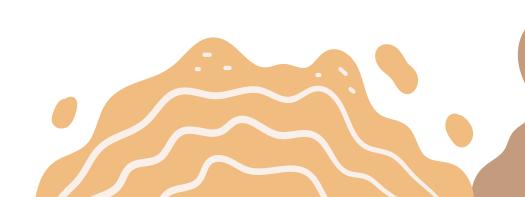






- Goal: Match SKU codes and correctly formatted names from the master file with userprovided product names.
- Process:
- 1. Extract & Preprocess Data: Normalize product names, handle variations.
- 2. Use Machine Learning Model: Apply the trained model to predict the best-matching SKU.
- 3. Assign Correct SKU: Return SKU with a similarity score and confidence level.



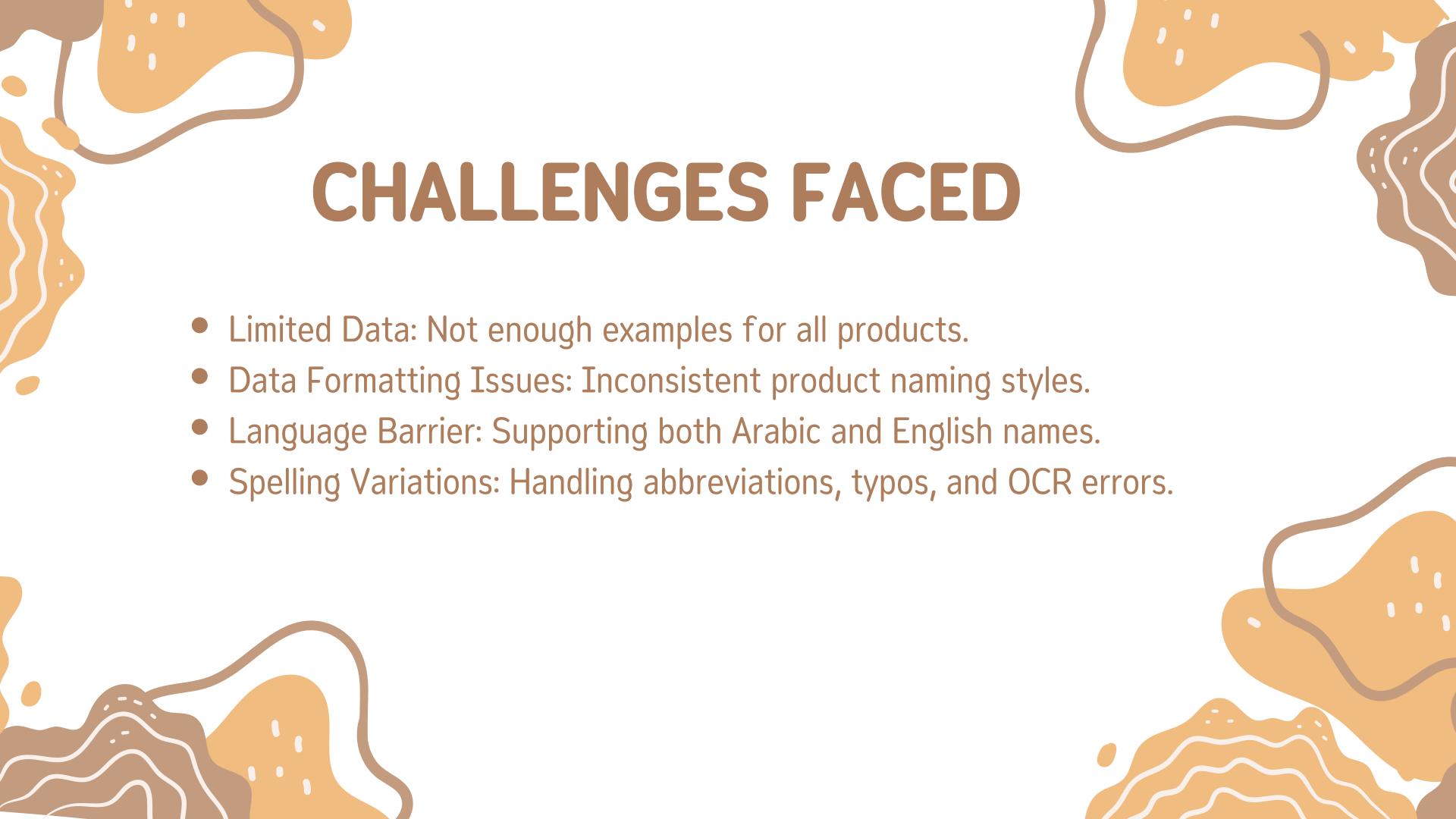


ACCEPTANCE CRITERIA & HOW THEY ARE MET

- Similarity Score: Between 0 and 1.
 - Achieved using TF-IDF vectorization and KNN-based distance calculations to quantify similarity.
- Robustness: Handles spelling mistakes, OCR errors (e.g., "Ibuprofn" \rightarrow "Ibuprofen").
 - Implemented text normalization, number-to-text conversion, and typo tolerance.
- Accuracy: At least 90% on validation data.
 - Model achieved 97% accuracy on test data and 94.3% mean accuracy in cross-validation.

ACCEPTANCE CRITERIA & HOW THEY ARE MET

- Efficiency:
 - Optimized for CPU execution.
 - Matching within ≤500ms per comparison.
 - Used precomputed TF-IDF vectors for fast lookups.
- Confidence Levels: High, Medium, or Low based on score.
 - Defined confidence thresholds based on predict_proba and similarity score, ensuring uncertain matches are flagged for review.



SOLUTIONS & APPROACHES TRIED



- Problem:
 - Too slow for large datasets.
 - Couldn't cover all variations in product names.
- 2. Machine Learning Approaches
 - Transfer Learning (Pretrained Models & Embeddings)
 - Issue: Overfitting due to the small dataset.
 - Random Forest & Similar Models
 - Issue: Required high computational power, but only CPU usage was allowed.
 - Word2Vec & Bag of Words
 - Issue: Did not guarantee high accuracy in matching.

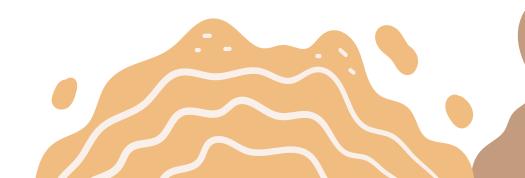


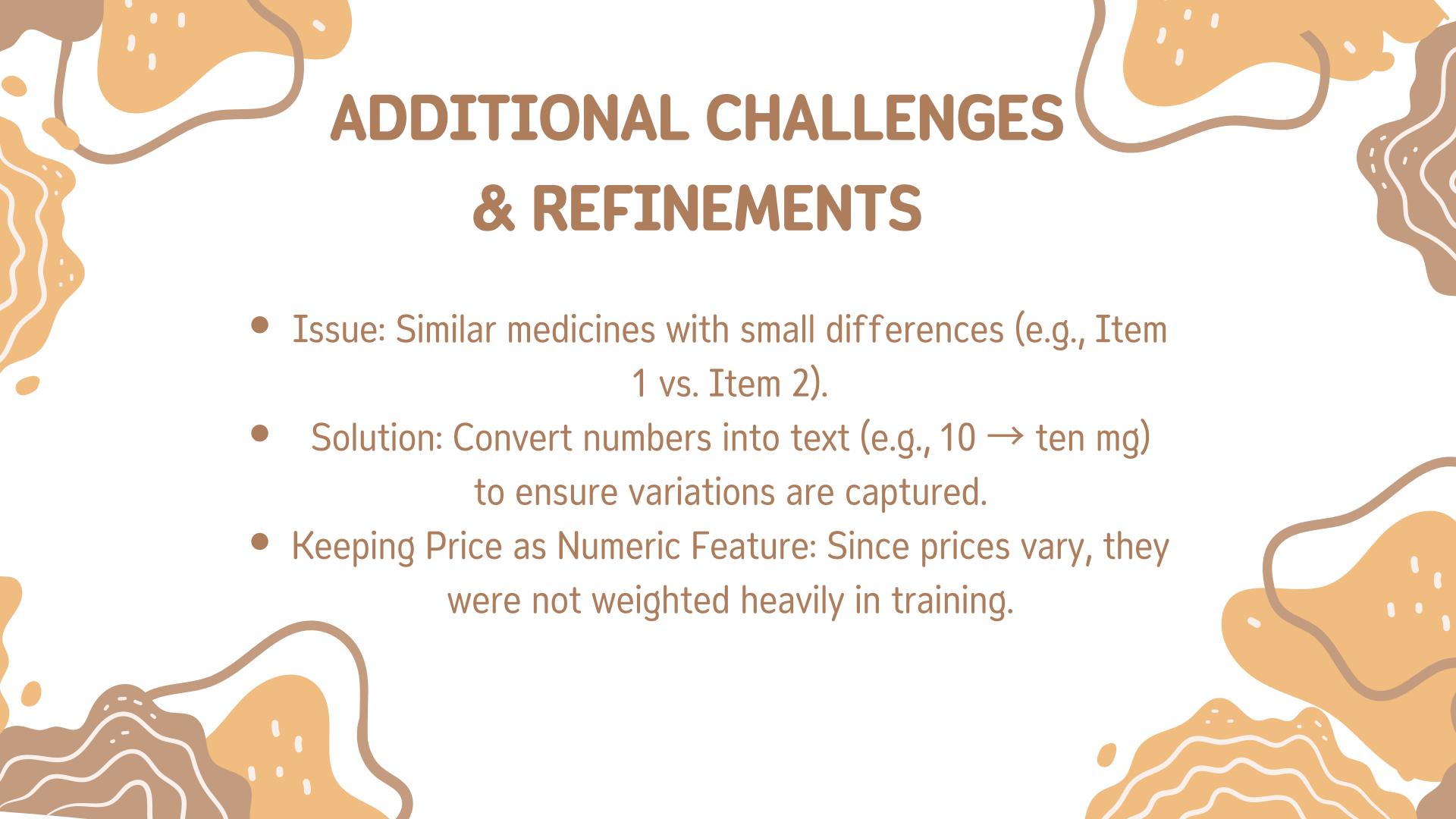
Why KNN?

- Finds the nearest match by minimizing distance.
- Naturally calculates similarity between product names.
- Handles new data easily without requiring retraining.

Challenge 1: Setting the Confidence Threshold

- Solution: Use minimum of predict_proba and similarity score
 - Ensures wrong predictions are labeled Low or Medium Confidence.
 - If High Confidence, the prediction is highly reliable.







Advantages:

- Training does not require much time or computational power.
- Naturally adapts to different product variations.

Limitations:

Confidence threshold may need periodic tuning.





- Execution Speed: Processes 2000 items in ~31 seconds.
- Confidence Check: 8.8% of predictions require manual review.
- Optimizations:
 - Precompute TF-IDF vectors.
 - Use efficient nearest neighbor search for fast matching.

