Tuxsa Master Degree

Business Digital Transformation (Data Science)

Kanyapak Kaewkeaw

Text Analytics Final Project

Dataset pre-processing techniques and learning algorithms

**Introduction**

On e-commerce websites, organizing products into the correct categories is essential in helping users find what they need quickly. This project studies a text classification dataset containing product descriptions from an e-commerce platform. The dataset focuses on four main categories: Electronics, Household, Books, and Clothing & Accessories, which together represent around 80% of typical e-commerce products.

The dataset is in .csv format and contains two columns: the first column shows the category name (class label), and the second column contains the product name along with its description. These descriptions are used as input for building machine learning models. Sampling process was used to extract a smaller yet representative subset of the original dataset

The aim of this study is to apply classification algorithms such as *k-Nearest Neighbors (k-NN)* and *Naïve Baye*s to automatically classify each product description into the correct category. *Clustering techniques* was also applied to identify patterns and similarities between different types of product descriptions. This approach can help improve the automatic tagging and organization of items on e-commerce websites.

This paper is divided into three sections. The first section provides an overview of the dataset and explores the data. The second section focuses on building a classification model to predict category distribution. The final section discusses the use of clustering algorithms to group similar categories.

**Dataset Overview**

The original dataset titled “Ecommerce Text Classification” was sourced from Kaggle and contains:

50,400 rows (product descriptions)

2 columns:

Category: The product category label Category distribution

Household: 38%, Books: 23% Other: 38%

Text: A descriptive sentence or paragraph about the product

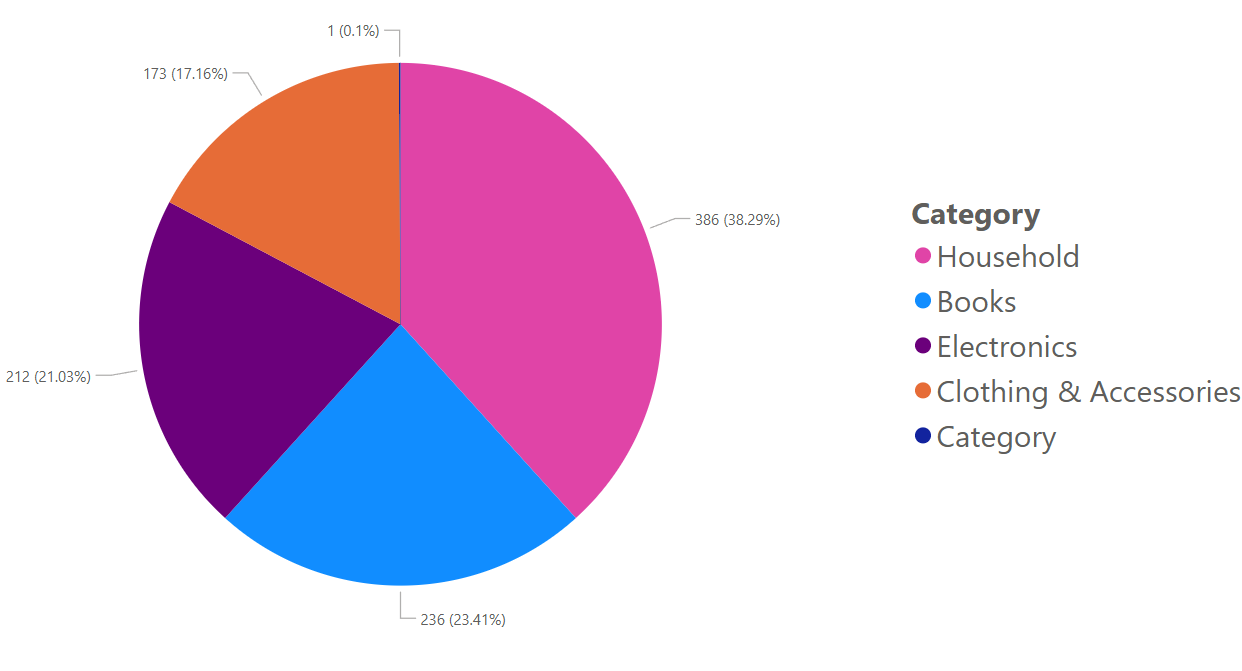
Sampling process was used to extract a smaller yet representative subset of the original dataset, preserving the original distribution of product categories. This smaller dataset (approximately 2% of the original size) is intended for efficient training, testing, and experimentation during the development of a text classification model.

**Sampling Steps in Excel**

1. Loaded the dataset into Excel
2. Inserted a Random column using =RAND() to generate random numbers for each row.
3. Calculated sample sizes per category based on 2% of the total:
4. Household ≈ 383 rows
5. Books ≈ 232 rows
6. Other ≈ 383 rows
7. Filtered each category, sorted by the Random column, and selected the top N rows.
8. Copied samples into a new sheet and combined all into a final 1008-row dataset.

**Data Exploration**

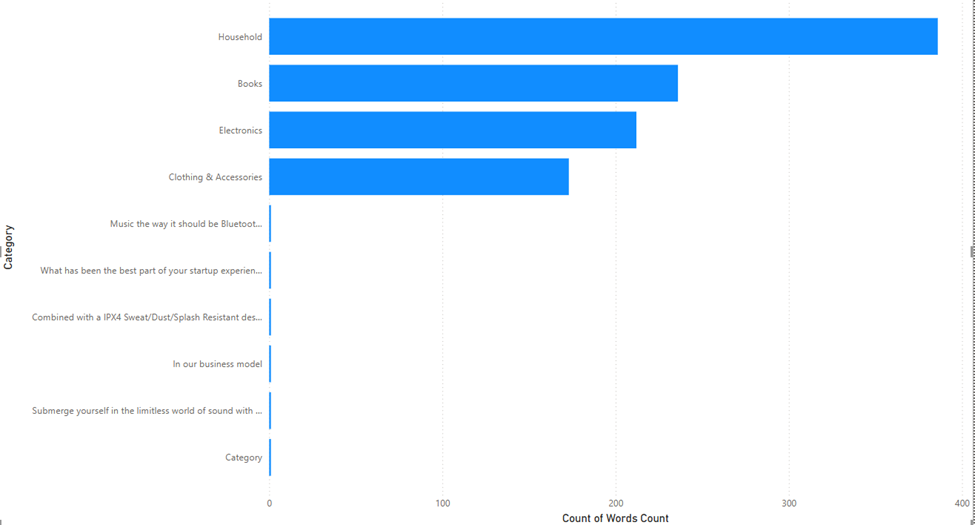
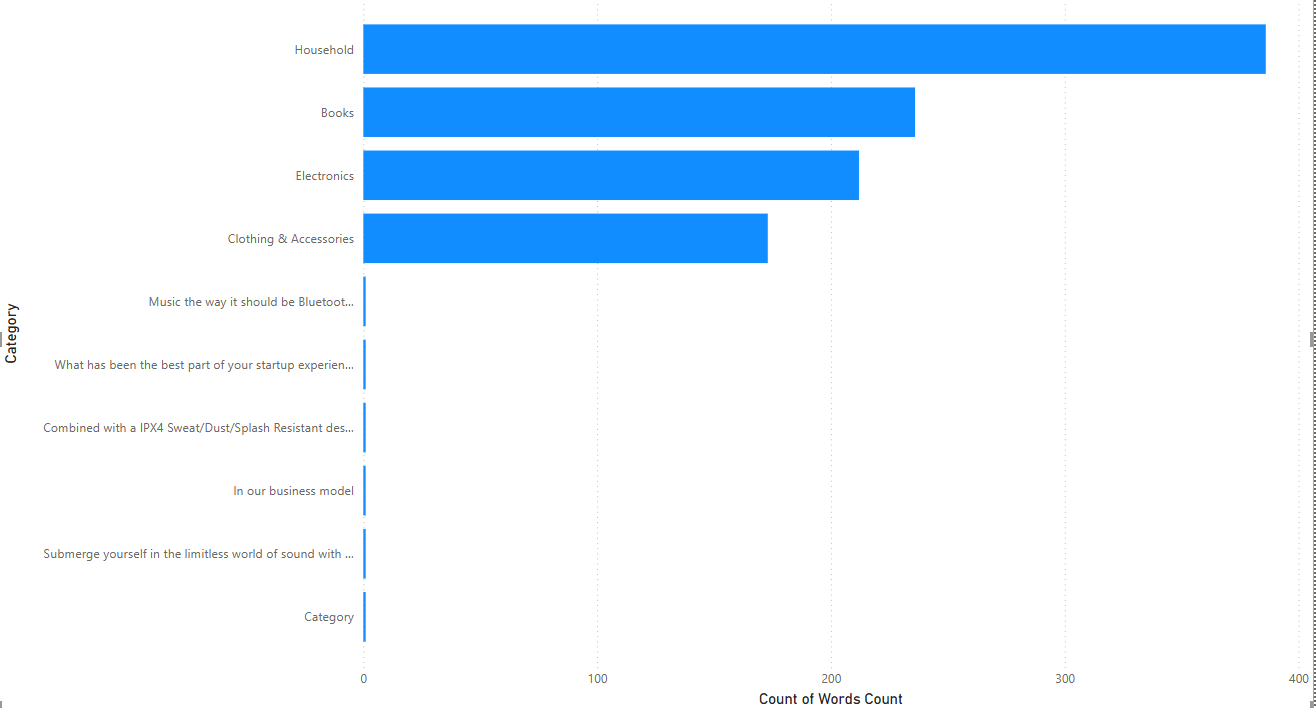
**Figure 1 Data Set Pie Chart**

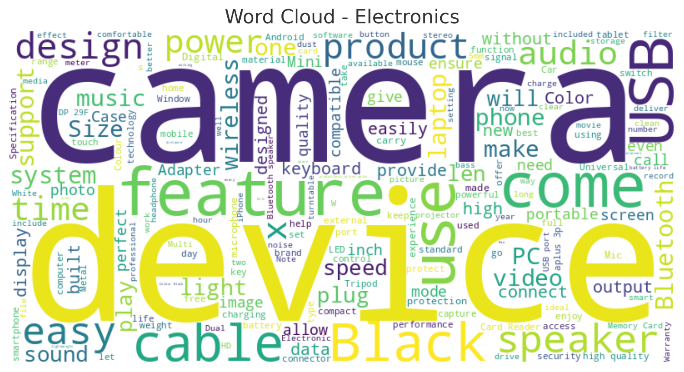
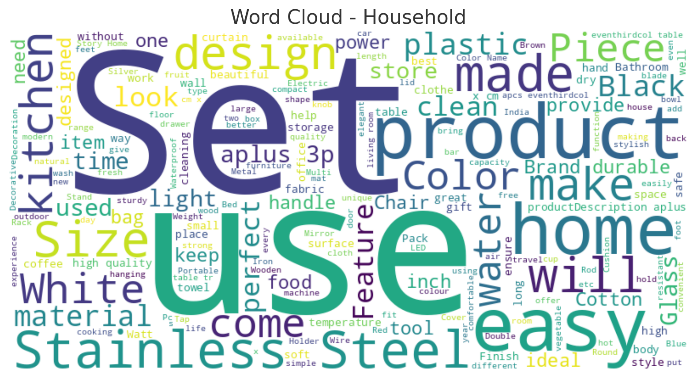
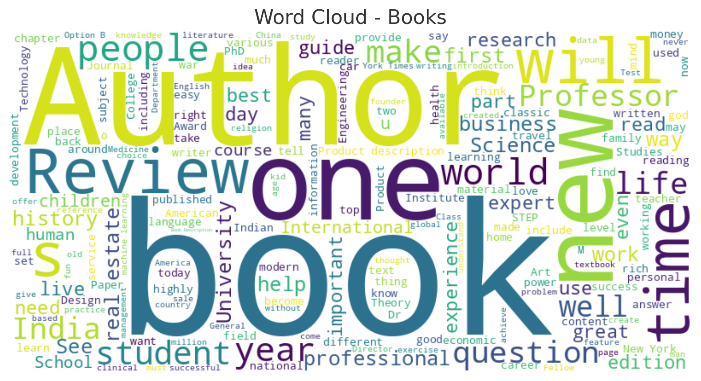


**Figure 2 Bar Chart Count of Category**

The dataset was then explored using Tableau, as illustrated in Figures 1 and 2. From the visualizations, the majority of product descriptions belong to the Household (38.3%) following by Books category (23.4%) , Electronic (21%) and clothing/accessories (17.3%).

**Figure 3 Count of word counts by categories**



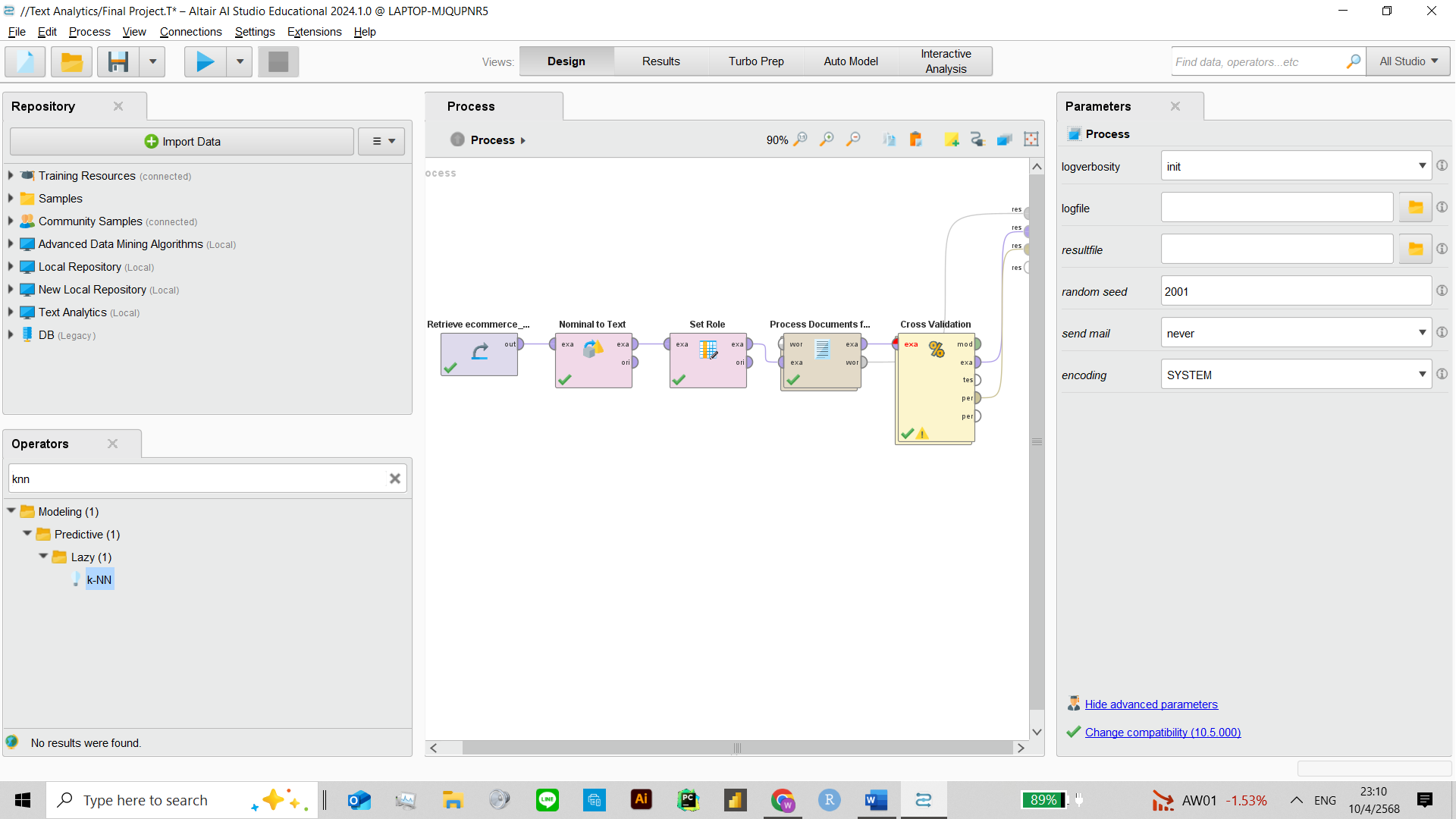
When analyzing word count across categories, product descriptions in the Household category tend to be longer, averaging around 140 words, while descriptions in the Books category are generally shorter, with an average of approximately 120 words.

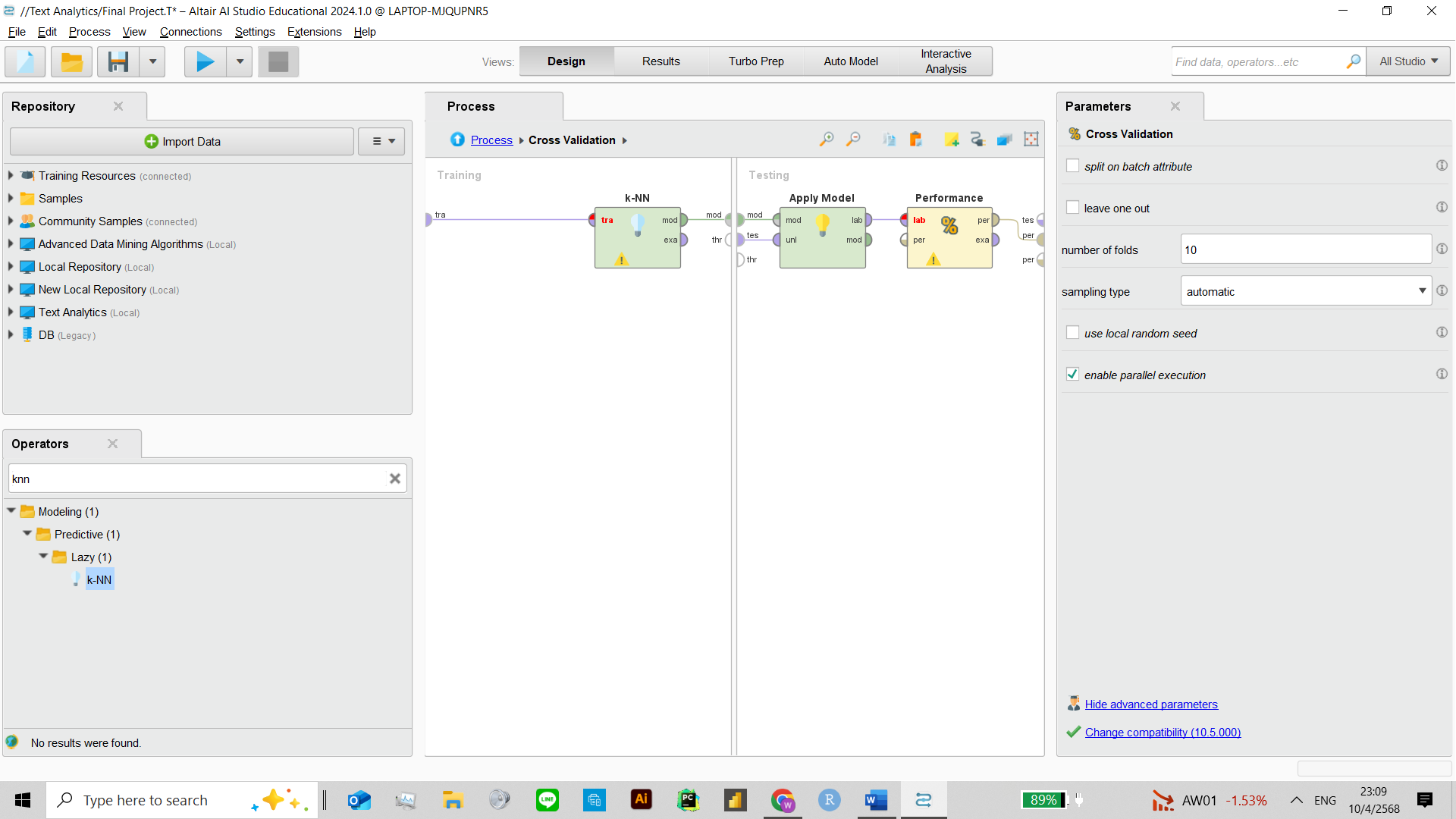
**Figure 4 Word Cloud of Product Descriptions**

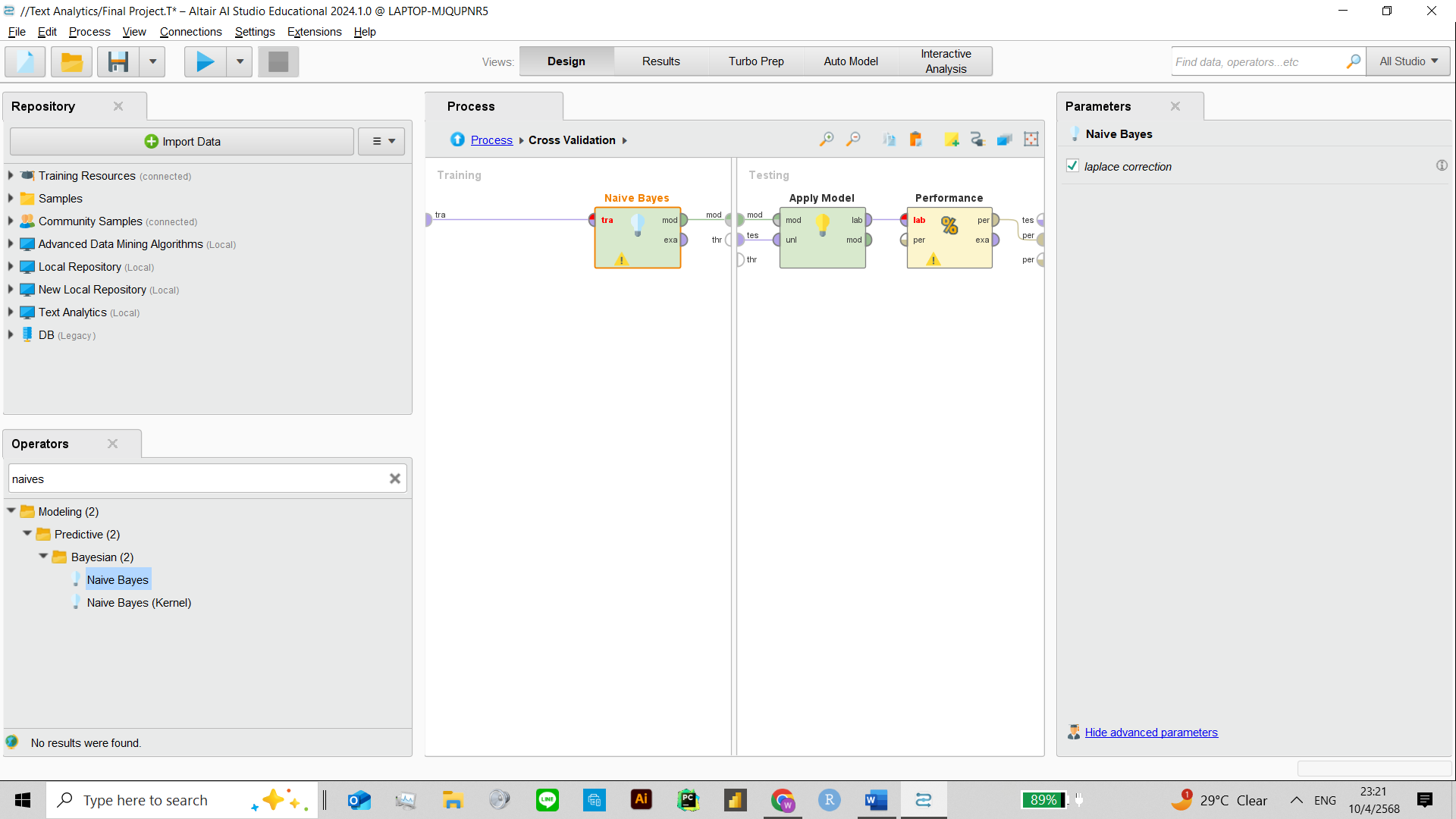
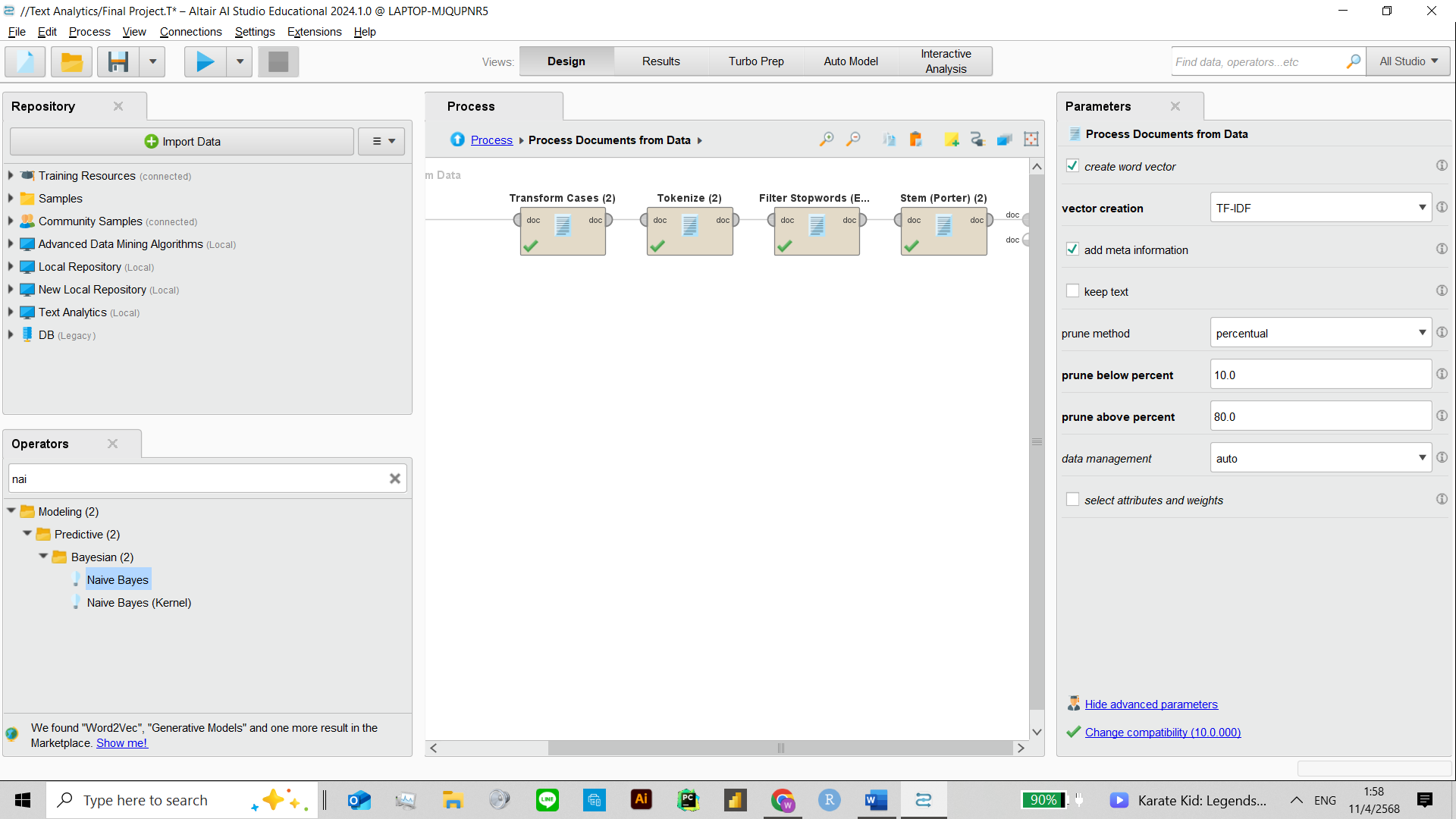
Figure 4 displays word clouds for different product categories. Positive product-related keywords such as “quality,” “durable,” “recommended,” and “stylish” are frequently seen in categories like Electronics and Clothing & Accessories. On the other hand, categories such as Household tend to include more functional terms like “cleaning,” “storage,” and “maintenance.”

Interestingly, specific product terms stand out more in certain categories. For example, Electronics prominently features words like “battery,” “wireless,” and “charger,” while Books often show terms such as “novel,” “author,” and “edition.” This variation in keyword prominence may reflect the priorities or expectations of customers in each category and can be useful for tuning classification models.

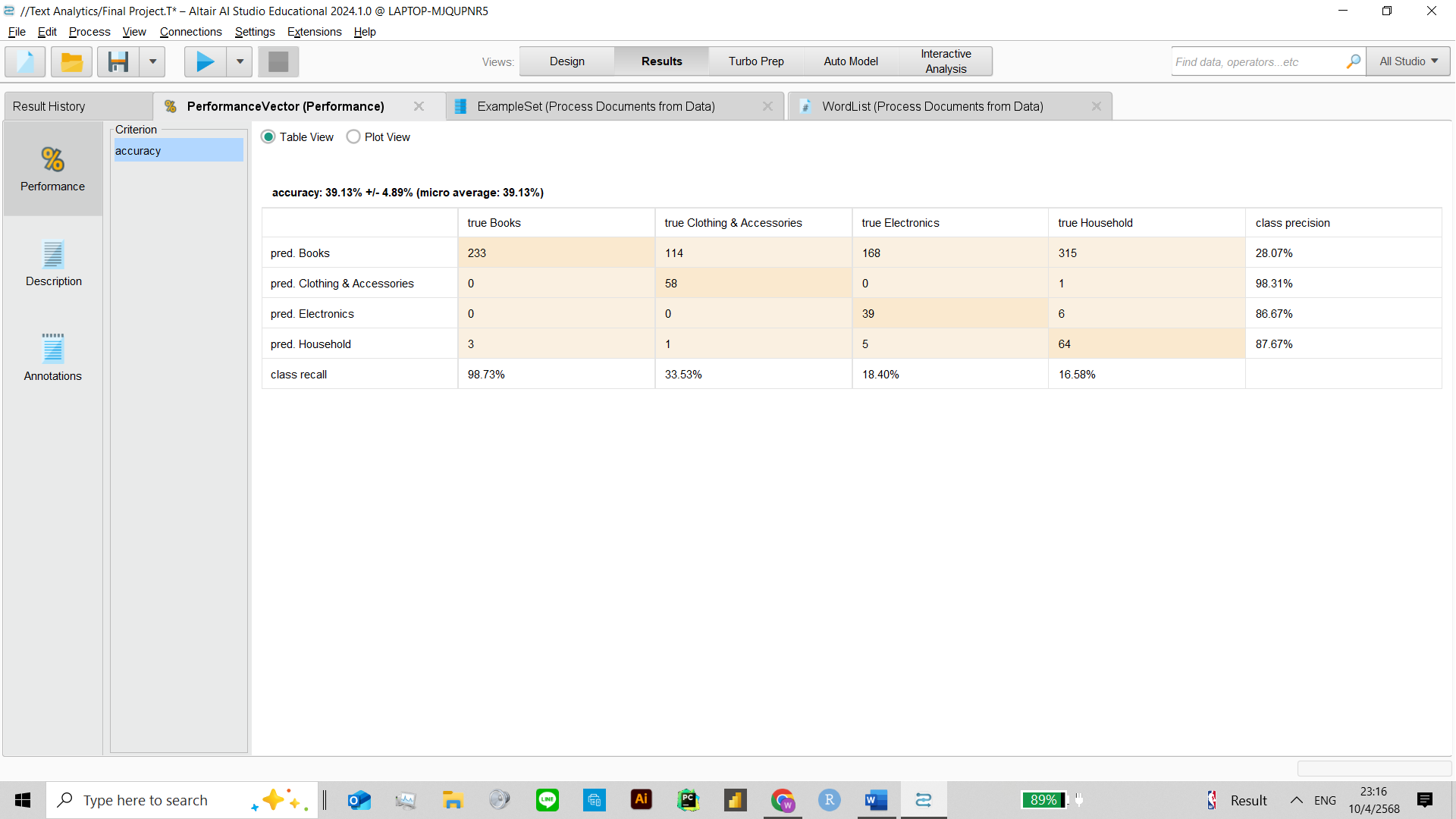
**Initial Exploration**

 A preliminary classification model was built using RapidMiner. The input data was preprocessed with standard text mining techniques, including case transformation (lowercase), tokenization, stopword removal, stemming, and pruning (keeping only terms with tf-idf scores between 3% and 30%). 2 algortihms were used 1) 5-class k-NN model (k=5) 2) Naive Bayes.



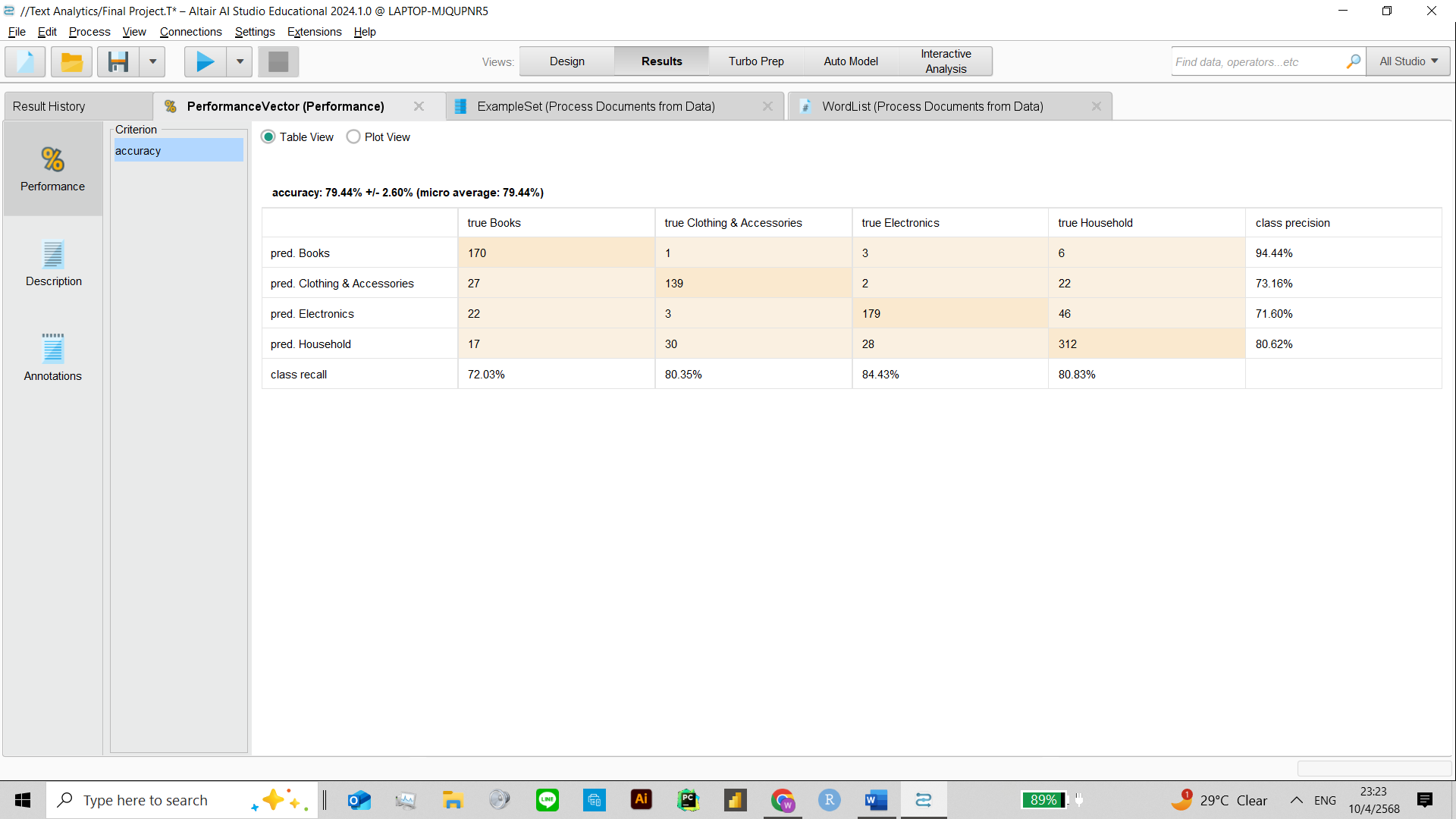


**Figure 5 Rapid Miner Process : Top K-NN, Bottom Naïve Bayes**



**Figure 6 K-** **NN k=5 prune 3-30 default**

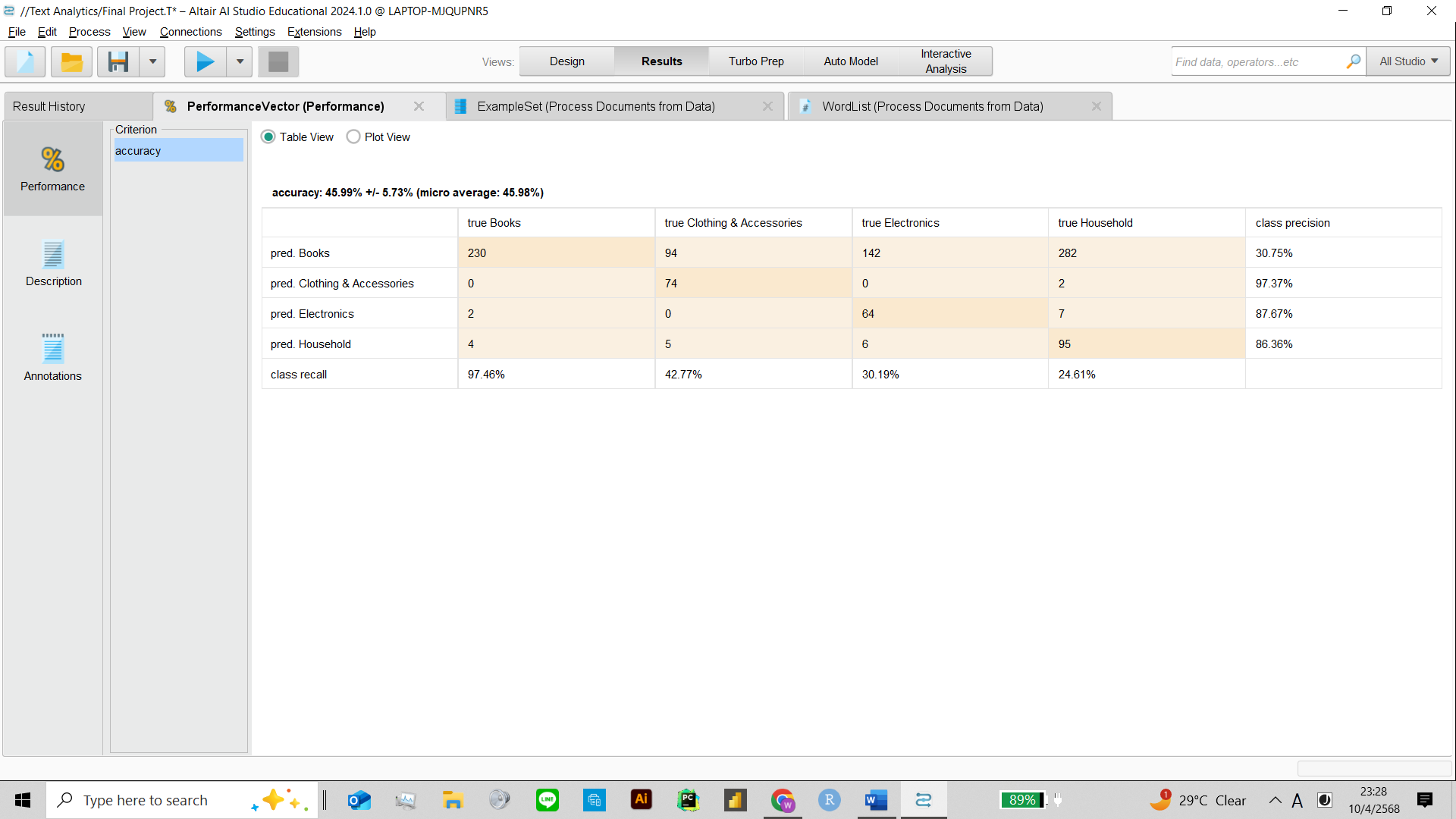
Figure 6 illustrates the initial model’s performance. The model achieved an accuracy of 39.13% ± 4.89%. The Books category had high recall (98.73%) but low precision (28.07%), indicating over-prediction. Household performed best in precision (87.67%) but had limited recall. Clothing & Accessories and Electronics showed weaker performance overall.

These results indicate that while the model identified some categories effectively, further improvements are needed for more balanced classification.These results suggested that while the initial preprocessing pipeline was effective for certain categories, further refinement was necessary to achieve more balanced classification across all product types.

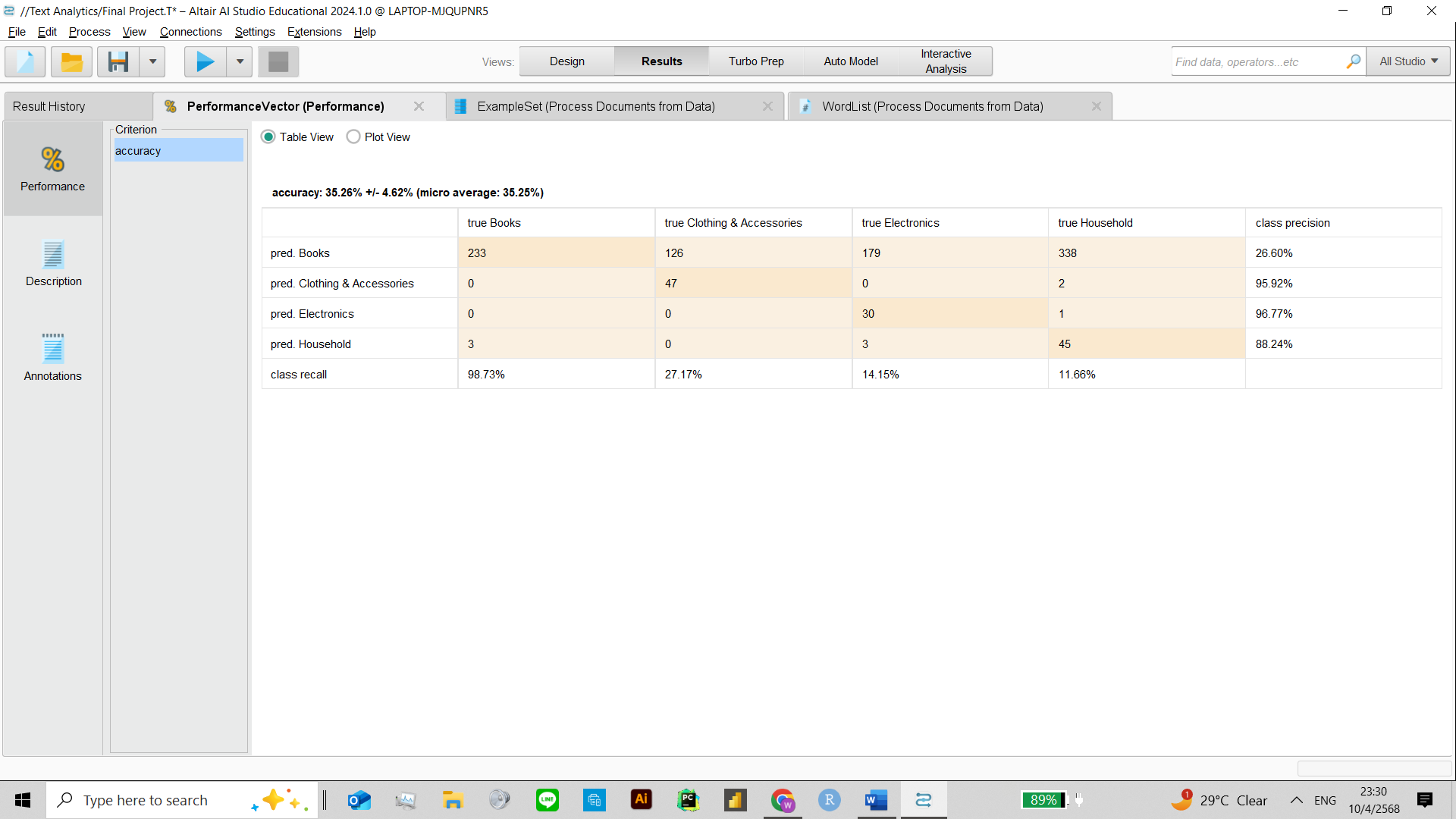
**Figure 7 Naïve Bayes prune 3-30 default**

In the second iteration, the classification model was switched to Naïve Bayes, keeping the same preprocessing setup: TF-IDF vectorization with 3–30% pruning. The model significantly improved, achieving an overall accuracy of 79.44% ± 2.60%. All four categories showed strong results: Electronics had the highest recall (84.43%) with solid precision (71.60%), while Clothing & Accessories and Household both showed balanced performance (around 80% for both metrics). Books achieved the highest precision (94.44%) with a recall of 72.03%.

Compared to the previous k-NN model, Naïve Bayes delivered a more balanced and accurate classification across all product categories, proving more effective for this dataset and feature setup.



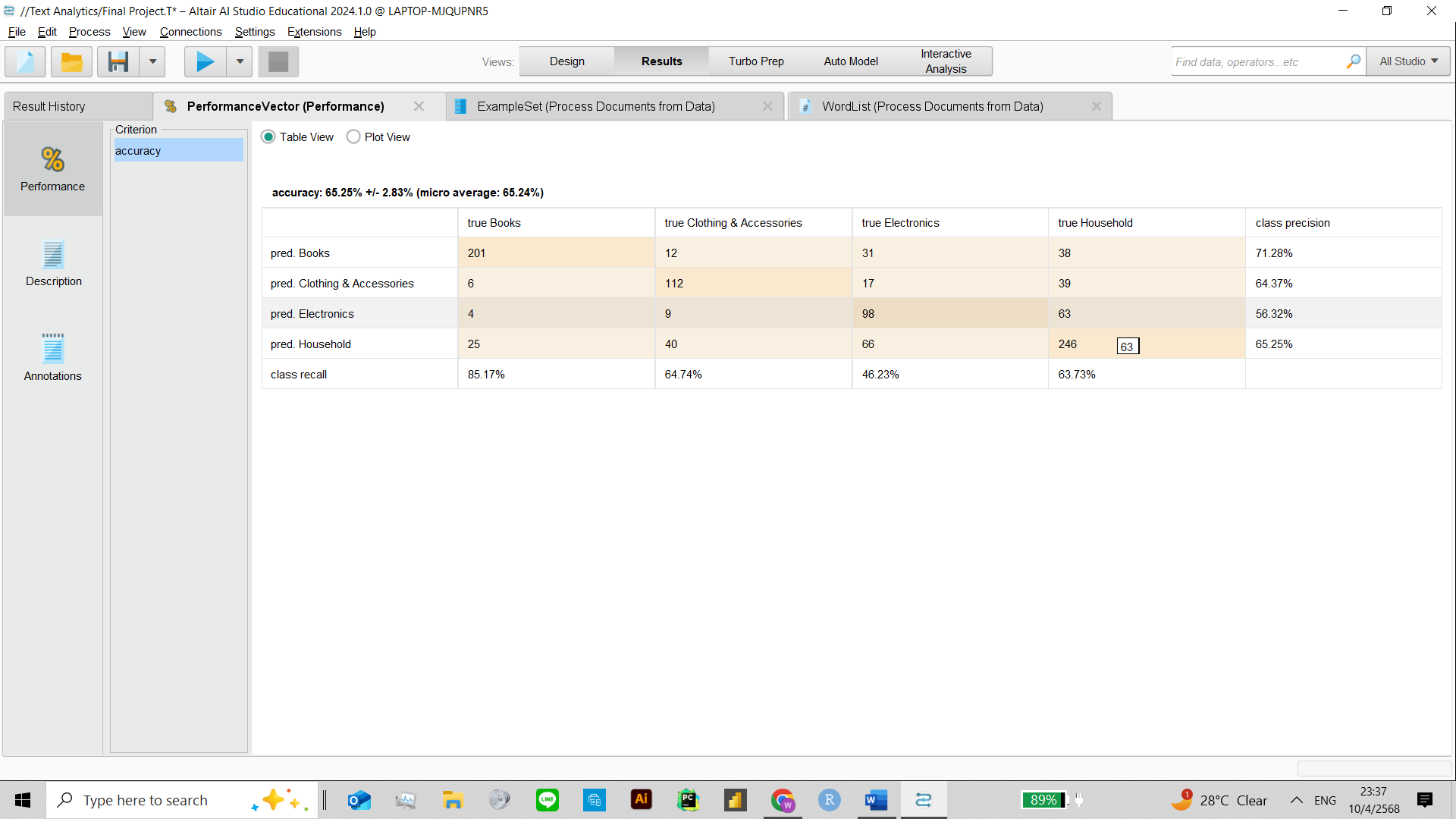
**Figure 8 K-NN , K=3 , prune**  **3-30**

**Figure 9 K-NN , K=7 , prune**  **3-30**

**Model Comparison – k-NN with Varying k**

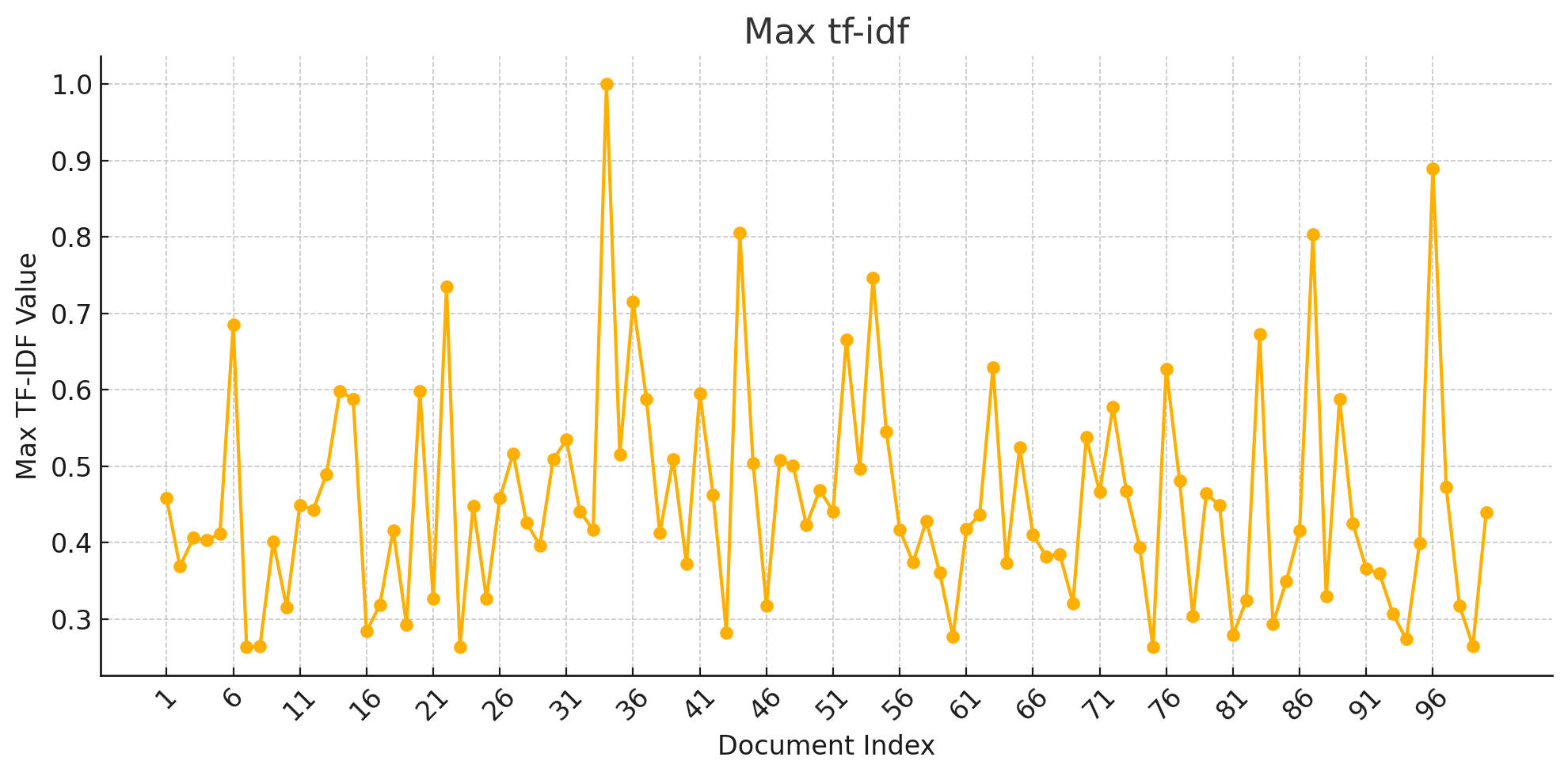
To evaluate the impact of different k values in k-NN, two additional tests were conducted using k=3 and k=7, with the same TF-IDF vectorization and 3–30% pruning settings. With k=3, the model reached 49.59% accuracy, showing improved balance compared to the initial k=5 setup. Notably, Books and Household achieved high precision (97.37% and 86.30% respectively).With k=7, accuracy dropped slightly to 38.95%, and category balance worsened.

These results suggest that smaller k values like 3 may offer better balance for this dataset, while larger k (7) leads to over-smoothing and poorer classification in minority classes.



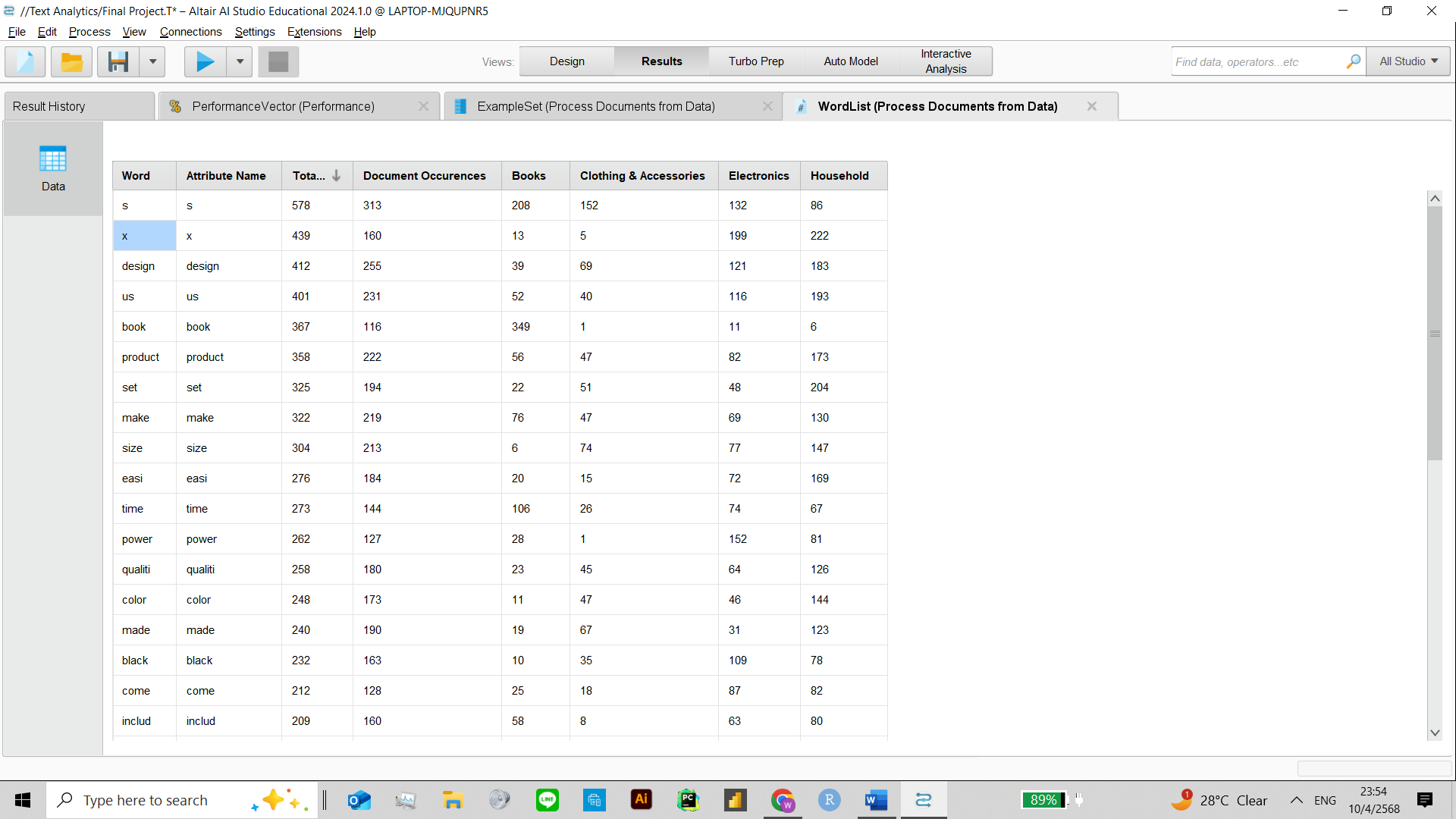
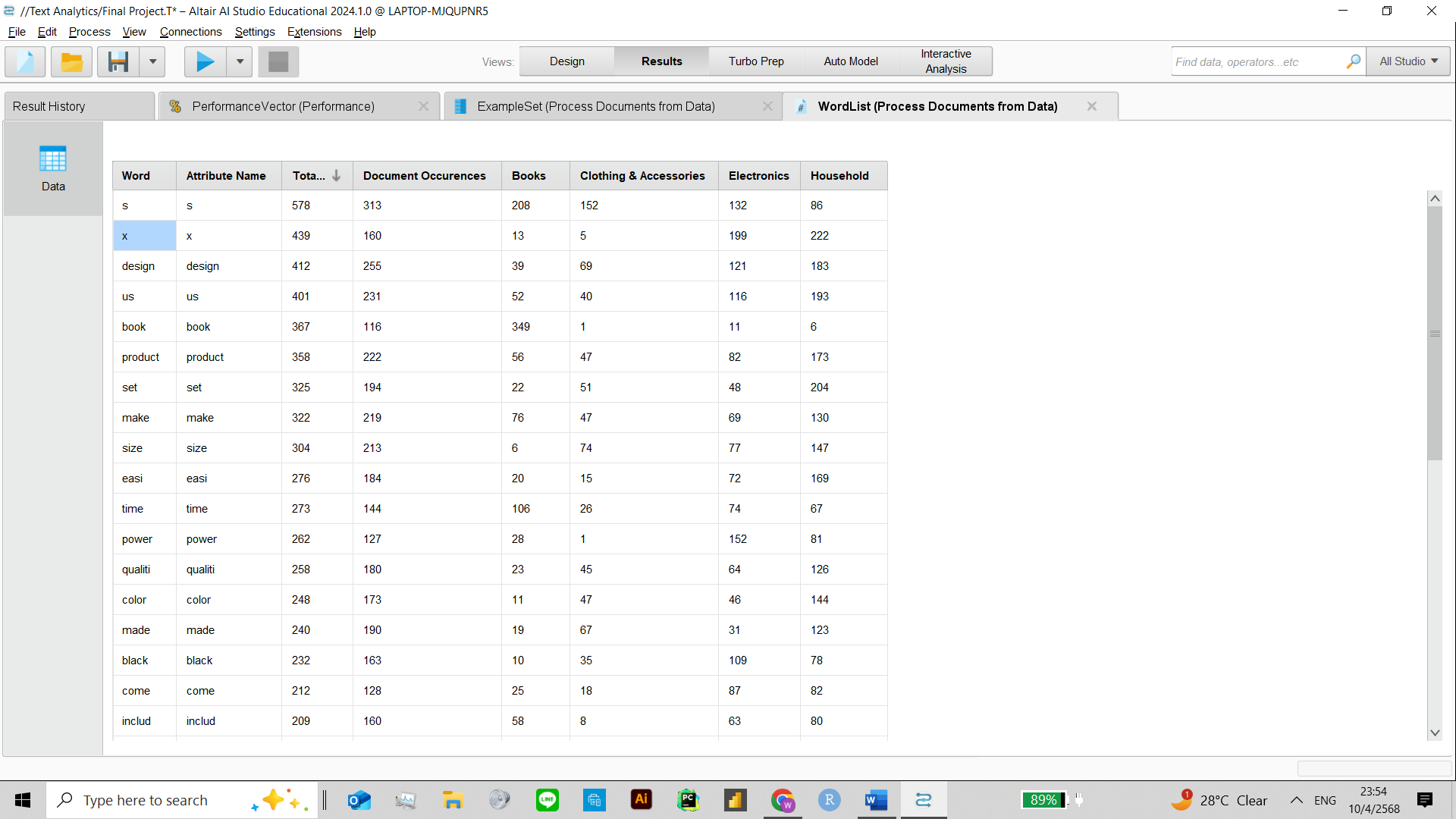
**Figure 9 K-NN , K=7 , prune**  **10-80**

**Results of Initial Exploration**

Several combinations of k values and pruning ranges were tested using the k-NN algorithm. The best result was achieved with k=7 and 10–80% pruning, reaching an accuracy of 66.25% ± 2.83%. This setup provided more balanced precision and recall across all categories — for example, Books had 71.28% precision and 85.17% recall, while other categories like Household, Clothing & Accessories, and Electronics also showed improved consistency. This combination proved to be the most effective configuration during this phase of the experiment.

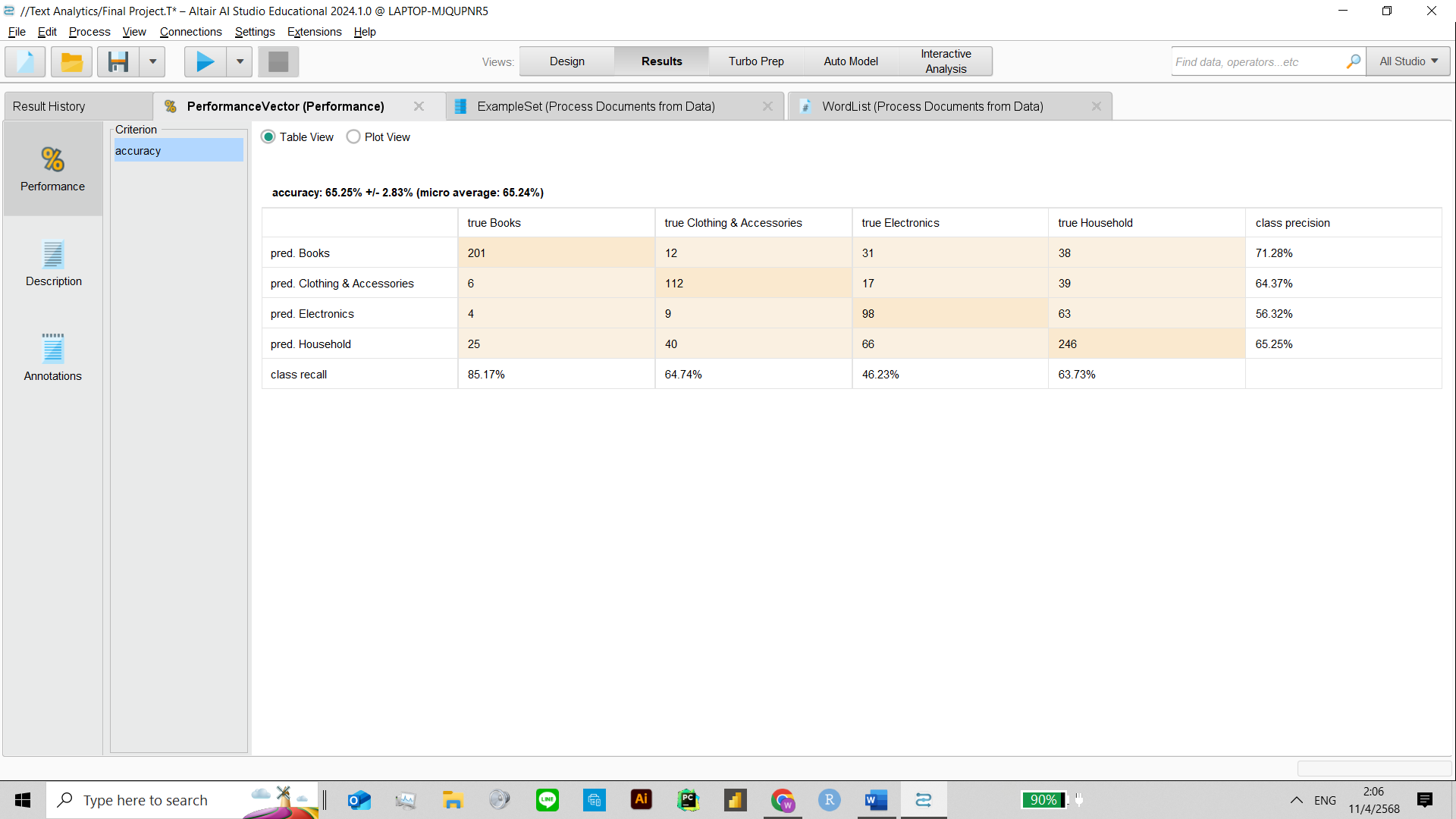
**Figure 10 Document Vector and Indexing**

**Figure 11 Most Frequent word**

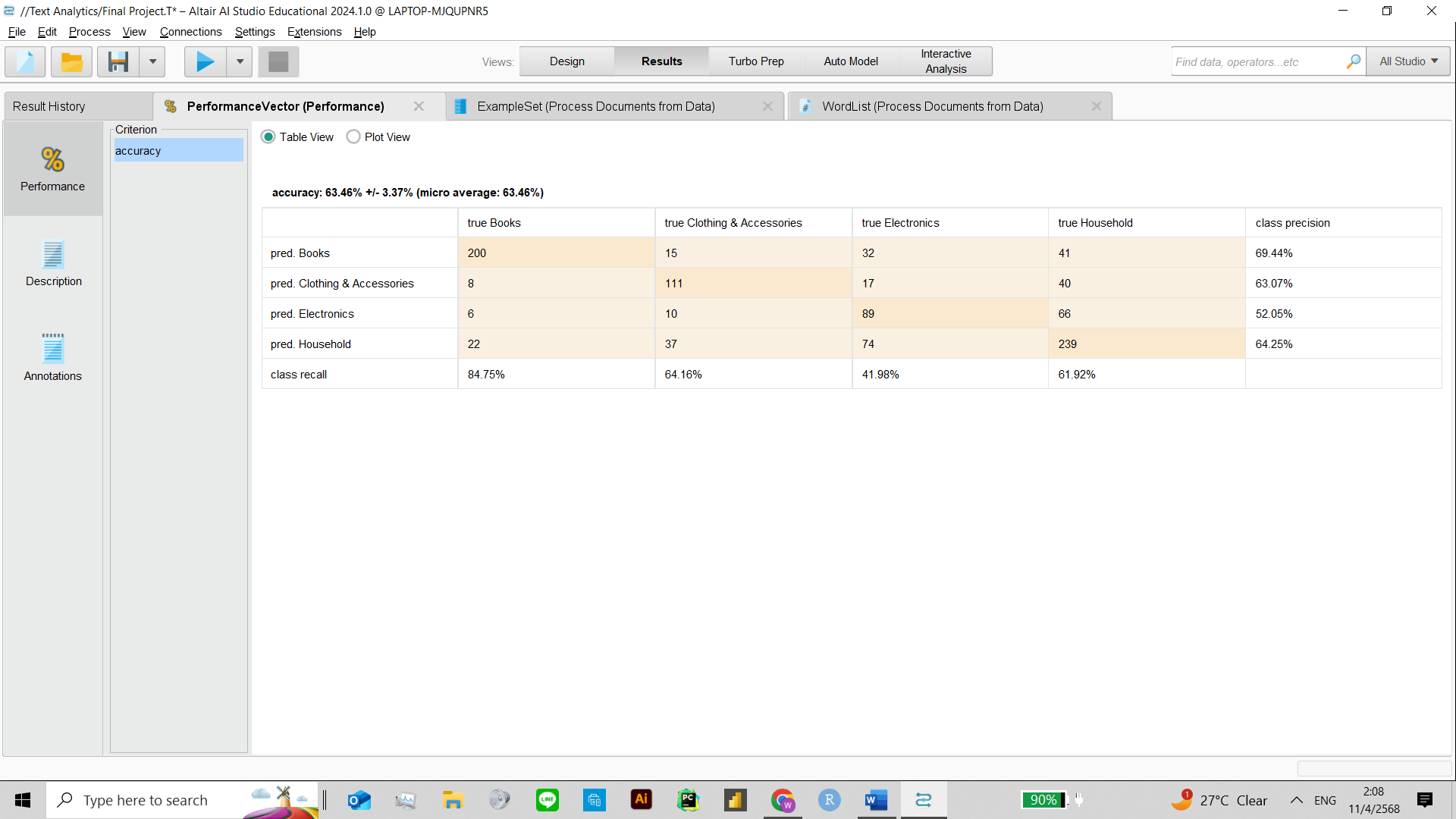


During the initial exploration, word frequencies across product categories were analyzed. Words like “design,” “product,” “quality,” and “set” appeared frequently across multiple categories, indicating their general relevance in e-commerce descriptions. For example, “design” occurred 412 times, especially in Electronics and Household, while “quality” was most frequent in Household.

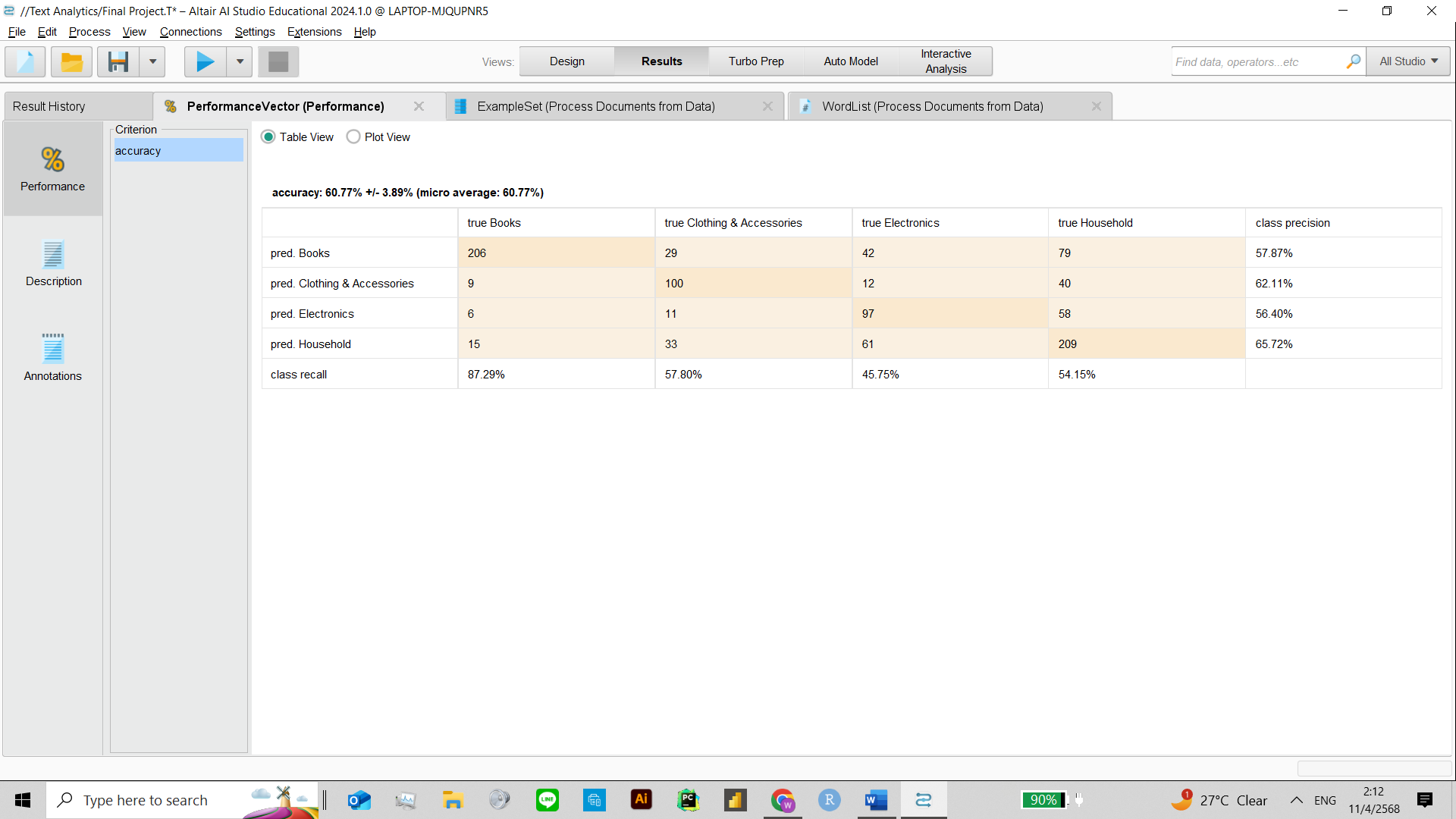
Some terms showed clear category-specific trends. The word “book” was dominant in the Books category (349 times) but nearly absent elsewhere, making it a strong classification feature. Similarly, “power” and “black” were more prominent in Electronics, while “set,” “make,” and “include” were frequent in Household items.

**2nd Iteration (Change Vector Creation Model) - KNN**

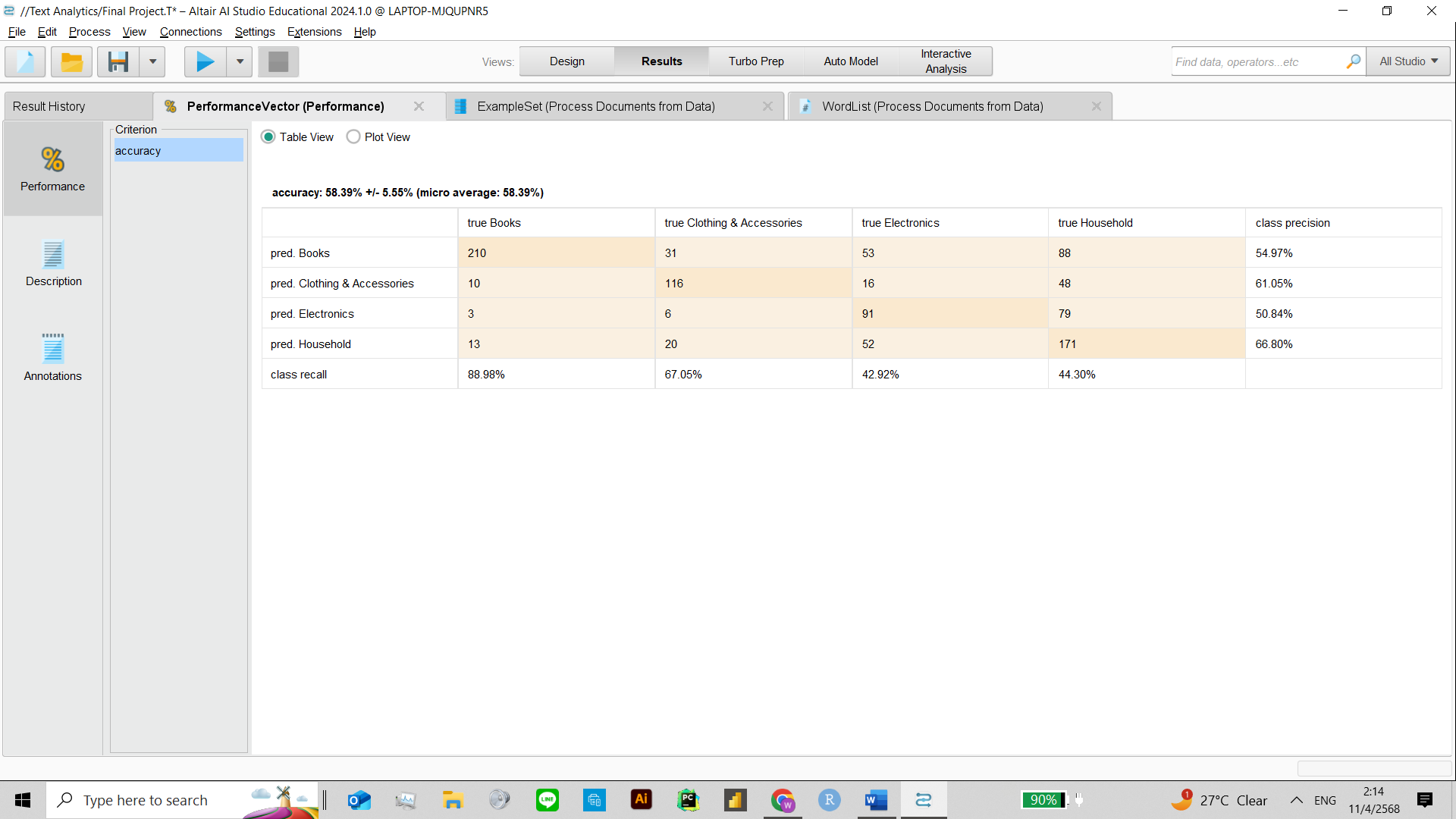
TF-IDF



Term Frequency



Term Occurrences



Binary Term Occurrences

**2nd Iteration (Change Vector Creation Model – k-NN)**

In the second iteration, four document vectorization methods were tested with the k-NN classifier to assess their impact on classification performance: TF-IDF, Term Frequency, Term Occurrence, and Binary Term Occurrence.

*TF-IDF* achieved the best accuracy at 66.25% ± 2.83%, with strong recall (85.17%) and balanced precision across categories. *Term Frequency* followed closely with 63.46% ± 3.37% accuracy. It offered slightly better recall for some categories but lower overall precision. *Term Occurrence* yielded 60.77% ± 3.89% accuracy, showing decreased performance, especially in identifying Electronics and Household items. *Binary Term Occurrence* had the lowest accuracy at 58.39% ± 5.55%, despite a relatively high recall rate of 88.98%, indicating it overestimated similarity between documents.

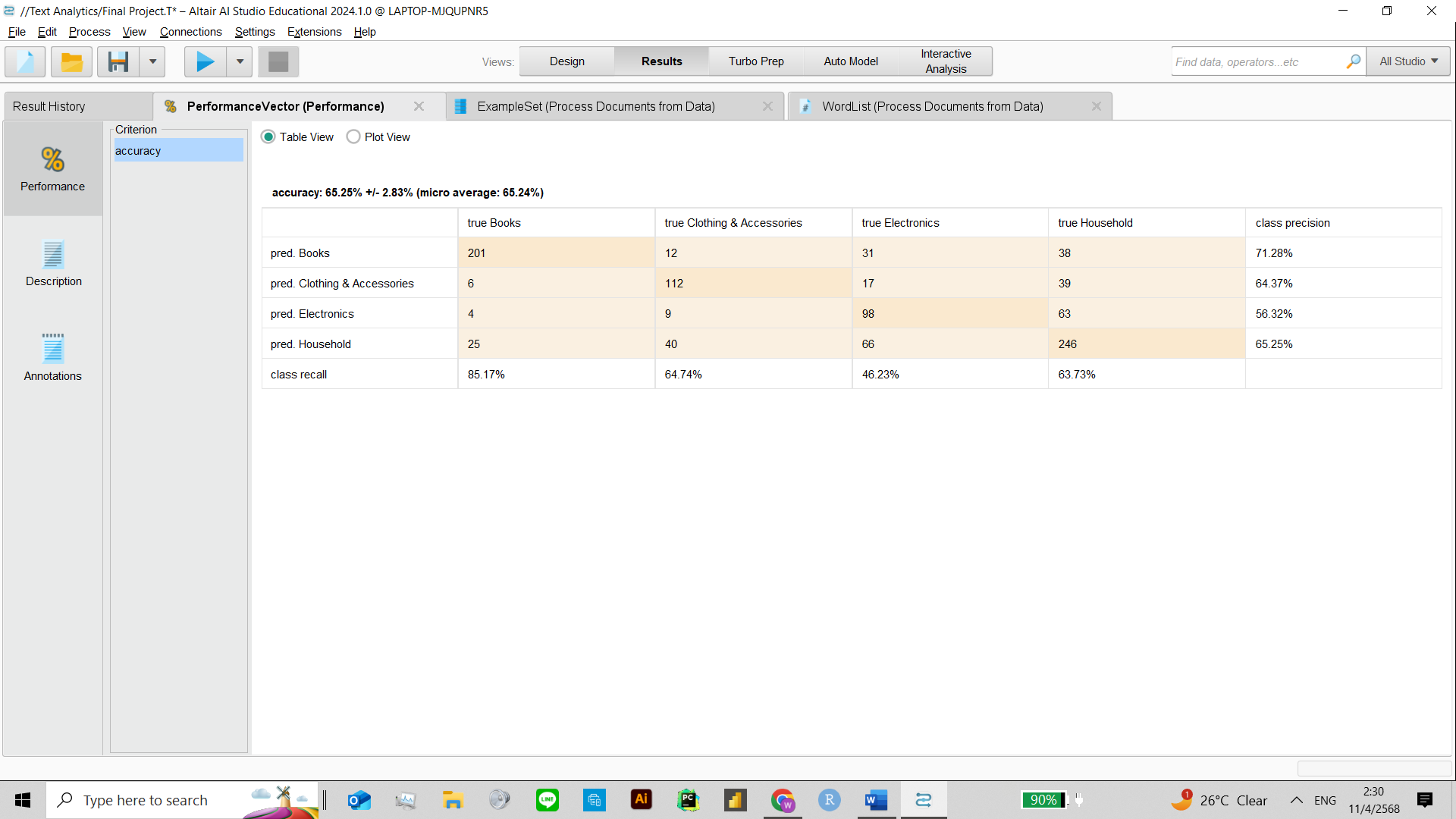
While all models captured key patterns in the data, TF-IDF remained the most reliable overall, balancing both precision and recall. Simpler representations like binary and occurrence-based vectors introduced more false positives, particularly in closely related categories.

**3rd Iteration (Using N-Gram)**



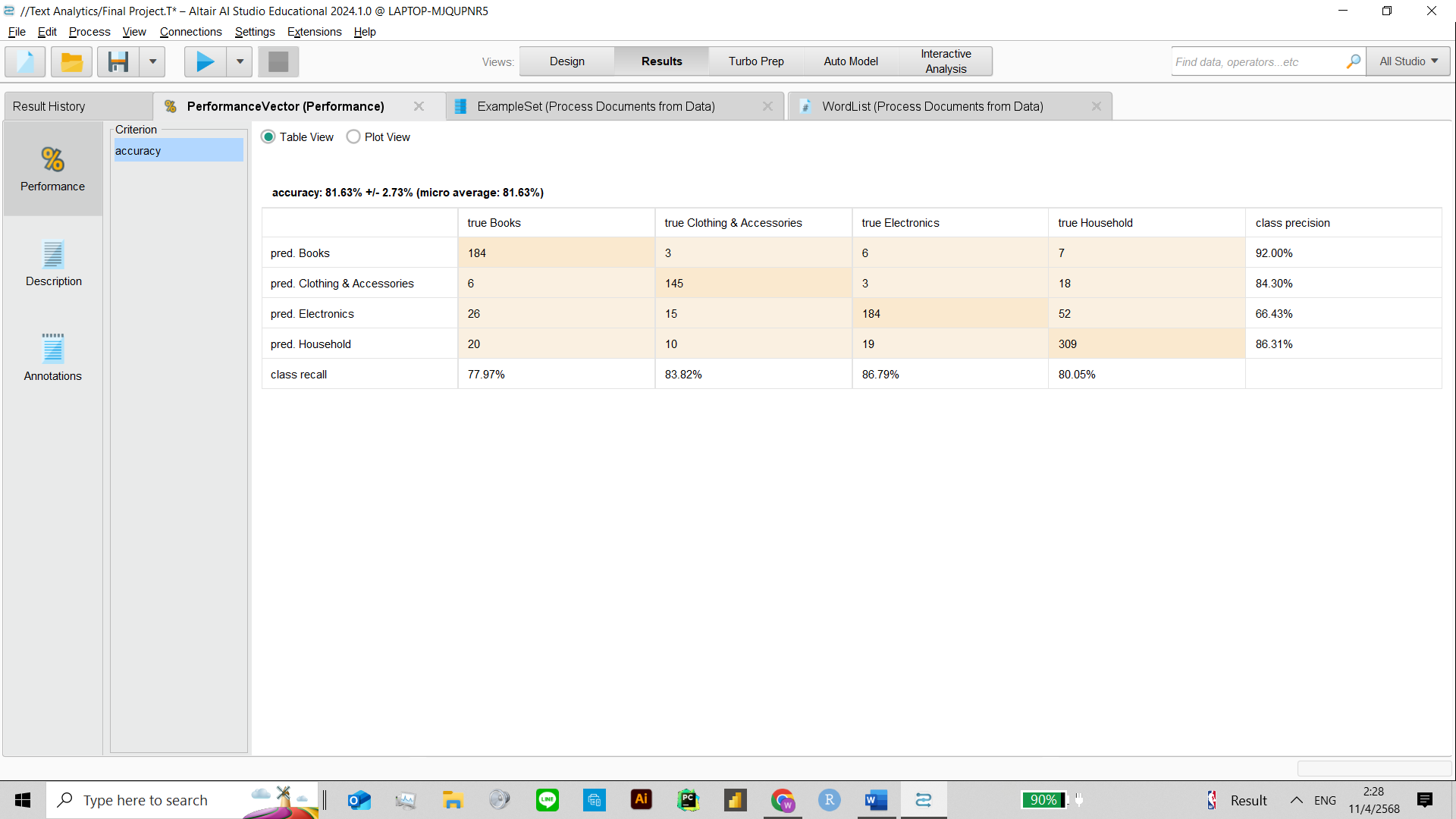
**Figure 12 Sub - processes Inside Process document**

In the final iteration, the text data was further tokenized using N-Gram techniques in RapidMiner. Two types of N-Grams were explored: Character N-Grams and Term N-Grams. This enhancement aimed to capture more contextual information and common word patterns that single-word tokens might miss.

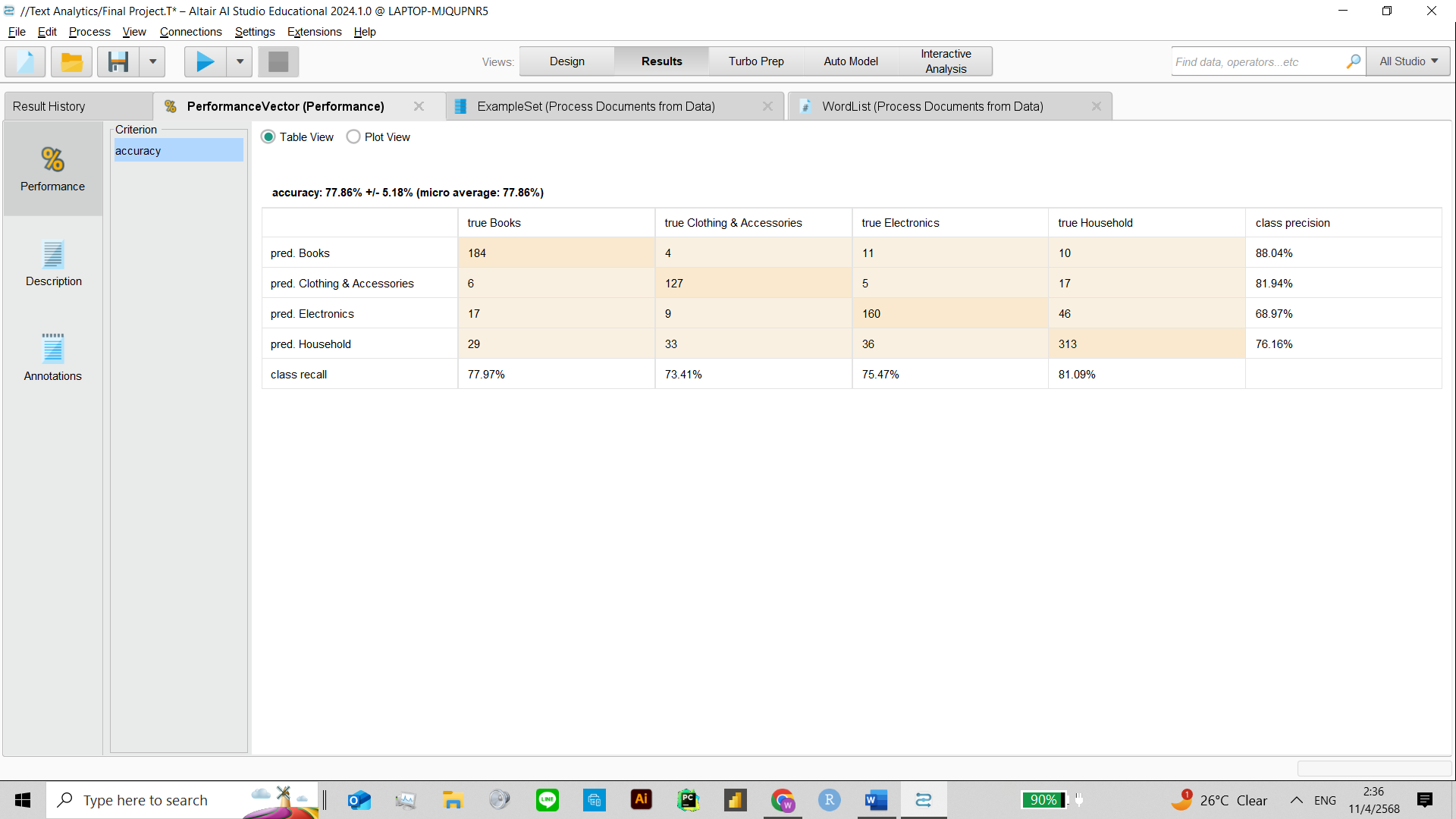


K-NN

N-Gram (Term)

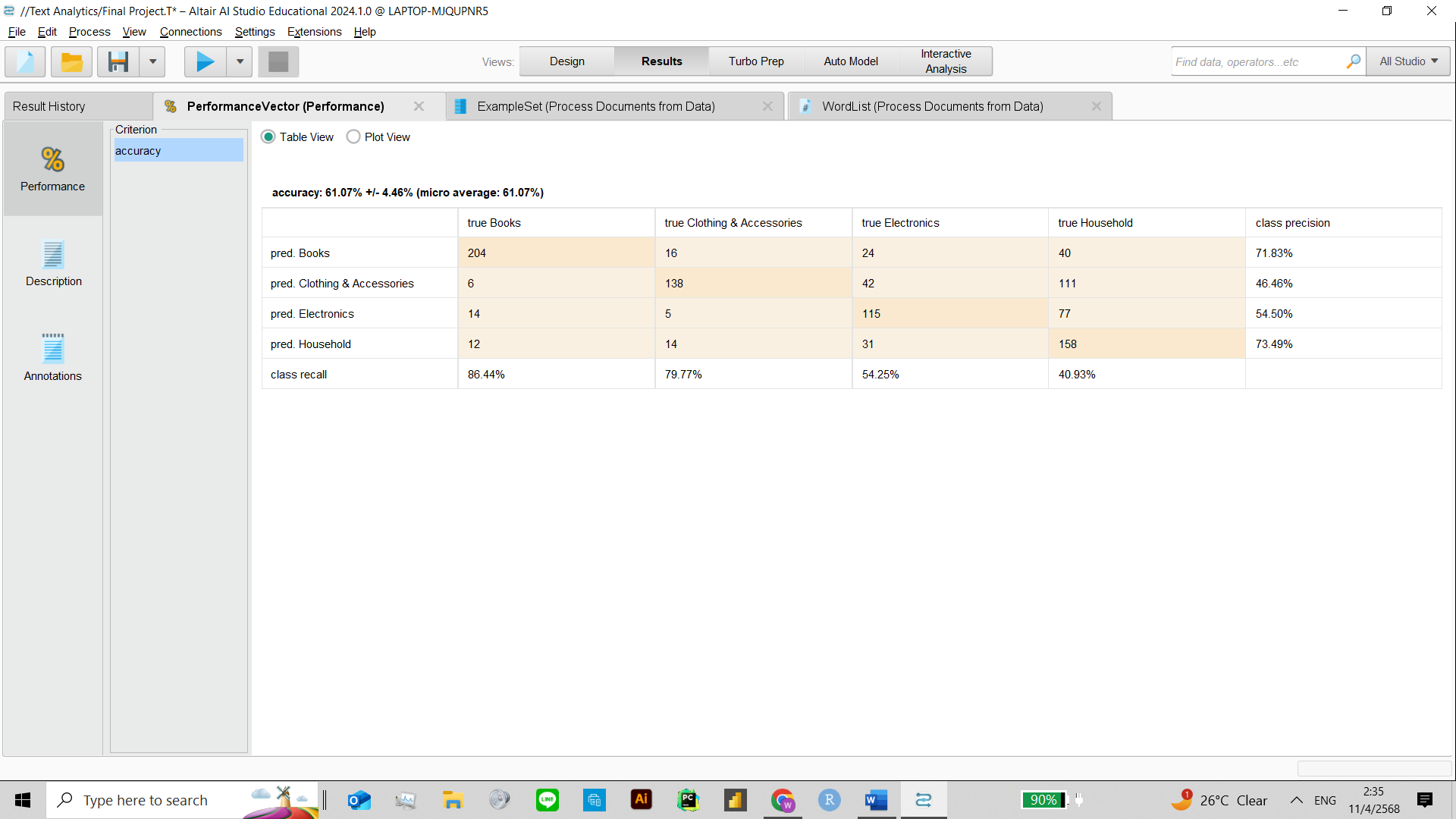


N-Gram (Characters)



N-Gram (Term)

Naïve-Bayes



N-Gram (Characters)

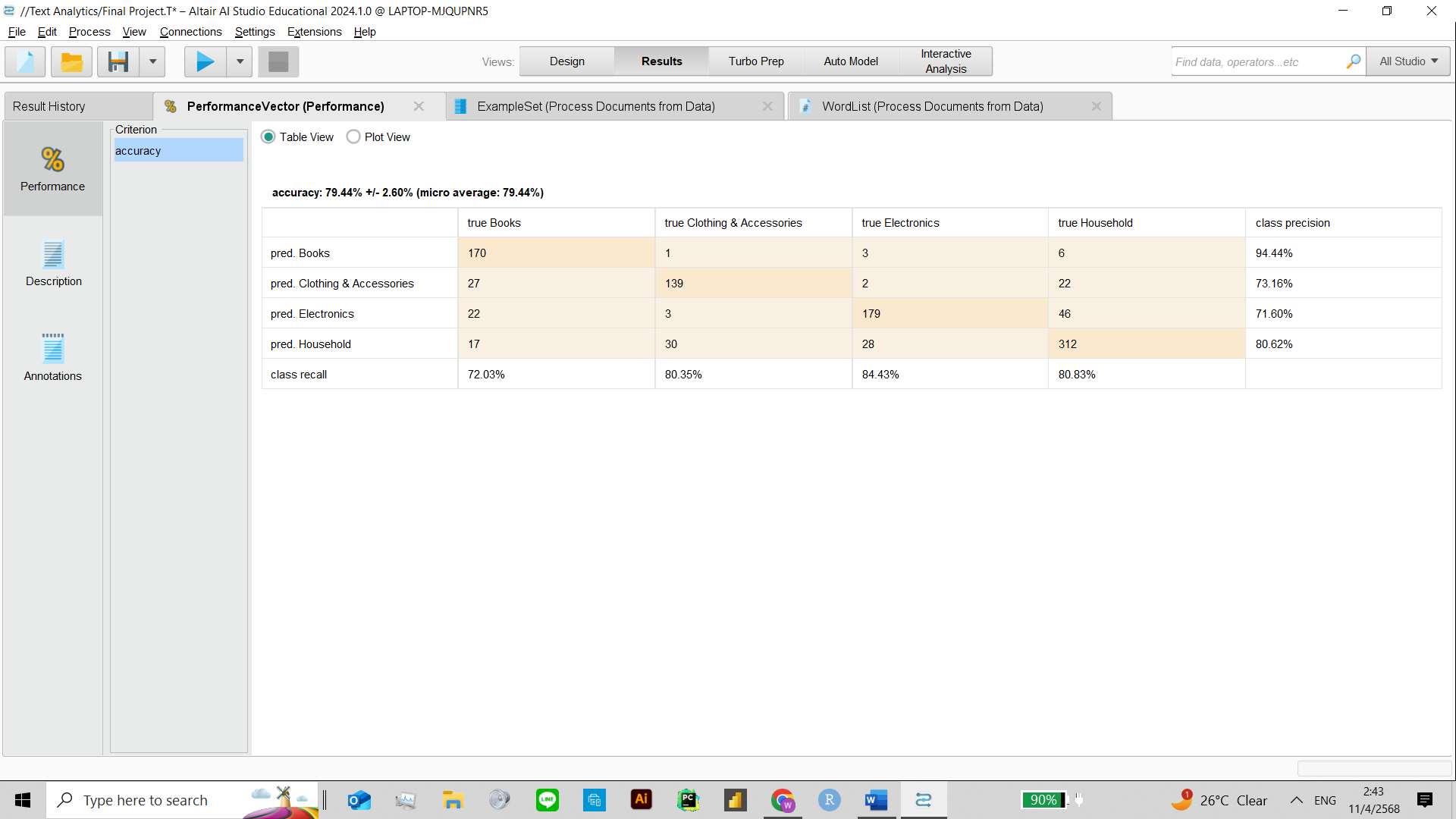
**N-Gram Comparison Summary**

Both Term N-Grams and Character N-Grams were applied to two classifiers—k-NN and Naïve Bayes—to evaluate their effectiveness.

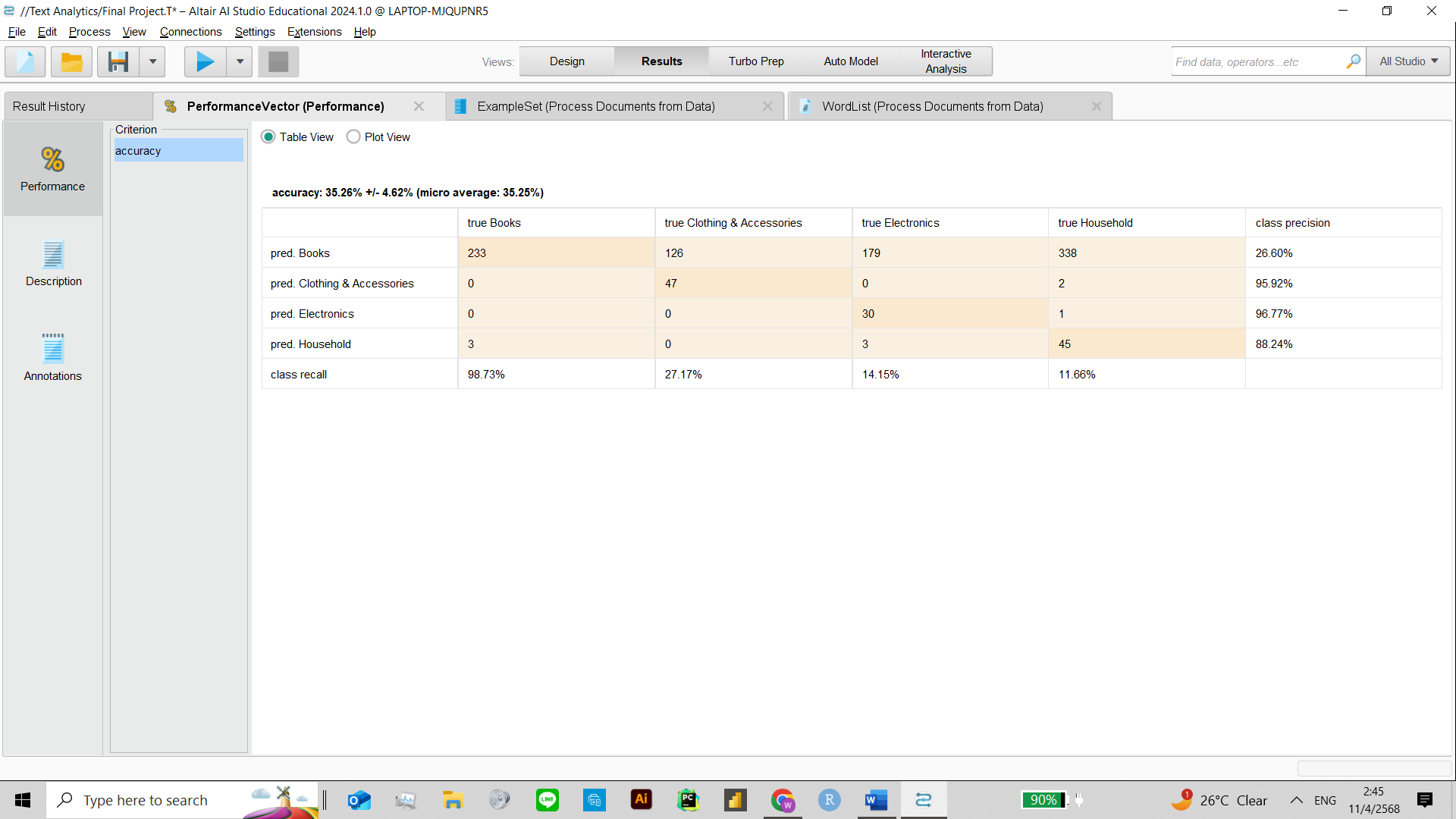
For k-NN, Character N-Grams significantly improved performance, achieving the highest overall accuracy of 81.63% ± 2.73%, with strong precision and recall across all product categories. In contrast, Term N-Grams produced an accuracy of 65.25% ± 2.83%, roughly on par with TF-IDF.

With Naïve Bayes, Term N-Grams performed best, reaching 77.86% ± 5.18% accuracy, while Character N-Grams dropped to 61.07% ± 4.46%, showing that Naïve Bayes benefited more from distinct word-level patterns rather than overlapping character sequences.

Overall, Character N-Grams were most effective with distance-based models like k-NN, while Term N-Grams proved more suitable for probabilistic models like Naïve Bayes.

**4th Iteration (Pruning)**

Naïve-Bayes

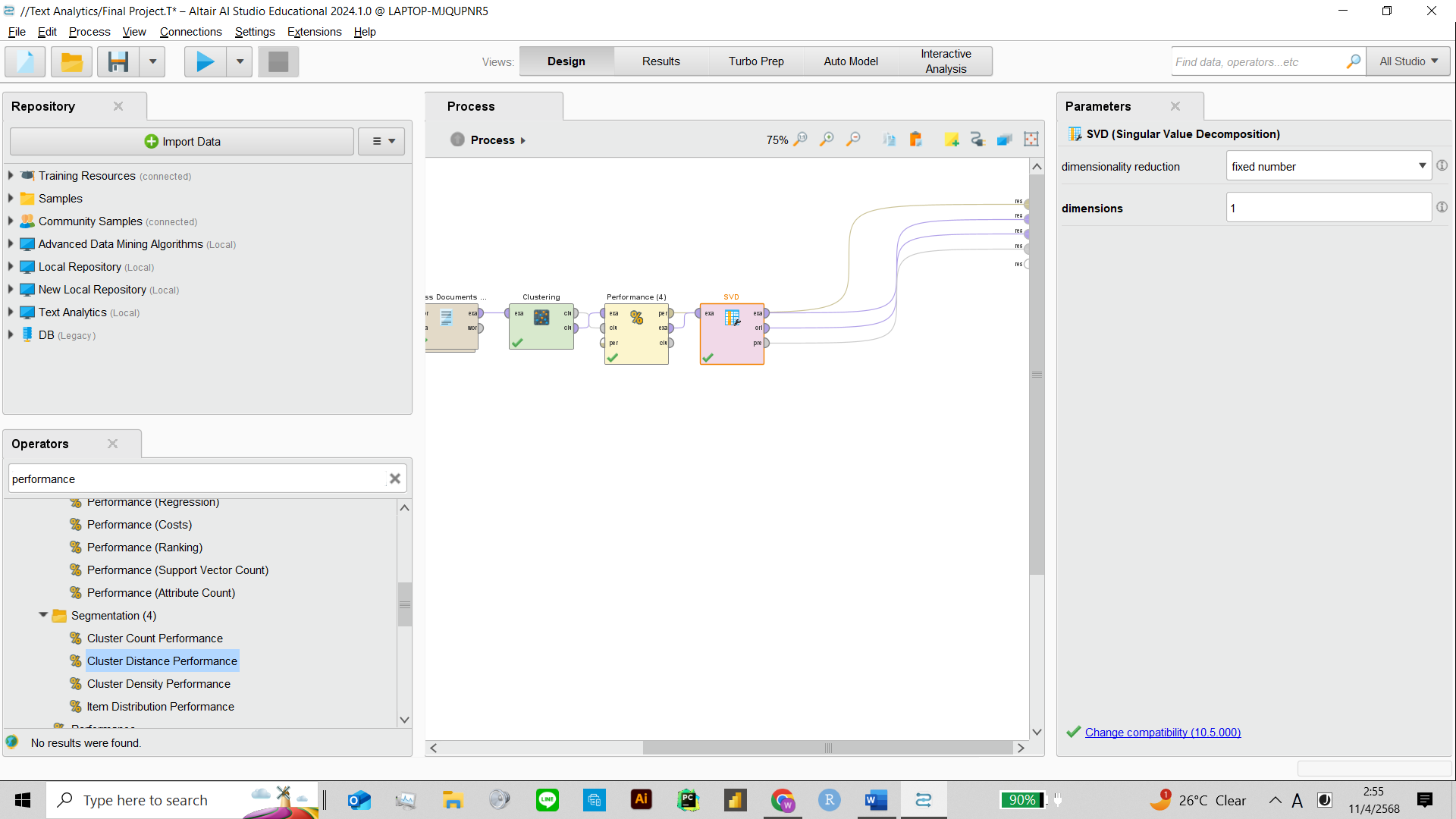


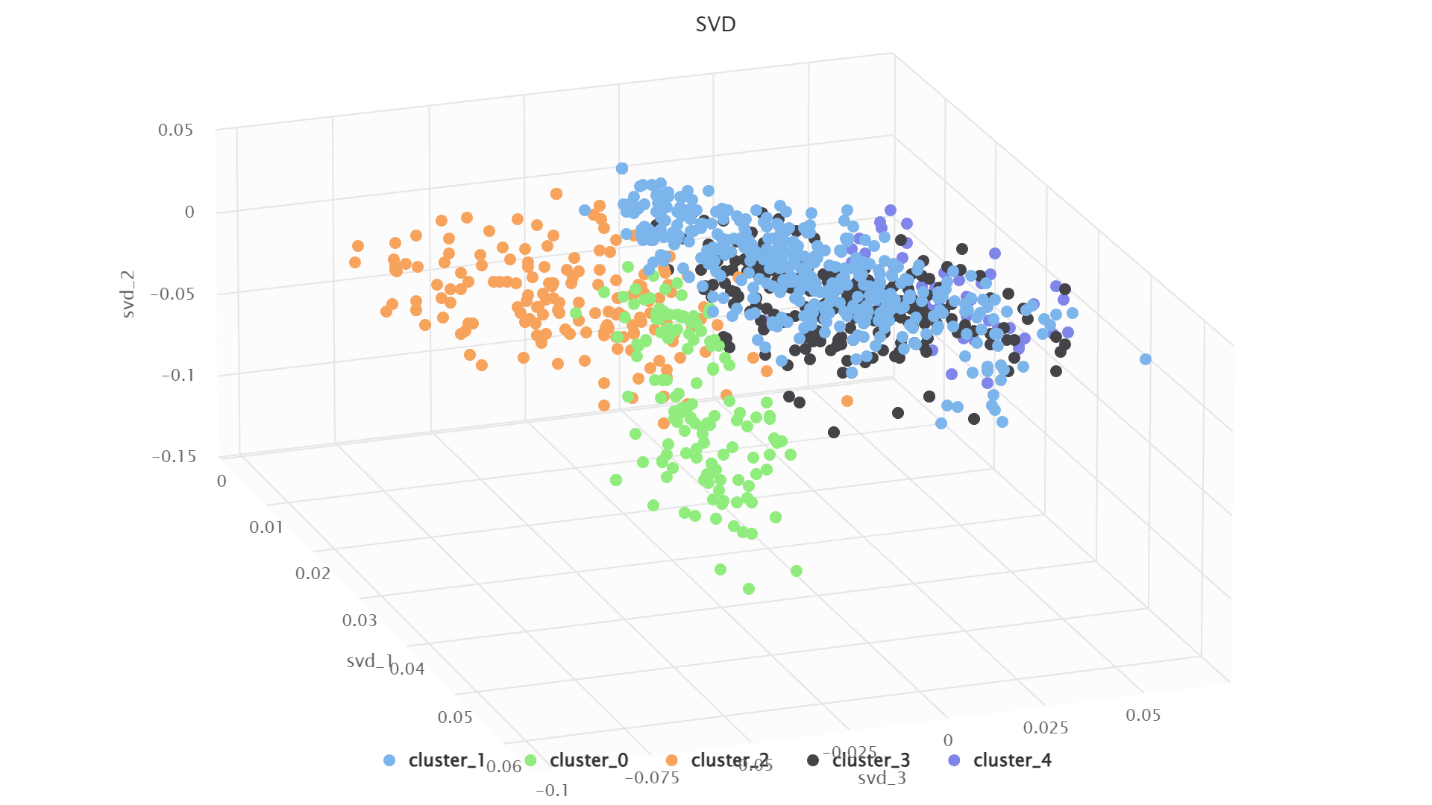
K-NN

**Figure 13 Performance after perform Pruning (3-30%) Clustering**

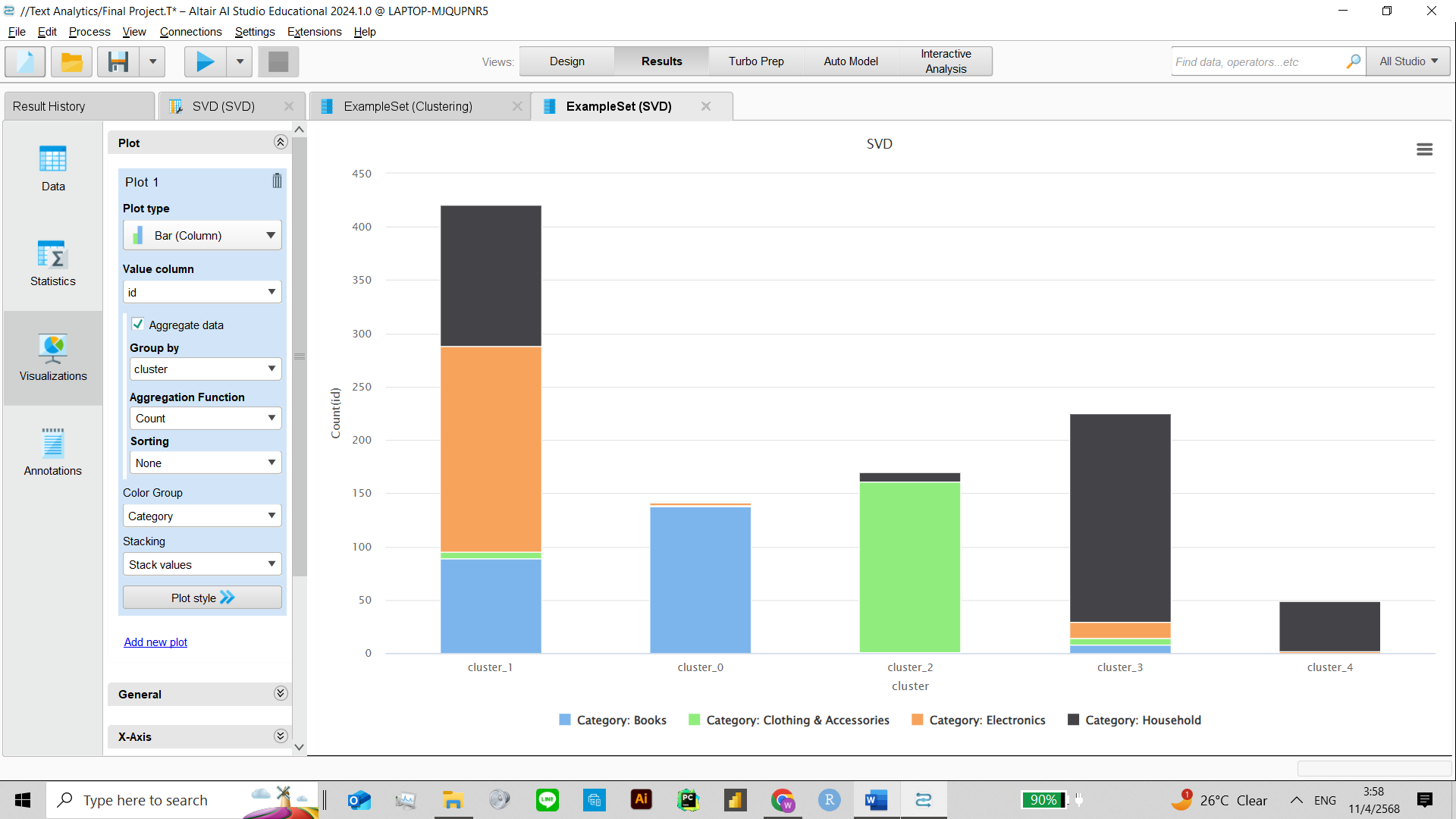
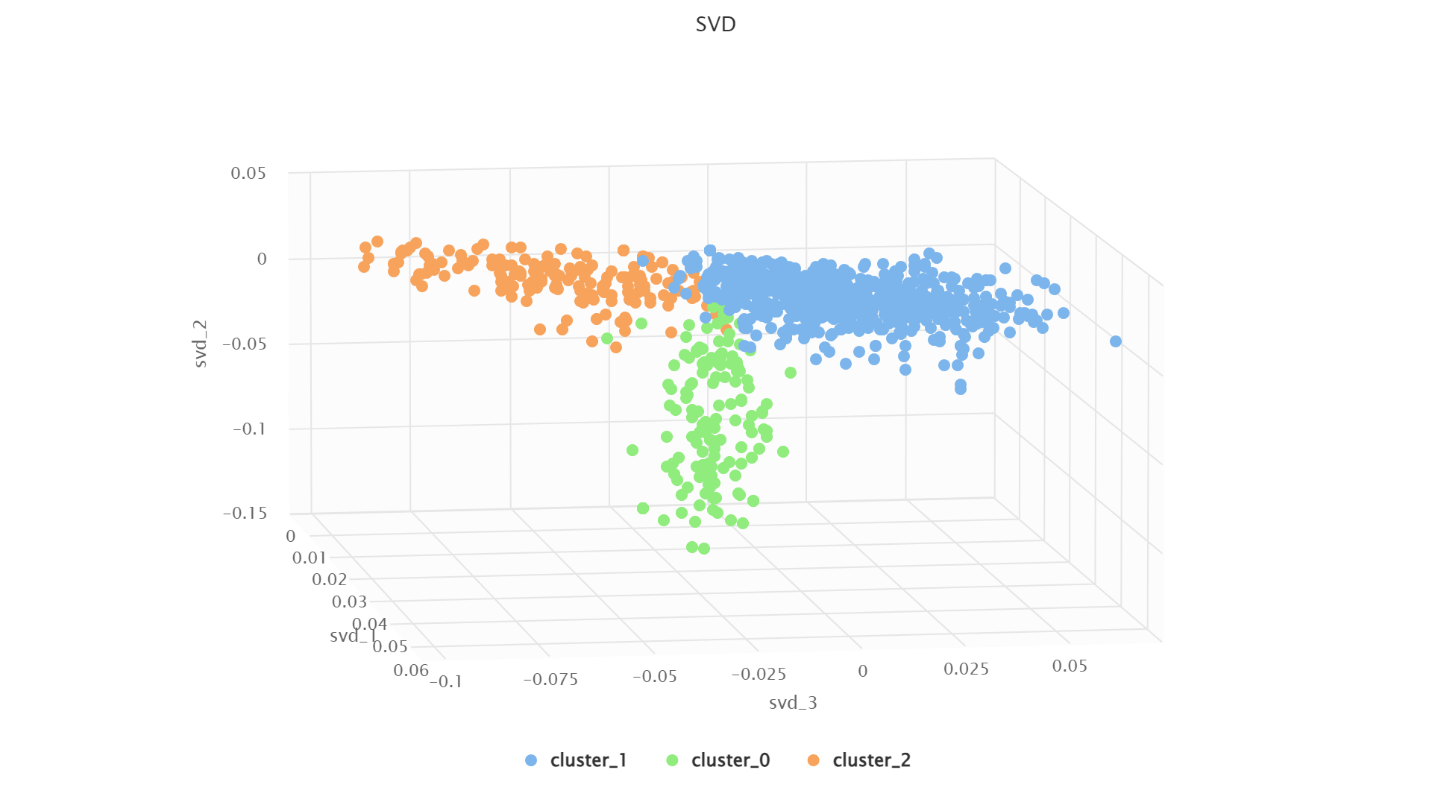
TF-IDF pruning (3%–30%) was applied to reduce noise and improve model performance. Results showed a clear contrast between models.

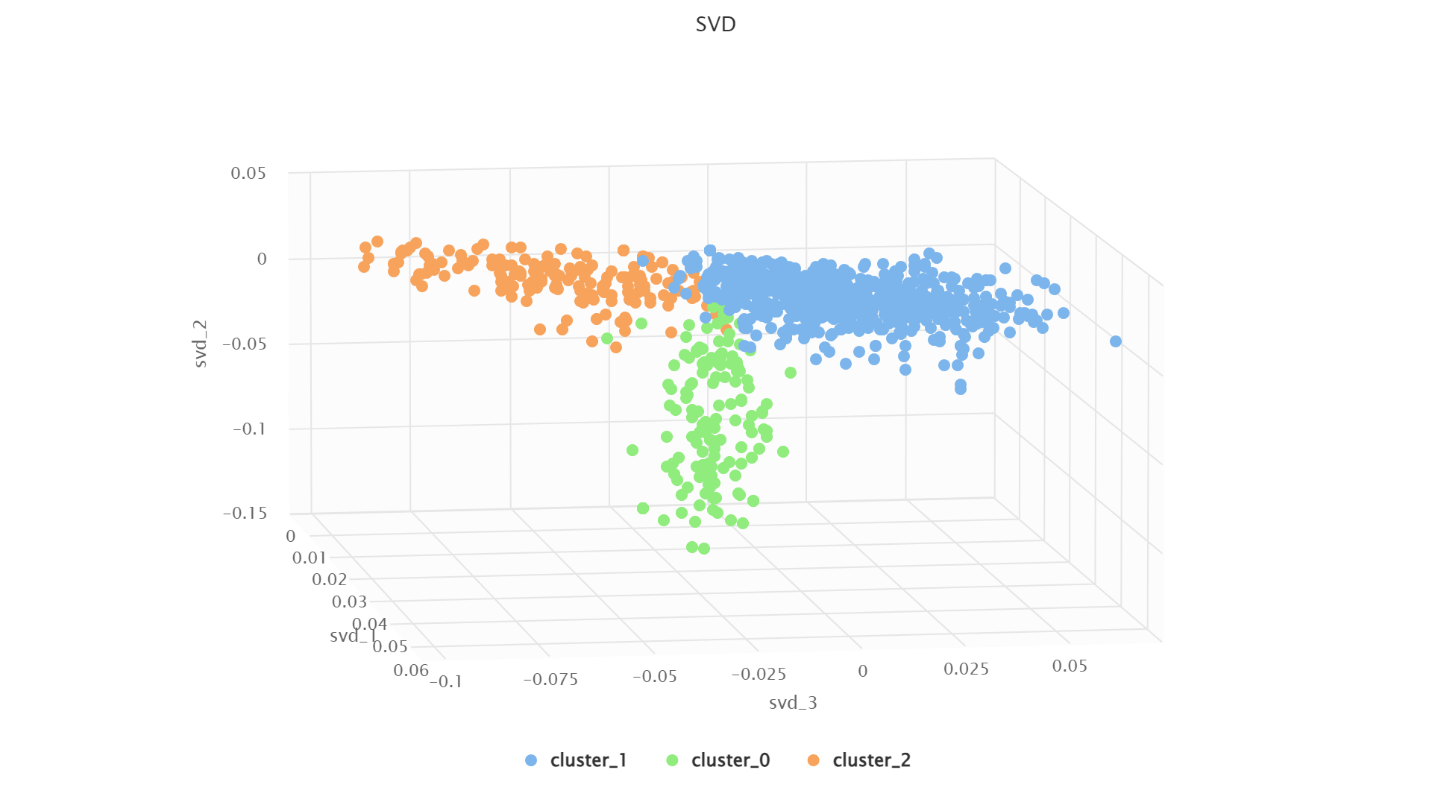
Naïve Bayes improved significantly, reaching 79.44% ± 2.60% accuracy, with high precision and recall across all categories. In contrast, k-NN performance dropped sharply to 35.26% ± 4.62%, as pruning removed key terms used for distance calculation.

This shows that pruning benefits probabilistic models like Naïve Bayes but can hinder distance-based models like k-NN.

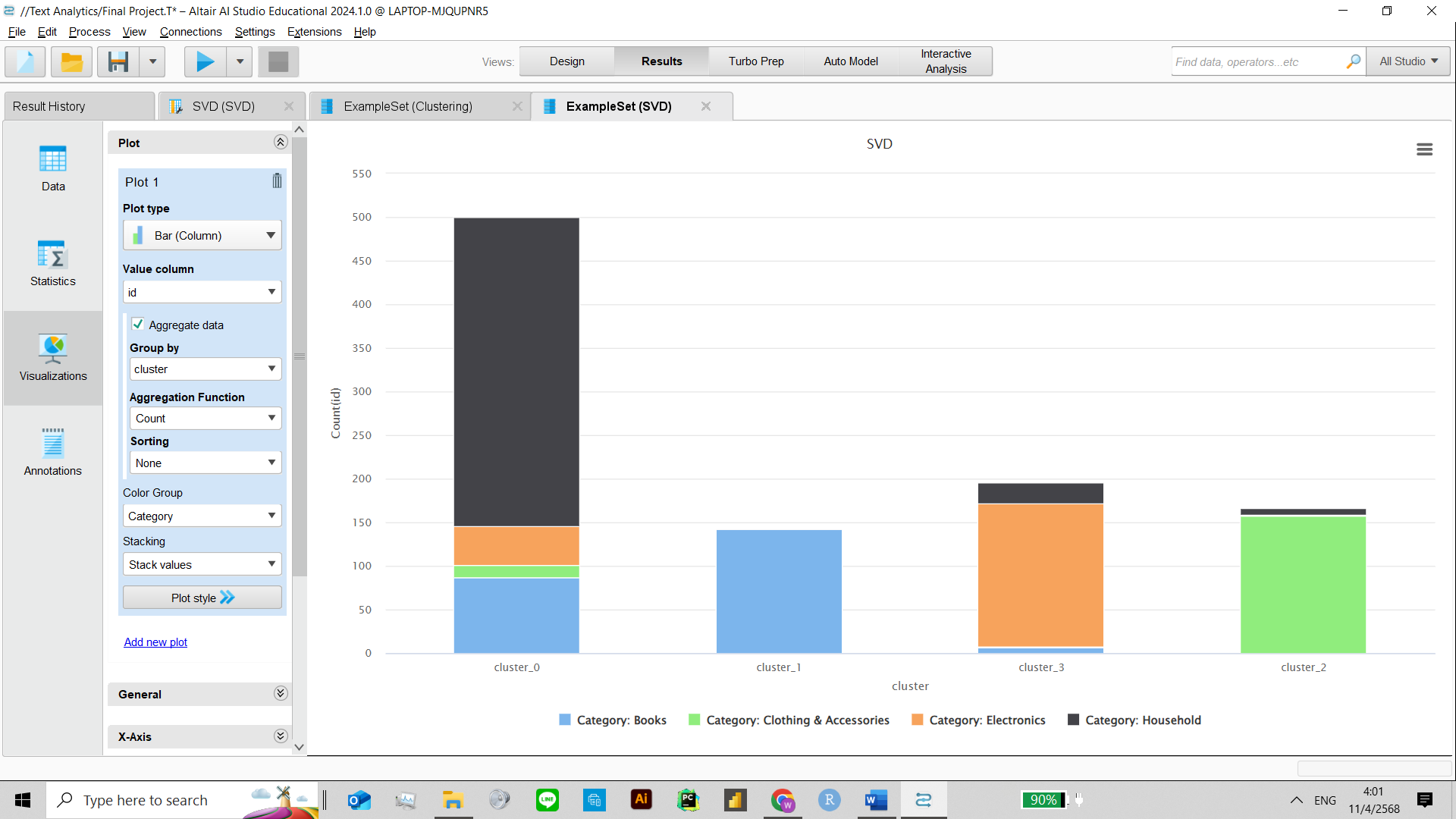
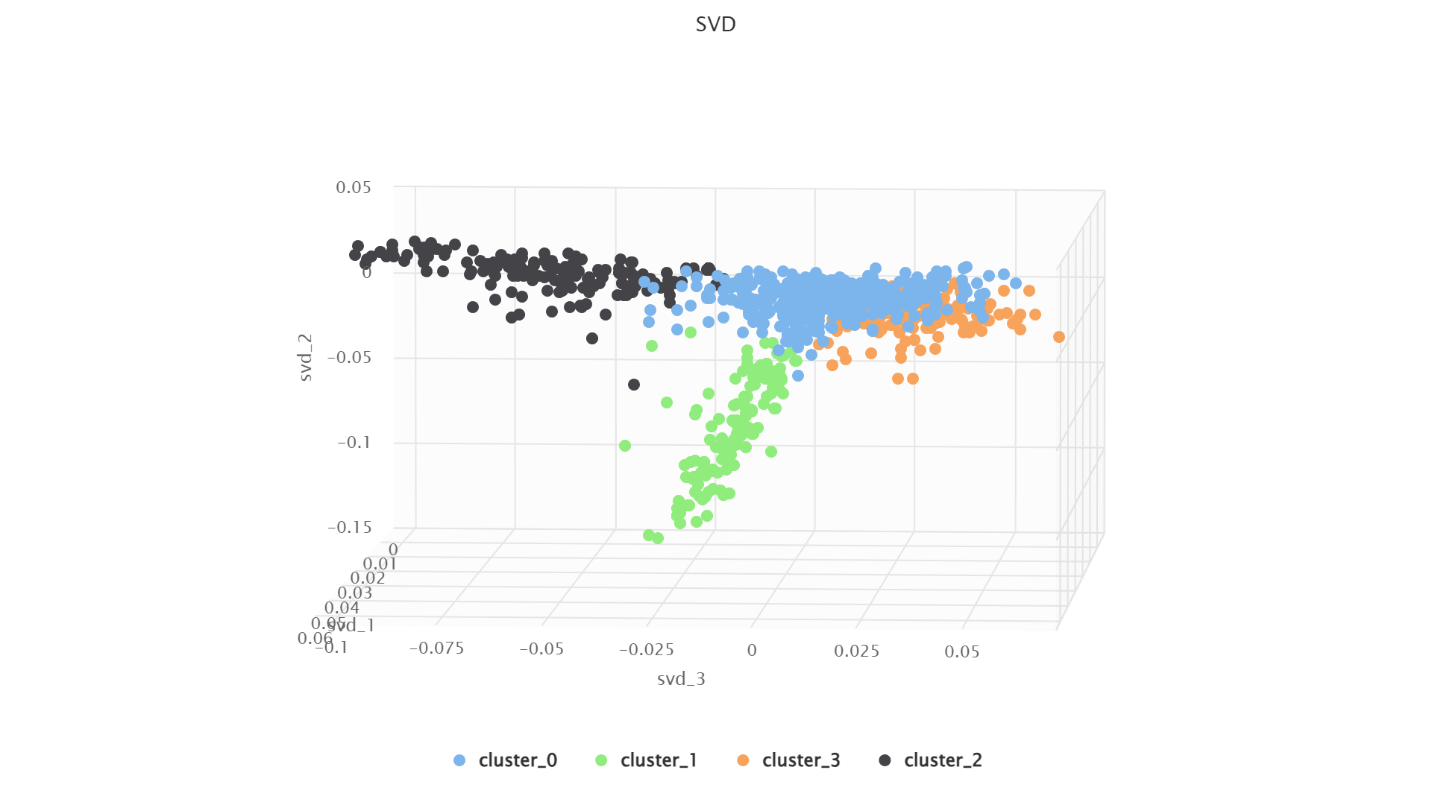
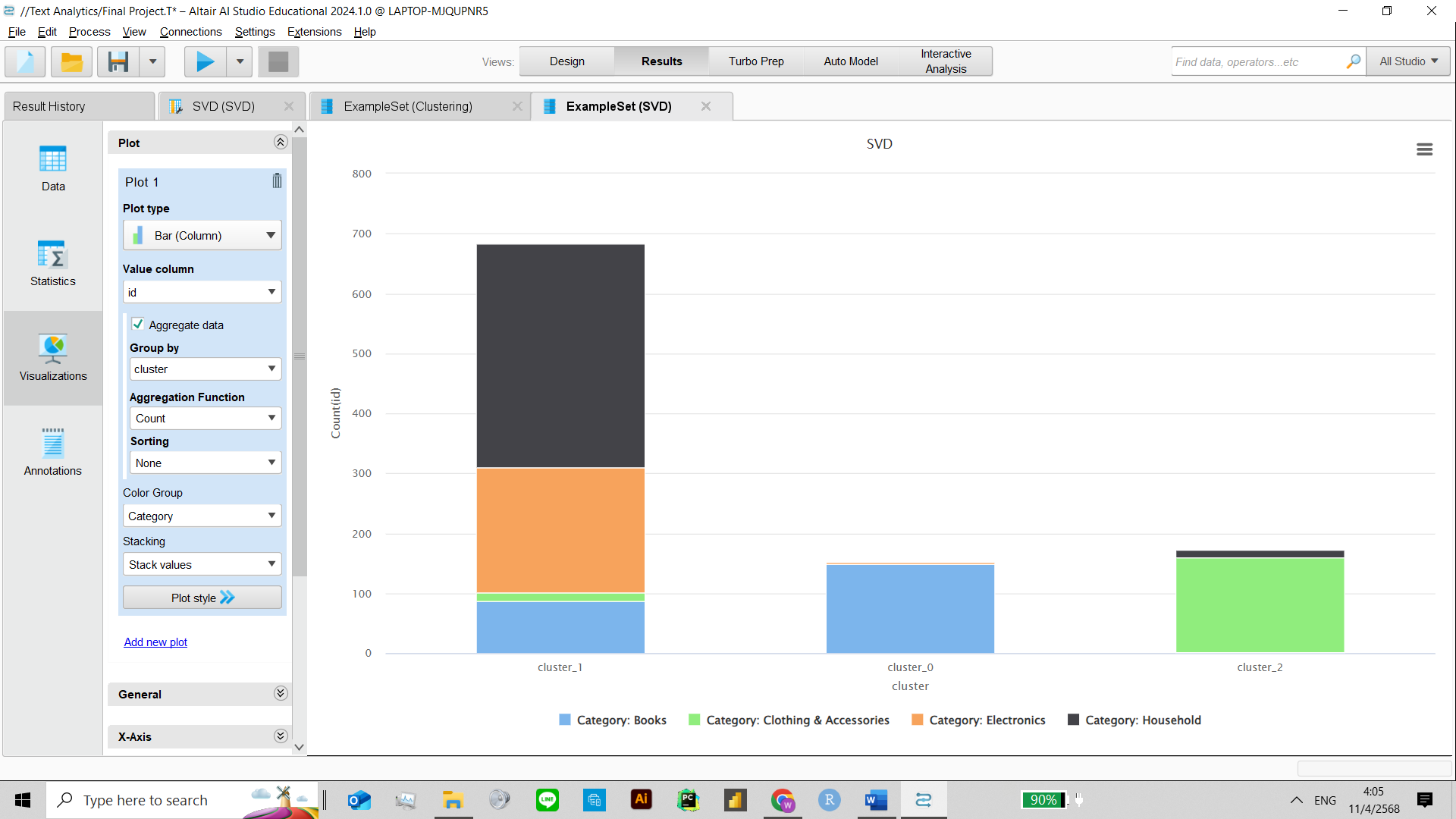
**Figure 15 Figure 16 RapidMiner Processes of Clustering**

K=5

****

****

K=3

****

K=4

**Clustering Analysis and Visualization (K=3, 4, 5)**

To evaluate the clustering performance, K-means was applied using K=3, 4, and 5. Each result was visualized with a 3D scatter plot based on the first three SVD components (svd\_1, svd\_2, svd\_3), and color-coded by cluster.

K=3: The clusters are fairly distinct in the 3D space. However, the bar chart shows an imbalance, with one dominant cluster containing most data points from all categories.

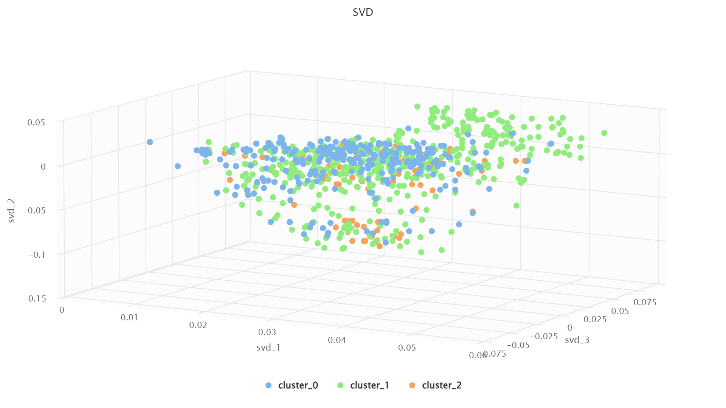
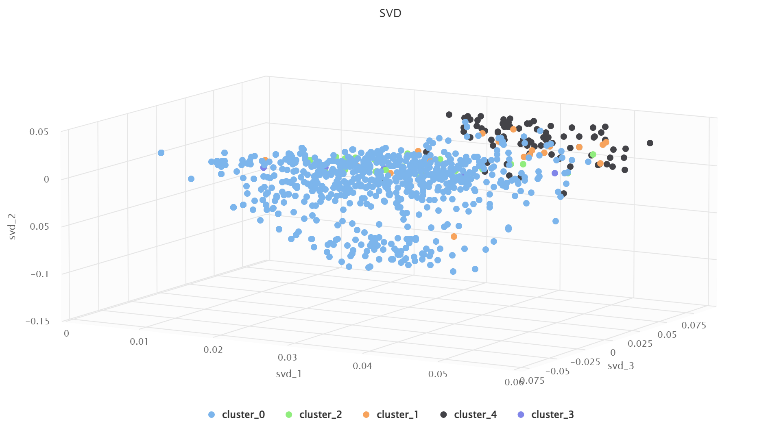
K=4: Clusters become more balanced, and category separation improves. The stacked bar chart reveals more evenly distributed product categories within clusters.

K=5: While clusters visually separate better in the 3D plot, some overlap remains. The bar chart shows the best differentiation of product categories, suggesting that K=5 offers more meaningful segmentation.

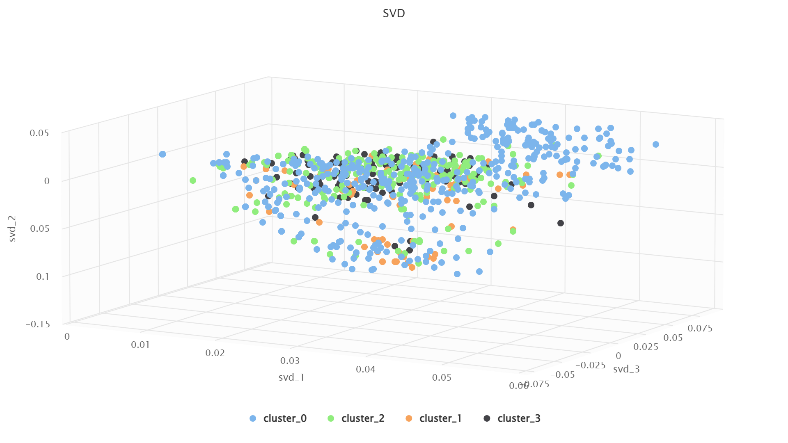
These visualizations help confirm that increasing K improves category representation across clusters, with K=5 performing best overall.

**Davies-Bouldin Index and Clustering Quality**

|  |  |
| --- | --- |
| **K** | **Davies Bouldin** |
| 2 | -4.617 |
| 3 | -5.030 |
| 4 | -5.668 |
| 5 | -0.873 |
| 10 | -0.834 |

****The Davies-Bouldin index was used to evaluate clustering quality for different values of k. Lower values indicate better separation between clusters. The best result was observed with k = 4, showing the lowest index at -5.668, suggesting the most distinct and compact clustering in comparison to other values.

K-medoids K=4

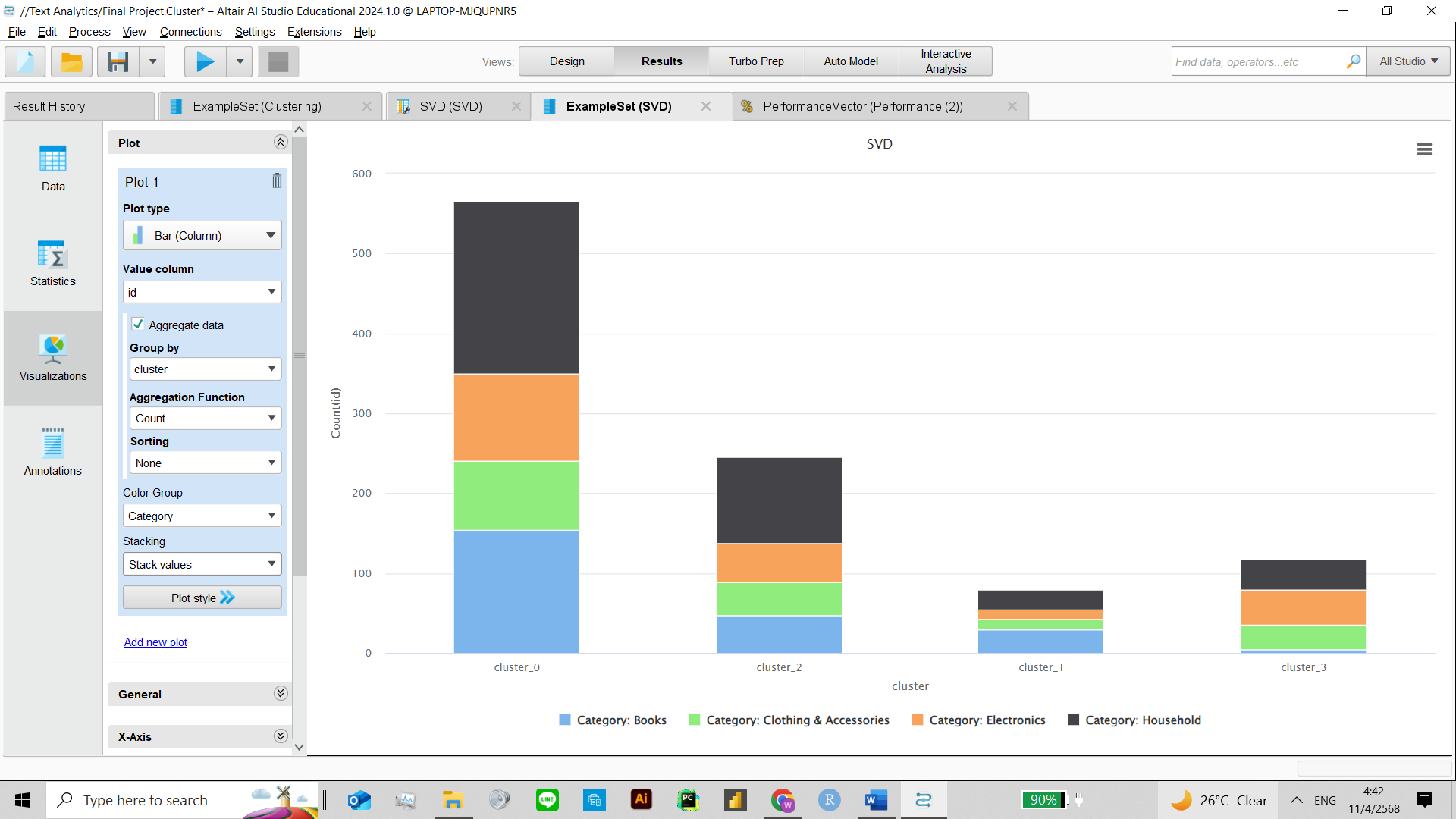


K-medoids K=5

K-medoids K=3

**Clustering with K-Medoids algorithm**

K-medoids clustering was applied with K = 3, 4, and 5 using SVD components for 3D visualization. Compared to K-means, the clusters formed were less distinct, with more overlap and weaker separation. K=3 showed scattered points, K=4 had slightly improved grouping, and K=5 offered the clearest structure, though some mixing remained. Overall, K-medoids produced less compact and well-separated clusters.



**Summary and Conclusion**

This project explored the classification and clustering of e-commerce product descriptions using machine learning. Starting with a sampled dataset from Kaggle, preprocessing techniques such as tokenization, stemming, and TF-IDF were applied. Classification models including k-NN and Naïve Bayes were tested with various parameters and vectorization methods. Naïve Bayes consistently outperformed k-NN, achieving up to 79.44% accuracy with TF-IDF and pruning. N-Gram models further improved results, with Character N-Grams yielding the highest accuracy (81.63%) using k-NN. Clustering was analyzed using K-means and K-medoids, visualized through 3D scatter plots and bar charts. K=4 showed the best Davies-Bouldin score, indicating the most compact clustering, while K=5 provided the clearest category separation in visualizations. Overall, the project highlights how proper preprocessing and model selection significantly impact text analytics performance in classifying and organizing product data.

These results have practical value in e-commerce, where automated classification improves product organization and search accuracy. Clustering also helps businesses group similar items and gain insights from product descriptions. Overall, this study shows how machine learning can make unstructured text data more useful and actionable in real-world applications.