E-Commerce Text Classification

DNSC 4280: Machine Learning

Jesse Mutamba, Henrique Cassol, Maya Serna, Matthew Wolf

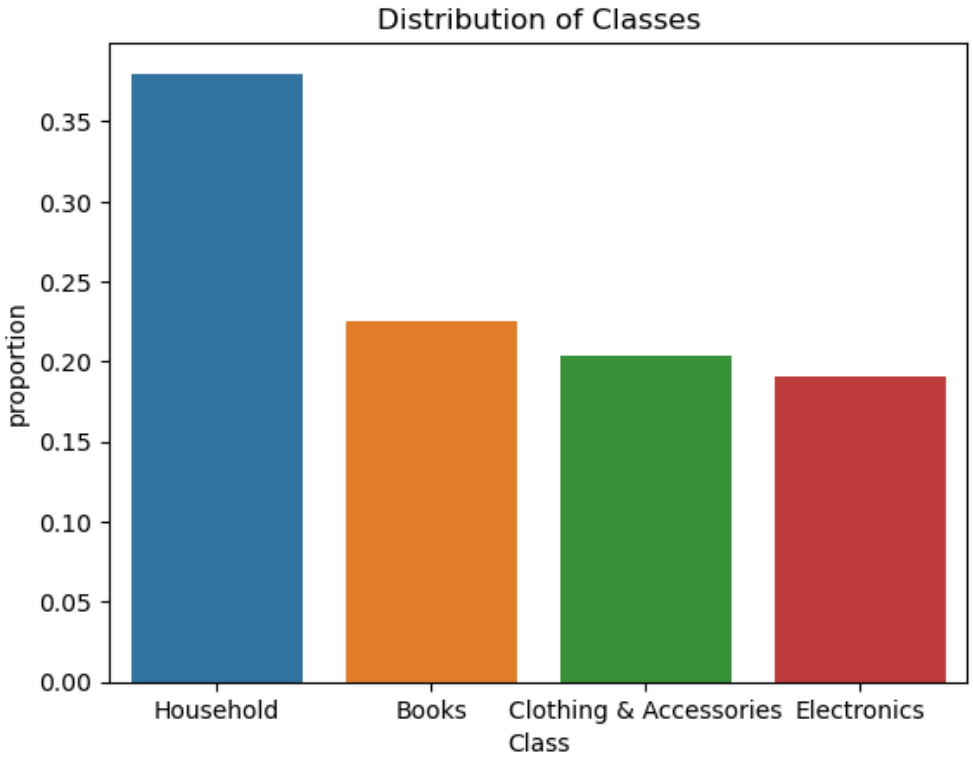
December 14, 2025

Introduction

Currently, E-Commerce platforms receive thousands if not millions of new products daily. If every product uploaded to one of these e-commerce platforms was done manually, it could be slow and costly. The goal of this project was to build a classifier that was as accurate as possible in order to distinguish between classes present in an e-commerce dataset. The dataset used was sourced from Kaggle, and contained product descriptions (text) and an associated label for each product type. The original source of the data was scraped from an Indian e-commerce platform. The product classes present in the data were as follows: household, books, clothing & accessories, & electronics.

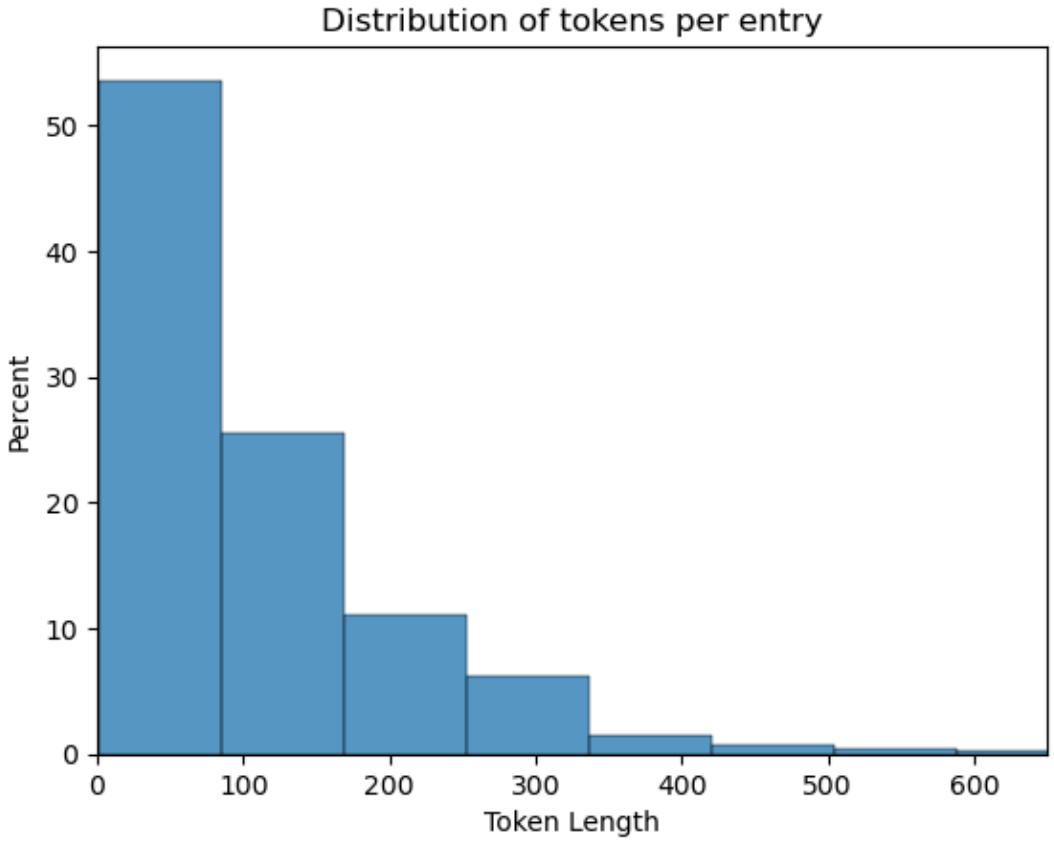
Methodology/EDA/Preprocessing/Tokenization

The technology used for this project was python for programming, keras/tensor-flow for constructing deep learning models, and sklearn for constructing "base" models. The first step taken before modeling was getting a better understanding of the dataset and the content included within it. Out of the ~50,000 entries within the data, 2 of the 4 classes, household was the most prominent at 38%, books were next at 22%, clothing and accessories after that at 20%, and finally electronics at 19% (see figure 1 below).



In order to select the best model for this task, multiple different models were created for comparison and selection. The "base" models which the deep learning models would look to improve upon were logistic regression, and a support vector machine. The deep learning models

used were a RNN, a LSTM, and transformers from HuggingFace. The base-learner models used a tf-idf word embedding, while the RNN and LSTM models used task specific learned embedding (hugging face models were pre-trained). The 95th percentile of tokens per entry was 312, so the training and test data was padded to 325 to try and capture the majority of information. Further, the average entry in the dataset contained 115 tokens, the full distribution is shown below



Results

Base Models:

Logistic Regression

The logistic regression model was fitted using a tf-idf vectorization with max features of 1000, and an n-gram range of (1,2). When tested on holdout data, it achieved an accuracy score of 0.926. Overall, this performance is very promising, and the next models will look to improve upon it.

SVM

The SVM model used the same tf-df vectorization as the logistic regression, although in presentation it was run with higher max feature (15,000) and no duplicates dropped, so it

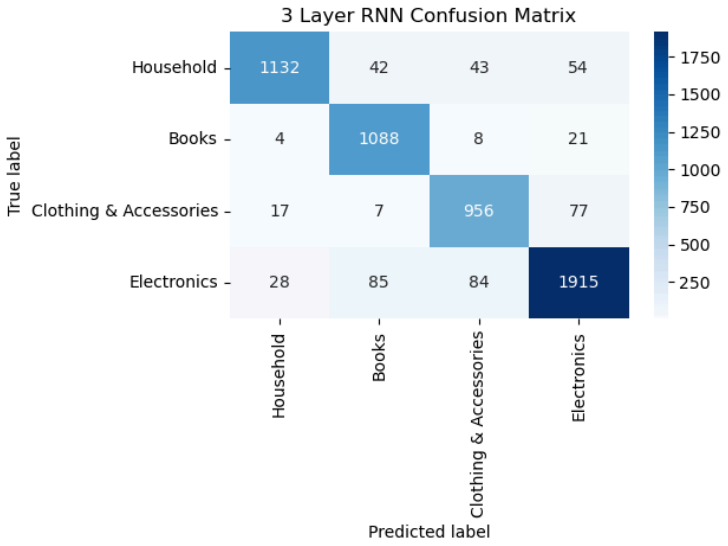
achieved an inflated score there, ~0.98. However, when run with the same methodology of the above logistic regression, the SVM attained a comparable accuracy score of 0.927. It is important to note that as the max_features of the tf-idf embedding increases, as does the model's accuracy.

Deep Learning Models:

The tokenizer used for the deep learning models had a vocabulary of 20,000 in order to match up with the input dimension used in these model's embedding layers.

RNN

The 3-layer RNN began with an embedding layer, with an input dimension of 20,000 and an output dimension of 64. The embedding layer was followed by 3 hidden RNN layers, all with 64 neurons and utilizing dropout learning with a rate of 0.3. After the RNN layers, there was a dense layer with 64 neurons before class predictions were generated by a softmax activation output layer. The model was fit using a batch size of 64, and 7 epochs. In order to help curb

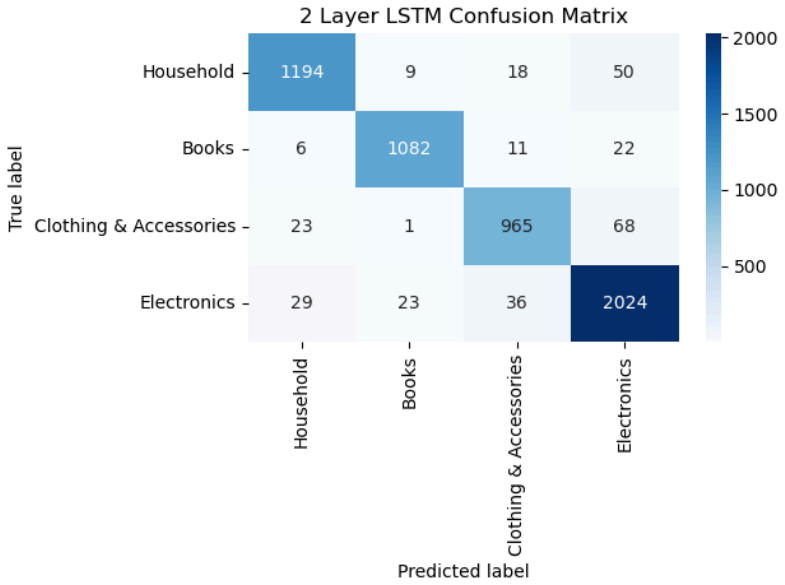


overfitting, this model also utilized early stopping. The RNN achieved an accuracy of 0.9784, and when it was used on unseen test data, a score of 0.911 was attained. The confusion matrix of the RNN was generated, and it appeared that the model in particular struggled distinguishing true household items. The RNN struggled with identifying the subtle differences between the other classes, possibly because it would not be unreasonable for books, clothing, and electronics to be under the umbrella of household items more generally.

LSTM

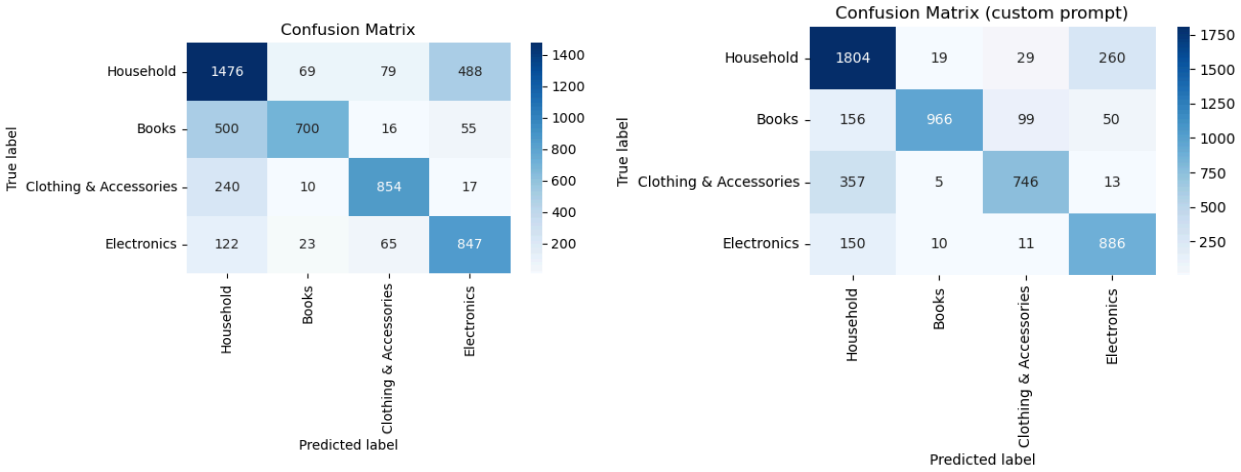
The LSTM model first had an embedding layer with an input dimension of 20,000 and an output dimension of 64. Following that, it had two LSTM layers each with 64 outputs and dropout learning with a rate of 0.3. Unlike the RNN model, the LSTM did not include a dense

layer before its softmax output layer. Further, the LSTM also did not employ early stopping. The LSTM was fitted on the training data using a batch size of 64 and 3 epochs. It achieved a training accuracy of 0.953, and a test accuracy of 0.9468, indicating that it was less overfit in comparison to the RNN. The LSTM confusion matrix reflected this accuracy increase, with household items being classified correctly at a rate ~5% higher than the RNN.



Transformers

The final model type that was fit was a pre-trained transformer from hugging face. Specifically, the one used was called bart-large-mnli, and it was trained on the MultiNLI dataset. Bart has 400 million parameters, and when it was run on the test dataset of ~5500 observations it took around 3 hours. Bart was run on the test data 2 times, once without a task specific prompt, and the next time with a prompt. When it was run without a prompt, it attained an accuracy of 0.697, indicating that although the model itself had a wide array of knowledge to pull from, it likely was not specific enough. The second time the model was run, a prompt was included to try and give it some more context: prompt was "This amazon product description is about {}." With this additional context, the model improved significantly, achieving an accuracy of 0.791 on the test set. By giving just a little more context related to the nature of the task and data available to the model, it was able to learn significantly. The confusion matrices for the two models are shown below(left no prompt, right prompted). In particular it can be seen how when the prompt was added to the model, true household accuracy went up significantly with the model being better able to distinguish between household and electronic items.

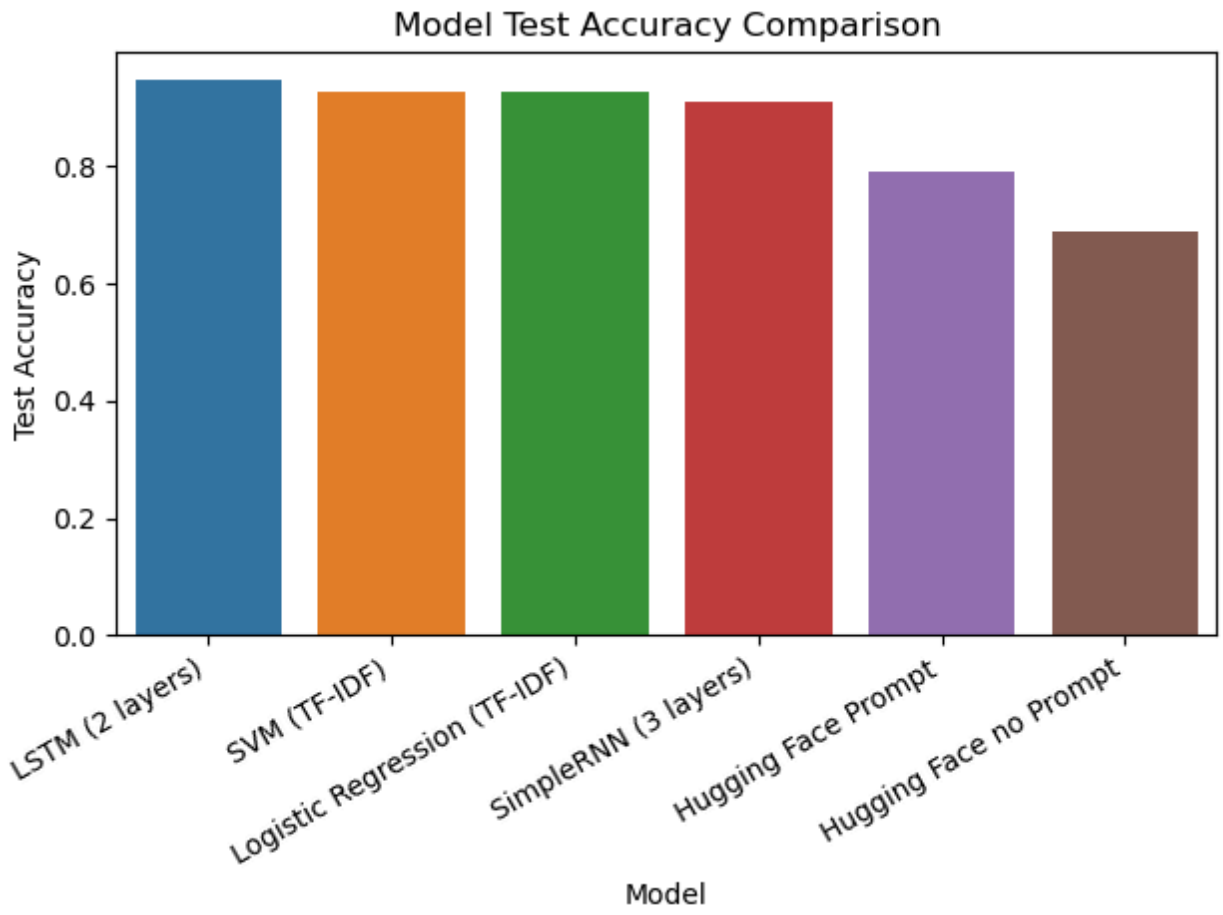


Limitations and Future Work

One of the main limitations of this project was the nature of the dataset. The dataset's information was underdocumented. The only information that was provided was descriptions of each column, and it was not indicated how the category labels were derived. Without knowing how the categories were developed, it is therefore unclear what *exactly* the accuracy of the models created are capturing. Given more time and resources, potentially implementing LoRA on the hugging face models used in order to update a small subset the model's weights for the task at hand would have been explored. Additionally, the use of different transformer architectures would have been explored as well.

Conclusion

Overall, the most advanced deep learning model ended up having the best success in terms of accuracy on holdout data. Specifically the LSTM achieved a 94.68% accuracy on the test set. "Base" learners (Logistic and SVM) both had accuracies in the ~92% range, with the more complex RNN underperforming in comparison with 91%. The most underwhelming models used were the hugging face transformers, however, when adding additional context, model accuracy jumped from 69% to 79%, signaling more room to potentially grow. The use case of models such as these can allow online retailers to optimize the labeling of online products, potentially saving potential energy and costs used on manually inputting product labels. Further advancement and improvements of these models should utilize LoRA or other parameter efficient fine tuning methods.



References

"Facebook/Bart-Large-Mnli · Hugging Face." *Huggingface.co*, huggingface.co/facebook/bart-large-mnli.

Gautam. "E Commerce Text Dataset." *Zenodo (CERN European Organization for Nuclear Research)*, 31 July 2019, www.kaggle.com/datasets/saurabhshahane/ecommerce-text-classification, https://doi.org/10.5281/zenodo.3355823. Accessed 18 Nov. 2024.