Decision Tree and Random Forest Evaluation Report

This section presents a comprehensive analysis of **Decision Tree** and **Random Forest** classifiers used in our **sentiment classification** project. We evaluate model performance using both **Bag-of-Words (BoW)** and **TF-IDF** feature representations and analyze the impact of **SVD dimensionality reduction** and key **hyperparameters**.

Model Training without SVD

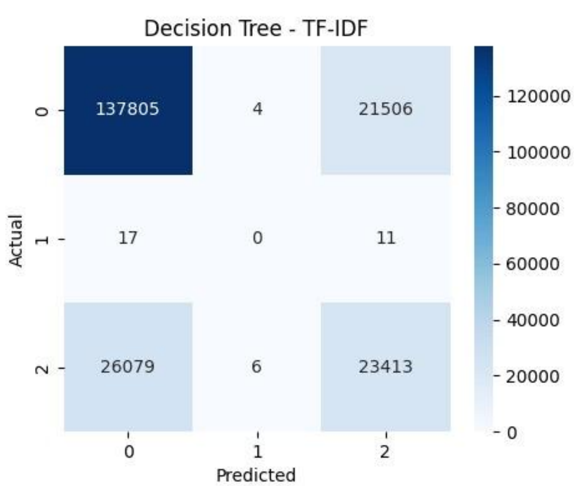
1. Decision Tree with TF-IDF Features

Accuracy :77.19%

Classification Report:

	Precision	Recall	F1-Score
Negative	0.8400	0.8600	0.8500
Postive	0.0000	0.0000	0.0000
Neutral	0.5200	0.4700	0.5000

Confusion Matrix:



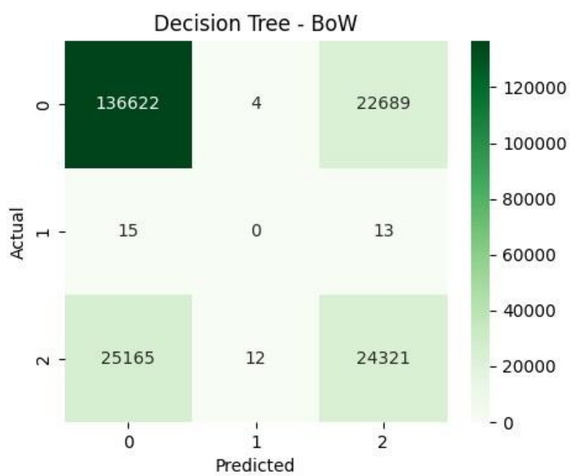
2. Decision Tree with BoW Features

Accuracy :77.06%

Classification Report:

	Precision	Recall	F1-Score
Negative	0.8400	0.8600	0.8500
Postive	0.0000	0.0000	0.0000
Neutral	0.5200	0.4900	0.5000

Confusion Matrix:



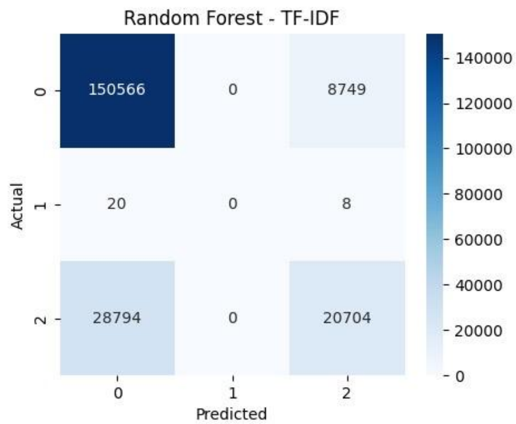
3. Random Forest with TF-IDF Features

Accuracy :82.01%

Classification Report:

	Precision	Recall	F1-Score
Negative	0.8400	0.9500	0.8900
Postive	0.0000	0.0000	0.0000
Neutral	0.7000	0.4200	0.5200

Confusion Matrix:



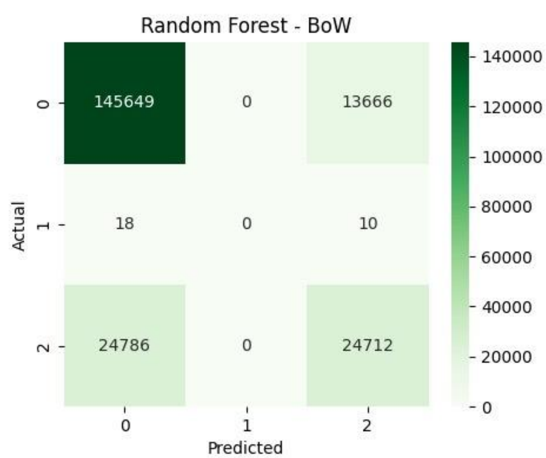
4. Random Forest with BoW Features

Accuracy : 81.57%

Classification Report:

	Precision	Recall	F1-Score
Negative	0.8500	0.9100	0.8800
Postive	0.0000	0.0000	0.0000
Neutral	0.6400	0.5000	0.5600

Confusion Matrix:



Model Training after applying SVD(Reduced to 100 Features).

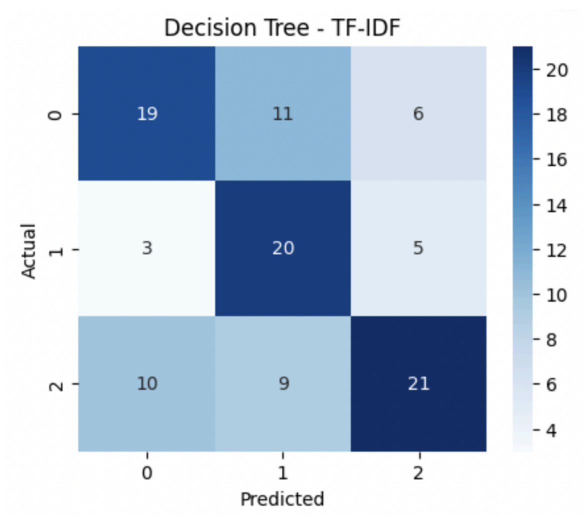
1. Decision Tree with TF-IDF Features

Accuracy :57.69%

Classification Report:

Sentiment	Precision	Recall	F1-Score
Negative	0.5900	0.5300	0.5600
Postive	0.5800	0.7100	0.5900
Neutral	0.6600	0.5300	0.5800

Confusion Matrix:



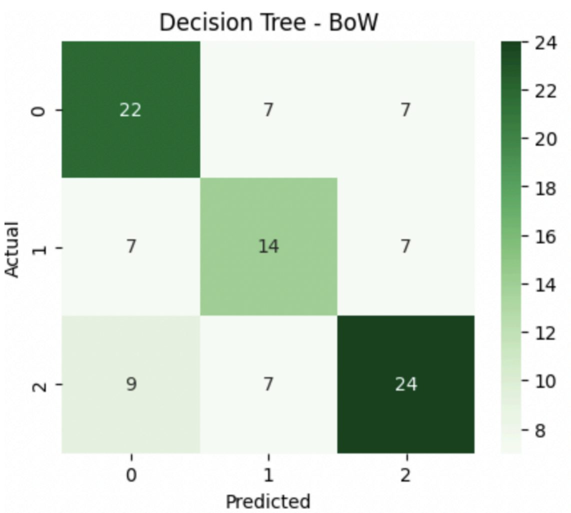
2. Decision Tree with BoW Features

Accuracy :57.69%

Classification Report:

Sentiment	Precision	Recall	F1-Score
Negative	0.5800	0.6100	0.5900
Postive	0.5000	0.5000	0.5000
Neutral	0.6300	0.6000	0.6200

Confusion Matrix:



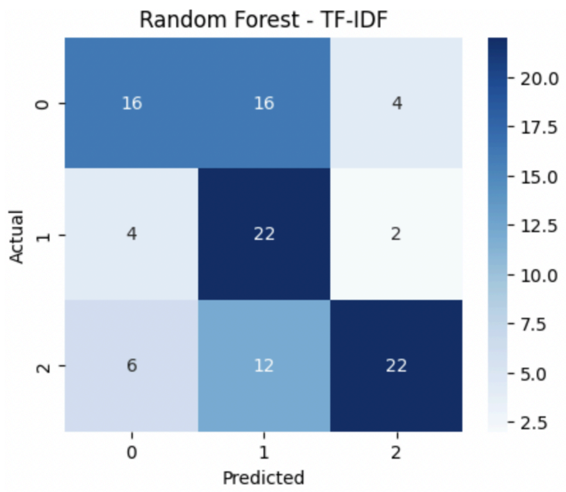
3. Random Forest with TF-IDF Features

Accuracy :57.69%

Classification Report:

Sentiment	Precision	Recall	F1-Score
Negative	0.6200	0.4400	0.5200
Postive	0.4400	0.7900	0.5600
Neutral	0.7900	0.5500	0.6500

Confusion Matrix:



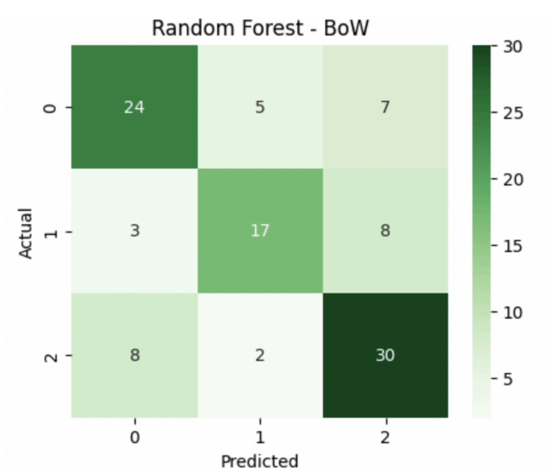
4. Random Forest with BoW Features

Accuracy : 68.27%

Classification Report:

Sentiment	Precision	Recall	F1-Score
Negative	0.6900	0.6700	0.6800
Postive	0.7100	0.6100	0.6500
Neutral	0.6700	0.7500	0.7100

Confusion Matrix:



Why Random Forest with BoW Outperforms TF-IDF in SVD training?

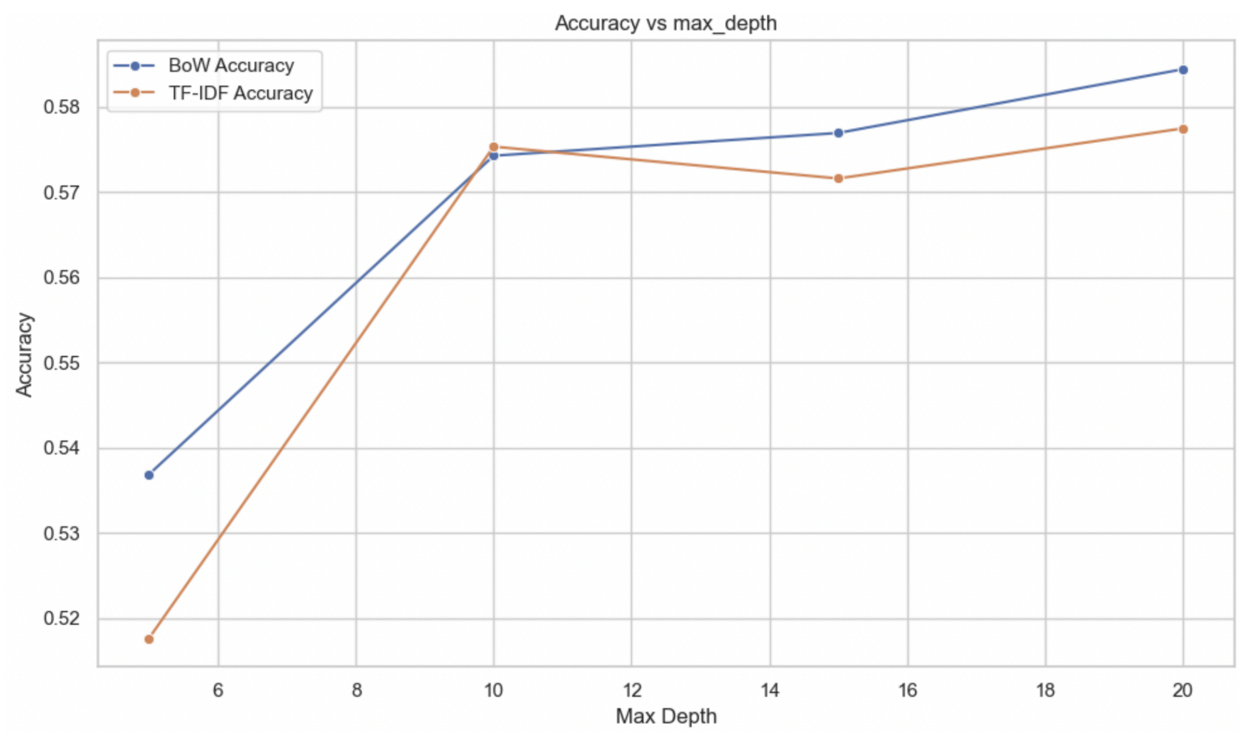
From our model evaluations, it is evident that Bag-of-Words (BoW) features outperform TF-IDF features in this sentiment classification task, particularly when paired with a Random Forest classifier. This may be attributed to the nature of our dataset — where common sentiment-indicative words (e.g., *good*, *bad*, *love*, *hate*) are crucial for classification. TF-IDF tends to down-weight these frequent terms, which can reduce their impact on model performance.

On the other hand, Random Forest classifiers consistently outperform Decision Trees due to their ensemble learning approach. By aggregating predictions from multiple decision trees, Random Forest reduces overfitting and improves generalization.

Thus, the best overall performance was achieved using BoW features with a Random Forest classifier, making it the recommended setup for this sentiment analysis task based on our current dataset.

Hyperparameter Analysis

Accuracy vs. Max Depth



Interpretation:

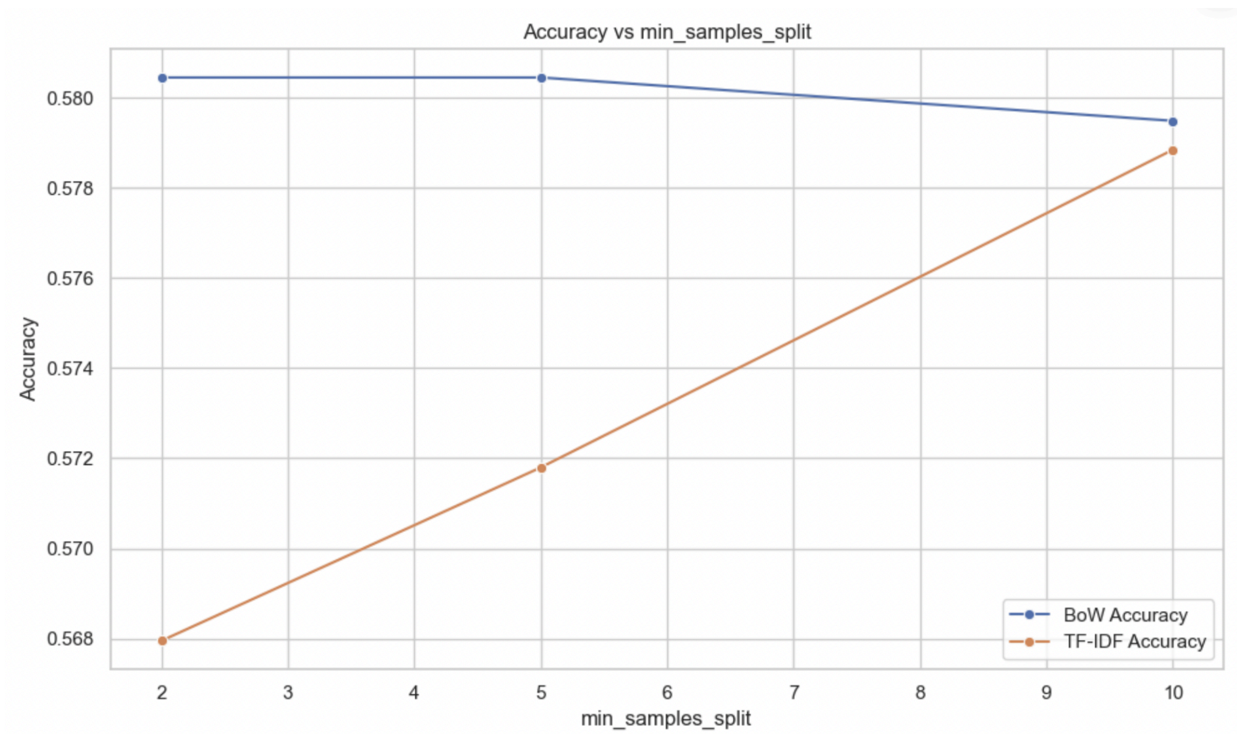
- Accuracy increases with depth for both BoW and TF-IDF initially.
- BoW consistently outperforms TF-IDF across all depth values.
- Beyond a depth of 10, both lines plateau, with BoW still slightly ahead.

Explanation:

- Shallow depths underfit the data, especially for TF-IDF.
- Optimal performance is reached around depth 10.
- Deeper trees risk overfitting, especially with TF-IDF.

Conclusion: A max depth between 10–15 is ideal, with BoW maintaining better generalization.

Accuracy vs. Min Samples Split



Interpretation:

- BoW starts with higher accuracy and remains stable.
- TF-IDF starts low but improves with increased min_samples_split.

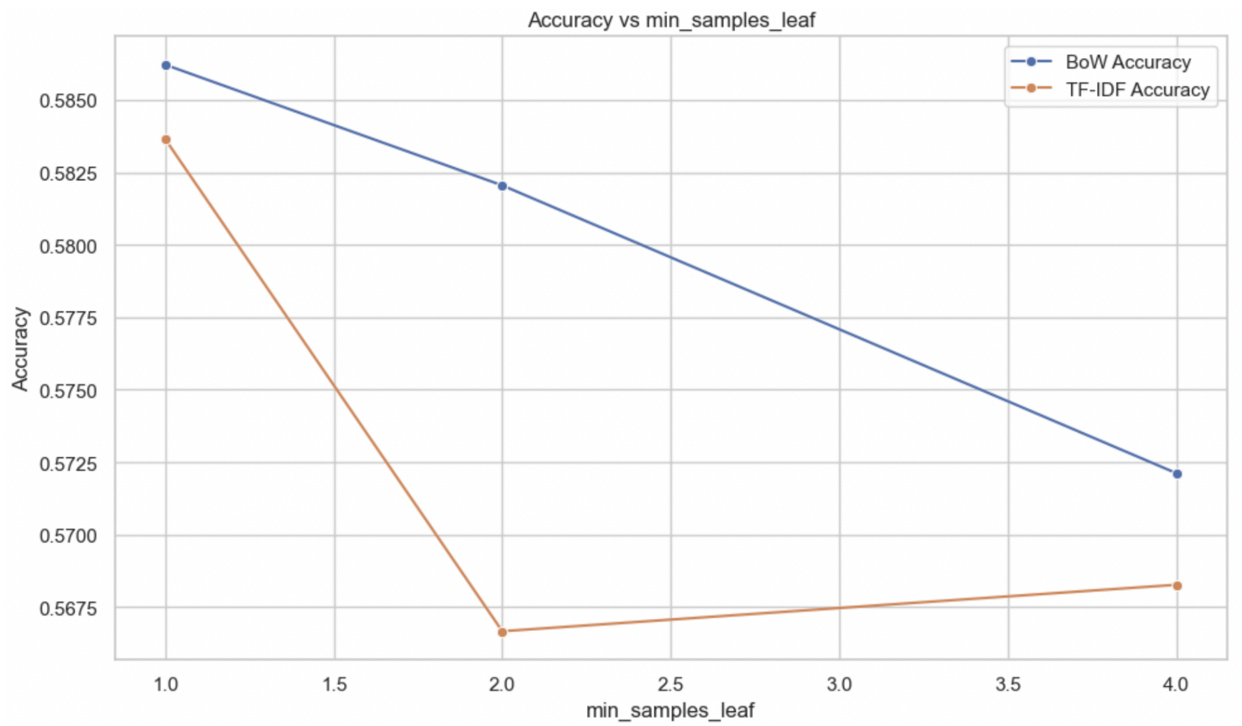
Explanation:

- Small splits increase model complexity and risk overfitting.

- BoW performs well at small splits; TF-IDF improves with regularization.

Conclusion: TF-IDF benefits from higher min_samples_split (~10) to reduce overfitting. BoW is robust even at lower values.

Accuracy vs. Min Samples Leaf



Interpretation:

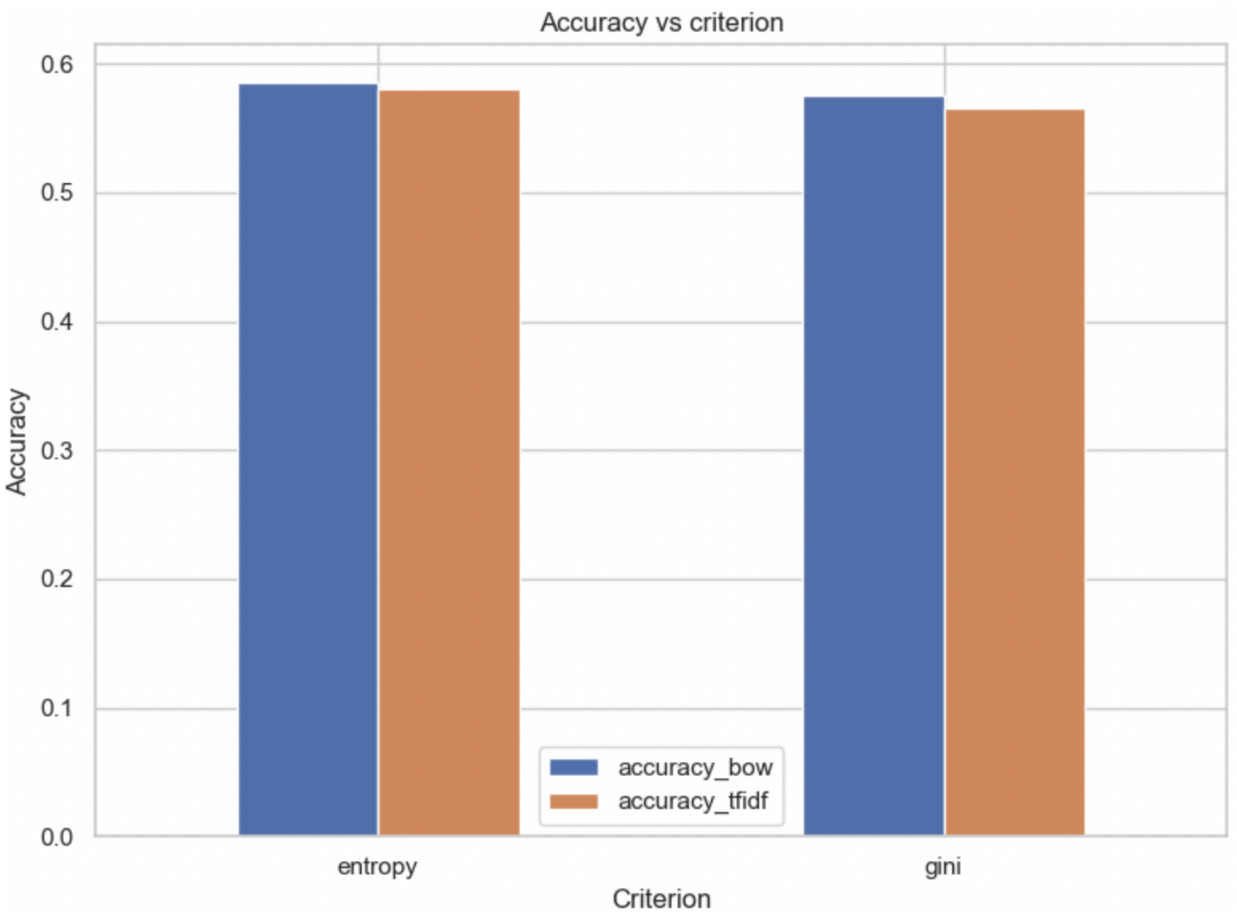
- BoW peaks at min_samples_leaf = 1 and declines steadily.
- TF-IDF shows a sharp drop at 2, then stabilizes at a lower level.

Explanation:

- BoW thrives with minimal leaf constraints due to frequent, strong features.
- TF-IDF over-regularizes easily, losing performance.

Conclusion: Keep min_samples_leaf low for BoW; apply slight regularization for TF-IDF.

Accuracy vs. Criterion (Entropy vs. Gini)



Criterion	BoW Accuracy	TF-IDF Accuracy
Entropy	~0.586	~0.581
Gini	~0.578	~0.567

Interpretation:

- Entropy consistently outperforms Gini across both feature sets.
- BoW outperforms TF-IDF for both criteria.

Explanation:

- Entropy captures subtle splits in sparse data better.
- Gini is faster but slightly less precise.

Conclusion: Entropy is the preferred criterion, especially with BoW, for capturing key sentiment-indicative patterns.

Why the Model Performs Better Without SVD

Applying SVD reduces the feature space to only **100 components**, significantly limiting the model's ability to capture nuanced textual patterns. This drop in performance is expected and can be explained by:

- **Information Loss**: Truncating the feature space compresses meaningful distinctions between sentiments, especially in sparse and context-dependent data like text.
- **Over-compression**: For tree-based models, which are naturally non-linear, reducing features can **over-regularize** the input, removing important split candidates.
- **Sparse Feature Strength**: BoW and TF-IDF in full form provide **thousands of distinct features**, capturing rich details; SVD compromises this advantage.

Conclusion

Tree-based models benefit more from the **original high-dimensional representation**, where they can leverage specific word patterns that get blurred after dimensionality reduction.