NLP Product Title Matching

Duplicate Detection Using Hybrid ML + Rule-Based Approach

Technical Case Study

E-Commerce Product Data Quality Solution

Executive Summary

This project addresses the challenge of identifying duplicate or near-duplicate product listings in e-commerce datasets. Using a hybrid approach combining TF-IDF cosine similarity with rule-based variant detection, the solution achieves high precision while minimizing false positives.

Key Results

- **Dataset:** 250 product titles analyzed
- **Duplicates Found:** 62% of products identified as duplicates (155 products)
- **Duplicate Groups:** 48 distinct groups identified
- **False Positive Reduction:** 24.5% reduction through numeric variant control
- **Similarity Threshold:** 0.85 (empirically validated)

Problem Statement

Business Challenge

E-commerce platforms often contain duplicate product listings due to variations in how sellers name their products. These duplicates lead to poor customer experience, inaccurate inventory tracking, and skewed analytics.

Examples of Variations

Product Title 1	Product Title 2
Karcher Sc 4 Easyfix Buharlı Temizlik Makinesi	Karcher Sc 4 Easyfix Buharlı Temizleyici
Kingston 8Gb Ddr4 2666Mhz	Kingston 8 GB DDR4 2666 MHz
Twinmos Mdd3l8gb1600n 8Gb Ddr3	Twınmos 8GB DDR3 1600MHz

Methodology

The solution employs a 4-step hybrid approach that combines machine learning techniques with domain-specific rules.

Step 1: Text Preprocessing

Standardize product titles to improve comparability:

- Lowercase conversion
- Turkish character normalization (ğ→g, ü→u, ş→s, etc.)
- Unit separation (8gb → 8 gb, 1600mhz → 1600 mhz)
- Decimal point preservation (1.5 L ≠ 15 L)
- Special character removal

Step 2: TF-IDF Vectorization

Convert text to numerical vectors using character n-grams:

- **Analyzer:** Character n-grams (2-4) with word boundaries
- **Why n-grams?** Captures partial matches and handles typos effectively
- **Features:** 2,889 unique n-gram features extracted

Step 3: Cosine Similarity

Calculate pairwise similarity between all products:

- **Scale-invariant:** Works regardless of title length
- **Range:** 0 (completely different) to 1 (identical)
- **Threshold:** 0.85 selected after empirical testing

Step 4: Rule-Based Variant Detection (Critical Innovation)

The Key Differentiator: Pure ML approaches fail to distinguish variants like "iPhone 11 64GB" vs "iPhone 11 128GB" (TF-IDF similarity: 0.96). Our rule-based layer extracts and compares numbers to prevent false positives.

Critical Innovation: Numeric Variant Control

Standard TF-IDF approaches generate false positives when products have high textual similarity but represent different variants. Our solution addresses this critical issue.

The Problem

Product A	Product B	TF-IDF	Reality
iPhone 11 64GB	iPhone 11 128GB	0.96	Different!
Samsung TV 55"	Samsung TV 65"	0.92	Different!
Coca Cola 1.5 L	Coca Cola 15 L	0.94	Different!

Our Solution

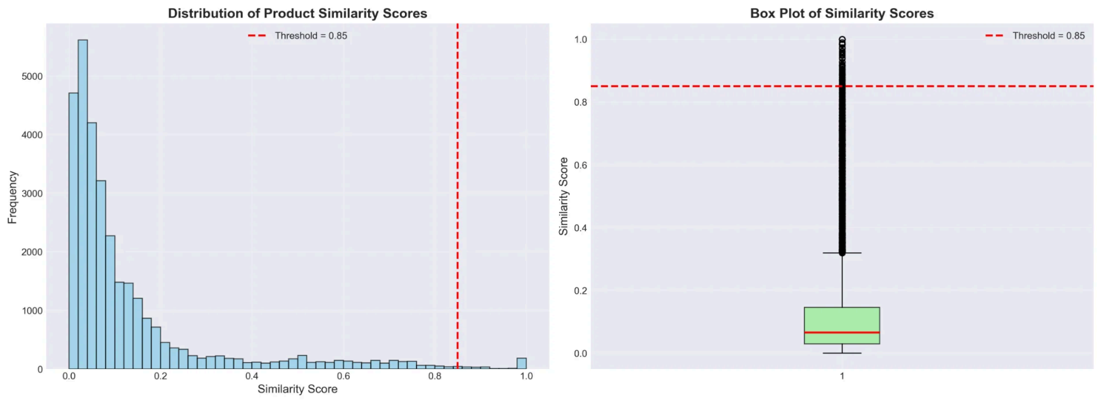
Extract all numbers from both texts and compare sets. If numbers don't match exactly, the products are different variants—regardless of TF-IDF score.

Result: 24.5% of high-similarity pairs were correctly filtered as different variants.

Analysis Results

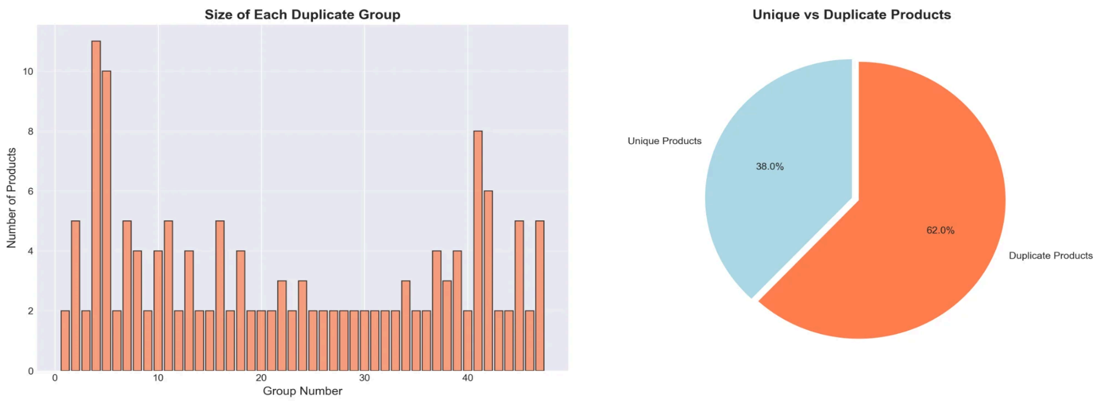
Similarity Score Distribution

The histogram shows that most product pairs have low similarity (< 0.3), with a small subset exceeding the 0.85 threshold.



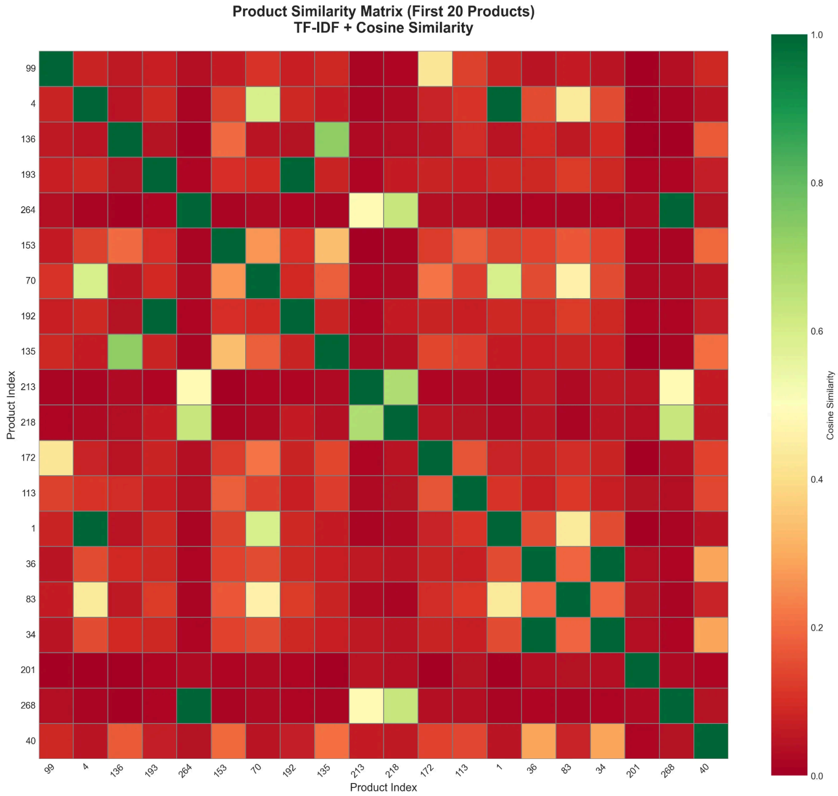
Duplicate Groups Statistics

62% of products were identified as duplicates across 48 distinct groups. Most groups contain 2-5 products, with some large clusters of up to 11 items.



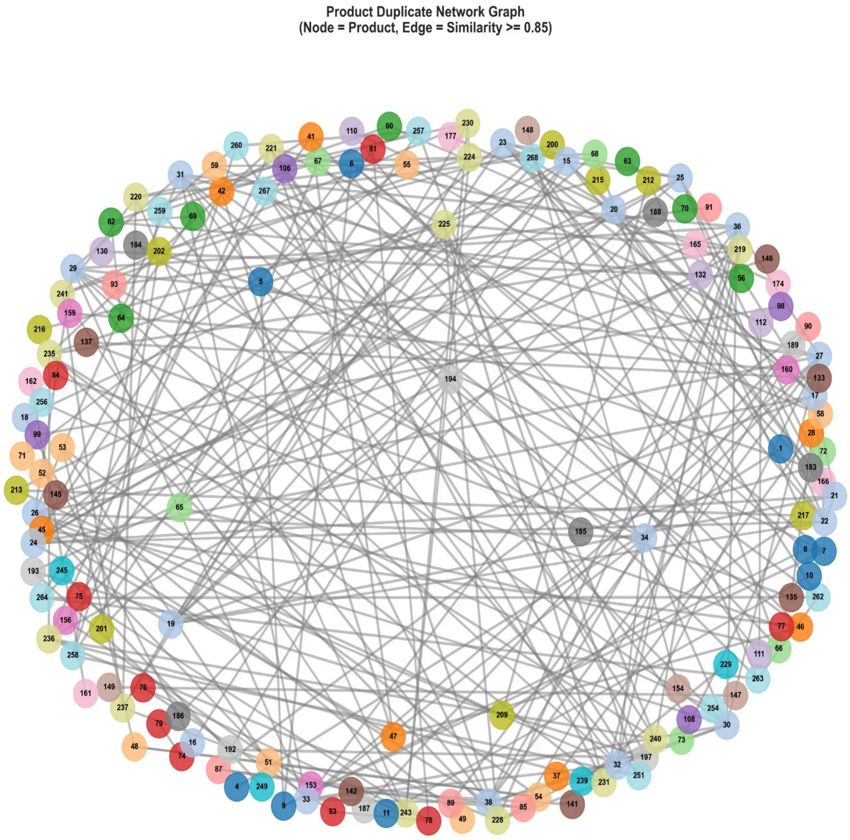
Similarity Matrix Heatmap

The heatmap visualizes pairwise similarities for a sample of 20 products. Green indicates high similarity (potential duplicates), while red indicates distinct products.



Product Duplicate Network Graph

Network visualization where nodes represent products and edges connect items with similarity ≥ 0.85. Connected components form duplicate groups.



Key Findings

Statistics Summary

Metric	Value
Total Products	250
Unique Products	95 (38%)
Duplicate Products	155 (62%)
Duplicate Groups	48
Pairs Above Threshold (Before Filter)	331
Pairs After Numeric Filter	250
False Positive Reduction	**24.5%**

Identified Patterns

1. **Exact Duplicates:** 104 products with identical titles
2. **Near-Duplicates:** Minor variations in spacing, Turkish characters, or word order
3. **Brand Clusters:** Karcher (multiple SC models), Kingston/Samsung RAM products
4. **Largest Group:** 11 products (Hi-Level DDR3 RAM variations)

Technical Highlights

Why This Approach?

Component	Choice	Rationale
Vectorization	TF-IDF + Char N-grams	Handles typos, abbreviations
Similarity	Cosine Similarity	Scale-invariant, efficient
Grouping	NetworkX Connected Comp.	Production-ready, O(V+E)
Variant Control	Number Extraction + Matching	Prevents 64GB vs 128GB errors

Threshold Selection

The 0.85 threshold was selected through empirical testing:

- **0.80:** Too many false positives
- **0.85:** **Optimal** — best precision/recall balance
- **0.90+:** Misses true duplicates (low recall)

Business Value & Recommendations

Impact Areas

- **Inventory Management:** Merge duplicate listings to avoid confusion
- **Search Quality:** Improve product search by consolidating variants
- **Data Quality:** Clean up product database
- **Customer Experience:** Prevent duplicate listings
- **Analytics:** More accurate sales and inventory metrics

Production Recommendations

1. **Automated Pipeline:** Deploy as a regular job to detect new duplicates
2. **Human-in-the-Loop:** High confidence (>0.95) → Auto-merge; Medium (0.85-0.95) → Human review
3. **Master Data Management:** Create canonical product IDs linking all variants
4. **Seller Guidelines:** Provide clear product naming guidelines

Scalability Considerations

Current approach: $O(n^2)$ — suitable for datasets up to ~10,000 products. For larger scale:

- Approximate Nearest Neighbors (Annoy, FAISS)
- Blocking strategies (group by brand/category first)
- Locality-Sensitive Hashing (LSH)

Conclusion

This project demonstrates a production-ready approach to product title deduplication that goes beyond simple ML solutions. The hybrid methodology combines the strengths of TF-IDF for semantic similarity with rule-based variant detection for precision.

Three Key Takeaways

- **Hybrid Approach:** ML + Domain Rules provides better results than either alone
- **Production Mindset:** Using established libraries (NetworkX, sklearn) over custom implementations
- **Data-Driven:** Threshold selection backed by empirical evidence

—— Thank You ——