One Hot Encoding

> 💡 **Not used in NLP.**
>
> **Useful in ML.**

What is One-Hot Encoding?

- Represents each word as a **binary vector** (0s and 1s)
- Vocabulary size = Vector dimension
- Only one "1" per vector (the "hot" bit), rest are "0"

Example:

- Vocabulary: `["cat", "dog", "bird"]`
- One-hot encoded:
  - `"cat"` → `[1, 0, 0]`
  - `"dog"` → `[0, 1, 0]`
  - `"bird"` → `[0, 0, 1]`

🧠 Why It's Called "One-Hot"?

- "Hot" means **active** = 1
- All others are **cold** = 0
- Only **one active value per word**

When to Use One-Hot Encoding?

✅ **Small vocabulary sizes** (e.g., categories, tags)

✅ **Simple baseline models**

❌ **Avoid for large vocabularies** (creates high-dimension sparse vectors)

Problem	Explanation
❌ No meaning	All words are equally distant — "king" and "queen" are just as far as "king" and "banana"
❌ Sparse	Vectors are mostly 0s — very inefficient
❌ No context	"bank" in “river bank” vs “money bank” is treated the same

✅ Where It's Still Useful

- In **simple models** (like Naive Bayes or Logistic Regression)
- As a **starting point** in text preprocessing
- For **categorical variables** in tabular data

Better Alternatives

- **Word Embeddings** (Word2Vec, GloVe) - Captures semantic meaning
- **TF-IDF** - Weights words by importance
- **Hashing Trick** - Reduces dimensionality

Python Code

```
from sklearn.preprocessing import OneHotEncoder
import numpy as np

# Sample text data
words = ["cat", "dog", "bird", "cat", "bird","YO"]

# Reshape for sklearn
words = np.array(words).reshape(-1, 1)

# Initialize and fit one-hot encoder
encoder = OneHotEncoder(sparse_output=False)
```

```
one_hot = encoder.fit_transform(words)

one_hot
```

```
array([[0., 0., 1., 0.],
       [0., 0., 0., 1.],
       [0., 1., 0., 0.],
       [0., 0., 1., 0.],
       [0., 1., 0., 0.],
       [1., 0., 0., 0.]])
```

reshape(-1, 1)

- Reshapes the array into a **2D matrix** with **1 column** and **automatic rows** (`1` means "infer the size").
- Why? Because `OneHotEncoder` expects **2D input** (each sample as a row).

Before Reshaping:

python Copy Download

```
array(['cat', 'dog', 'bird'])  # Shape: (3,)
```

(1D array, incompatible with scikit-learn)

After Reshaping:

python Copy Download

```
array([['cat'],
       ['dog'],
       ['bird']])  # Shape: (3, 1)
```

(2D array, where each word is a separate sample)

🔄 Compared To Embeddings

Feature	One-Hot Encoding	Word Embedding
Vector Type	Binary (0s and 1s)	Continuous numbers
Length	Vocabulary size	Fixed size (e.g., 100, 300)
Captures Meaning?	❌ No	✅ Yes
Sparse/Dense	Sparse	Dense
Context-Aware	❌	✅ (for contextual embeddings)