# E-commerce Product Text Classification

Using TF-IDF and Word2Vec for Text Representation and Classification

**Pourya Alvani** **Hossein Arianmehr**

Ferdowsi University of Mashhad
Department of Computer Engineering
Course: Fundamentals of Computational Intelligence
Summer Semester 2025

## Contents

# 1 Introduction

The main objective of this project is to develop a classification system for product descriptions from an online e-commerce store. Each product belongs to one of the predefined categories such as:

- **Electronics**
- **Household**
- **Books**
- **Clothing**

In this context, we focus on Natural Language Processing (NLP) techniques to convert raw text into numerical representations. Two widely-used and effective methods are implemented and compared:

- TF-IDF (Term Frequency–Inverse Document Frequency)
- Word2Vec (Word Embedding Model)

The dataset used for this project is the **E-commerce Text Classification Dataset**, which contains thousands of product descriptions along with their corresponding categories. The goal is to analyze and compare the performance of these two vectorization techniques on the task of text classification.

# 2 Phase 1: Data Preprocessing

In the first phase of the project, the primary goal was to clean and preprocess the dataset to make it suitable for text representation and classification tasks. Raw product descriptions usually contain noise such as URLs, HTML tags, punctuation, and irrelevant words that must be removed.

## Objectives

- Prepare raw product texts for Natural Language Processing (NLP)
- Remove noise and standardize text format
- Create a clean textual representation for each product description

## Methods

The preprocessing process involved the following key techniques:

- **Lowercasing:** Convert all text to lowercase to ensure consistency.
- **URL and HTML Removal:** Detect and remove URLs and HTML tags using regular expressions.
- **Non-alphabetic Filtering:** Exclude non-alphabetic characters to reduce noise.
- **Token Filtering:** Only retain meaningful tokens with at least 3 characters.
- **Stopword Removal:** Eliminate common English stopwords that carry little semantic meaning.
- **Lemmatization:** Reduce each word to its base form to unify word variants.

## Outcome

After preprocessing, each product description was transformed into a clean, normalized text. The dataset was saved as a new CSV file for use in subsequent phases. A preview of the cleaned dataset is shown below.

| **Label** | **Clean Text** |
|---|---|
| Electronics | wireless bluetooth headphone long battery life |
| Books | beginner python programming guide |
| Clothing | cotton shirt breathable summer wear |
| Household | vacuum cleaner high suction power |
| Electronics | usb flash drive fast speed durable |

Table 1: Sample of Cleaned Data

## 3 Phase 2: Text Vectorization

After the dataset was cleaned in Phase 1, the next step involved transforming the textual data into numerical representations suitable for machine learning classifiers. Two distinct vectorization techniques were implemented and compared in this phase:

### 1. TF-IDF (Term Frequency-Inverse Document Frequency)

The TF-IDF method quantifies the importance of words in each document relative to the entire corpus. In this project, the TF-IDF representation was implemented manually, without relying on external libraries such as `scikit-learn`. The key steps were:

- Constructing the vocabulary from the cleaned corpus
- Calculating document frequency (DF) for each token
- Computing the inverse document frequency (IDF) values
- Generating the TF-IDF sparse matrix using raw term frequency (TF) times IDF

The final output was a compressed sparse row (CSR) matrix representing each document in a high-dimensional space where each column corresponds to a unique token in the vocabulary.

### 2. Word2Vec Embedding

To capture the semantic similarity between words, Word2Vec was also applied using the `gensim` library. A skip-gram model was trained on the tokenized corpus to produce dense word embeddings of dimension 100.

Each document was then represented as the mean of the embeddings of its constituent words. This produced a dense matrix where each row corresponds to a document vector with semantic information.

### Output

The results of this phase included:

- **TF-IDF Matrix:** Sparse matrix with shape $(N, V)$, where $N$ is the number of documents and $V$ is the vocabulary size.
- **Word2Vec Matrix:** Dense matrix with shape $(N, 100)$.
- **Labels:** Array of original product categories for use in classification.

All vectorized outputs were saved to the `./vectorised/` directory for reuse in later classification tasks.

# 4 Phase 3: Classification and Visualization

With the vectorized representations of the dataset obtained in Phase 2, this phase focused on building classification models to predict the category of a product description. The goal was to assess which vectorization method—TF-IDF or Word2Vec—yields better classification performance.

## Approach

A baseline classification pipeline was implemented using **Logistic Regression**. The following steps were followed for both vector types:

- Split the dataset into training and test sets (80% / 20%)
- Train the model on the training set
- Predict the test set
- Evaluate using accuracy, precision, recall, F1-score, and confusion matrix

## Class Labels

The classification task involved four product categories:

- Electronics
- Household
- Books
- Clothing

## Results Summary

### TF-IDF + Logistic Regression

- **Training Accuracy:** 99.98%
- **Test Accuracy:** 95.07%
- **Macro F1-score:** 0.95

### Word2Vec + Logistic Regression

- **Training Accuracy:** 92.59%
- **Test Accuracy:** 92.93%
- **Macro F1-score:** 0.93

### Remarks on Visualization

The classification performance was also evaluated using confusion matrices (not included here) which confirmed high precision and recall across all classes. Additionally, dimensionality reduction using t-SNE was applied to Word2Vec vectors, revealing distinct clustering patterns for each category in a 2D space.

The visualizations provide further qualitative support for the numerical evaluation.

## 5 Phase 4: Evaluation and Conclusion

In the final phase, the two vectorization methods were compared in terms of their effectiveness in downstream classification.

### Findings

- TF-IDF demonstrated slightly superior accuracy and consistency across all metrics.
- Word2Vec embeddings captured semantic similarity and yielded comparable performance with fewer dimensions.
- Logistic Regression proved effective and interpretable for both vector types.

### Conclusion

Both TF-IDF and Word2Vec are effective for text classification in the context of e-commerce product descriptions. While TF-IDF achieved higher overall accuracy, Word2Vec offers flexibility and better semantic representation. Future work could explore ensemble models, transformer-based embeddings, or class-specific tuning.