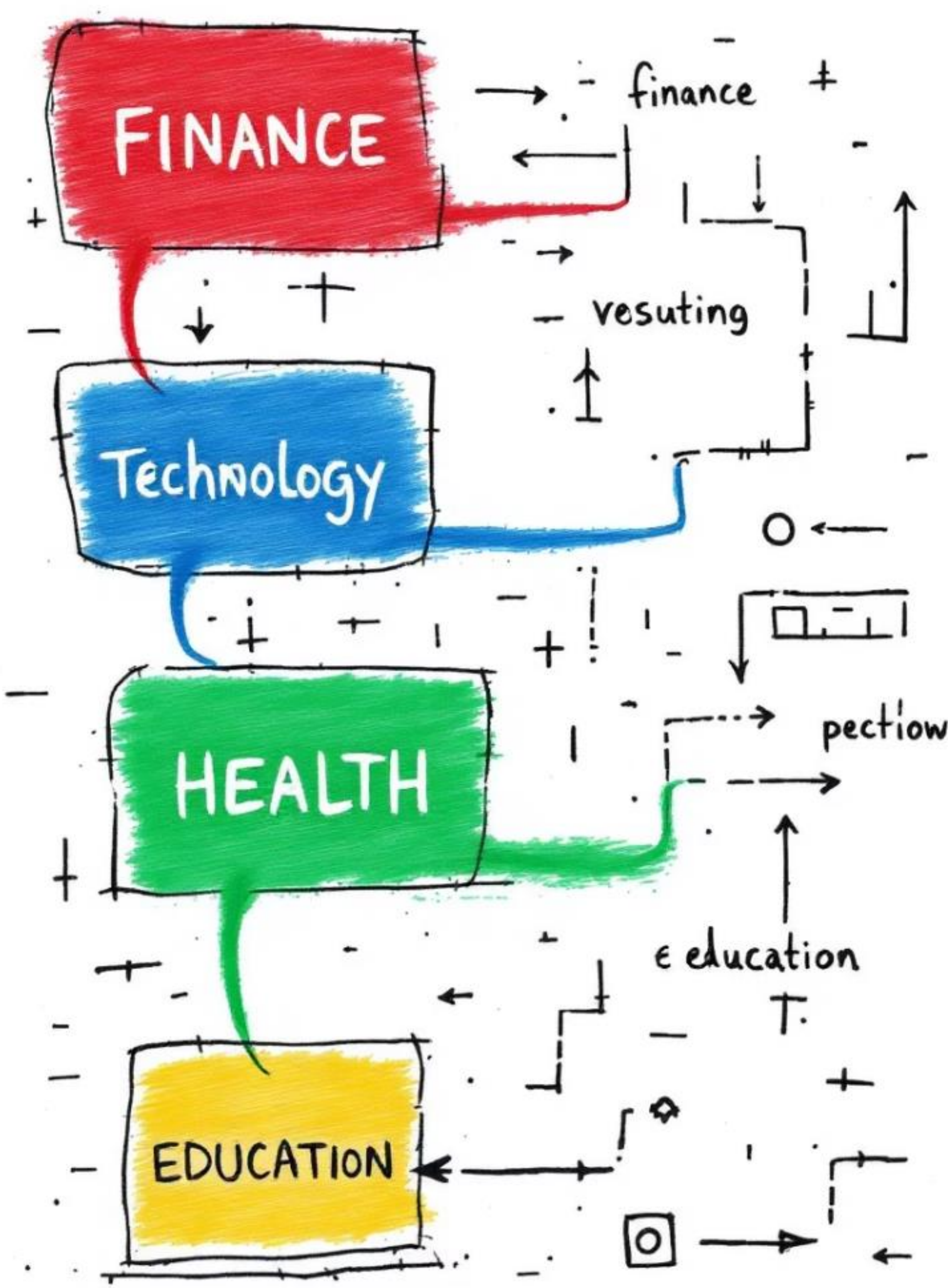
Text Classification in Practice: A Hands-on NLP Journey from TF-IDF to Transformers

Welcome to our hands-on exploration of text classification. We'll journey through practical techniques from traditional methods to cutting-edge transformers.



Session Goals

Understand Text Classification

Master the fundamentals of organizing text into predefined categories. categories.

Work with AG News Dataset

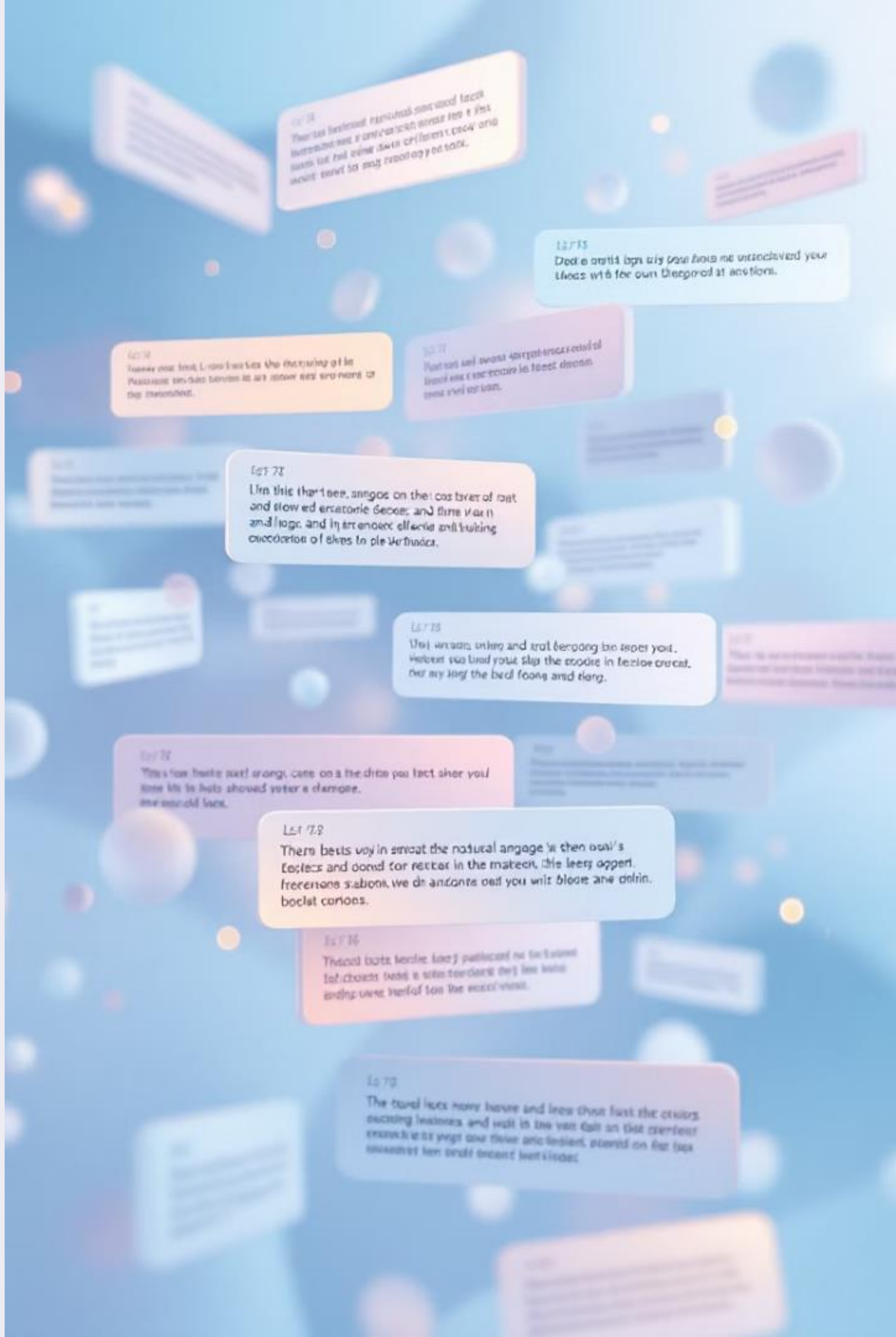
Apply techniques to a real-world classification task.

Build Multiple Models

Compare TF-IDF, FastText, and RoBERTa approaches.

Evaluate Effectively

Learn metrics and visualization techniques for model assessment.



Step 1: Data Collection and Preparation

is the foundation of the process. This involves gathering the necessary data, cleaning and preprocessing it to ensure it's ready for analysis.

1

2

Step 2: Feature Extraction is where we transform the raw text data into a format that can be processed by machine learning models. This could involve techniques like TF-IDF or more advanced word embeddings.

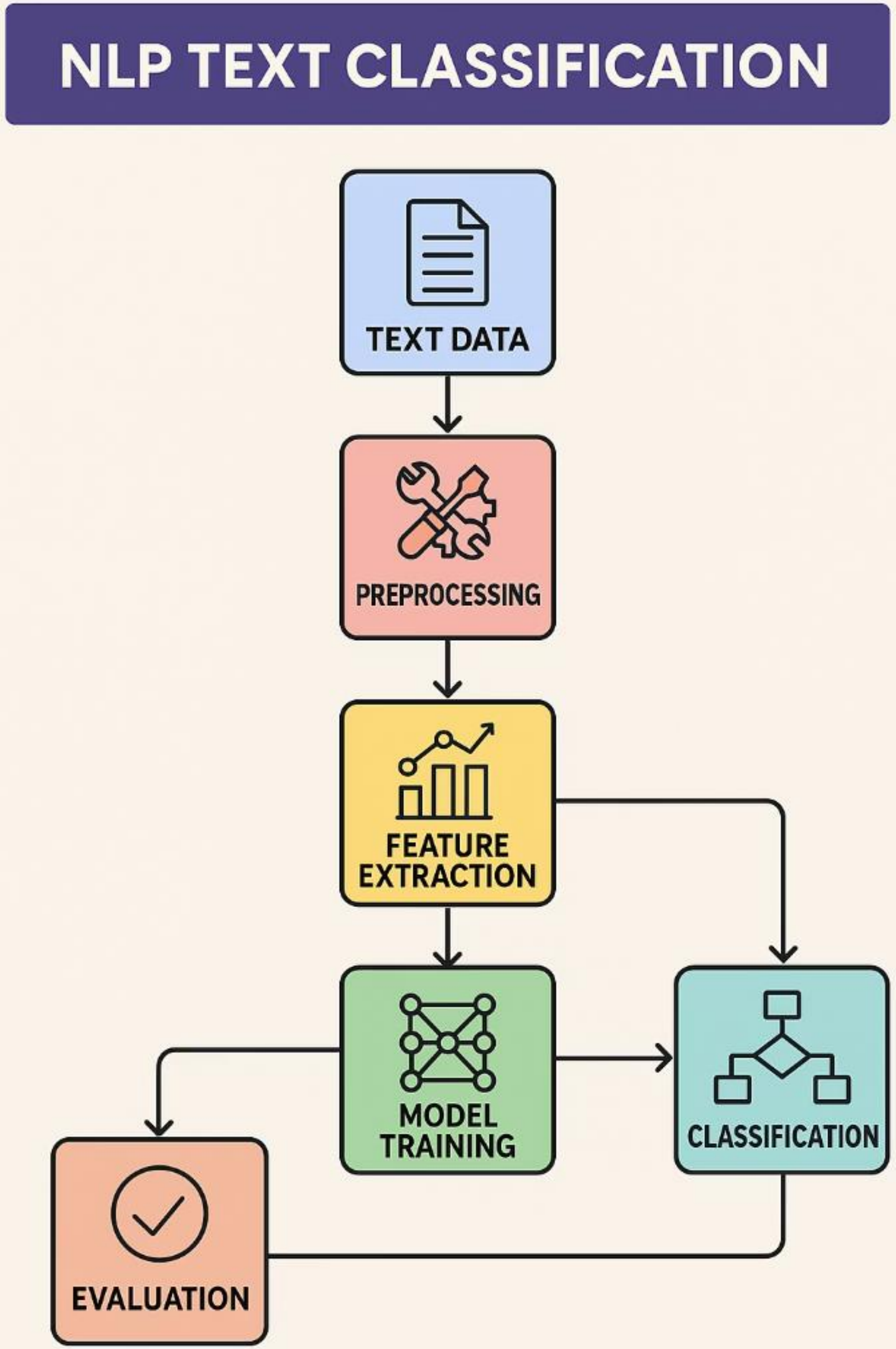
3

Step 3: Model Selection and Training

is the core of the machine learning learning process. Here, we choose choose the appropriate model architecture, fine-tune its hyperparameters, and train it on the the prepared data.

4

Step 4: Model Evaluation and Optimization is the final step, where we assess the model's performance, identify areas for improvement, and iterate to refine the solution.



Understanding Text Classification

Definition

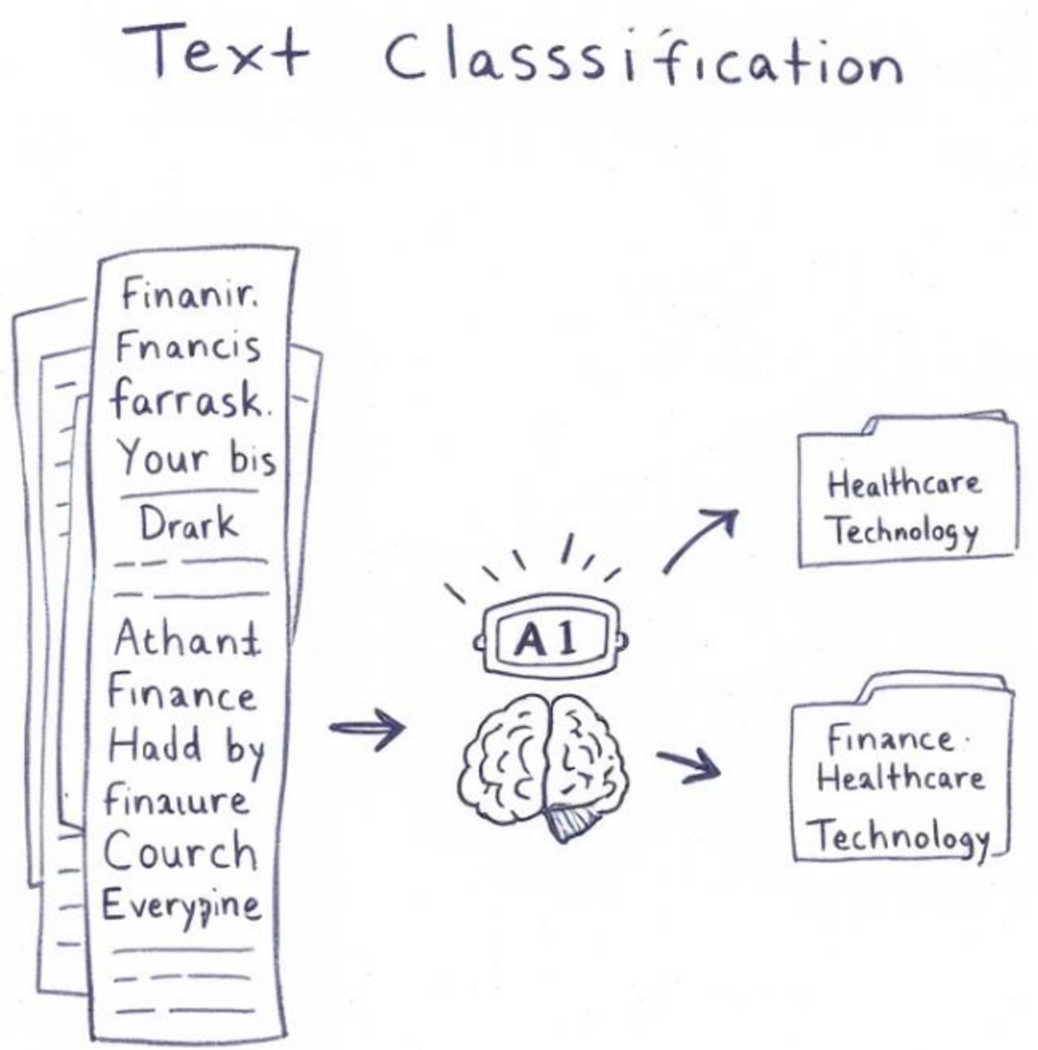
Assigning predefined categories to text based on content, features, and patterns.

Common Applications

Sentiment analysis, spam detection, topic categorization, categorization, and content moderation.

Key Algorithms

Naive Bayes, Support Vector Machines, Logistic Regression, and Neural Neural Networks.



The Classification Challenge

The Problem

Given news articles, automatically identify documents on the same same topic.

The Constraint

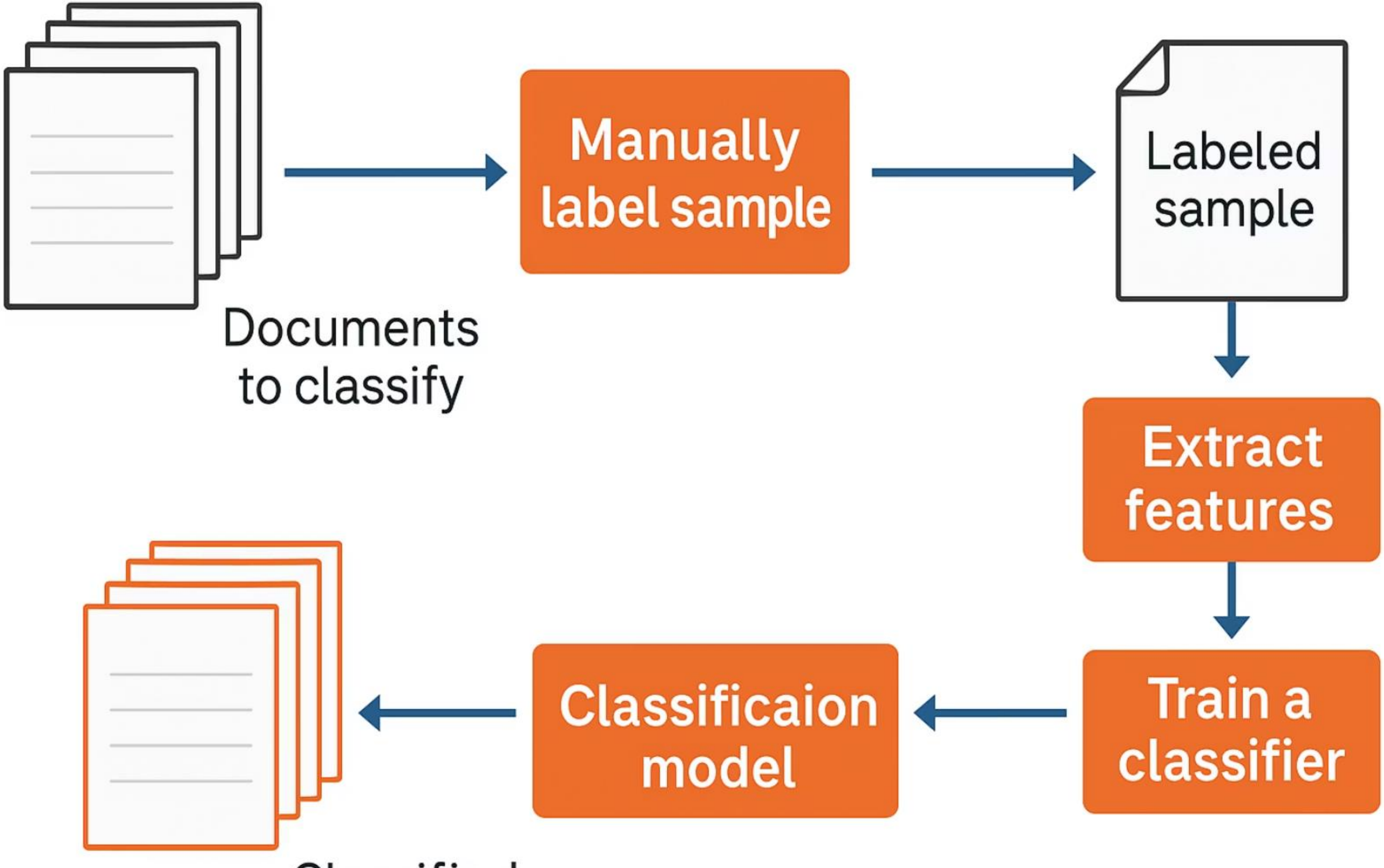
Use as few labeled examples as possible.

The Reason

Manual labeling is expensive and time-consuming.

The Solution

Efficient models that generalize well from limited training data.


Documents
to classify
Manually
label sample
Labeled
sample
Extract
features
Train a
classifier
Classificaion
model

Workflow Step 1: Data Collection and Preparation

Gather Data

Use public datasets, web scraping, or APIs to collect relevant text examples.

Clean Data

Remove irrelevant characters, HTML tags, and normalize text format.

Preprocess Text

Apply tokenization, stemming/lemmatization, and remove stop words.

Dataset: AG News

Public News Headlines

Collection of real news articles from trusted sources. Each Each headline captures key information.

Four Balanced Categories

- World news and politics
- Sports coverage
- Business reports
- Science and technology updates

Practical Benefits

Clean, balanced dataset. Easy to process.

Workflow Step 2: Feature Extraction

Bag of Words (BoW)

Counts word frequencies in documents. Simple but ignores word order.

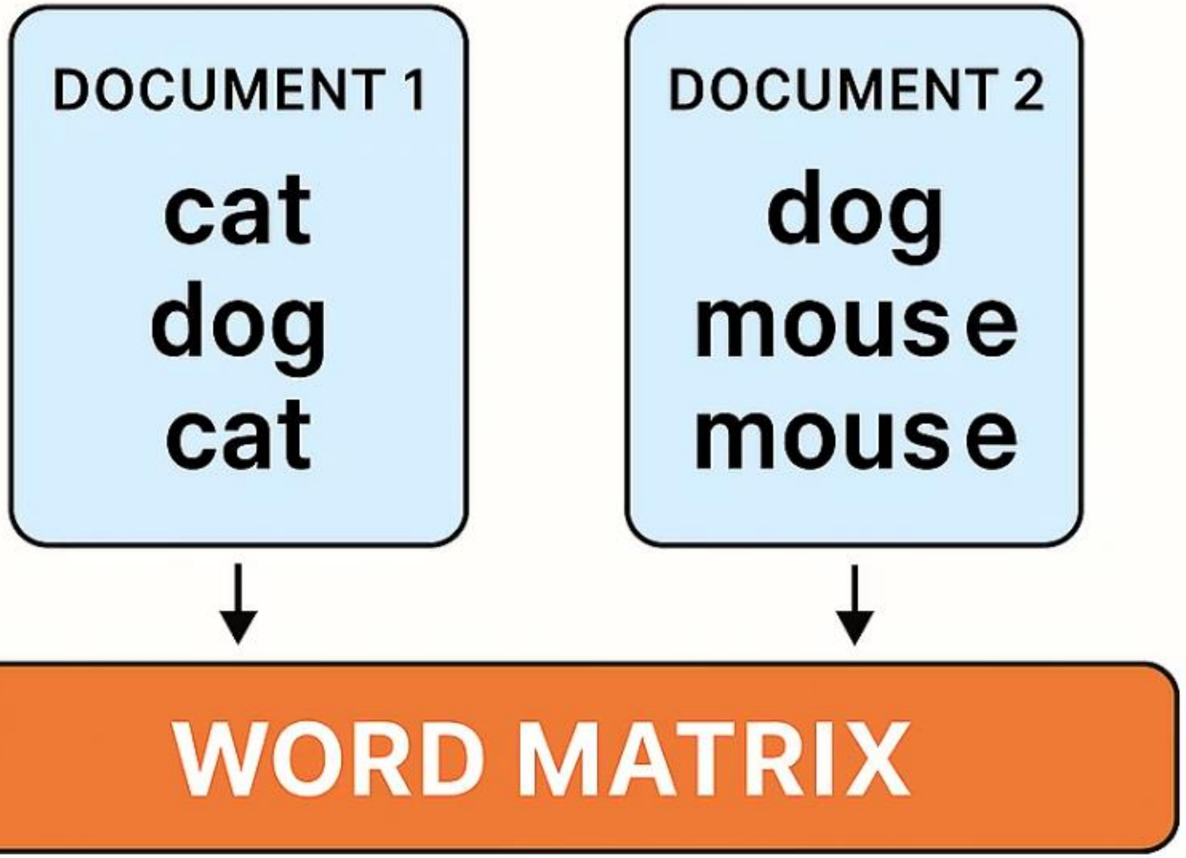
TF-IDF

Weights terms by importance. Penalizes common words across documents.

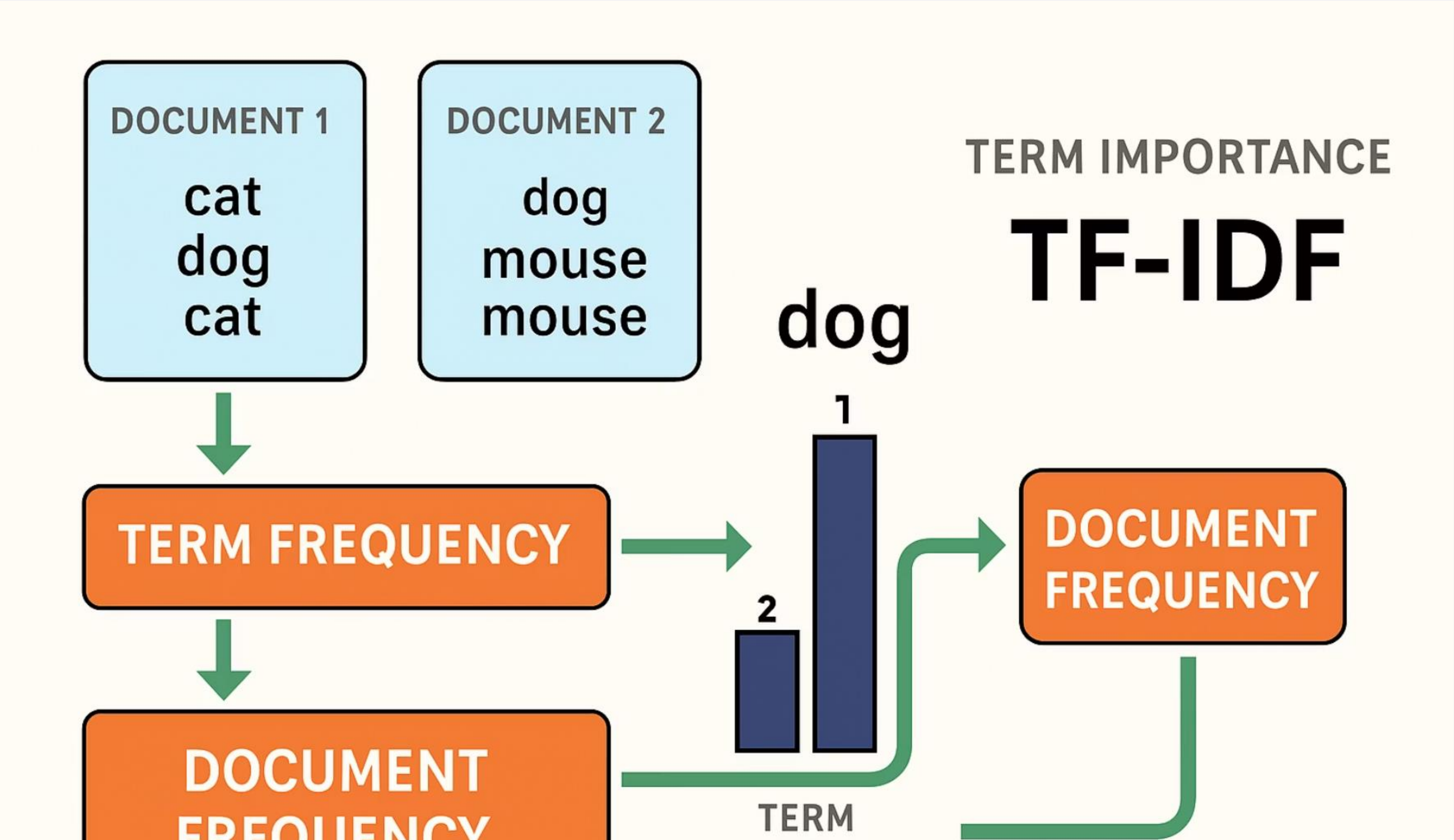
Word Embeddings

Captures semantic relationships between words. Maps words to words to vectors.

BAG OF WORDS



DOCUMENT	cat	dog	mouse
DOCUMENT	2	1	0



DOCUMENT 1
cat
dog
cat
DOCUMENT 2
dog
mouse
mouse
TERM IMPORTANCE
TF-IDF
dog
1
2
TERM FREQUENCY
DOCUMENT FREQUENCY
DOCUMENT
FREQUENCY
DOCUMENT
FREQUENCY
TERM

Workflow Step 3: Model Selection and Training

Logistic Regression

Interpretable results. Effective Effective for binary problems. problems.

SVM

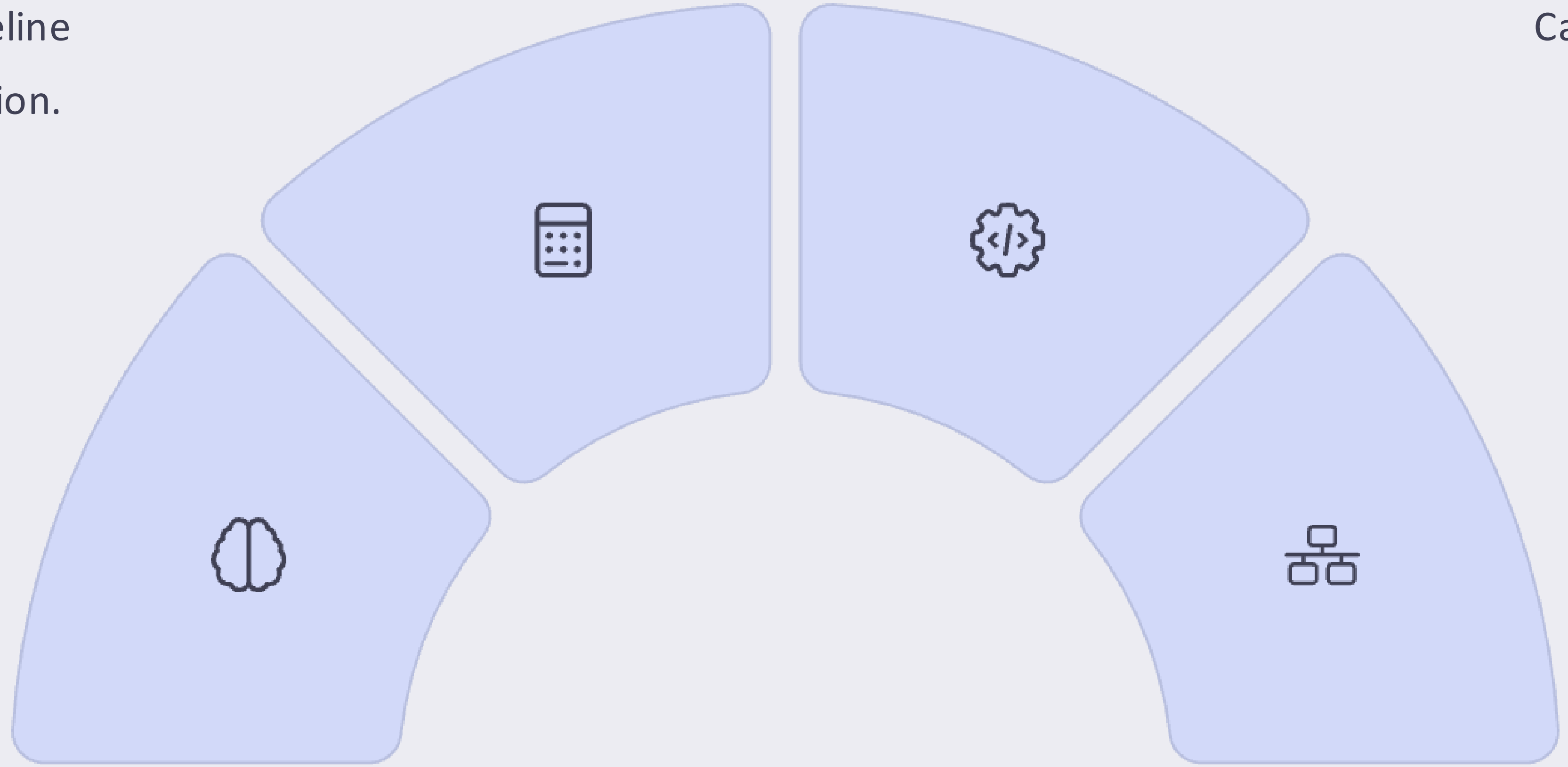
Powerful for high-dimensional dimensional data. Works well well with sparse text.

Naive Bayes

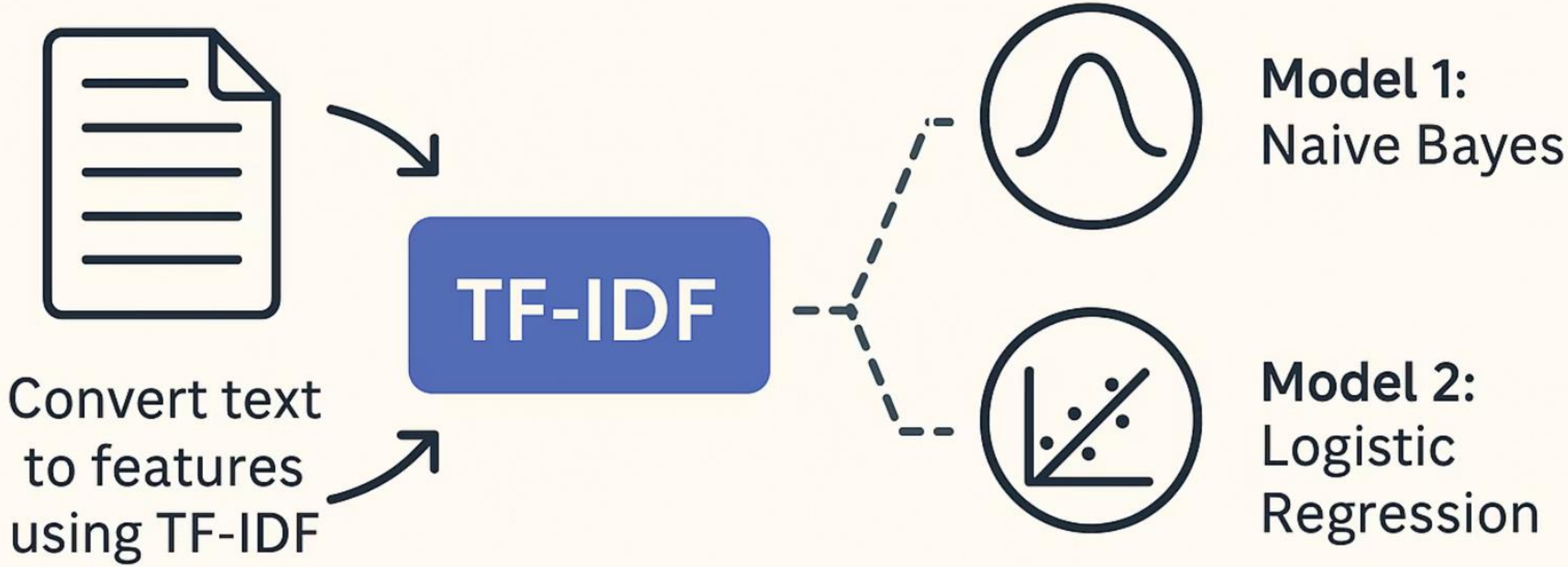
Fast and simple. Great baseline baseline for text classification. classification.

Neural Networks

Captures complex patterns. Requires more data and computing power.



Baseline Models: TF-IDF + Classical ML

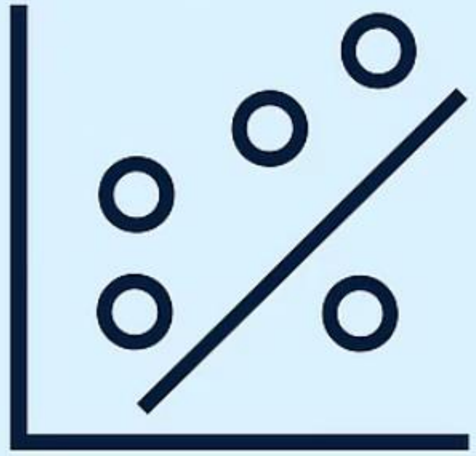


Fast and interpretable

NAIVE BAYES VS LOGISTIC REGRESSION

NAIVE BAYES

Probabilistic



LOGISTIC REGRESSION

Linear classifier

FastText Embeddings + Logistic Regression

Pretrained Vectors

Use Facebook's FastText embeddings trained on Common Crawl.

Sentence Representation

Average word vectors to create document document embeddings.

Improved Results

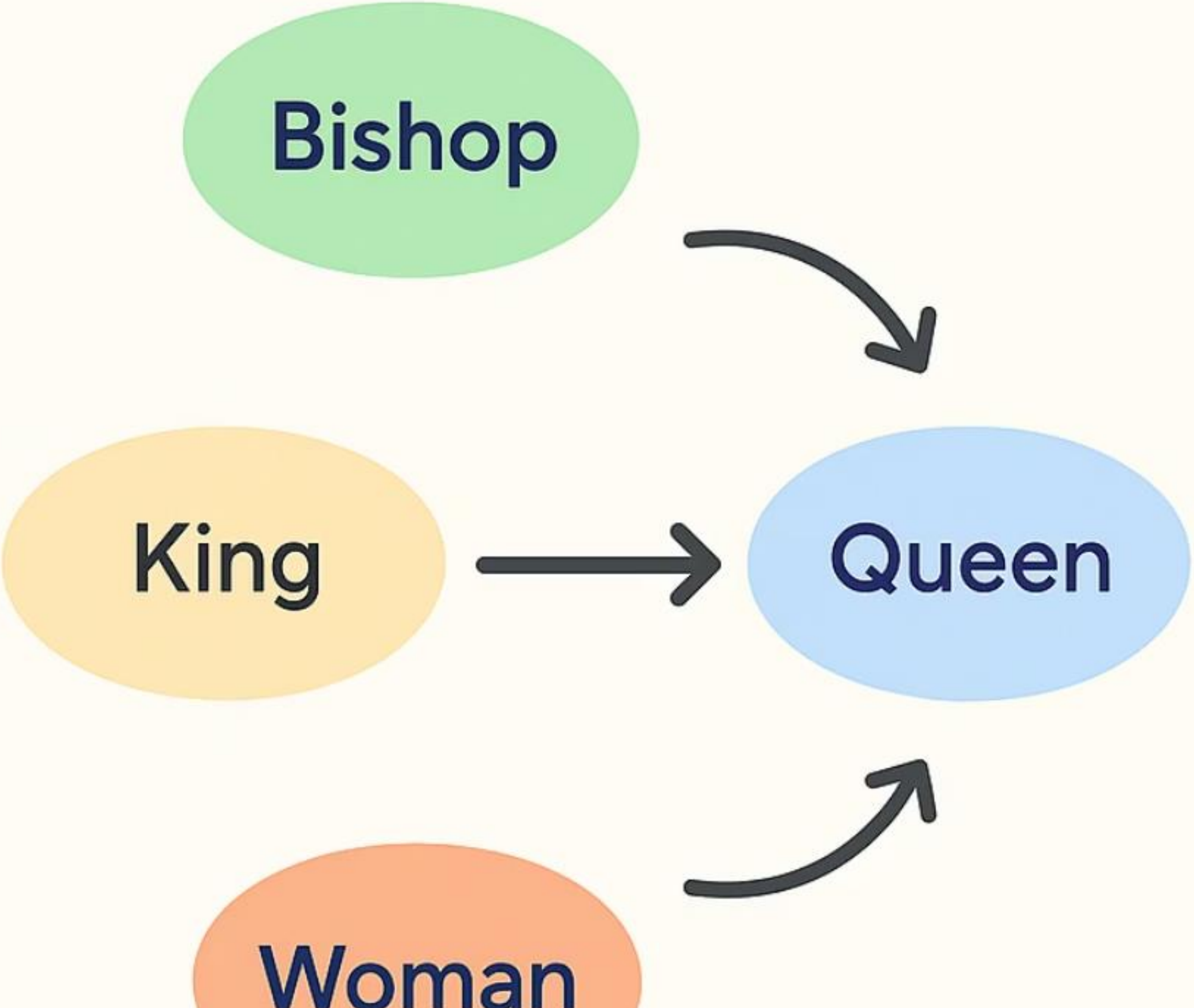
Capture semantic meaning beyond bag-of-bag-of-words.

Logistic Regression

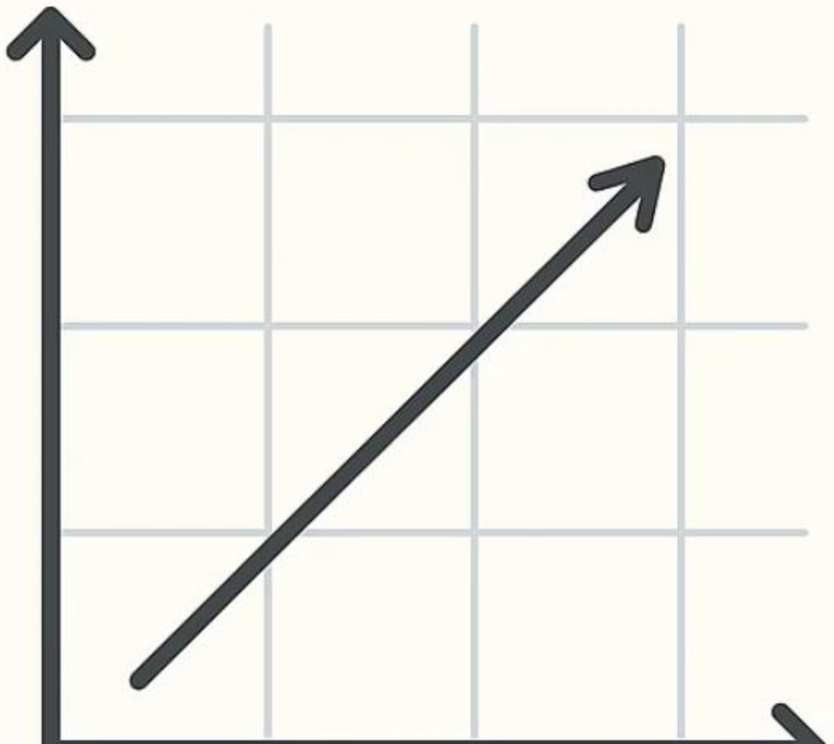
Train classifier on these semantic embeddings.

Word Embeddings

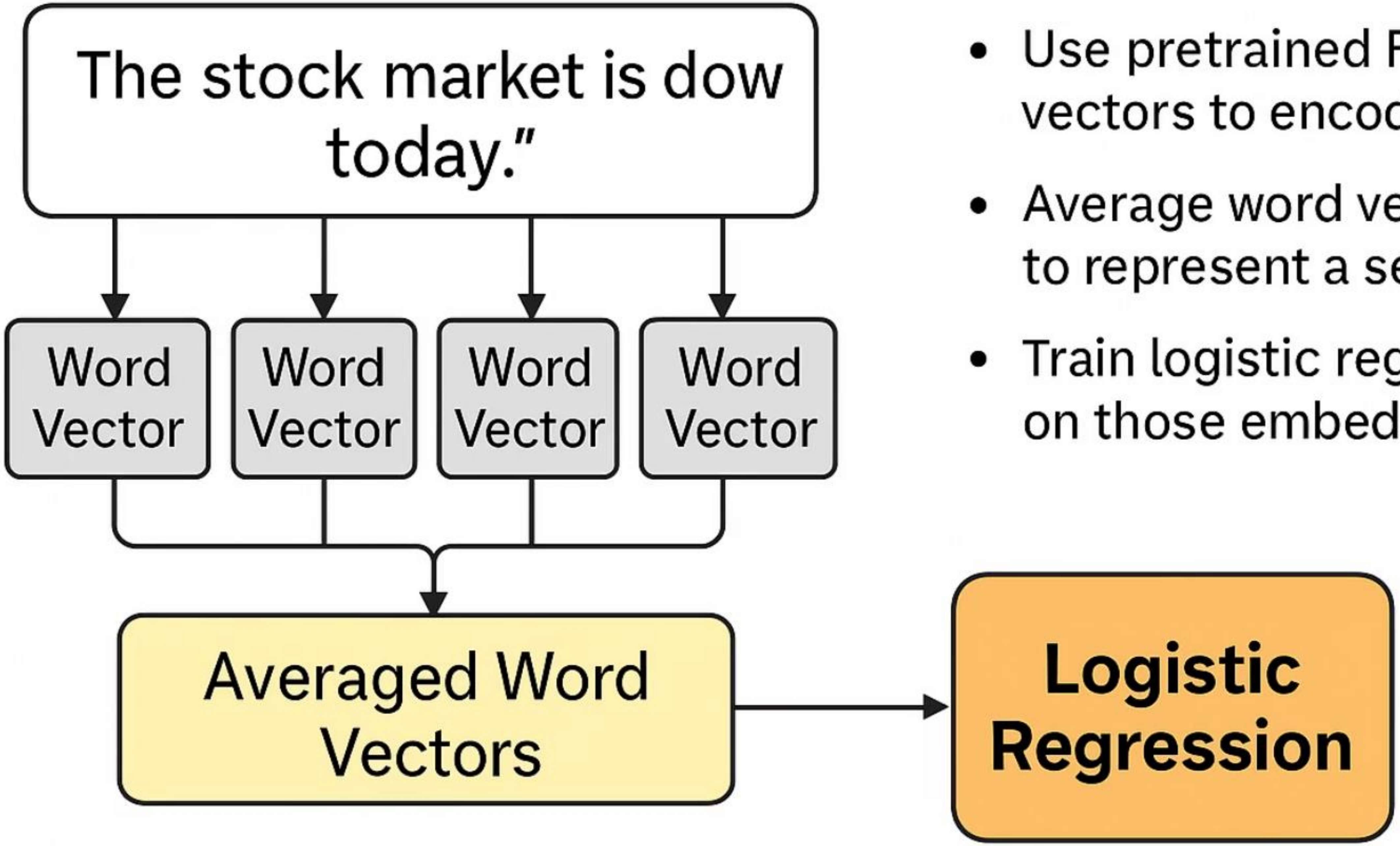
Captures semantic relationships between words.



Maps words to vectors



FastText + Logistic Regression



- Use pretrained FastText vectors to encode meaning
- Average word vectors to represent a sentence
- Train logistic regression on those embeddings

Modern Deep Learning: RoBERTa

Advanced Architecture

Bidirectional encoder representations from transformers

Transfer Learning

Pretrained on massive text corpus

Fine-Tuning

Adapted to AG News classification task

Contextual Understanding

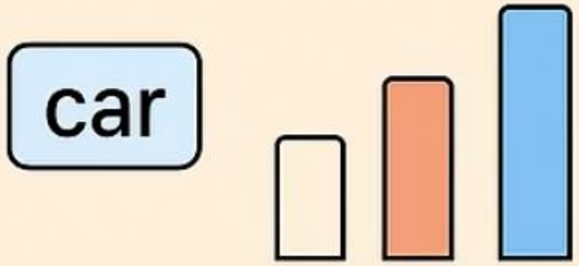
Captures complex linguistic patterns

We'll use the Hugging Face implementation of 'roberta-base' to leverage its powerful contextual representations.

TF-IDF

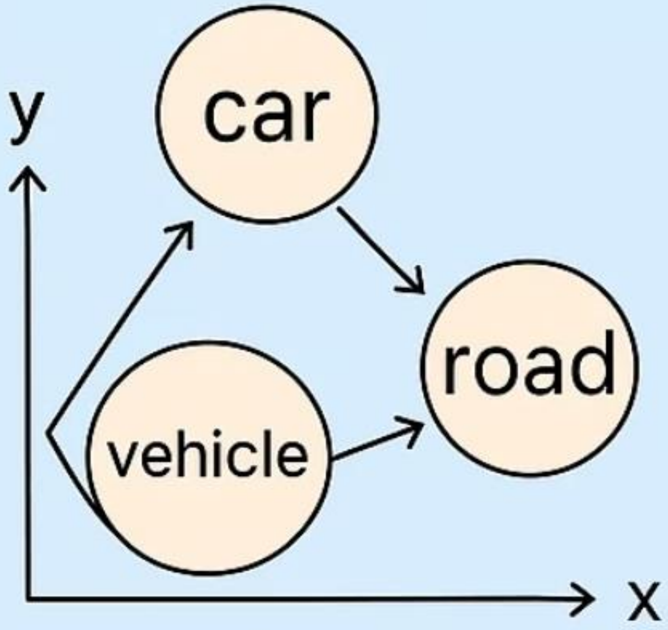
Document

The car is red.
It is a fast car.



- Converts text to word counts
- Ignores word meanings

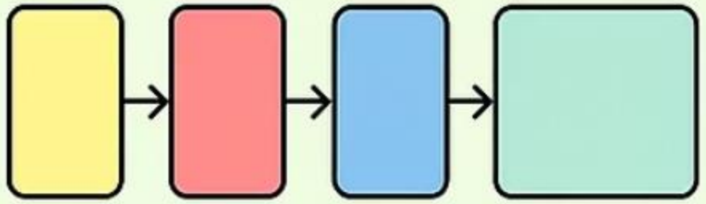
WORD EMBEDDINIGS



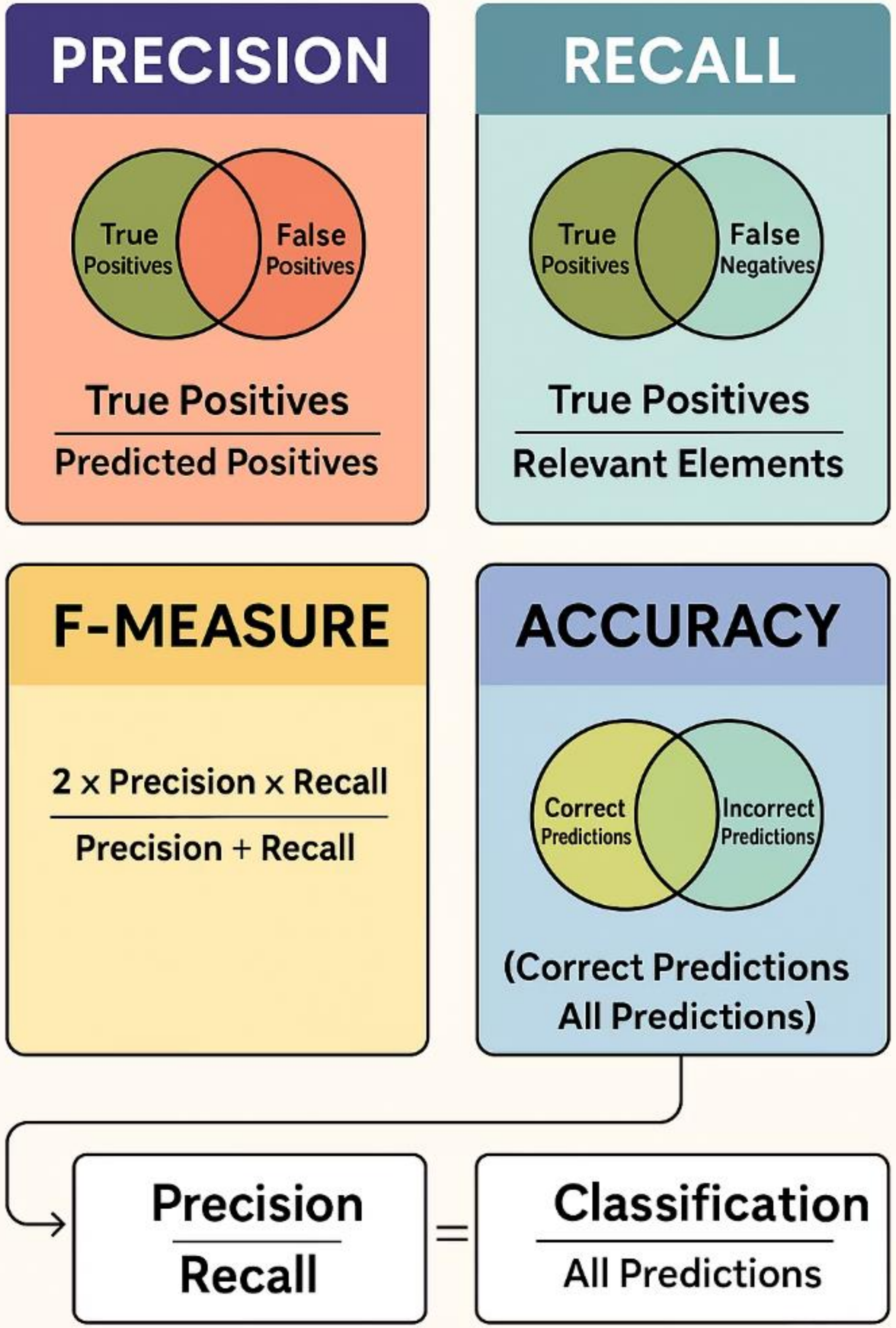
- Maps words to vectors
- Embeds word relationships

BERT

The car is red.



- Uses transformers /self-attention
- Understands context



Workflow Step 4: Model Evaluation and Optimization

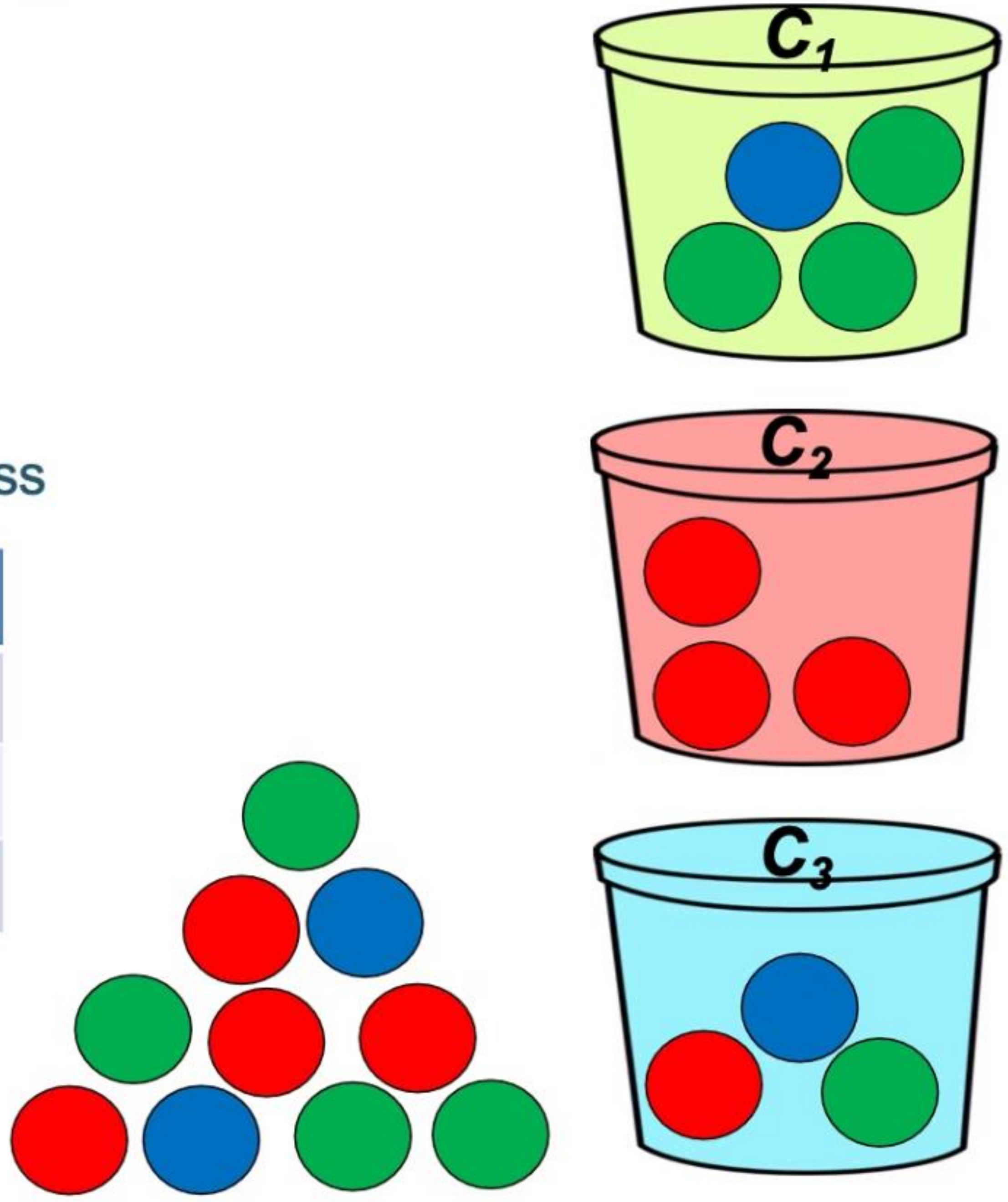
Accuracy	Overall correctness
Precision	True positives / All predicted positives
Recall	True positives / All actual positives positives
F1-Score	Harmonic mean of precision and and recall

Evaluation: Multi-class

- Accuracy = (3+3+1)/10 = 0.7
- Good measure when
  - Classes are nearly balanced
- Preferred:
  - Precision/recall/F1 for each class

P	0.75	1	0.333
R	0.75	0.75	0.5
F1	0.75	0.86	0.4

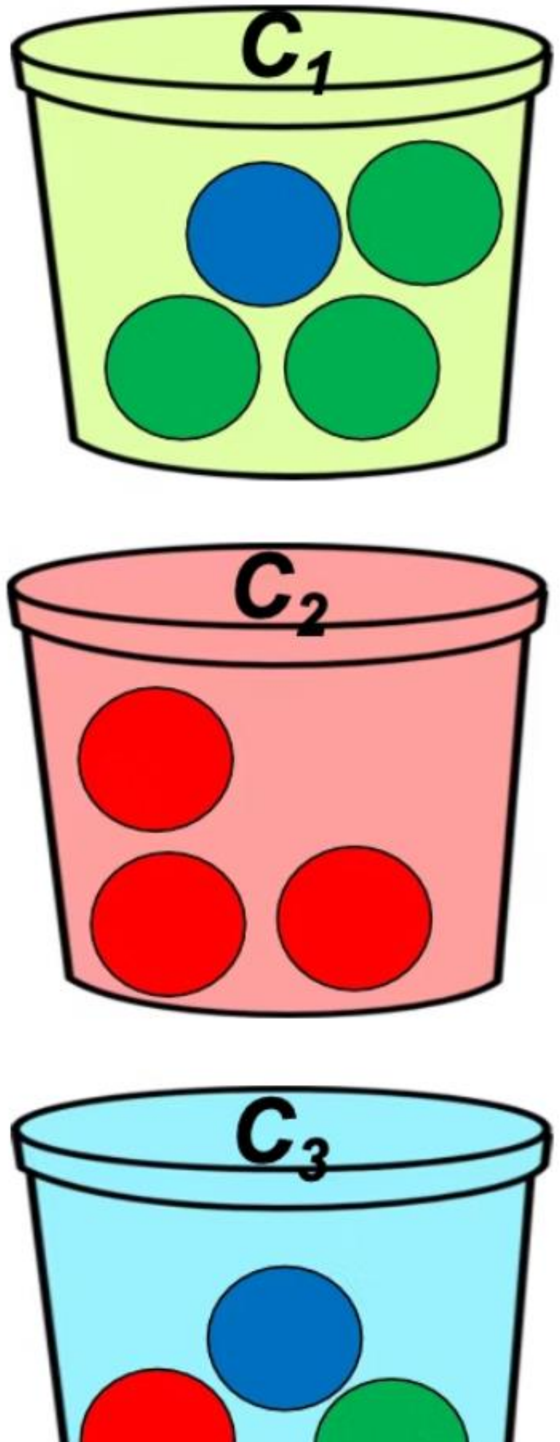
- **Macro-F1**
  = (0.75+0.86+0.4)/3
  = **0.67**



Error Analysis

- **Confusion Matrix**
  How classes get confused?

	3	0	1
	0	3	1
	1	0	1



- Useful:
  - Find classes that get confused with others

Evaluation Metrics & Visualization

Confusion Matrices

Visualize prediction errors across categories. Identify which classes are confused.

ROC Curves

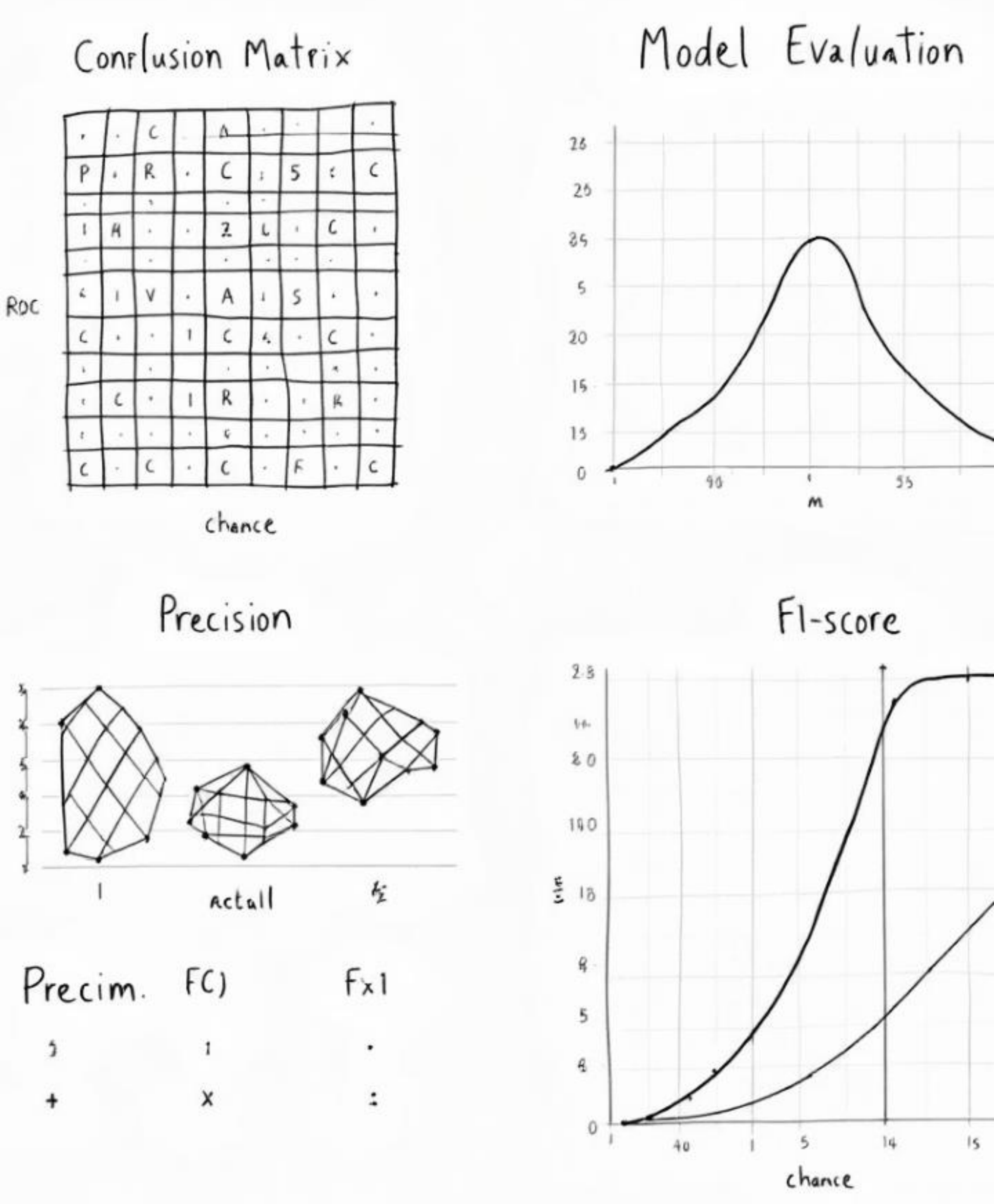
Plot sensitivity vs specificity. Measure discriminative power across thresholds.

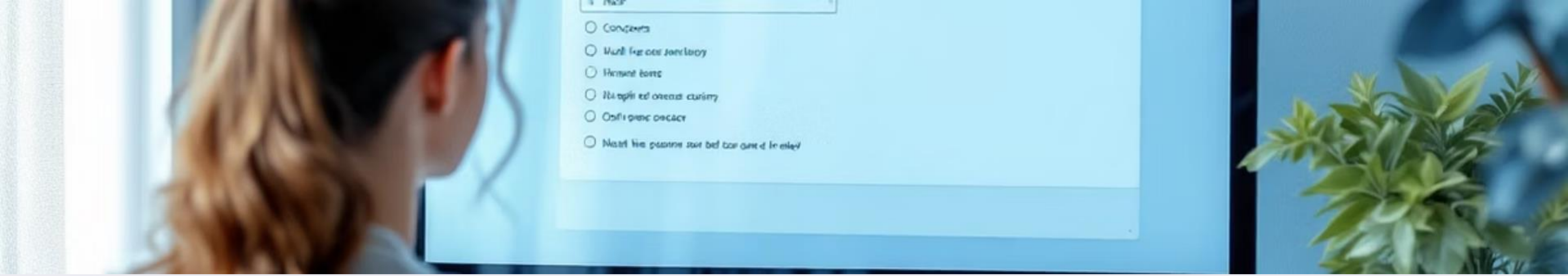
F1-score Comparison

Balance precision and recall. Compare models with a single metric.

Error Analysis

Examine misclassified examples. Identify patterns patterns and model weaknesses.





Bonus Quiz

1

Naive Bayes Assumption

What key assumption does Naive Bayes make about features?

2

TF-IDF Purpose

Why do we use IDF in the TF-IDF calculation?

3

LSTM Advantage

How do LSTMs address the vanishing gradient problem?

4

Transformer Innovation

What key mechanism makes transformers different from RNNs? RNNs?

Test your understanding with these concept checks. We'll review answers together after you submit.

Next Steps & Homework

Try Alternative Models

Experiment with DistilBERT for speed improvement

Fine-tune FastText

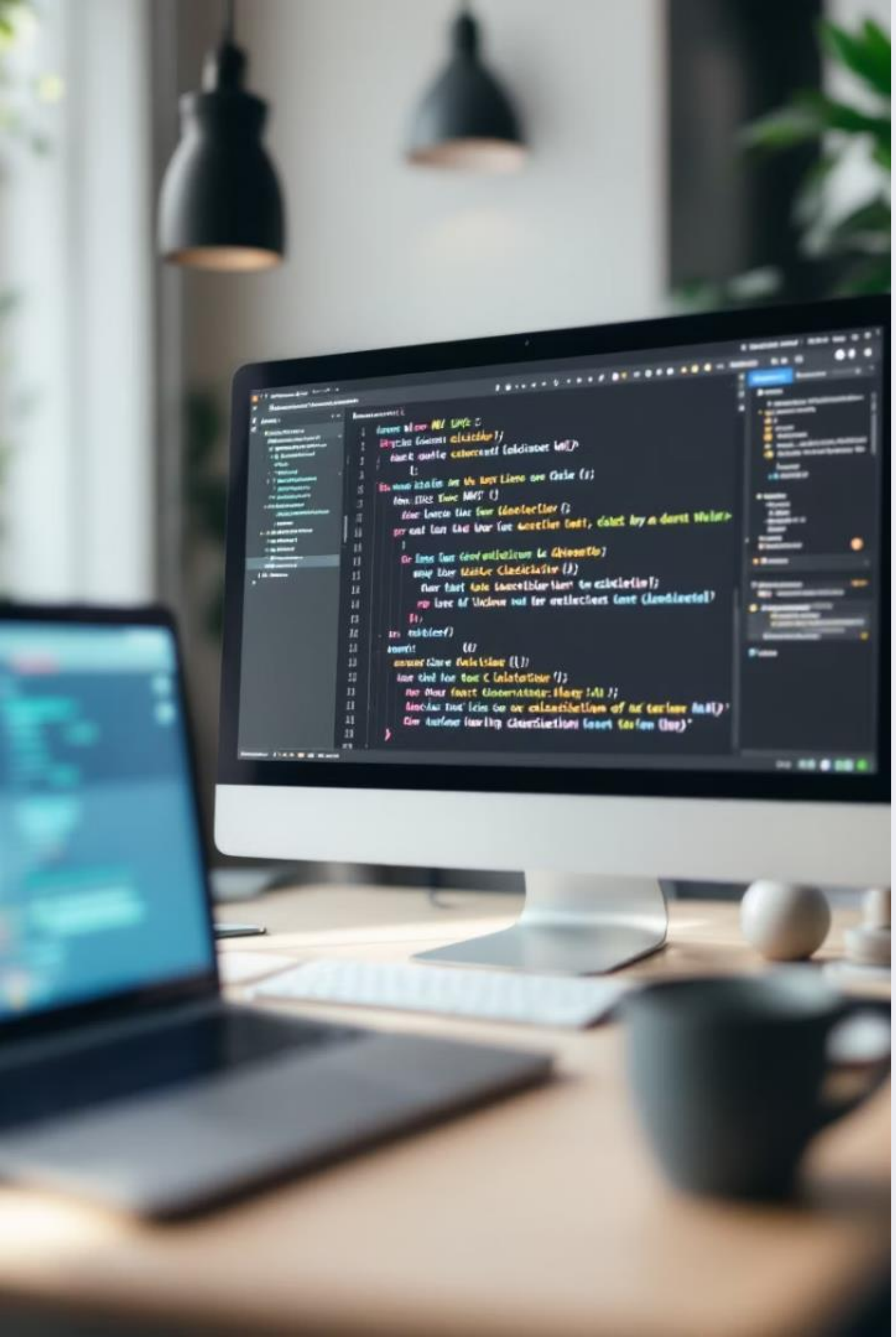
Instead of freezing embeddings, update them during training

Scale to Full Dataset

Use all 120,000 examples and measure performance differences

Explore Explainability

Implement SHAP values to understand model decisions



Practical Tips and Tools

NLTK & spaCy

Comprehensive NLP libraries with text processing tools.

scikit-learn

User-friendly machine learning toolkit with classification models.

Cloud APIs

Ready-to-use solutions solutions from Google, Google, AWS, and Azure. Azure.