Comparative Analysis of ML Models for Sentiment Classification on Amazon Product Reviews

Kai Gowers
gowers@bc.edu

Abstract

This project compares four machine learning models for binary sentiment classification of product reviews: Logistic Regression, a Basic Neural Network, an LSTM Recurrent Neural Network, and a fine-tuned DistilBERT Transformer. I evaluated model performance using two datasets: a domain-specific dataset of 5,000 dog food reviews and a general dataset of 100,000 Amazon product reviews. Preprocessing included binary labeling of reviews, tokenization, padding, and a standard data split. My results demonstrate that while Logistic Regression performs well on smaller clean datasets, model performance improves with architectural complexity and data scale. DistilBERT achieves the highest accuracy (95%) but requires significant computational resources. My study highlights trade-offs between model complexity, interpretability, data requirements, and training cost.

1. Introduction

Customer reviews are overwhelmingly numerous online, especially on platforms like Amazon, but manually filtering through thousands of them is inefficient. My motivation stems from personal experience: searching reviews is tedious and lacks structure. To address this, I designed a system to automatically classify reviews as positive or negative using machine learning. This project investigates:

- How different ML models perform on sentiment classification.
- How well they generalize across datasets of different sizes and domains.
- The balance between simplicity, accuracy, and computational cost. My contributions include a comparative analysis of four model types and dataset-specific insights on performance and resource efficiency.

2. Related Work

Sentiment analysis, or opinion mining, has evolved significantly from rule-based systems and keyword spotting to sophisticated machine learning and deep learning models. Early tools like NLTK's SentimentIntensityAnalyzer or TextBlob offered fast lexicon-based approaches but struggled with understanding nuance, sarcasm, or contextual sentiment shifts. These tools assign polarity scores based on predefined dictionaries, which limits their effectiveness on diverse or domain-specific texts.

Traditional machine learning methods such as Logistic Regression and Support Vector Machines (SVM) became popular for text classification due to their interpretability and performance on structured, clean datasets. When paired with vectorization techniques like TF-IDF (Term Frequency-Inverse Document Frequency), these models can achieve surprisingly good results by identifying the importance of words across documents. However, they assume word independence and ignore word order, which limits their ability to capture complex linguistic structures.

With the rise of deep learning, neural networks began outperforming traditional models, particularly for NLP tasks. Basic feedforward neural networks with word embeddings improved upon earlier methods by learning distributed word representations and nonlinear decision boundaries. However, they still fall short in modeling sequential dependencies.

To address this, researchers introduced Recurrent Neural Networks (RNNs) and, more effectively, Long Short-Term Memory (LSTM) networks, which are designed to retain memory across sequences. LSTMs are capable of modeling long-range dependencies and have been successfully applied to tasks such as machine translation and document classification. However, their training is slow and they require substantial data to generalize well.

The latest advancement in NLP is the adoption of Transformer architectures, which use self-attention mechanisms to model relationships between all words in a sentence simultaneously, regardless of their distance. BERT (Bidirectional Encoder Representations from Transformers) and its

distilled version DistilBERT represent the state-of-the-art in contextual language modeling. Pretrained on massive corpora and fine-tuned on specific tasks, these models achieve unparalleled performance but at high computational costs.

My project draws from this lineage of research, deliberately comparing models across this spectrum—from classic linear models to powerful Transformer-based architectures—to better understand trade-offs in performance, interpretability, and efficiency across datasets of different sizes and domains.

3. Data Preparation

I used two primary datasets.

Dog Food Reviews Dataset:

- A domain-specific set of 5,000 Amazon reviews focused on dog food products.
- Categorizes reviews with "id", "date", "author url", "rating", "review title", "review text", and "meta data".
- Appended with 300 synthetic reviews generated via ChatGPT to introduce informal and colloquial language variations.

Amazon Product Reviews Dataset:

- A large, general-purpose collection of 100,000 reviews spanning diverse categories, used primarily to support training of data-hungry architectures.
- Each data point contains values such as "Product ID", "User ID", "ProfileName", "Summary", and "Text".

4. Preprocessing

All reviews were assigned binary sentiment labels by converting star ratings:

- 1–2 stars → Negative (0)
- 3–5 stars → Positive (1)

Texts were tokenized using a shared vocabulary of the most frequent 20,000 tokens, padded or truncated to a length of 200 tokens, and split into 70% training, 20% validation, and 10% test sets. I applied the same splits across models to ensure comparability.

5. Logistic Regression

I implemented Logistic Regression as a baseline model. Input text was vectorized using a TF-IDF vectorizer, which transforms raw text into a sparse matrix representation reflecting the relative importance of each word. Hyperparameters used were:

- TF-IDF max_features = 5000
- max_iter = 1000

```
Validation Results
Accuracy: 0.8665254237288136
              precision    recall  f1-score   support

           0       0.87      0.47      0.61       211
           1       0.87      0.98      0.92       733

    accuracy                           0.87       944
   macro avg       0.87      0.73      0.77       944
weighted avg       0.87      0.87      0.85       944

Test Results
Accuracy: 0.8919491525423728
              precision    recall  f1-score   support

           0       0.94      0.51      0.66        98
           1       0.89      0.99      0.94       374

    accuracy                           0.89       472
   macro avg       0.91      0.75      0.80       472
weighted avg       0.90      0.89      0.88       472
```

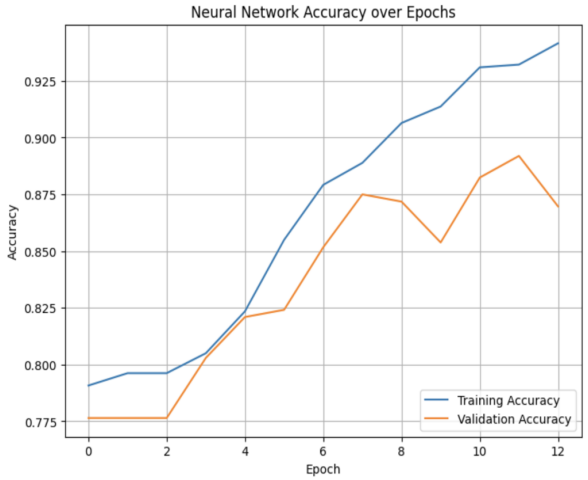
This model was trained on the Dog Food Review Dataset. It achieved 89% accuracy, performing well on positive reviews but with lower recall on negative examples due to class imbalance. For negative reviews, it had a 0.94 precision, but only a 0.51 recall and a 0.66 f1-score. For positive reviews, it had a 0.89 precision, and a 0.99 recall and 0.94 f1-score. Its simplicity and fast training time made it ideal for small, clean datasets.

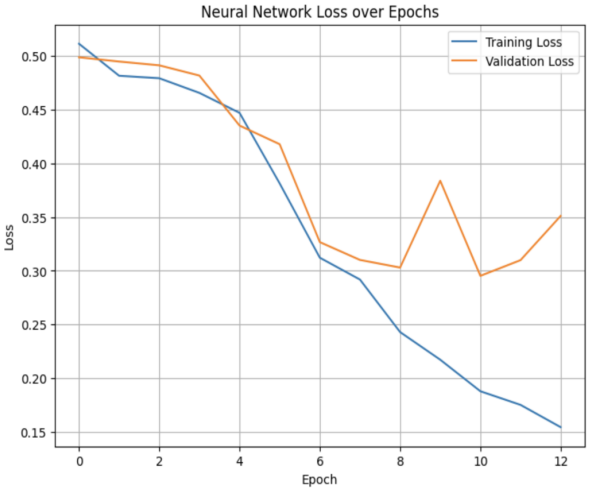
6. Basic Neural Network

The second model introduced nonlinearity through a shallow feedforward neural network. In my application, each dog-food review is first converted into a sequence of word indices, which an embedding layer maps into dense vectors. I then aggregate those vectors into a fixed-length summary (via global average pooling) and feed that summary through a pair of fully connected layers with ReLU activations before producing a single sigmoid-activated output for binary sentiment classification. Hyperparameters used were:

- Tokenizer(num_words = 5000)
- input_dim = 5000
- output_dim = 128
- Adam(learning_rate = 0.0005)
- patience = 2

This model was also trained on the Dog Food Review Dataset. It reached 90% accuracy and learned complex patterns more effectively than the linear model. It started



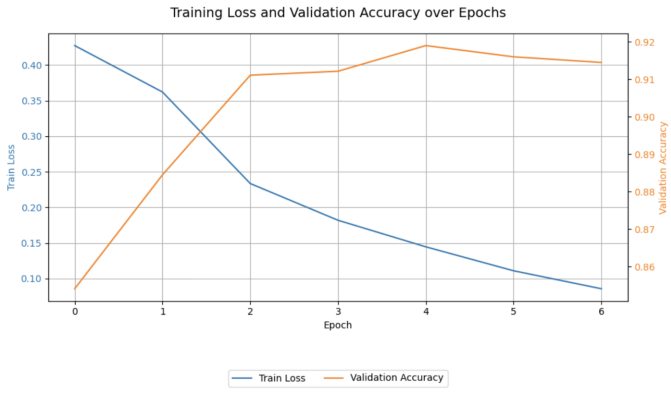


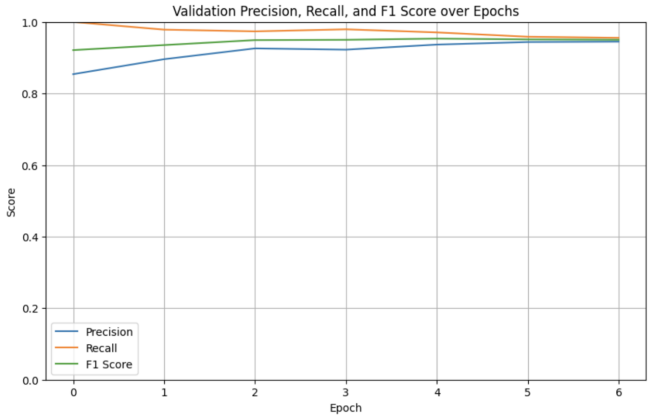
with a <0.800 training accuracy, but increased to over 0.925. However, slight overfitting emerged after 7 epochs, as shown in training/validation loss divergence.

7. LSTM Recurrent Neural Network

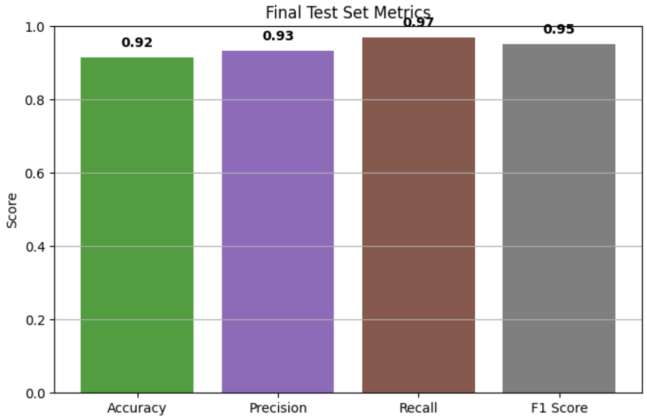
The third model I introduced was a Long Short Term Memory Recurrent Neural Network. LSTM models are a type of recurrent neural network designed to capture long-range dependencies in sequential data by using gated memory cells that learn when to "remember" or "forget" information. In my pipeline, each review is first tokenized and converted into a fixed-length sequence of word indices, which an embedding layer then maps into dense vectors. Those embeddings feed into an LSTM layer, and I extract the last hidden state and pass it through a small fully-connected network with a sigmoid output for binary sentiment prediction. Because LSTMs thrive on large datasets, I trained this model on my 100000-sample Amazon review dataset (70% train / 20% val / 10% test), minimizing binary cross-entropy via the Adam optimizer and applying early stopping on validation accuracy to prevent overfitting. Hyperparameters used were:

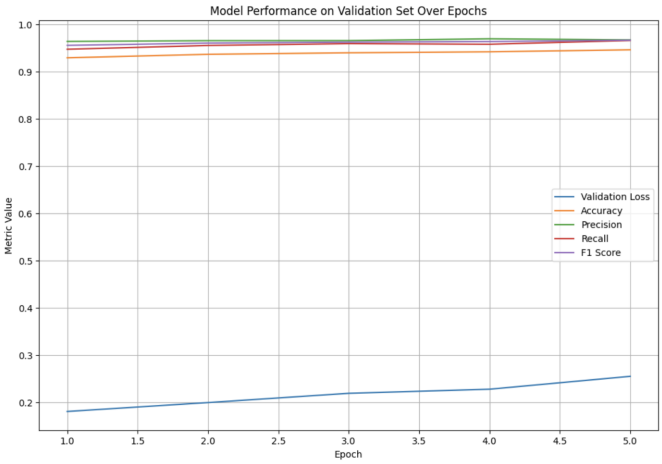
- VOCAB_SIZE = 5000
- EMBEDDING_DIM = 128
- HIDDEN_DIM = 128
- OUTPUT_DIM = 1
- patience = 2
- best_val_acc = 0
- epochs_no_improve = 0
- early_stop = False
- EPOCHS = 10





The LSTM model required substantial training time but achieved high precision, recall, and F1-scores across classes. It quickly increased its validation accuracy over the first 2 epochs, then leveled out at over 0.91. Its final test set metrics were a 0.92 accuracy, 0.93 precision, 0.97 recall, and 0.95 f1-score. It handled long-term dependencies well and significantly outperformed previous models on more complex, diverse input.



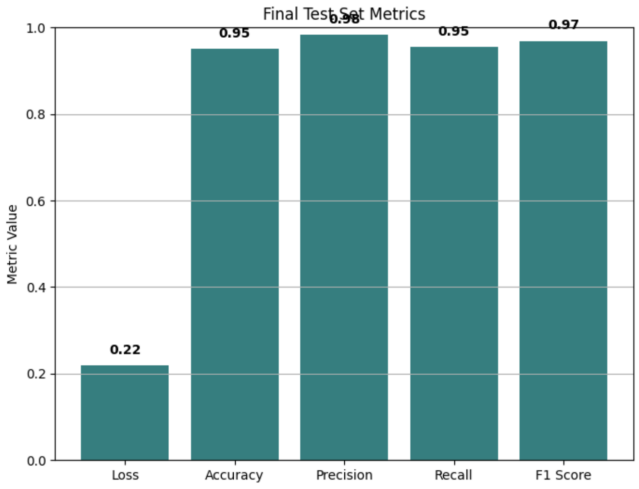


8. DistilBERT Transformer Model

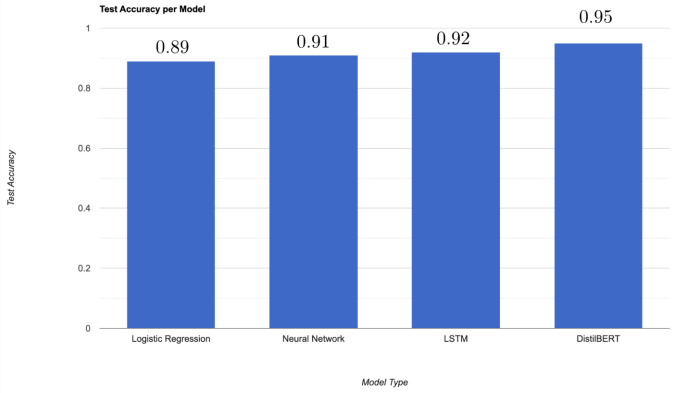
My final model was DistilBERT, a distilled version of the BERT model pre-trained on a large English corpus. I fine-tuned it on the Dog Food Review Dataset using the HuggingFace Transformers library. To adapt it for dog-food sentiment classification, I tokenized each review with the "distilbert-base-uncased" tokenizer—padding and truncating to the model's maximum length—then fine-tune the pre-trained model (with a two-label classification head) on my 4,719-sample dog food review dataset (70% train / 20% validation / 10% test). During fine-tuning I optimized binary cross-entropy loss with a low learning rate (1×10^{-5}), small batch sizes (8), and weight decay (0.01) across five passes over the data, tracking accuracy and F1 on the validation set after each epoch. This end-to-end approach transfers rich language representations from large-scale pretraining directly to my domain, yielding state-of-the-art sentiment accuracy with minimal manual feature engineering. Hyper-parameters used were:

- AutoTokenizer.from_pretrained(model_checkpoint)
- learning_rate = 1e-5
- per_device_train_batch_size = 8
- per_device_eval_batch_size = 8
- num_train_epochs = 1
- weight_decay = 0.01

Despite its compute demands and longer training time (20 minutes with GPU), DistilBERT achieved 95% accuracy with excellent generalization, precision (0.98), and F1-score (0.97). It started at above 0.90 on the 1st epoch, and only increased 0.01-0.02 accuracy. It was the top performer overall.



9. Final Results



The final test accuracy results for each model were:

- Logistic Regression - 0.89
- Neural Network - 0.91

- LSTM - 0.92
- DistilBERT - 0.95

10. Conclusion

This comparative study demonstrates that model selection for binary sentiment classification entails nuanced trade-offs between simplicity, performance, and computational cost. Logistic Regression, with TF-IDF features, provides a fast and interpretable baseline, achieving near-90% accuracy on small, clean datasets. Introducing nonlinearity via a shallow neural network yields modest gains but requires hyperparameter tuning and vigilance against overfitting. LSTM networks leverage sequential information effectively when trained on large, diverse corpora, delivering robust performance across metrics; however, their longer training times can be prohibitive for rapid prototyping. Finally, DistilBERT sets a new performance ceiling by harnessing contextualized embeddings from large-scale pretraining, though fine-tuning demands significant GPU resources and careful scheduling.