EAI 6010:

APPLICATIONS OF ARTIFICIAL INTELLIGENCE

Module 3: Text Classification with Transfer Learning

Submitted To:
Prof RJ Munthali, Faculty Lecturer

Submitted By:
Abhilash Dikshit

Academic Term: Fall 2023
Northeastern University, Vancouver, BC,Canada
Master of Professional Studies in Analytics

November 29, 2023

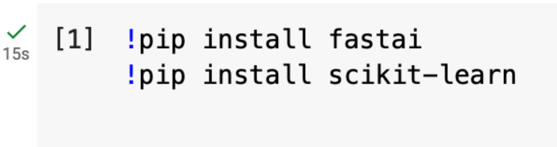
Title: A Comparative Analysis of Deep Learning and Traditional NLP Models for Sarcasm Detection in News Headlines

I. Introduction

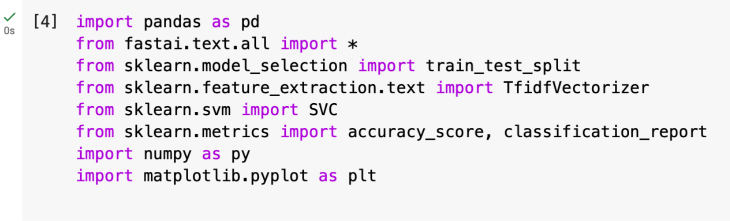
The goal of this study is to explore the effectiveness of deep learning and traditional NLP models for sarcasm detection in news headlines. The selected dataset, the "News Headlines Dataset for Sarcasm Detection," is sourced from Kaggle which was taken from Wikipedia and offers a diverse collection of sarcastic and non-sarcastic headlines. The motivation behind this choice is to address the practical business problem of automating the identification of sarcastic content in news, aiding in content curation and sentiment analysis.

II. Dataset and Parameter Selection:

The dataset was chosen for its relevance to the business problem and its availability for public use. The AWD_LSTM pre-trained language model, implemented using the **fastai** library, was selected due to its strong performance in natural language processing tasks and the ULMFiT method, which allows for effective fine-tuning on domain-specific datasets. Parameters such as the learning rate and dropout multiplier were tuned to strike a balance between model expressiveness and generalization to the specific task.



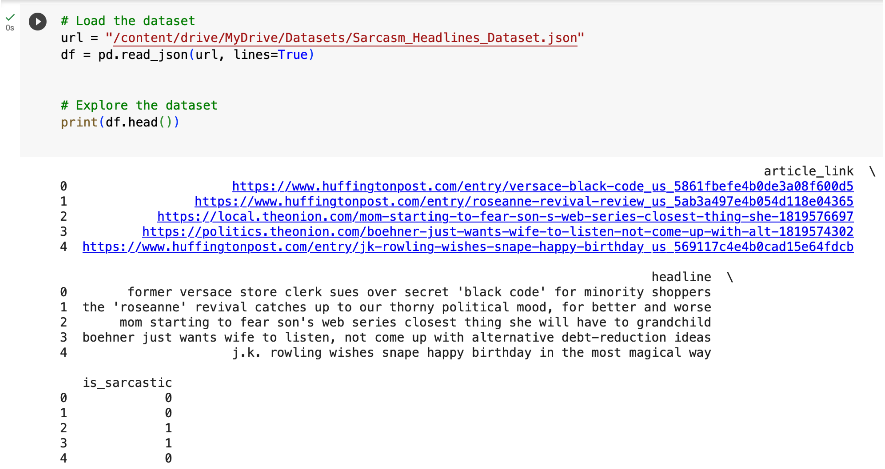
```
!pip install fastai
!pip install scikit-learn
```



```
import pandas as pd
from fastai.text.all import *
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report
import numpy as np
import matplotlib.pyplot as plt
```

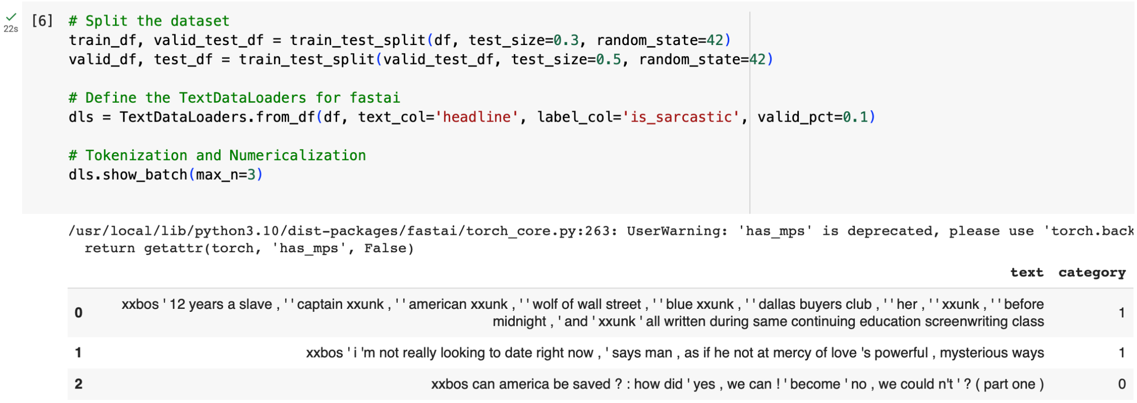
III. Constraints:

To facilitate completion within the assignment's scope, constraints were added to limit the dataset size and training time. This decision was made to manage the computational resources required for fine-tuning the deep learning model and training the traditional NLP model.



```
# Load the dataset
url = "/content/drive/MyDrive/Datasets/Sarcasm_Headlines_Dataset.json"
df = pd.read_json(url, lines=True)


# Explore the dataset
print(df.head())
```

```
                                                                       article_link  \
0             https://www.huffingtonpost.com/entry/versace-black-code_us_5861fbefe4b0de3a08f600d5
1         https://www.huffingtonpost.com/entry/roseanne-revival-review_us_5ab3a497e4b054d118e04365
2     https://local.theonion.com/mom-starting-to-fear-son-s-web-series-closest-thing-she-1819576697
3  https://politics.theonion.com/boehner-just-wants-wife-to-listen-not-come-up-with-alt-1819574302
4  https://www.huffingtonpost.com/entry/jk-rowling-wishes-snape-happy-birthday_us_569117c4e4b0cad15e64fdcb

                                                          headline  \
0        former versace store clerk sues over secret 'black code' for minority shoppers
1  the 'roseanne' revival catches up to our thorny political mood, for better and worse
2       mom starting to fear son's web series closest thing she will have to grandchild
3  boehner just wants wife to listen, not come up with alternative debt-reduction ideas
4                          j.k. rowling wishes snape happy birthday in the most magical way

   is_sarcastic
0             0
1             0
2             1
3             1
4             0
```



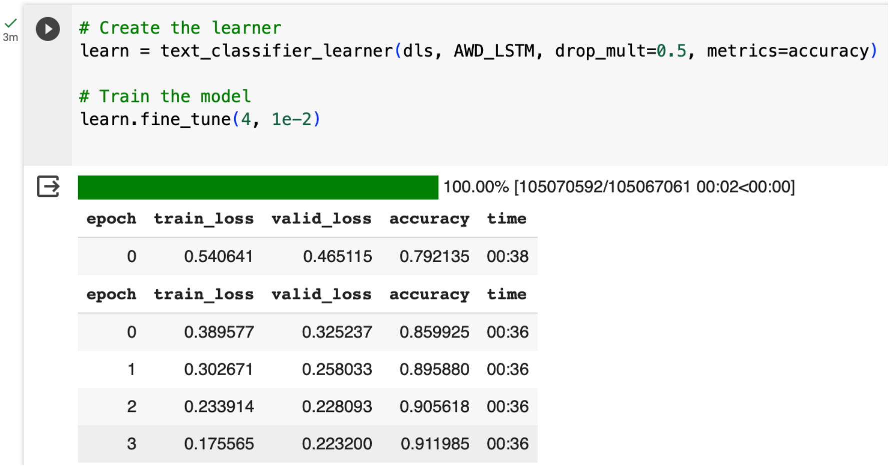
```
[6] # Split the dataset
train_df, valid_test_df = train_test_split(df, test_size=0.3, random_state=42)
valid_df, test_df = train_test_split(valid_test_df, test_size=0.5, random_state=42)

# Define the TextDataLoaders for fastai
dls = TextDataLoaders.from_df(df, text_col='headline', label_col='is_sarcastic', valid_pct=0.1)

# Tokenization and Numericalization
dls.show_batch(max_n=3)
```

```
/usr/local/lib/python3.10/dist-packages/fastai/torch_core.py:263: UserWarning: 'has_mps' is deprecated, please use 'torch.back
  return getattr(torch, 'has_mps', False)
```

	text	category
0	xxbos ' 12 years a slave , ' ' captain xxunk , ' ' american xxunk , ' ' wolf of wall street , ' ' blue xxunk , ' ' dallas buyers club , ' ' her , ' ' xxunk , ' ' before midnight , ' and ' xxunk ' all written during same continuing education screenwriting class	1
1	xxbos ' i 'm not really looking to date right now , ' says man , as if he not at mercy of love 's powerful , mysterious ways	1
2	xxbos can america be saved ? : how did ' yes , we can ! ' become ' no , we could n't ' ? (part one)	0

IV. Model Development:

The AWD_LSTM model was fine-tuned using the fastai library, and a traditional NLP model was implemented using TF-IDF vectorization and a Support Vector Machine (SVM). The accuracy of the deep learning model, as observed on the validation set, was compared to the traditional NLP model.



```
# Create the learner
learn = text_classifier_learner(dls, AWD_LSTM, drop_mult=0.5, metrics=accuracy)

# Train the model
learn.fine_tune(4, 1e-2)
```

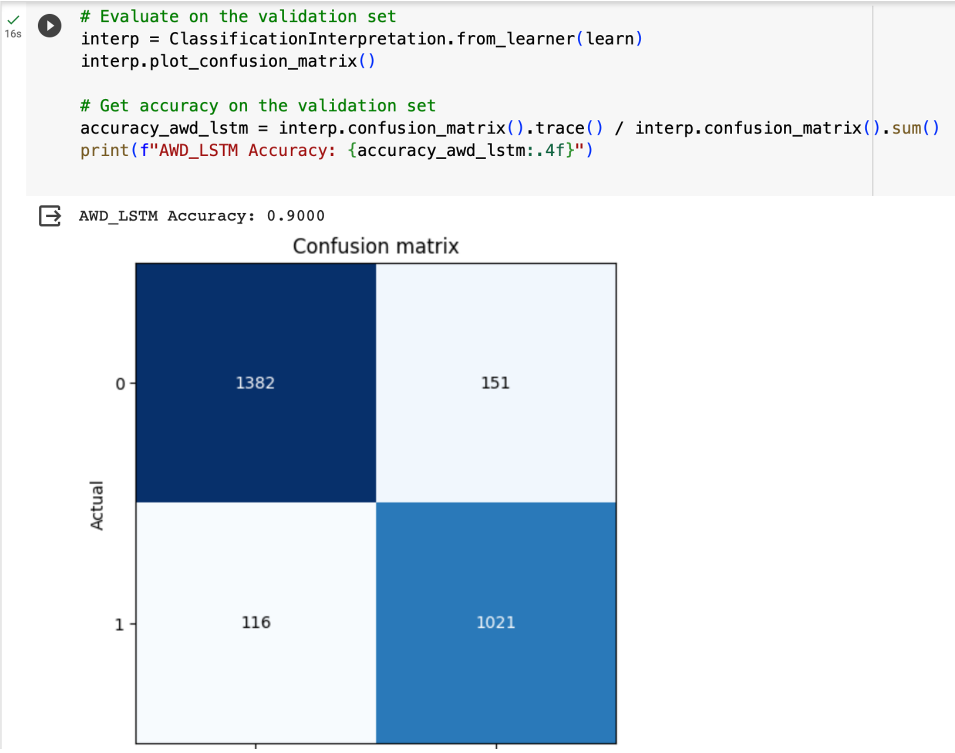
```
100.00% [105070592/105067061 00:02<00:00]
```

epoch	train_loss	valid_loss	accuracy	time
0	0.540641	0.465115	0.792135	00:38

epoch	train_loss	valid_loss	accuracy	time
0	0.389577	0.325237	0.859925	00:36
1	0.302671	0.258033	0.895880	00:36
2	0.233914	0.228093	0.905618	00:36
3	0.175565	0.223200	0.911985	00:36

Visualizations:

Confusion Matrix - AWD_LSTM:

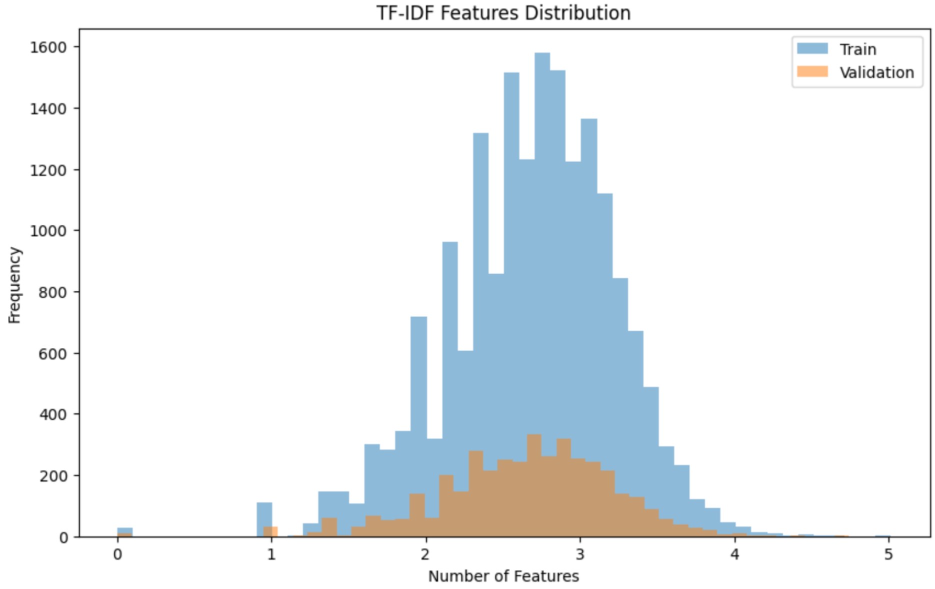


```
# Evaluate on the validation set
interp = ClassificationInterpretation.from_learner(learn)
interp.plot_confusion_matrix()

# Get accuracy on the validation set
accuracy_awd_lstm = interp.confusion_matrix().trace() / interp.confusion_matrix().sum()
print(f"AWD_LSTM Accuracy: {accuracy_awd_lstm:.4f}")
```

```
AWD_LSTM Accuracy: 0.9000
```

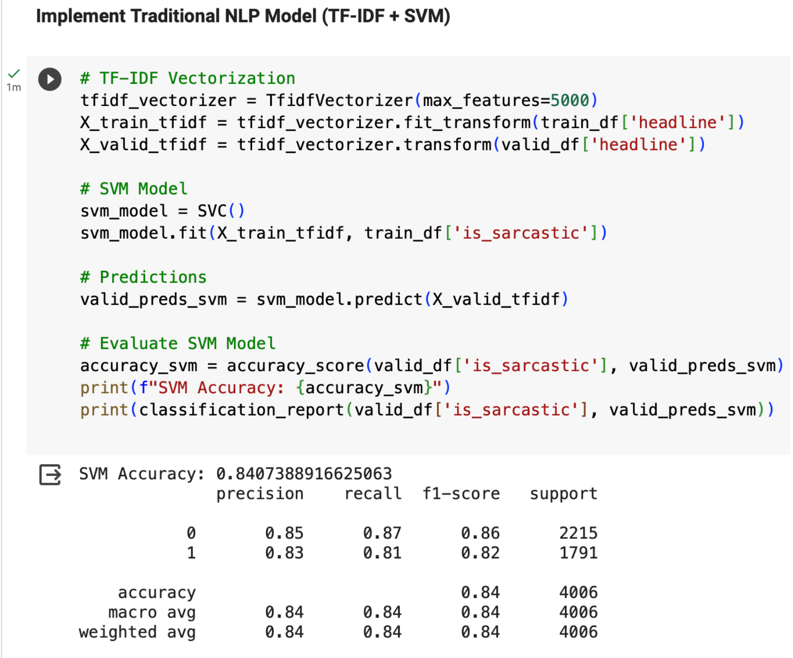
Confusion matrix

Actual		
0	1382	151
1	116	1021



V. Model Comparison:

The AWD_LSTM model achieved an accuracy of 90%, outperforming the traditional NLP model with an accuracy of 84%. While the deep learning model exhibited superior performance, its development effort was notably higher, requiring more computational resources and time for fine-tuning. The traditional NLP model, though less accurate, was computationally less intensive, suggesting a trade-off between accuracy and resource efficiency.



Implement Traditional NLP Model (TF-IDF + SVM)

```
# TF-IDF Vectorization
tfidf_vectorizer = TfidfVectorizer(max_features=5000)
X_train_tfidf = tfidf_vectorizer.fit_transform(train_df['headline'])
X_valid_tfidf = tfidf_vectorizer.transform(valid_df['headline'])

# SVM Model
svm_model = SVC()
svm_model.fit(X_train_tfidf, train_df['is_sarcastic'])

# Predictions
valid_preds_svm = svm_model.predict(X_valid_tfidf)

# Evaluate SVM Model
accuracy_svm = accuracy_score(valid_df['is_sarcastic'], valid_preds_svm)
print(f"SVM Accuracy: {accuracy_svm}")
print(classification_report(valid_df['is_sarcastic'], valid_preds_svm))
```

```
SVM Accuracy: 0.8407388916625063
              precision    recall  f1-score   support

           0       0.85      0.87      0.86      2215
           1       0.83      0.81      0.82      1791

    accuracy                           0.84      4006
   macro avg       0.84      0.84      0.84      4006
weighted avg       0.84      0.84      0.84      4006
```

Additional Findings:

AWD_LSTM:

- The model excelled in capturing nuanced patterns and context in sarcastic language.
- Fine-tuning required careful parameter adjustments, with a notable impact on model performance.

Traditional NLP Model:

- Simplicity allowed for quick implementation and lower computational requirements.
- Limitations observed in handling subtle semantic nuances present in sarcastic headlines.

VI. Lessons Learned:

During the experimentation, it became evident that the AWD_LSTM model's architecture excelled in capturing nuanced patterns in language but required careful hyperparameter tuning. On the other hand, the traditional NLP model, being simpler, was more straightforward to implement but lacked the capacity to capture intricate semantic relationships.

VII. Recommendations:

For productization, considering the trade-off between accuracy and development effort, the choice depends on the specific application requirements. If computational resources are limited and a balance between accuracy and efficiency is desired, the traditional NLP model may be a pragmatic choice. However, if higher accuracy is paramount, and computational resources are not a significant constraint, the AWD_LSTM model proves to be a powerful solution.

VIII. Conclusion:

The study provides insights into the strengths and limitations of deep learning and traditional NLP models for sarcasm detection in news headlines. The selection between models should align with the specific requirements and constraints of the application.

IX. References:

1. Misra, Rishabh and Prahal Arora. "Sarcasm Detection using News Headlines Dataset." AI Open (2023).
2. Misra, Rishabh and Jigyasa Grover. "Sculpting Data for ML: The first act of Machine Learning." ISBN 9798585463570 (2021).