Sentiment Data Classification using ANN: Analysis Report

1. Data Preprocessing

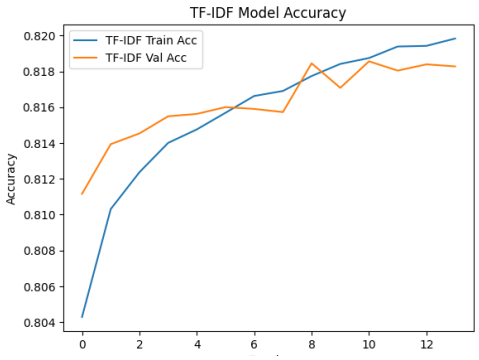
In the initial stage, the dataset underwent comprehensive text cleaning to ensure the input was suitable for machine learning models. The preprocessing steps included:

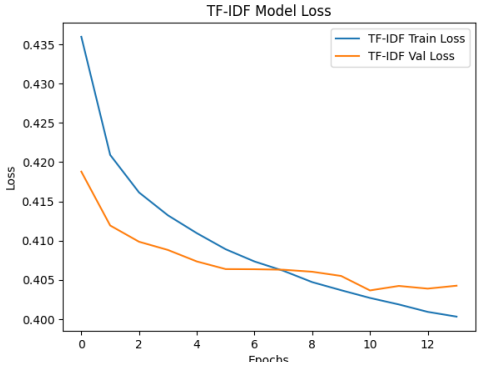
- Cleaning: Removal of unwanted characters, punctuations, numbers, and converting all text to lowercase.
- Stopword Removal: Eliminated common stopwords that do not contribute significant meaning to sentiment analysis.
- Lemmatization: Reduced words to their base or dictionary form to maintain consistency.
- N-grams: Applied bigram (2-gram) transformation to capture contextually significant word pairs that may better represent sentiment than single words.

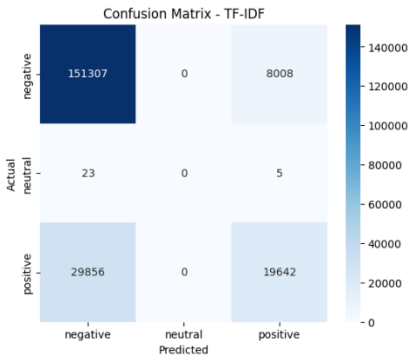
2. TF-IDF vs. Bag of Words (BoW)

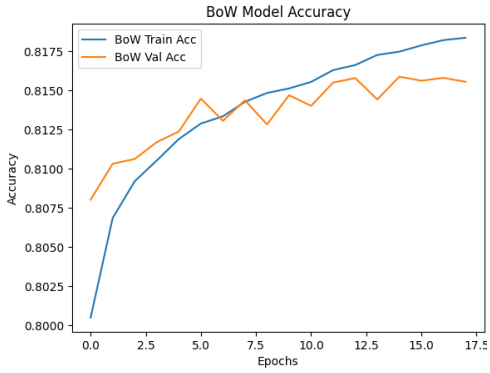
Two popular text vectorization techniques were compared:

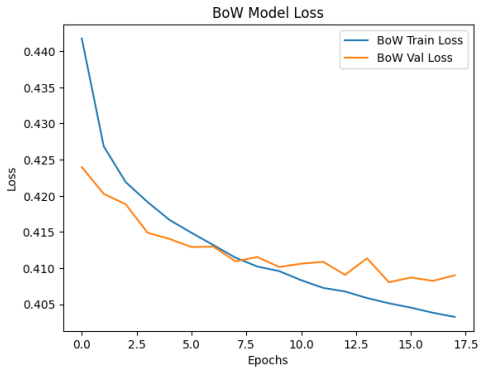
- Bag of Words (BoW): Represented text as the frequency of words without considering their relative importance or position.
- TF-IDF (Term Frequency-Inverse Document Frequency): Enhanced representation by weighing terms based on their frequency across documents, giving more importance to rarer terms.

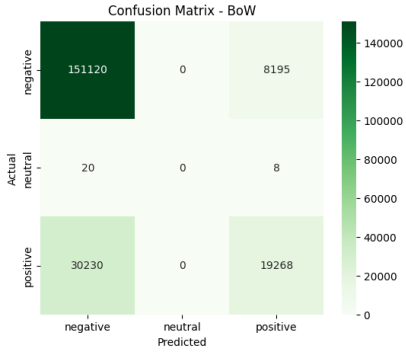












Comparison Metrics:

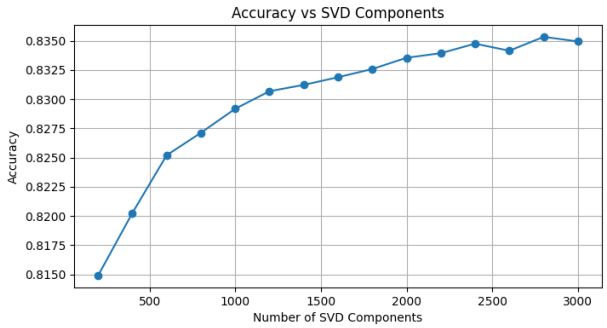
Metric	TF-IDF	BoW
Accuracy	**0.8186**	**0.8159**
F1 (Negative)	**0.89**	**0.89**
F1 (Positive)	**0.51**	**0.50**
F1 (Neutral)	**0.00**	**0.00**
Macro F1	**0.47**	**0.46**
Weighted F1	**0.80**	**0.80**

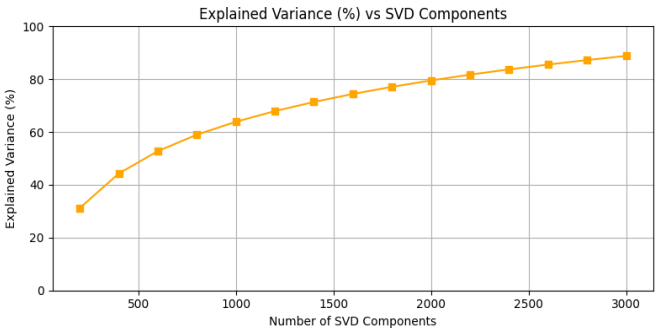
Although the Bag of Words (BoW) model achieved slightly higher test accuracy than TF-IDF (by just 0.27%), I chose to proceed with TF-IDF for further analysis because its training and validation accuracies are more closely aligned. This suggests better generalization and lower risk of overfitting compared to BoW, where the gap between training and validation accuracy was more pronounced. Hence, TF-IDF offers a more reliable representation for building a robust model

3. SVD Component Selection

To reduce the dimensionality of TF-IDF vectors and extract the most relevant features, Truncated Singular Value Decomposition (SVD) was applied. Experiments were conducted using a wide range of SVD components (200 to 3000). For each configuration:

- The percentage of explained variance was recorded.
- Classification accuracy using the transformed features was evaluated.
- Plots were generated to visualize the relationship between the number of components, explained variance, and accuracy.





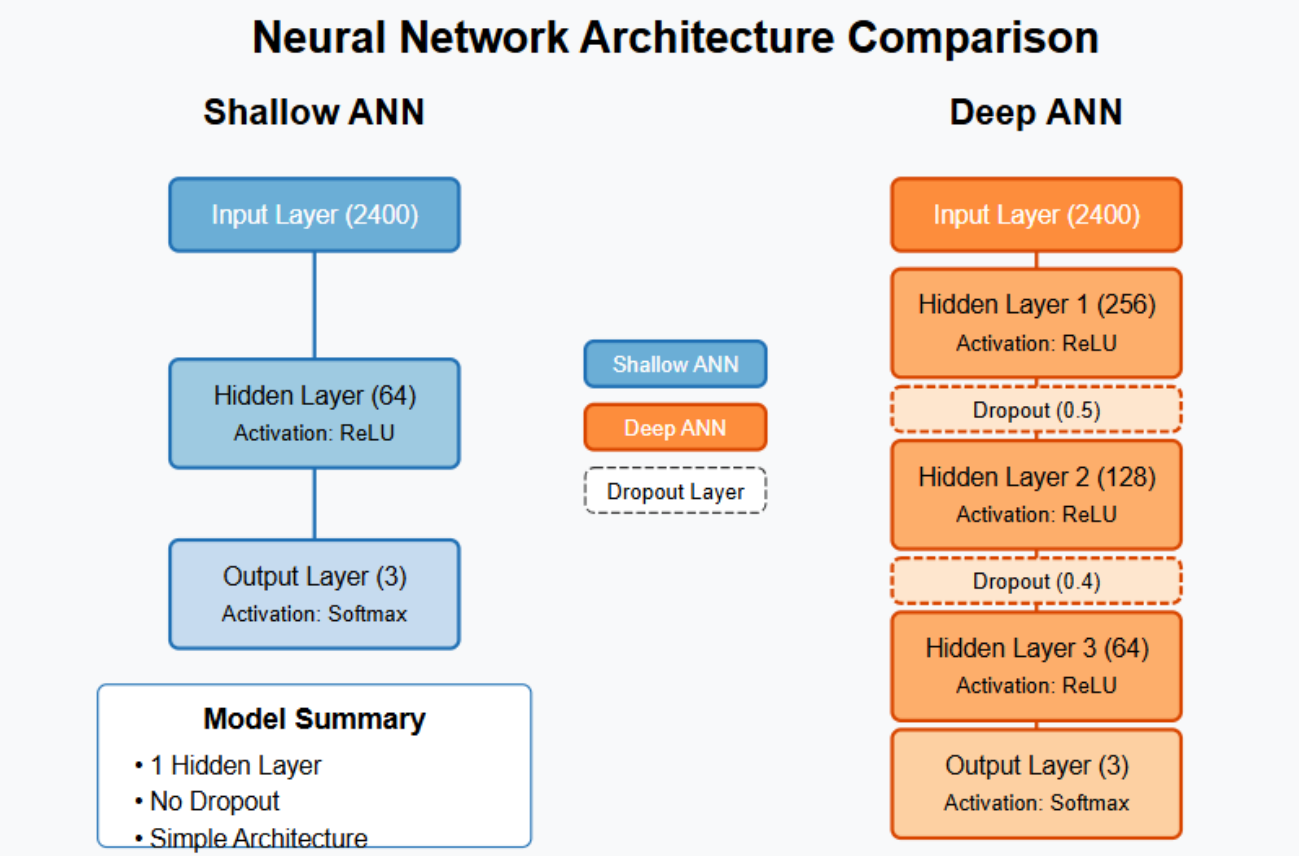
The explained variance plot shows a steady increase with the number of SVD components, reaching around 89% at 3000 components. However, the rate of gain slows down significantly beyond 2000 components, indicating diminishing returns. The accuracy plot follows a similar

trend—accuracy improves consistently up to around 2400 components, beyond which it saturates and exhibits minor fluctuations. These observations suggest that increasing components beyond a certain point offers limited benefits, and around 2400 components provides a good balance between dimensionality and performance.

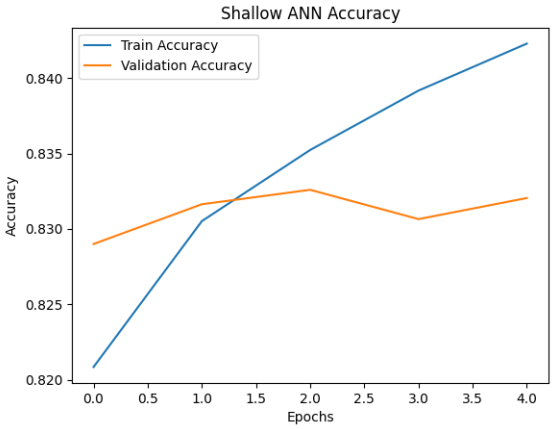
4. Shallow vs. Deep ANN

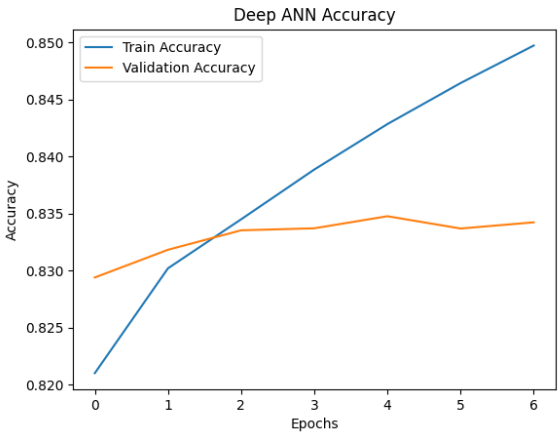
Two different Artificial Neural Network (ANN) architectures were evaluated to assess their performance on sentiment classification:

- **Shallow ANN**: Featured fewer layers and neurons, representing a simpler architecture.
- **Deep ANN**: Included multiple hidden layers, higher neuron counts, and dropout layers to enhance learning capacity and mitigate overfitting.



Both models were trained using the same input features derived from **TF-IDF vectorization followed by Truncated SVD (dimensionality reduction)**. The comparison focused on **training/validation accuracy** and **classification metrics** on the test set.





Accuracy Trends:

- **Training Accuracy**: The deep ANN showed a steady increase in training accuracy, eventually surpassing the shallow model (85.0% vs. 84.2%).
- **Validation Accuracy**: Both models plateaued around ~83.3%, with the deep ANN maintaining slightly more stability across epochs, suggesting better generalization.

Metric	Shallow ANN	Deep ANN
Accuracy	83.26%	83.48%
F1 Score	81.93%	82.35%
Precision	82.20%	82.46%
Recall	83.26%	83.48%
F1 (Negative)	90%	90% (approx)
F1 (Positive)	59%	59% (approx)
F1 (Neutral)	0%	0%
Macro Avg F1	50%	50%
Weighted Avg F1	82%	82%

Classification Performance:

- The deep ANN achieved marginally higher **overall accuracy (83.48%)** compared to the shallow ANN (**83.26%**).
- The **F1-score**, a balanced indicator of precision and recall, improved from **0.8193 (shallow)** to **0.8235 (deep)**.

- The deep model also slightly outperformed the shallow model in **precision (82.46% vs. 82.20%)** and **recall (83.48% vs. 83.26%)**.

Class-Wise Insight:

- Both models performed strongly on the **"negative"** class, which dominated the dataset.
- The **"positive"** class showed moderate performance.
- The **"neutral"** class had negligible representation and was poorly predicted in both models, which skewed the macro-average metrics.

While both ANNs achieved strong results, the **deep ANN demonstrated marginally better generalization and robustness**, as seen in the validation performance and classification report. The improvement, though slight, suggests that added depth can be beneficial when paired with regularization strategies like dropout.

5. Final Performance metrics of the Deep ANN architecture used

ANN Accuracy: 0.8348

Class	Precision	Recall	F1-Score	Support
Negative	0.86	0.94	0.90	159,315
Neutral	0.00	0.00	0.00	28
Positive	0.72	0.50	0.59	49,498
Accuracy			0.83	208,841
Macro avg	0.53	0.48	0.50	208,841
Weighted avg	0.82	0.83	0.82	208,841

The Deep ANN model shows strong overall performance with **83.48% accuracy**. It performs best on the **negative class** (F1: 0.90) and reasonably well on the **positive class** (F1: 0.59). However, it fails to capture the **neutral class**, likely due to the very low number of samples (only 28).

The **macro average F1-score** is **0.50**, reflecting the poor neutral class performance, while the **weighted F1-score** is **0.82**, indicating good overall effectiveness. To improve, addressing class imbalance is recommended.

In conclusion, the Deep ANN performs well for the majority classes (negative and positive), but addressing the **neutral class imbalance** through data augmentation, re-weighting, or synthetic sampling could further enhance its performance