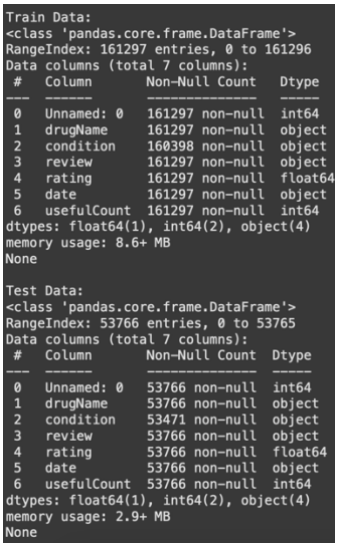
Dataset Selection and Business Use Case

The Drug Review Dataset from Drugs.com, which was obtained from the UCI Machine Learning Repository. It has over 200,000 reviews and separated into 161,297 training reviews and 53,766 test reviews. offering ample data for both deep learning and traditional NLP models. These reviews consist of valuable patient feedback that can be used for sentiment analysis—a critical task to comprehend patient experiences, determining concerns, and enhancing drug offerings. The dataset consists of 6 variables where 5 are categorial and 1 date. Based on ratings, sentiment variable is created. Two methods were used to analyze the dataset which are Random Forest with TF-IDF and AWD_LSTM with ULMFiT. Random Forest was picked for its effectiveness and interpretability in feature-engineered tasks, while AWD_LSTM was chosen for its capacity to use transfer learning to capture intricate semantic patterns in text. Figure 1 illustrates a preview of the data.

Business Problem: Pharmaceutical businesses and healthcare providers can do the following by using sentiment analysis of drug reviews:

- Evaluate the side effects and efficacy of drugs.
- Improve customer service by resolving frequent problems.
- Adapt medication formulas to patient input.



```
Train Data:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 161297 entries, 0 to 161296
Data columns (total 7 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   Unnamed: 0   161297 non-null  int64  
 1   drugName     161297 non-null  object 
 2   condition    160398 non-null  object 
 3   review       161297 non-null  object 
 4   rating       161297 non-null  float64
 5   date         161297 non-null  object 
 6   usefulCount  161297 non-null  int64  
dtypes: float64(1), int64(2), object(4)
memory usage: 8.6+ MB
None

Test Data:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 53766 entries, 0 to 53765
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Unnamed: 0   53766 non-null  int64  
 1   drugName     53766 non-null  object 
 2   condition    53471 non-null  object 
 3   review       53766 non-null  object 
 4   rating       53766 non-null  float64
 5   date         53766 non-null  object 
 6   usefulCount  53766 non-null  int64  
dtypes: float64(1), int64(2), object(4)
memory usage: 2.9+ MB
None
```

Image 1: Test and train dataset

Preprocessing Steps and Constraints

Preprocessing is a crucial step for effective text classification, as it directly impacts model performance. For this assignment, key preprocessing procedures comprised the removal of non-ASCII characters and HTML entities, such as '&#039', for clean and uniformed text input. Text translation methods based on Python were used to accomplish this (GeeksforGeeks, 2023). HTML entities were also decoded to human-readable language (W3Schools, 2023). According to Dey (2021), the content was then normalized by applying tokenization, deleting unnecessary whitespace, and changing it to lowercase. Image 2 displays the text after preprocessing techniques.

Based on the rating variable, a sentiment column was created, classifying reviews with a rating of $\geqslant$5 as positive and those with a rating of <5 as negative. The supervised learning process relied heavily on this binary label. To guarantee balanced sentiment labels, the dataset was divided into training and testing subsets (80:20) .

To balance computational efficiency and feature coverage, text vectorization using TF-IDF was carried out for the conventional NLP approach with a constrained vocabulary size. On the other hand, because of computational limitations, the AWD_LSTM model used tokenized sequences, which resulted in a smaller dataset.

```
[ ] train_data.head()
```

	cleaned_review	sentiment	drugName	condition	rating	usefulCount
0	it ha no side effect i take it in combination ...	positive	Valsartan	Left Ventricular Dysfunction	9.0	27
1	my son is halfway through his fourth week of i...	positive	Guanfacine	ADHD	8.0	192
2	i used to take another oral contraceptive whic...	negative	Lybrel	Birth Control	5.0	17
3	this is my first time using any form of birth ...	positive	Ortho Evra	Birth Control	8.0	10
4	suboxone ha completely turned my life around i...	positive	Buprenorphine / naloxone	Opiate Dependence	9.0	37

```
test_data.head()
```

	cleaned_review	sentiment	drugName	condition	rating	usefulCount
0	ive tried a few antidepressant over the year c...	positive	Mirtazapine	Depression	10.0	22
1	my son ha crohn disease and ha done very well ...	positive	Mesalamine	Crohn's Disease, Maintenance	8.0	17
2	quick reduction of symptom	positive	Bactrim	Urinary Tract Infection	9.0	3
3	contrave combine drug that were used for alcoh...	positive	Contrave	Weight Loss	9.0	35
4	i have been on this birth control for one cycl...	positive	Cyclafem 1 / 35	Birth Control	9.0	4

Image 2: Test and Train data after preprocessing

Accuracy Comparison: Deep Learning vs. Traditional NLP

The accuracy of the deep learning model, AWD_LSTM with ULMFiT, exceeds that of the traditional NLP model using Random Forest with TF-IDF features. In particular, the Random Forest classifier obtained an accuracy of 77.6%, whilst the AWD_LSTM model achieved 92.8%. This suggests that deep learning has a distinct advantage when it comes to identifying intricate syntactic and semantic patterns in textual data.

The effectiveness of the AWD_LSTM model in capturing intricate linguistic patterns and modeling sequential dependencies is what makes it outstanding. According to Seth (2018), AWD_LSTM's robust architecture, which uses strategies like embedding dropout to avoid overfitting, makes it superior to conventional NLP methods. However, the Random Forest classifier uses artificial features, like TF-IDF, which can only identify superficial patterns in text data. According to ML Archive (2018), Random Forest's efficacy for complicated tasks is limited by its incapacity to dynamically adjust to contextual changes despite its computational efficiency and interpretability. Its confusion matrix and categorization report make these flaws clear.

Strong performance is demonstrated by the AWD_LSTM confusion matrix in both positive and negative classes, with little misclassification. For example, it incorrectly classifies 235 good reviews as negative and 175 negative reviews as positive, but correctly predicts 1,242 positive and 348 negative reviews (image 3). In contrast, the Random Forest model has a greater rate of misclassification. 6,734 positive and 1,026 negative reviews are correctly identified, whereas 1,950 positive and 290 negative reviews are incorrectly classified as positive and negative, respectively. The difference demonstrates the enhanced semantic understanding offered by AWD_LSTM.

This analysis is further supported by the classification report. The AWD_LSTM model's strong generalization is demonstrated by its F1-scores of 0.93 for the positive class and 0.85 for the negative class. The Random Forest model, on the other hand, struggles to manage complicated or unbalanced

sentiment structures, as evidenced by its F1-score of 0.48 for negative reviews and 0.86 for good ones. These conclusions are supported by screenshots (Images 3,4,5) of the classification reports and confusion matrices, which show the relative advantages and disadvantages of each method.

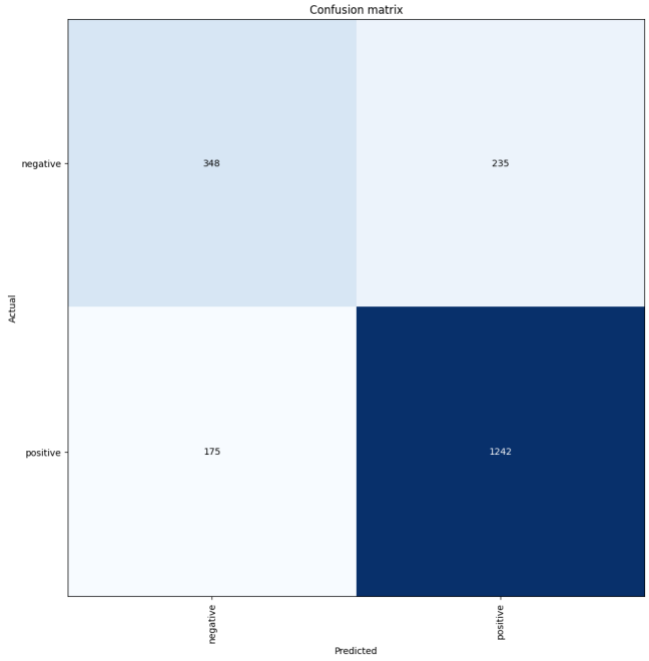


Image 3: Confusion matrix for AWD_LSTM

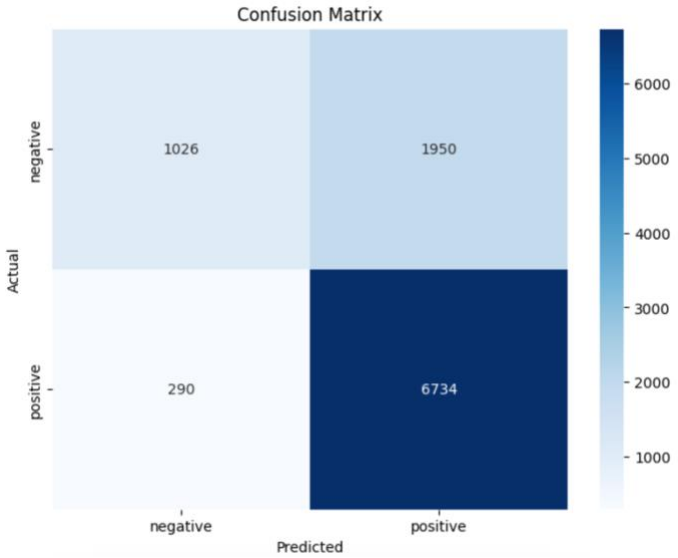


Image 4: Confusion Matrix for RF

```
Test Accuracy: 0.7760

Classification Report:
              precision    recall  f1-score   support

    negative       0.78      0.34      0.48      2976
    positive       0.78      0.96      0.86      7024

    accuracy                           0.78     10000
   macro avg       0.78      0.65      0.67     10000
weighted avg       0.78      0.78      0.74     10000
```

Image 5: Classification Matrix for Random Forest

Development Effort: Deep Learning vs. Traditional NLP

The AWD_LSTM model with ULMFiT took a large amount of work and computational resources to construct. Issues like GPU disconnections and session timeouts that occurred when using Google Colab interfered with training and reduced process efficiency. The complexity of the AWD_LSTM model was increased by the multi-stage training required to fine-tune it, which included controlling layer freezing,

learning rates, and dropout tactics. On the condensed dataset, each training run took approximately 3 to 4 hours. According to Seth (2018), AWD_LSTM uses sophisticated methods like embedding dropout and DropConnect to enhance prediction, which makes it very successful for complex sentiment classification tasks, while having high computing requirements.

Model Recommendation for Productization

AWD_LSTM with ULMFiT is the suggested model for productization because of its high accuracy (92.8%) and strong generalization across sizable and dynamic datasets. It is perfect for real-world applications needing high precision, such sentiment analysis in healthcare, because of its fine-tuning capabilities, which allow it to react to new or changing data. Furthermore, complex language patterns that are difficult for standard algorithms like Random Forest to handle are successfully captured by AWD_LSTM. However, because of its simplicity, quick training time, and interpretability, the Random Forest model with TF-IDF offers a good substitute in settings with limited resources or for smaller-scale deployments. In the end, Random Forest is better for faster, resource-efficient solutions, while AWD_LSTM is more appropriate for situations where scalability and high performance are critical.

References

Dey, S. (2021). Text Cleaning: The Secret Weapon for Smarter NLP Models. Medium. https://deysusovan93.medium.com/text-cleaning-the-secret-weapon-for-smarter-nlp-models-part-2-9c22b2f1bcd

GeeksforGeeks. (2023). Transliterating Non-ASCII Characters with Python. https://www.geeksforgeeks.org/transliterating-non-ascii-characters-with-python/

W3Schools. (2023). HTML Entities. https://www.w3schools.com/html/html_entities.asp

Seth, Y. (2018, September 12). AWD-LSTM explanation: Understanding language model. *Machine Learning Archive*. Retrieved from https://yashuseth.wordpress.com/2018/09/12/awd-lstm-explanation-understanding-language-model/

ML Archive. (2018). Sentiment analysis with Random Forest. *Machine Learning Archive*. Retrieved from https://mlarchive.com/machine-learning/sentiment-analysis-with-random-forest/