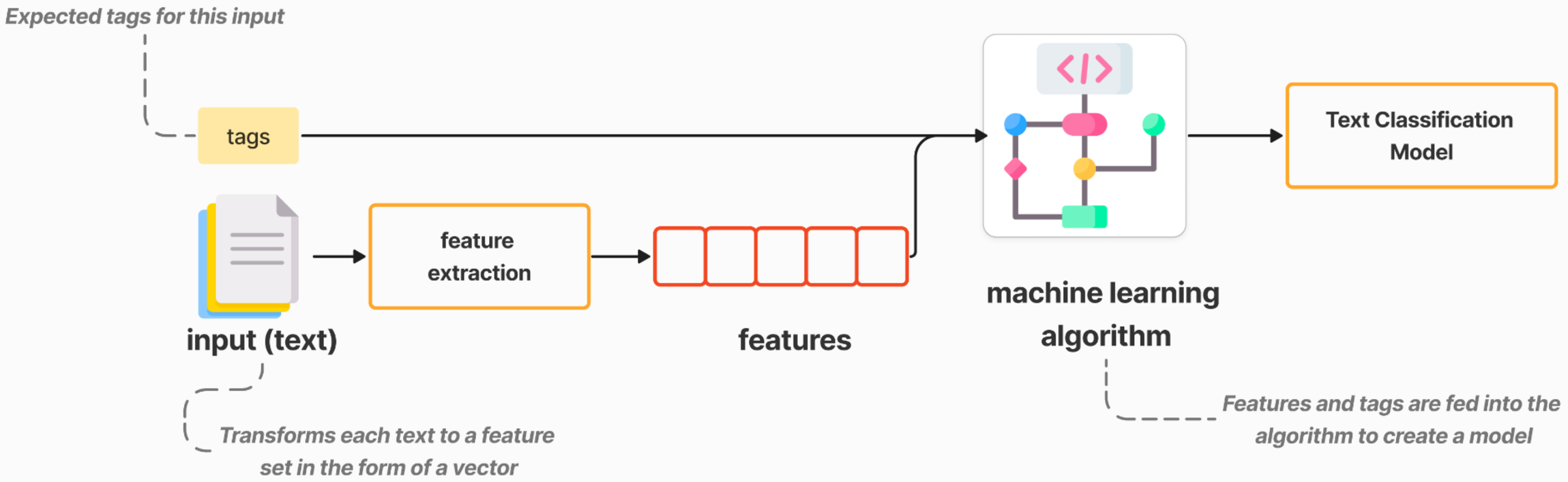
1. A dataset to provide examples for training the classifier. (Training)
2. A tool for generating and consuming the classifier. (Actual Application)

Machine learning-based system



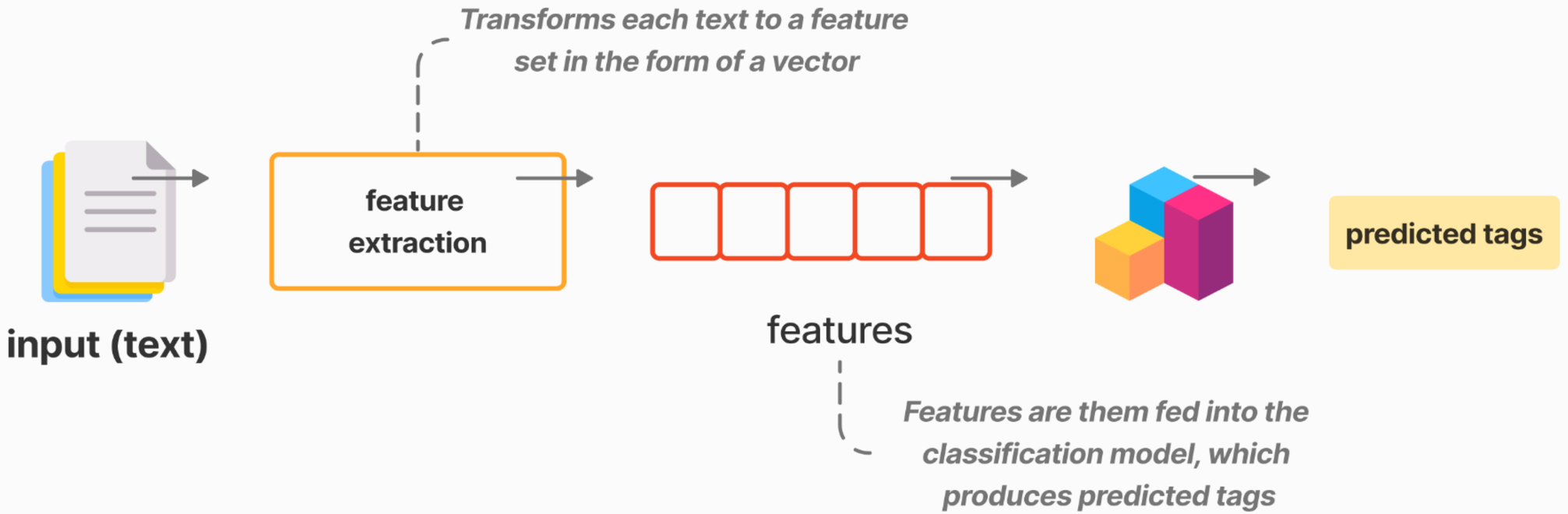
Machine learning-based system

TRAINING



Machine learning-based system

PREDICTION / ACTUAL APPLICATION



Text Classification Pipeline

DATA COLLECTION

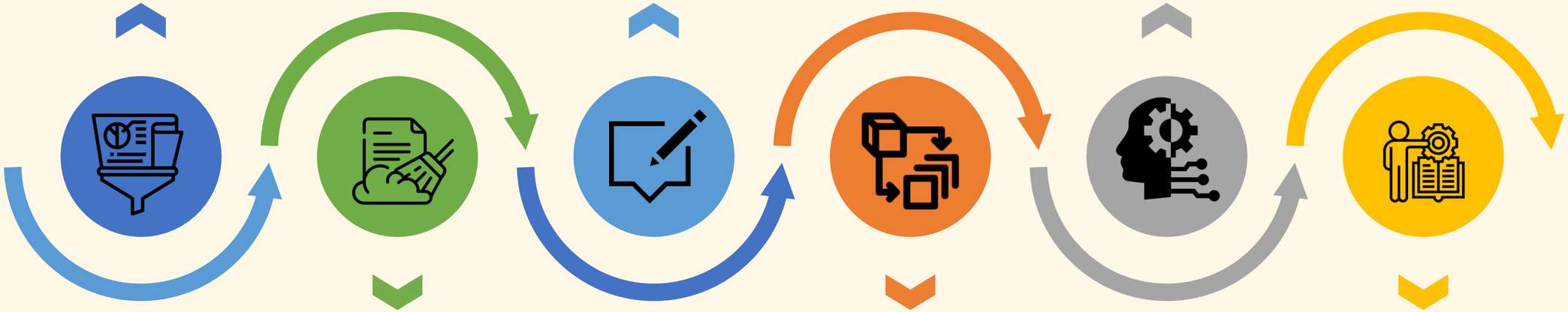
- Tweets, game chat; news articles, facebook posts, blogs

DATA ANNOTATION

POS tagging and assigning categories

MODEL TRAINING

Train the model using the training data



DATA PROCESSING

Word parsing and tokenization, stop words removal, lemmatization and Data Cleaning stemming

DATA SPLITTING

Divide the dataset into training and testing sets

FEATURE EXTRACTION

Generate word vectors

PERFORMANCE EVALUATION

Analyze results of experiments; Accuracy; Precision; Recall; F-score

Text Data


Text Classification
Tokenization
Word2Vec
RMDL
HDLTex
Deep Learning
Naive Bayes Classifier
FastText
A Survey
Logistic Regression
Random Forest
K-nearest Neighbor
Support Vector Machine (SVM)
Recurrent Neural Networks (RNN)
Conditional Random Field (CRF)
Contextualized Word Representations
Term Frequency-Inverse Document Frequency
Recurrent Convolutional Neural Networks (RCNN)
Deep Belief Network (DBN)
Hierarchical Deep Learning for Text (HDLTex)
Convolutional Neural Networks (CNN)

Where to get data?

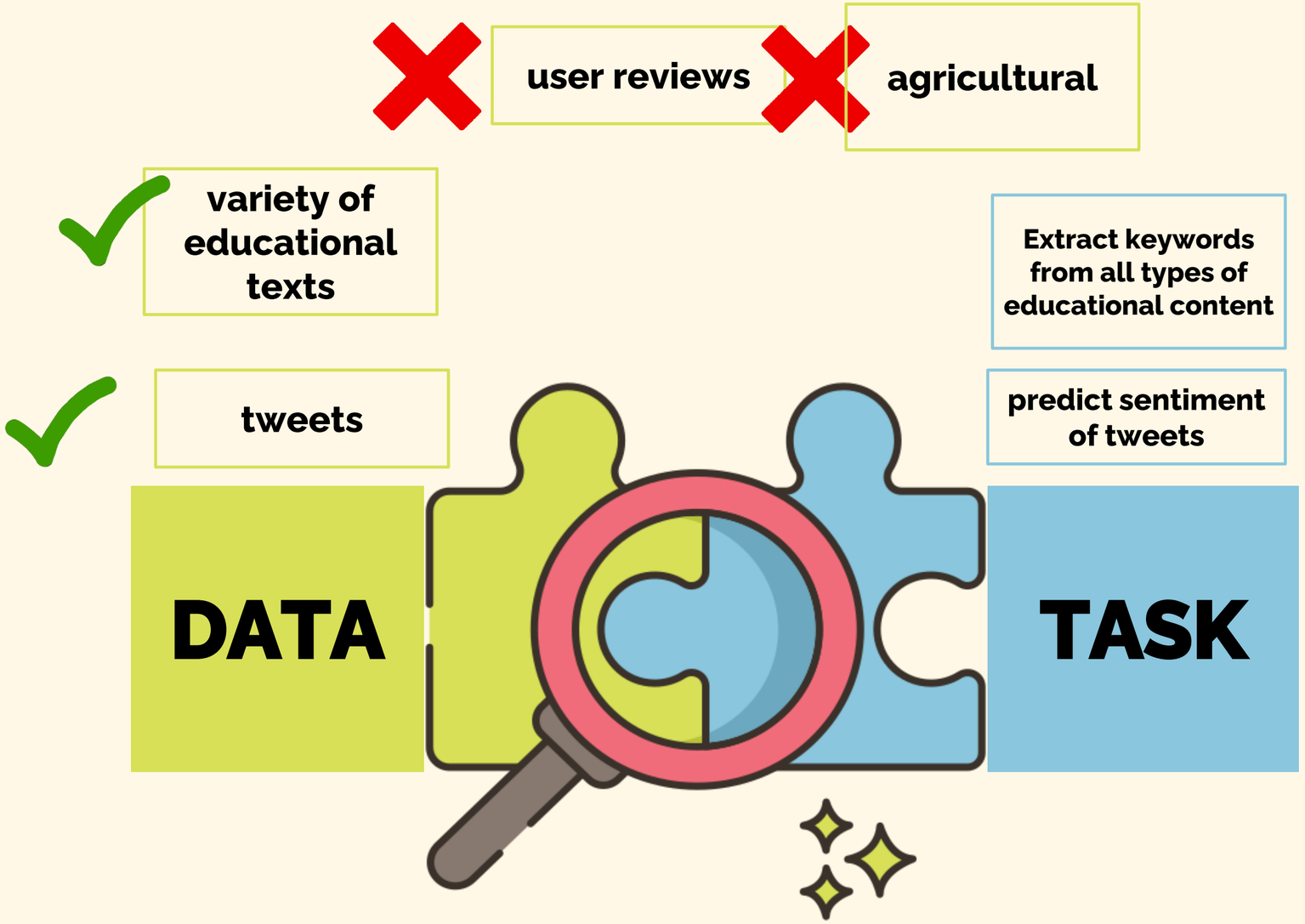
- **Internal Data**
  - Generated from the apps and tools
    - Customer Relationship Management systems
    - chat apps
    - help desk software
    - survey tools
- **External Data** from the web
  - Web scraping or public datasets
  - Kaggle, Hugging Face Datasets

Text Data

Use **QUALITY** training data

- ❑ **Compatible with the task**
- ❑ **Fairly balanced**
- ❑ **Representative**

Data must FIT IN the task



Text Data

Data should be fairly balanced between classes

Student Concerns Data:

50%

scholarship

40%

financial assistance

5%

counselling

Possible high accuracy, but useless classifer.

5%

Academic appeals

Techniques for Imbalanced Datasets

- Collect **more** data
- **Resample** the dataset (over-sampling and under-sampling)
- Generate **synthetic** samples (e.g. Synthetic Minority Over-sampling Technique (SMOTE))
- Try **different algorithms**

Data that is representative

Pre-processing

transforms the data into a format that is more easily and effectively processed

1. Data cleaning
2. Tokenization
3. Lowercasing
4. Stop words removal
5. Lemmatization
6. Stemming
7. POS tagging

Pre-processing – Data Cleaning

Remove unwanted characters, punctuation, numbers, or extra spaces that do not contribute to meaning.

Process:

✓ Remove punctuation and numbers
✓ Strip whitespace
✓ Keep only relevant characters

Input: "The cats are running faster than the dogs!"
Output: "The cats are running faster than the dogs"

Clean text is easier to process and reduces noise in the dataset.

Pre-processing – Tokenization

Split text into smaller units — typically **words** or **tokens**.

Input: "The cats are running faster than the dogs"
Output: ['The', 'cats', 'are', 'running', 'faster', 'than', 'the', 'dogs']

Tokens are the basic units used in most NLP algorithms.

Pre-processing – Lowercasing

Convert all text to lowercase so the model treats words like “The” and “the” as the same.

Input: ['The', 'cats', 'are', 'running', 'faster', 'than', 'the', 'dogs']
Output: ['the', 'cats', 'are', 'running', 'faster', 'than', 'the', 'dogs']

Avoids duplication of word features due to case differences.

Pre-processing – Stop Words Removal

Remove common words (like *the, is, are, of, in*) that don’t add much meaning to the sentence.

Input: ['the', 'cats', 'are', 'running', 'faster', 'than', 'the', 'dogs']
Output: ['cats', 'running', 'faster', 'dogs']

Reduces dimensionality and focuses on more meaningful words.

Stemming vs. Lemmatization

- Words are stemmed or diminished to their root/base form.

word	stemming	lemmatization
information	inform	information
having	hav	have
am	am	be
computers	comput	computer

Stemming vs Lemmatization

- Stems the word but makes sure that it does not lose its meaning. Lemmatization has a pre-defined dictionary that stores the context of words and checks the word in the dictionary while diminishing

Pre-processing – Lemmatization

Convert words to their **base or dictionary form (lemma)** — using grammar and vocabulary context.

Input: ['cats', 'running', 'faster', 'dogs']
Output: ['cat', 'run', 'fast', 'dog']

Lemmatization uses morphological analysis (e.g., “running” → “run”).

Pre-processing – Stemming

Cut words down to their **root form** (often by removing suffixes).
It’s a more *mechanical* process than lemmatization.

Input: ['cats', 'running', 'faster', 'dogs']
Output: ['cat', 'run', 'fast', 'dog']

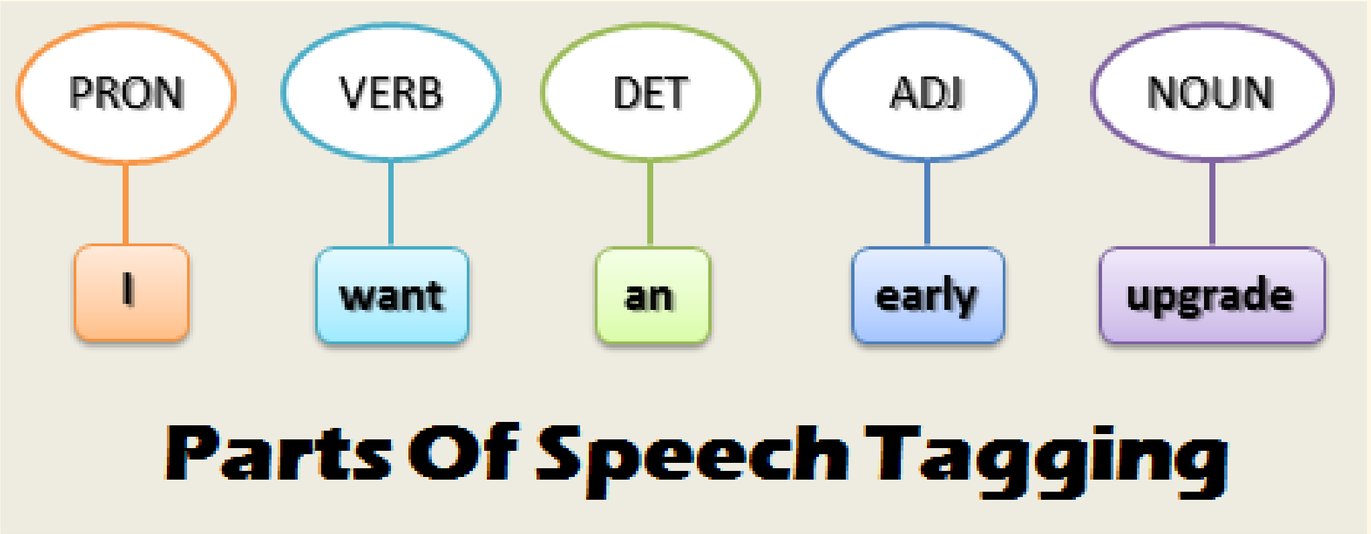
Faster but less accurate than lemmatization.

Pre-processing – Stemming and Lemmatization

Word	Lemmatized	Stemmed
running	run	run
better	good	better
studies	study	studi
wolves	wolf	wolv
geese	goose	gees
mice	mouse	mic
corpora	corpus	corpora
driving	drive	drive
nationality	nationality	nation
organizational	organizational	organ

Part of Speech (POS) Tagging

A task of labelling each word in a sentence with its appropriate part of speech. It converts a sentence into a list of words with their tags. (word, tag).



https://thinkinfi.com/extract-custom-keywords-using-nltk-pos-tagger-in-python/

Pre-processing – POS

Assign grammatical tags (noun, verb, adjective, etc.) to each word.

Input: ['the', 'cats', 'are', 'running', 'faster', 'than', 'the', 'dogs']

```
('the', 'DT'),      # Determiner
('cats', 'NNS'),    # Plural noun
('are', 'VBP'),     # Verb (present tense)
('running', 'VBG'), # Verb (gerund)
('faster', 'RBR'),  # Comparative adverb
('than', 'IN'),     # Preposition
('the', 'DT'),
('dogs', 'NNS')     # Plural noun
```

Useful for syntactic analysis, entity recognition, or semantic role labeling.

Feature Extraction

Feature extraction step means to extract and produce feature representations that are appropriate for a type of NLP task.

- Bag of Words - text is represented as the bag of its words, disregarding grammar and even word order but keeping multiplicity.
- N-grams - built by counting how often word sequences occur in corpus text and then estimating the probabilities
- TF-IDF stands for Term Frequency Inverse Document Frequency of records. It can be defined as the calculation of how relevant a word in a series or corpus is to a text.
- Word2vec - group of related models that are used to produce word embeddings

Bag-of-words Model

row row row your boat
gently down the stream
merrily merrily merrily
life is but a dream

word	count
row	3
your	1
boat	1
gently	1
down	1
the	1
...	...

N-Gram

“Natural language processing is fun.”

1. Unigrams (1-gram)

Natural
language
processing
is
fun

2. Bigrams (2-gram)

Natural language
language processing
processing is
is fun

3. Trigrams (3-gram)

Natural language processing
language processing is
processing is fun

Why N-grams matter (simple explanation)

- They help computers understand context by looking at word patterns.
- Used in text prediction, speech recognition, machine translation, and text classification.
- Example: If the bigram *“thank you”* appears frequently, a model may predict *“very much”* next.

TF-IDF

Term Frequency - Inverse Document Frequency (TF-IDF) is a statistical measure used to determine the mathematical significance of words in documents

- **TF** tells you:
"How often does the word appear in this document?"
- **IDF** tells you:
"How rare or unique is this word across all documents?"
- **TF-IDF** highlights words that are:
✓ Important inside the document
✓ Not too common across all documents

Examples of words with **low TF-IDF**:
- "and", "are", "the" (appear everywhere → not important)

Examples of words with **high TF-IDF**:
- "cats", "dogs", "loyal" (more unique)

TFIDF

For a term i in document j:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

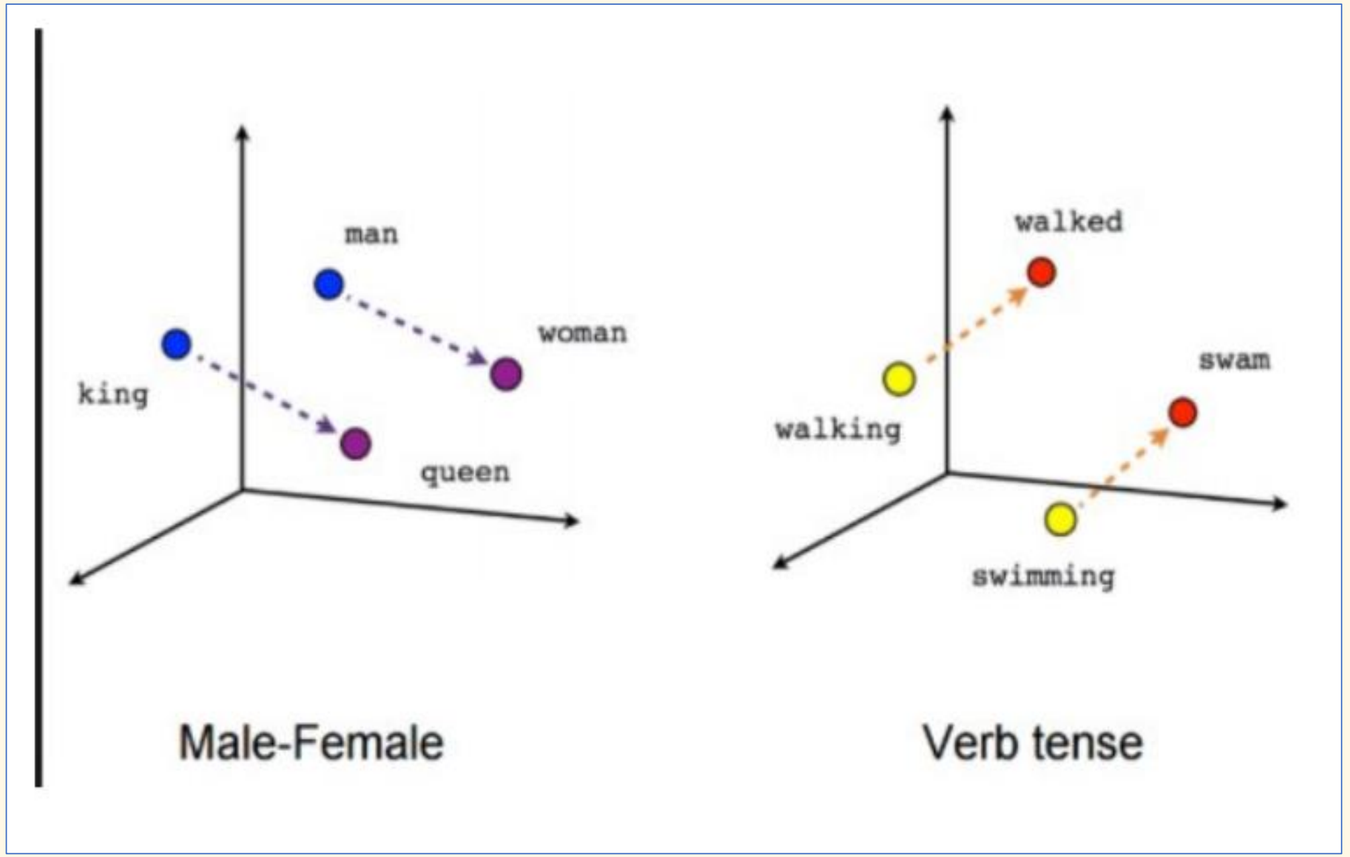
TF-IDF

Suppose you have 3 documents:
Document 1 (D1): *“Cats are cute and playful.”*
Document 2 (D2): *“Dogs are playful and loyal.”*
Document 3 (D3): *“Cats and dogs can be playful.”*

- **TF** tells you:
  “How often does the word appear in this document?”
- **IDF** tells you:
  “How rare or unique is this word across all documents?”
- **TF-IDF** highlights words that are:
  ✓ Important inside the document
  ✓ Not too common across all documents
- Examples of words with **low TF-IDF**:
- “and”, “are”, “the” (appear everywhere → not important)
- Examples of words with **high TF-IDF**:
- “cats”, “dogs”, “loyal” (more unique)

Word2vec

- **Word2vec is a group of related models that are used to produce word embeddings.**
- **Words that share common contexts in the corpus are located in close proximity to one another in the space**



Word2Vec will place these in vector space such that:
•king is close to queen
•man is close to woman
•king – man ≈ queen – woman

It captures:
•**Similarity** (“big” ~ “large”)
•**Analogies** (“king – man + woman = queen”)

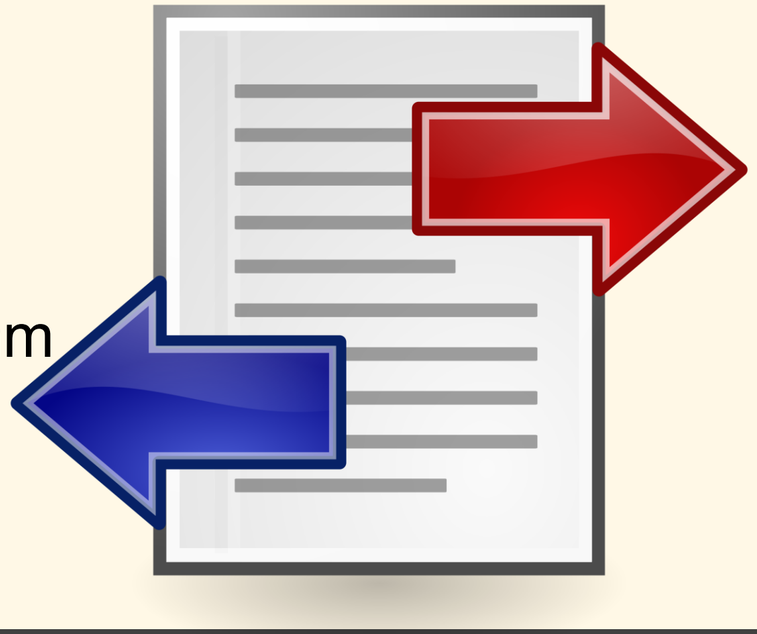
Data Splitting

Divide the dataset into **training** and **testing** sets. The training set is used to train the model, and the testing set is used to evaluate its performance.

training dataset

80%

train the model using ML algorithm

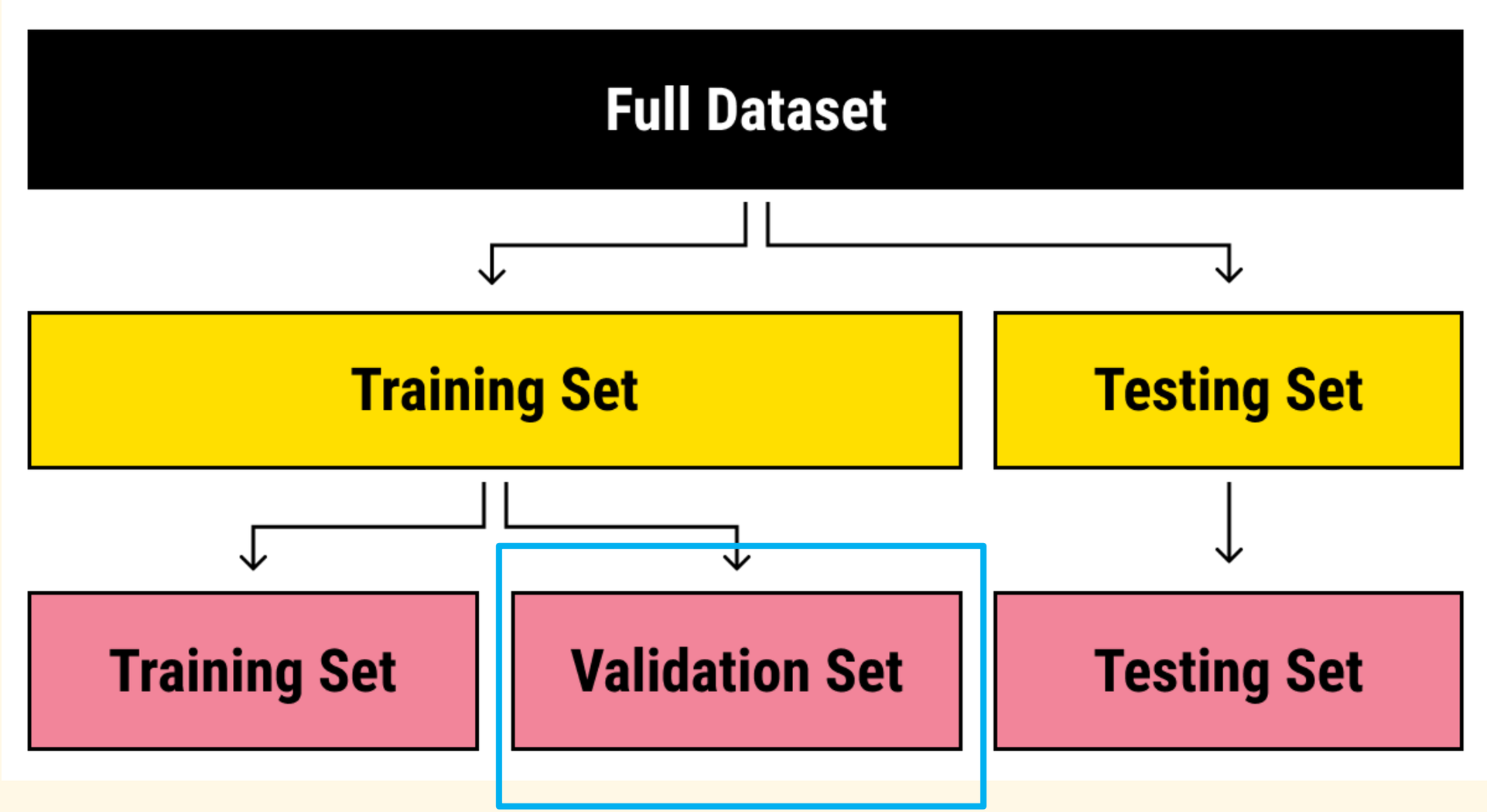


testing dataset

Evaluate the model using the **labeled** and **unseen** data

20%

Data Splitting



Training the Model

- **Traditional ML Algorithms**
  - SVM - Support Vector Machine
  - Naïve Bayes
  - K-nearest Algorithm
  - Logistic Regression

- **Deep learning**
  - Neural Networks

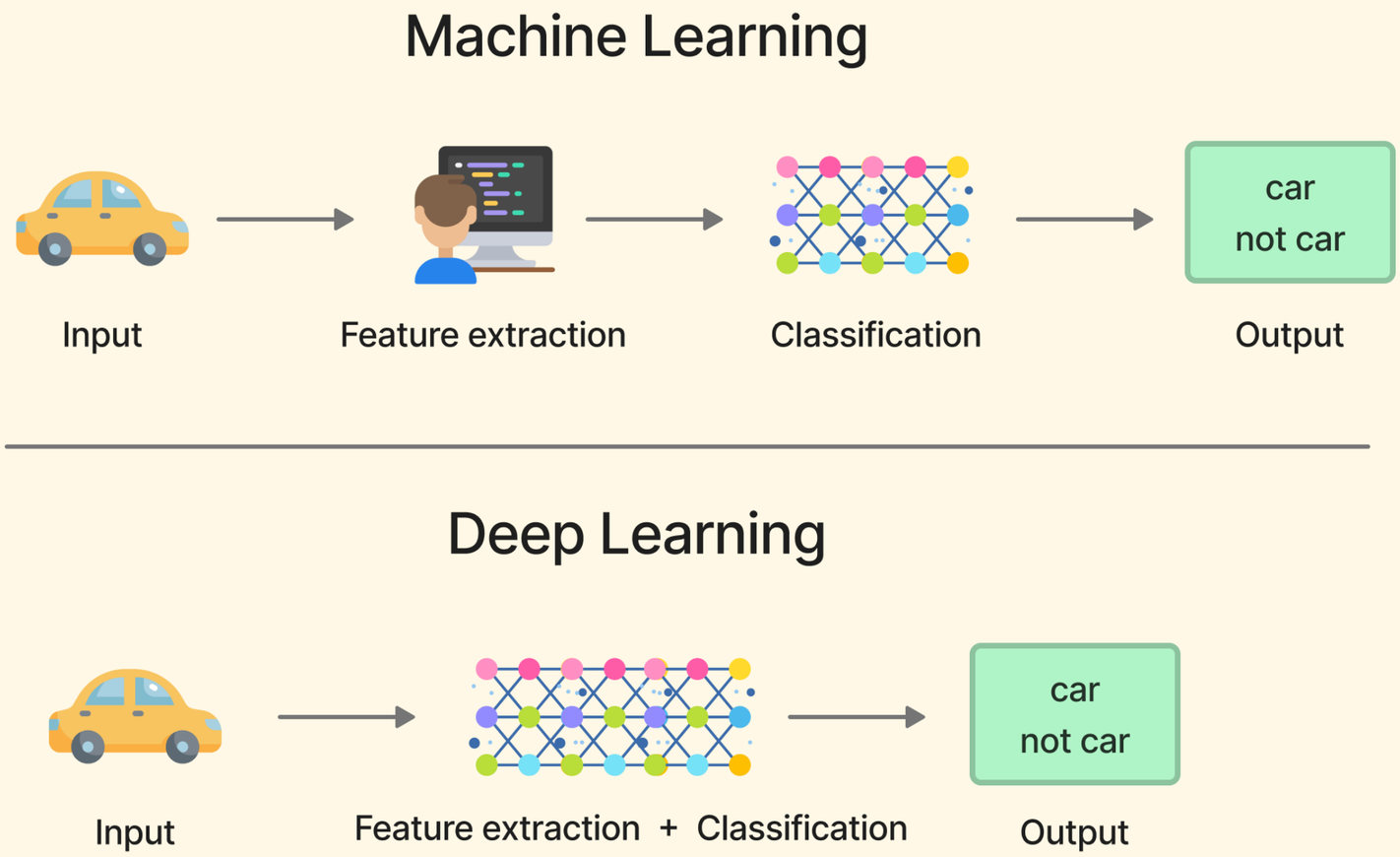
- **Transfer learning**
  - Pre-trained language models (BERT, RoBERTa, Electra)

Traditional ML + NLP

- intersection of **computer science and statistics** where algorithms are used to perform a specific task without being explicitly programmed
- recognizes **patterns** in the data and **make predictions** once new data arrives.

Deep Learning + NLP

- Uses neural network-based methods
- Introduced the concept of **contextual understanding**.
- Automatic feature engineering
- Requires a massive amount of data, but requires little human intervention



Transfer Learning + NLP

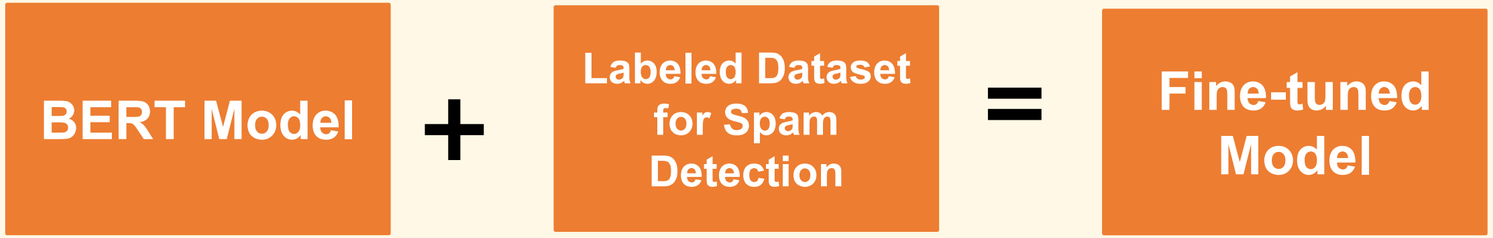
- Enables the **transfer of knowledge** from one machine to another.
- Needs **less training data**
- use of **pre-trained models**
- BERT and GPT-2

Transfer Learning + NLP

Transfer Learning with Pre-trained Model

General knowledge

Domain-specific knowledge

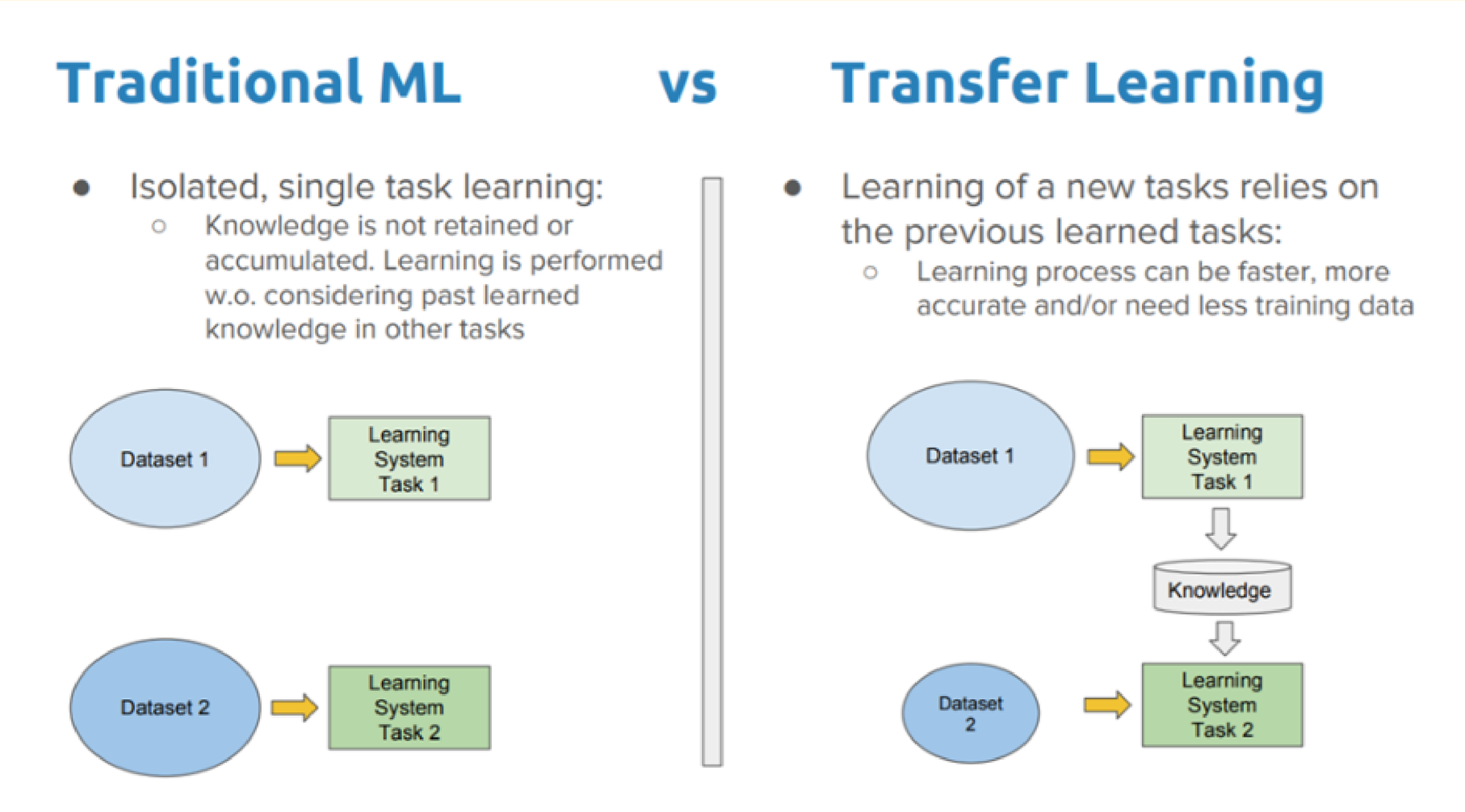


Pre-trained model

Finetuning process using task-specific dataset

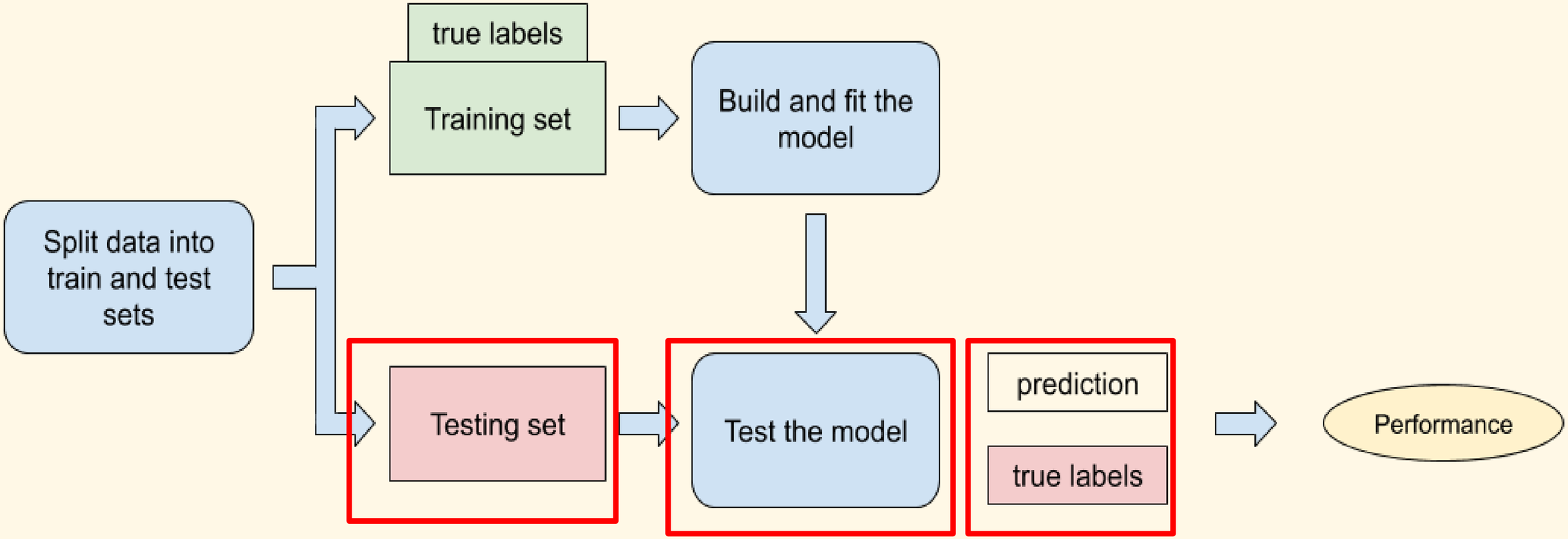
Final model

Deep Learning + NLP



A comparison of traditional learning and transfer learning (Image credits: Dipanjar Sarkar)

Performance Evaluation



Performance Evaluation

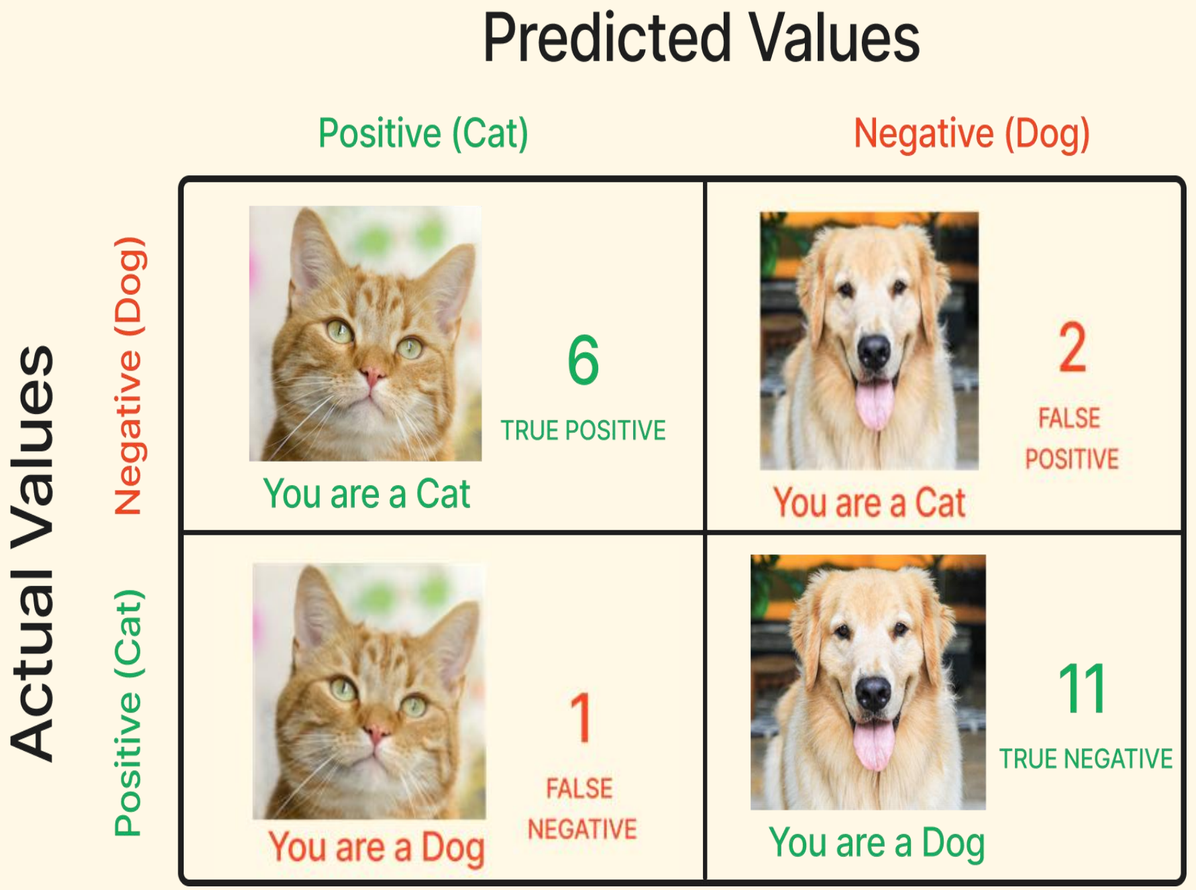
This is the process where we use the trained model to make predictions on previously unseen, labelled data

- Accuracy
- Confusion Matrix
- Precision
- Recall
- F1 Score

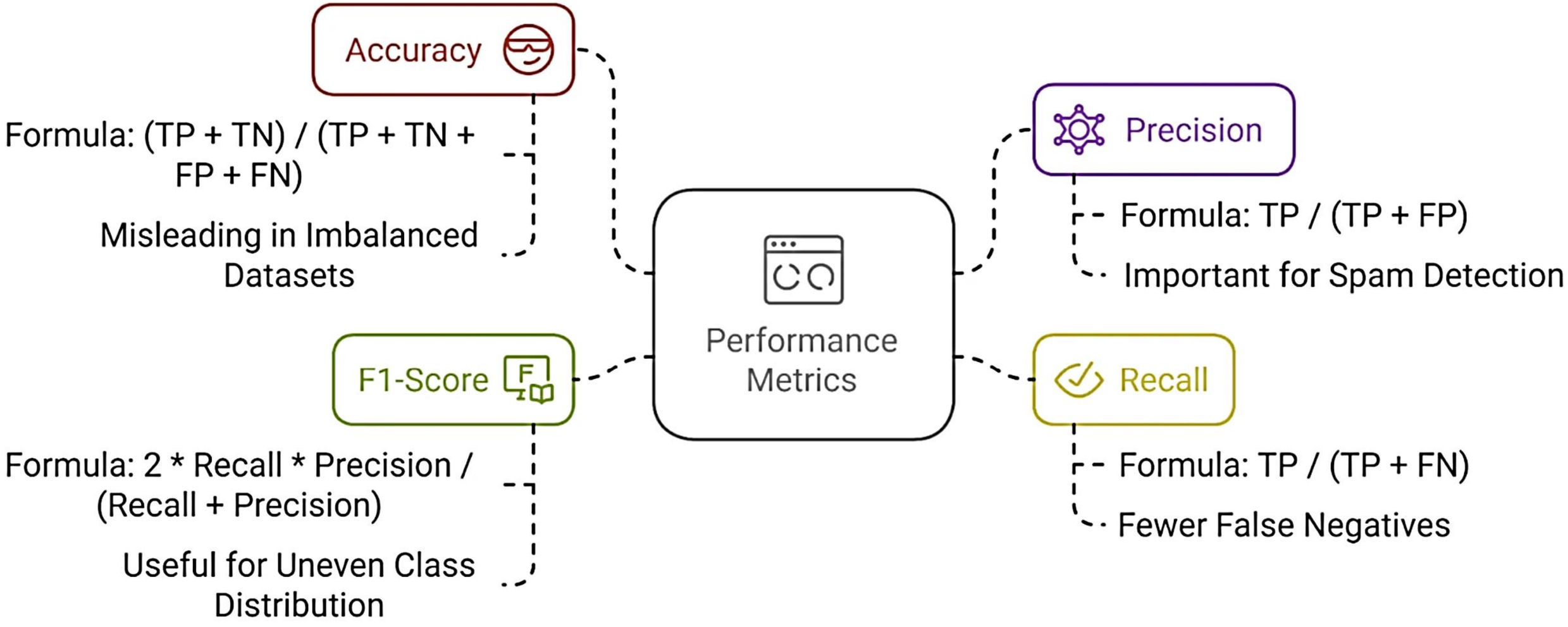
https://towardsdatascience.com/8-metrics-to-measure-classification-performance-984d9d7fd7aa

Performance Evaluation | Confusion Matrix

Actual Value \ Predicted Value		
Actual Value	**True Positive (TP)** The actual value is positive, and the model correctly predicts it as positive	**False Positive (FP)** The actual value is negative, but the model predicts it as positive
True Negative	**False Negative (FN)** The actual value is positive, but the model predicts it as negative (Type II error)	**True Negative (TN)** The actual value is negative, and the model correctly predicts it as negative



Performance Evaluation | Confusion Matrix



https://www.mygreatlearning.com/blog/confusion-matrix-an-overview-with-python-and-r/

Performance Evaluation | Accuracy

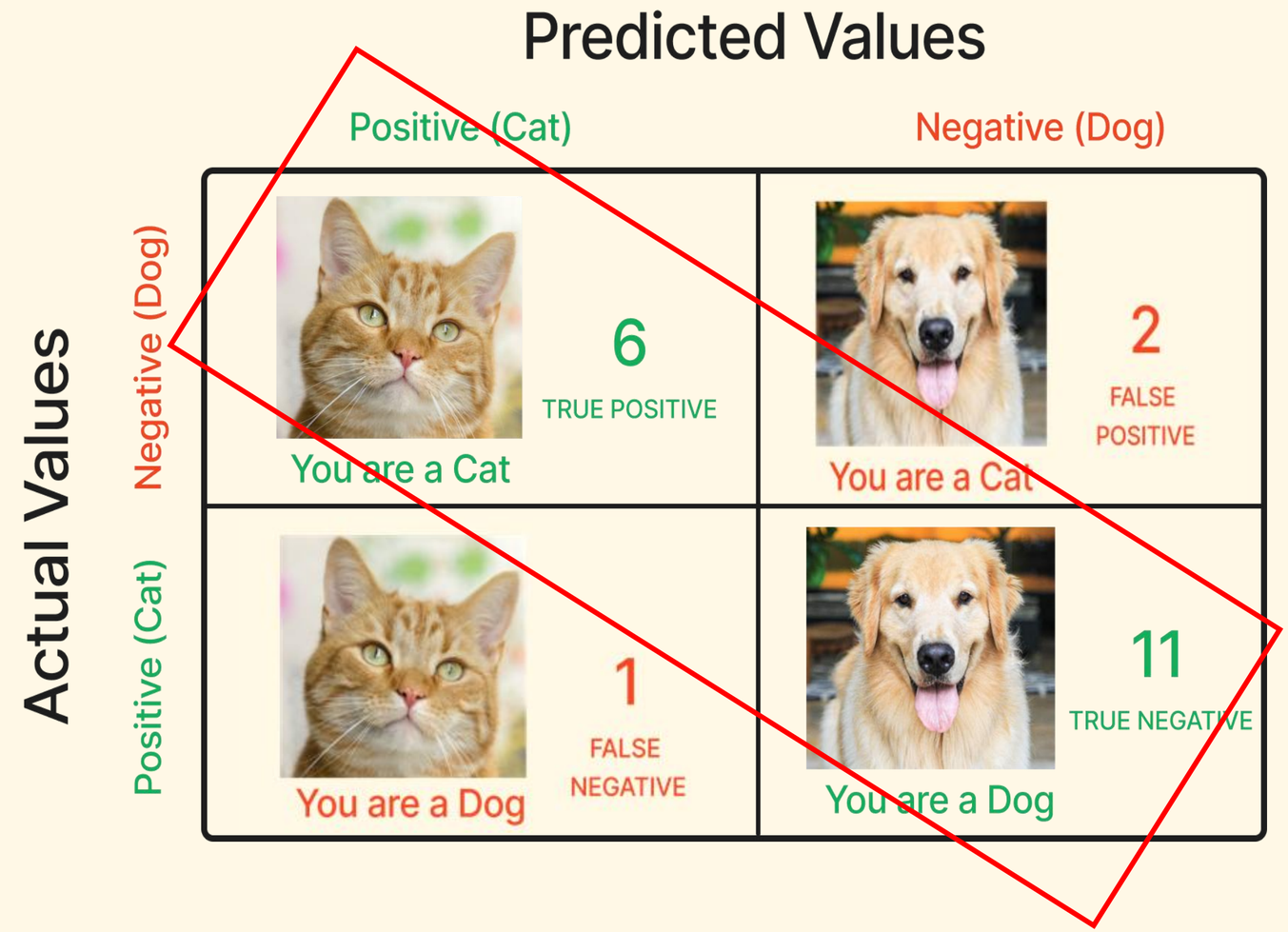
Formula: Accuracy = (TP + TN) / (TP + TN + FP + FN)

- ✓ Accuracy measures how often the model makes correct predictions. However, in imbalanced datasets, high accuracy might be misleading.
- ✓ Valid choice of evaluation for classification problems which are well balanced and not skewed or there is no class imbalance.

Accuracy = (6 + 11) / 6 + 11 + 2 + 1)

= 17 / 20

= 0.85 or 85%



Performance Evaluation | Precision

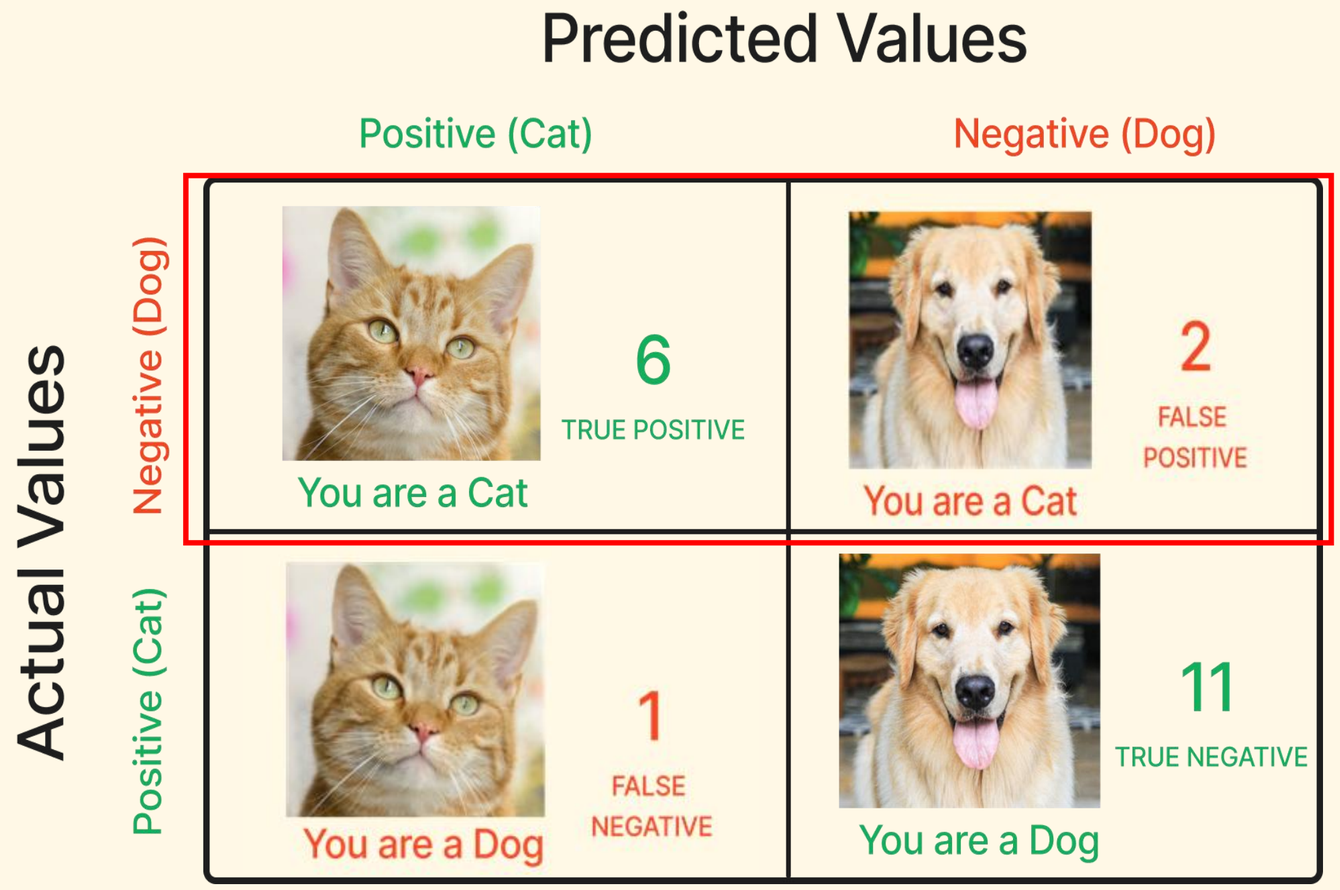
Formula: Precision = TP / (TP + FP)

- ✓ Precision measures how many predicted positive values are actually correct. It is crucial in applications like spam detection.
- ✓ Measure of correctness that is achieved in true prediction
- ✓ It tells us how many predictions are actually positive out of all the total positive predicted
- ✓ Also known as ***specificity***

Precision = 6 / (6 + 2)

= 6 / 8

= 0.75 or 75%



Performance Evaluation | Recall (Sensitivity)

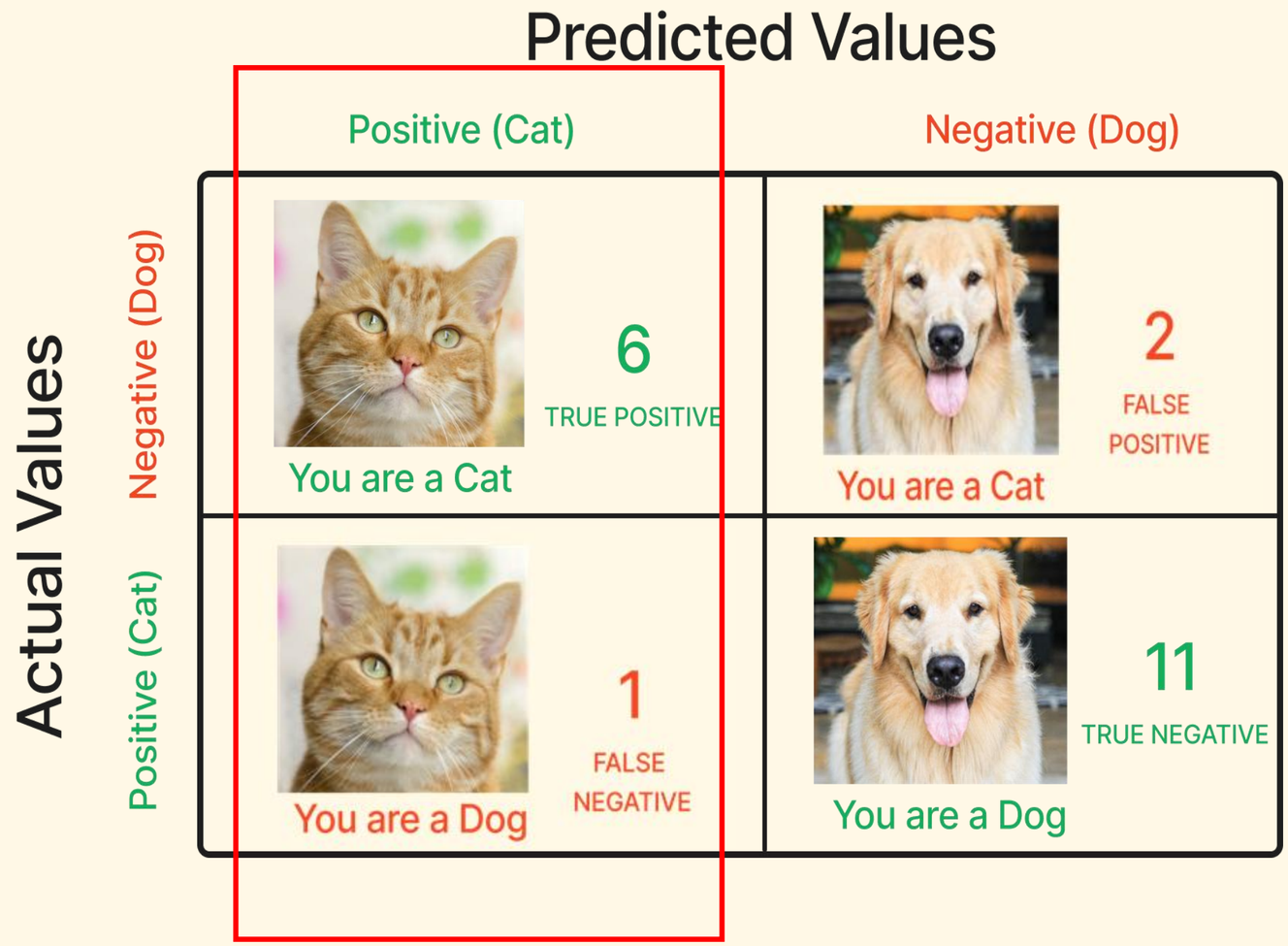
Formula: Recall = TP / (TP + FN)

- ✓ Recall indicates how well the model identifies positive instances. A high recall means fewer false negatives.
- ✓ Measure of actual observations which are predicted correctly, i.e. how many observations of positive class are actually predicted as positive.
- ✓ It is also known as ***sensitivity***

Precision = 6 / (6 + 1)

= 6 / 7

= 0.857 or 86%



Performance Evaluation | F1 Score

a number between 0 and 1 and is the harmonic mean of precision and recall.

F-1 Score = 2 * Recall * Precision / Recall + Precision

= 2 * 0.86 * 0.75 / 0.86 + 0.75

= 2 * 0.645 / 1.61

= 2 * 0.400

= 0.80 or 80%

POST-ASSESSMENT – CASE 1

Hospital Diagnostics
A hospital has a labeled dataset of patients' symptoms, lab tests, and diagnosis (e.g., flu, dengue, pneumonia).
They want to create a system that suggests a probable diagnosis for new patients.

Questions
1. Is this supervised or unsupervised learning? Why?
2. What type of supervised learning task is this: classification or regression?
3. Name two algorithms that could be used.
4. Why is labeled data critical in this case?

POST-ASSESSMENT – CASE 2

Online Retail Store

A store wants to segment customers into groups based on behavior:

- ✓ Purchase frequency
- ✓ Amount spent
- ✓ Browsing patterns
- ✓ Cart abandonment rate

- There are no predefined labels.

Questions

1. Which learning approach applies here?
2. Which unsupervised method is most appropriate?
3. How can segmentation benefit marketing strategies?

POST-ASSESSMENT – CASE 3

School Chatbot

A university wants to create a chatbot that categorizes student questions into topics such as:

- ✓ Enrollment
- ✓ Tuition fees
- ✓ Grades
- ✓ Guidelines
- ✓ Student services

They have thousands of **labeled** example questions.

Questions

1. What type of machine learning applies?
2. What supervised algorithm could be used for text classification?
3. What advantage do labeled examples give in this context?

CODING EXERCISE

1. **Pre-Processing and Feature Extraction**
2. **Sentiment Analysis**
3. **Topic Modeling**

THANK YOU!

ANY QUESTIONS?