# TF-IDF Vectorization

> **TF-IDF** stands for **Term Frequency–Inverse Document Frequency**.
> It transforms textual data (like movie overviews or genres) into numerical vectors, highlighting terms that are **important in a specific movie but uncommon across all movies**.

## Key-points

We combine fields like **title, overview, genres, cast, crew, keywords** into one text blob per movie.

We apply TF-IDF to convert this text into a **high-dimensional vector**.

Common words across many movies get lower weight (like “the” or “film”).

Unique, meaningful words like “wizard”, “mission”, “time travel” get higher scores.

### 📘 What is Term Frequency (TF)?

TF measures **how often a word appears** in a single document.

> **Example – Movie 1 Overview:**
> *"The wizard enters the magical school of Hogwarts."*

| Term | TF (count in movie 1) |
|---|---|
| wizard | 1 |
| magical | 1 |
| school | 1 |
| hogwarts | 1 |
| the | 2 |

### 📚 What is Inverse Document Frequency (IDF)?

IDF gives **higher weight to rare words** and **penalizes common words** across all movies.

Let’s say we have **3 movies**:

| Word | Appears in how many movies? | IDF Score |
|---|---|---|
| wizard | 1 | High |
| hogwarts | 1 | High |
| the | 3 | Low |
| school | 2 | Medium |

**Rare words** → **high IDF** → more important
**Common words** → **low IDF** → less useful

### 🔢 TF-IDF Calculation (Simplified)

TF-IDF=TF×IDF\text{TF-IDF} = \text{TF} \times \text{IDF}TF-IDF=TF×IDF

| Term | TF | IDF | TF-IDF (Importance) |
|---|---|---|---|
| wizard | 1 | High | ⭐ High |
| the | 2 | Low | 🔻 Low |
| hogwarts | 1 | High | ⭐ High |

---

## Cosine Similarity

> **Cosine Similarity** measures how similar two movies are by comparing the **angle between their TF-IDF vectors**.

### 🔍 Why We Use It:

- We want to recommend movies **with similar themes, tone, or plot**.
- Cosine similarity gives high score to vectors (movies) pointing in the same direction — even if their magnitudes differ.

### 📈 Example:

- `Cars` vector vs. `Cars 2` vector → small angle → high similarity

- `Cars` vs. `Shawshank Redemption` → large angle → low similarity

---

# Jaccard Evaluation

> **Jaccard Similarity** compares two **sets** — in our case, the **genres** of two movies.

**Jaccard (A, B) = (| A ∩ B |)/ (| A U B |)**

### 🔍 Why We Use It:

- After recommending movies using cosine similarity, we **filter results based on genre similarity**.
- We ensure the recommendations are not just textually similar, but **thematically aligned**.

### 📈 Example:

- Input Movie Genres: `{"Drama", "Fantasy", "Mystery"}`
- Recommended Movie Genres: `{"Drama", "Fantasy", "Romance"}`

**Jaccard score = 2/4 -> This shows as Good thematic overlap**

We **discard movies with Jaccard < 0.2** to improve recommendation quality.

---

# Combined flow

User inputs a movie
|
Get its TF-IDF vector
|
Find cosine similarity to all other movies
|
Filter by genre using Jaccard similarity

Return Top-N most similar & relevant movies