CLICK2BUY – PREDICTIVE INTEREST RECOMMENDATION SYSTEM

- Melisa Maldonado Melenge
- Jean Pierre Mora Cepeda
- Juan Diego Martínez Beltrán
- Luis Felipe Suárez Sánchez

Universidad Distrital Francisco José de Caldas
School of Engineering

BESTBUY

Which product are you going to choose ?
How do you know which products to show each customer?

THE E-COMMERCE CHALLENGE

- Modern e-commerce catalogs are massive.
- Users rely on **Search** to navigate, not just browsing categories.
- **The Expectation**: Immediate, relevant results. If they don't find it, they leave.
- **The Core Difficulty**: Understanding user intent from short, often ambiguous text queries

WHY TRADITIONAL SEARCH FAILS

AMBIGUITY

Different users describe the same product differently (e.g., "cheap laptop" vs. "inexpensive notebook")

KEYWORD LIMITATIONS

Traditional keyword matching misses the context.

SPARSITY

Many products have very few historical clicks, making "Collaborative Filtering" (people who bought X also bought Y) difficult for new items or users.

CONSTRAINTS:

MUST BE FAST

MUST BE ACCURATE

Recommend relevant products

MUST SCALE

Handle thousands of queries per day

SENSITIVITY ANALYSIS

Understanding which inputs most affects the output

DIFFERENT QUERIES → **DIFFERENT PREDICTIONS**

DIFFERENT CATEGORY → **DIFFERENT MODEL USED**

OUR PROPOSAL – THE CLICK2BUY SYSTEM

- We proposed a **Machine Learning-based recommendation system.**
- **Goal**: Predict the Top 5 most relevant products (SKUs) for any given search query.

Approach: Supervised learning using historical interaction data (August–October 2011).

Key Differentiator: We did not use one giant model. We used **Category-Specific Models.**

DECODING USER INTENT

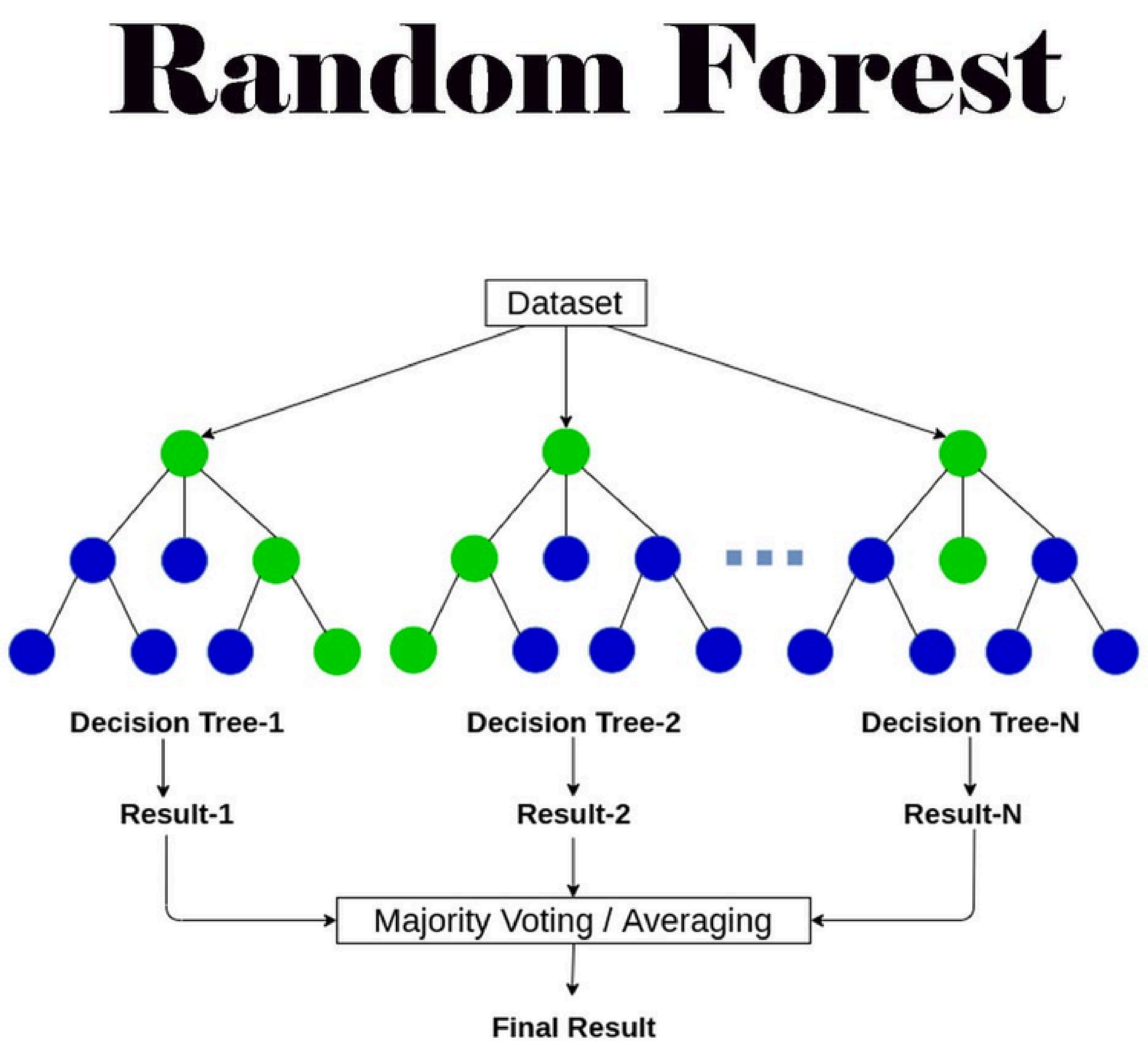
- **Cleaning the Noise:** We automatically standardize user inputs. For example, "running" and "run" are treated as the same concept to avoid confusion.
- **Finding Meaningful Phrases:** The system looks for pairs of words that belong together (like "Free Shipping" or "Hard Drive") rather than reading words in isolation.
- **Extracting Signals**: We convert the text into **84 distinct behavioral signals.** These signals tell the model not just what the user typed, but how specific or popular their request is within a category.

RANDOM FOREST & CATEGORY STRATEGY

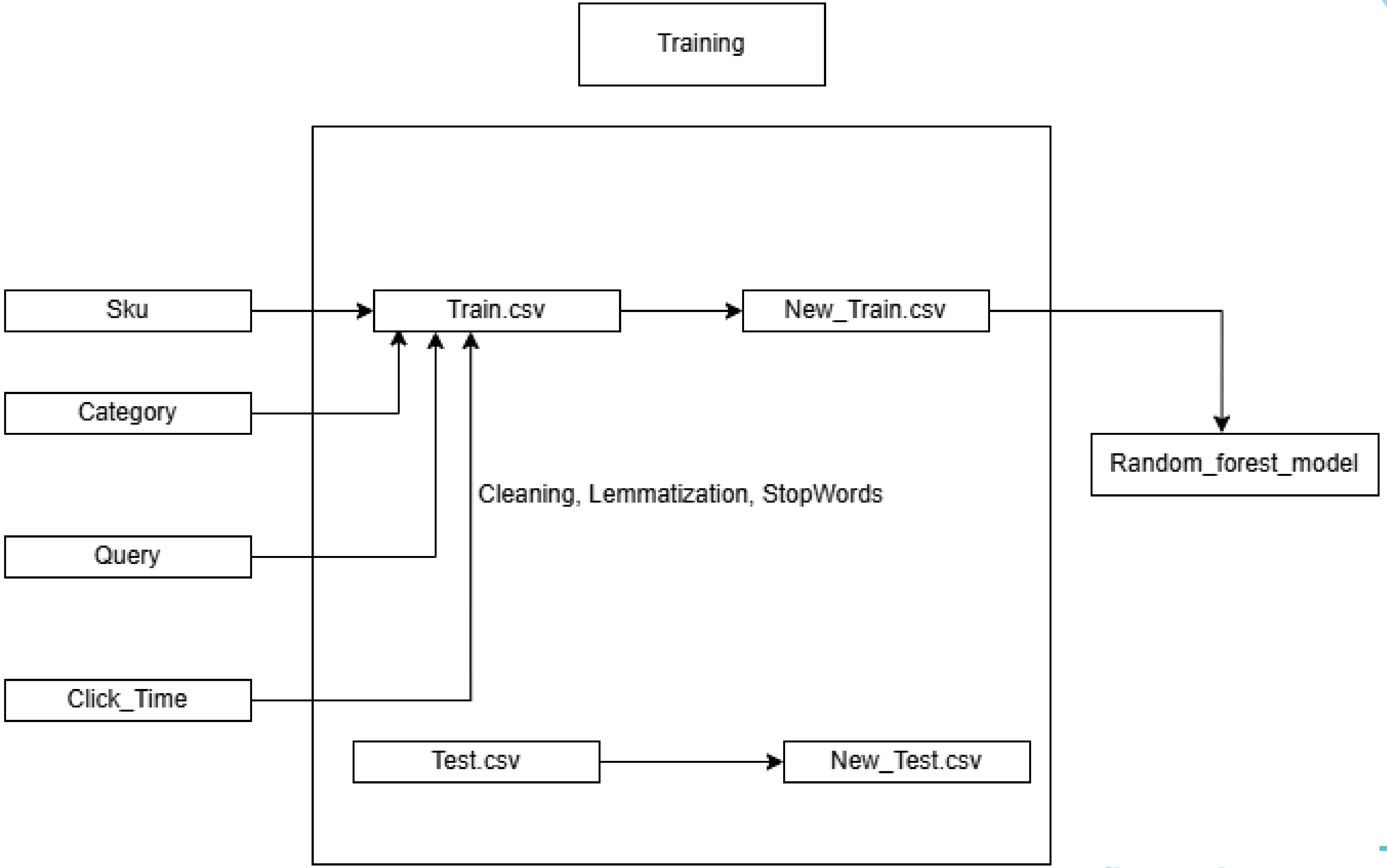
- **Algorithm**: We utilized **Random Forest Classifiers.**

Why? It handles noise well, avoids overfitting, and provides "feature importance" (interpretability).

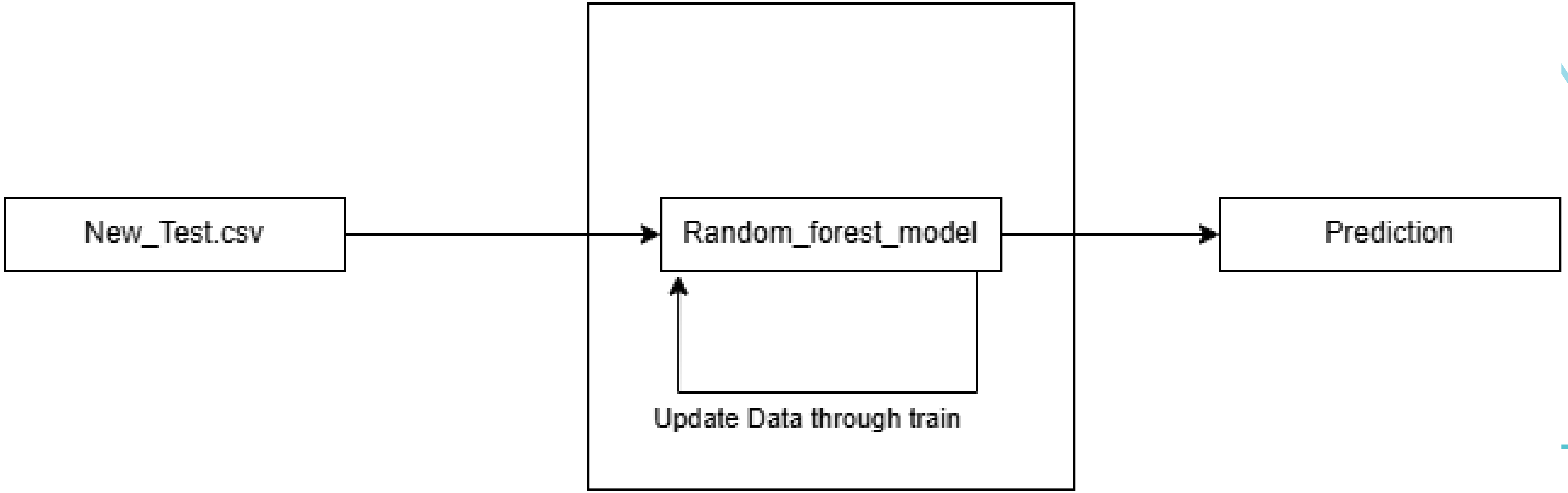
- **The Category Strategy**: Instead of one global model, we trained a separate model for each product category.
- **Reasoning**: The vocabulary for "Electronics" is completely different from "Apparel." Separating them improves accuracy



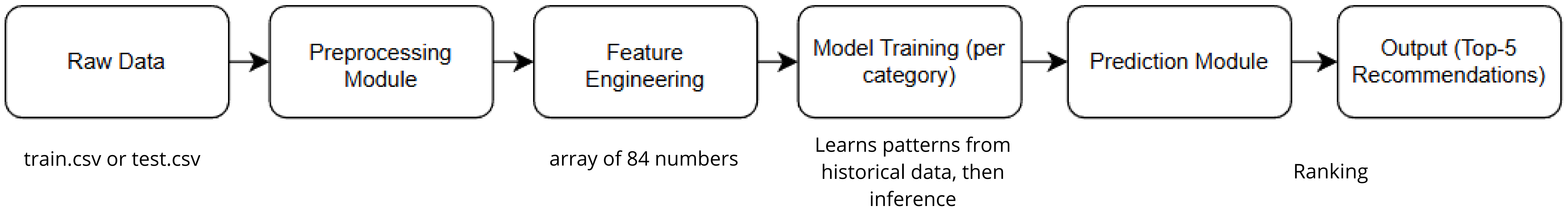
DATA PROCESSING



Prediction


New_Test.csv
Random_forest_model
Update Data through train
Prediction

System Architecture Overview



- **Preprocessing:** Cleans the user input.
- **Feature Extraction:** Calculates the 84 data points.
- **Inference:** The trained category model calculates probability scores for all products and ranks them, using patterns.
- **Ranking:** We sort by probability and present the top 5

WHAT DID WE FIND?

1598166 8669078 19498576 17240521 19498567

9132379 9699159 9124262 1283713 14536718

3108172 9755322 1534115 3108109 1535836

0

Modularity

Easy to understand, test, modify one part without breaking the others.

WHAT DID WE FIND?

- We analyzed what users actually type.
- **Top Terms**: "Free," "New," "Shipping" were the most frequent.
- **Category Dominance**: Electronics and Apparel were the most searched domains.
- **Efficiency**: We successfully reduced raw queries from 3.2 words to 2.8 meaningful tokens per search.

LIMITATIONS

- **Category Coverage**: We only trained models for categories with sufficient data (10+ samples). This meant niche categories (Long Tail) didn't get a model.
- **Cold Start:** The system struggles to recommend brand-new products that were not in the training data.
- **Semantic Depth**: While bigrams help, the model doesn't fully understand deep semantic synonyms (e.g., nuanced differences between "cheap" and "budget")

FUTURE WORK & IMPROVEMENTS

- **Embeddings:** Moving from word counts to "Word Embeddings" (Vectors) to capture semantic meaning better.
- **Deep Learning:** Exploring Neural Networks (Transformers) for complex query understanding.
- **A/B Testing:** Moving from historical data evaluation to live user testing to measure real engagement.

CONCLUSION

- We successfully built a pipeline from raw data to prediction without using "black box" deep learning, proving that Classical Machine Learning (Random Forest) is still highly effective.
- The "Category-Specific" approach was the key to handling the diversity of a massive catalog.
- The system provides a strong foundation for real-time personalization in e-commerce.

THANK YOU