THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Fake or Real News Classification

Group 7:

Aasish Kumar Immadisetty
Greeshmanjali Bandlamudi
Pranay Bhakthula

Introduction

- The goal of our project is to classify real and fake news based on the news data.
- This was implemented using NLP models.
- We have used one classical method i.e., Logistic Regression
- We have used 3 non-classical methods (Transformers) :
    1. DistilBERT
    2. DeBERTa
    3. RoBERTa

Dataset Description

- There are two sorts of articles in the dataset: fake and true news. The true articles were retrieved via crawling articles from Reuters.com. The fake news items were gathered from untrustworthy websites that Politifact (a fact-checking organization in the United States) and Wikipedia had highlighted
- The dataset contains a variety of articles on various themes; however, the majority of the articles are about politics and world events.
- Two CSV Files make up this dataset:
  1. True.csv (21,417 observations)
  2. Fake.csv (23,502 observations)

- The dataset contains 'title', 'text', 'subject', 'date', 'target’ columns.

Dataset Preprocessing

1. Initially, combined the Fake.csv and True.csv files into one dataset by assigning Fake news to ‘0’ and Real news to ‘1’.
2. Then merged ‘title’ and ‘text’ into one column called ‘text’.
3. Dropped ’title’, ‘subject’ and ‘date’ columns as they are not necessary.
4. Dropped rows with NULL values

Dataset Preprocessing

Step 1: Removing URL

Original text :

```
political hack: https://www.youtube.com/watch?v=h4xzSKcArdcThe Democrats
```

Function:

```
def remove_url(text):
    return re.sub(r'http\S+', '', text)
```

Cleaned text :

```
this political hack:  Democrats so desperately
```

Step 2: Apply Lower

Original text :

```
“This Is TOTALLY FAKE!” Tucker Loses His Temper
```

Function:

```
def to_lower(text):
    return text.lower()
```

Cleaned text :

```
“this is totally fake!” tucker loses his temper
```

THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

Dataset Preprocessing

Step 3: Applying Contractions

Original text :

```
take down this president that it's getting out of control
```

Function:

```
def remove_contractions(text):
    return ' '.join([contractions.fix(word) for word in text.split()])
```

Cleaned text :

```
'take down this president that it is getting out of control'
```

Step 4: Removing Punctuations

Original text :

```
'gets his point across to this political hack:'
```

Function:

```
def remove_punctuations(text):
    return re.sub(r'[^\w\s]', '', text)
```

Cleaned text :

```
gets his point across to this political hack
```

Dataset Preprocessing

Step 5: Removing Special Characters

Original text :

```
the Russia conspiracy with Mustafa Tameez, former
```

Function:

```
def remove_characters(text):
    return re.sub('[^a-zA-Z]', ' ', text)
```

Cleaned text :

```
the Russia conspiracy with Mustafa Tameez  former
```

Step 6: Removing Stopwords

Original text :

```
former consultant for the Department of Homeland Security
```

Function:

```
def remove_stopwords(text):
    return ' '.join([word for word in nltk.word_tokenize(text) if word not in stop_words])
```

Cleaned text :

```
former consultant Department Homeland Security
```

Dataset Preprocessing

Step 7: Stemming

Original text :

```
Tucker Carlson debates the Russia conspiracy with Mustafa Tameez,
```

Function:

```
def stemming_words(text):
    return ' '.join(stemmer.stem(word) for word in text.split())
```

Cleaned text :

```
tucker carlson debat the russia conspiraci with mustafa tameez
```

Step 8: Lemmatization

Original text :

```
Tucker goes off the rails after this political hack tries to pull
```

Function:

```
def lemmatize_words(text):
    return ' '.join(lemmatizer.lemmatize(word) for word in text.split())
```

Cleaned text :

```
tucker goe off the rail after thi polit hack tri to pull
```

Modeling:

Classical:

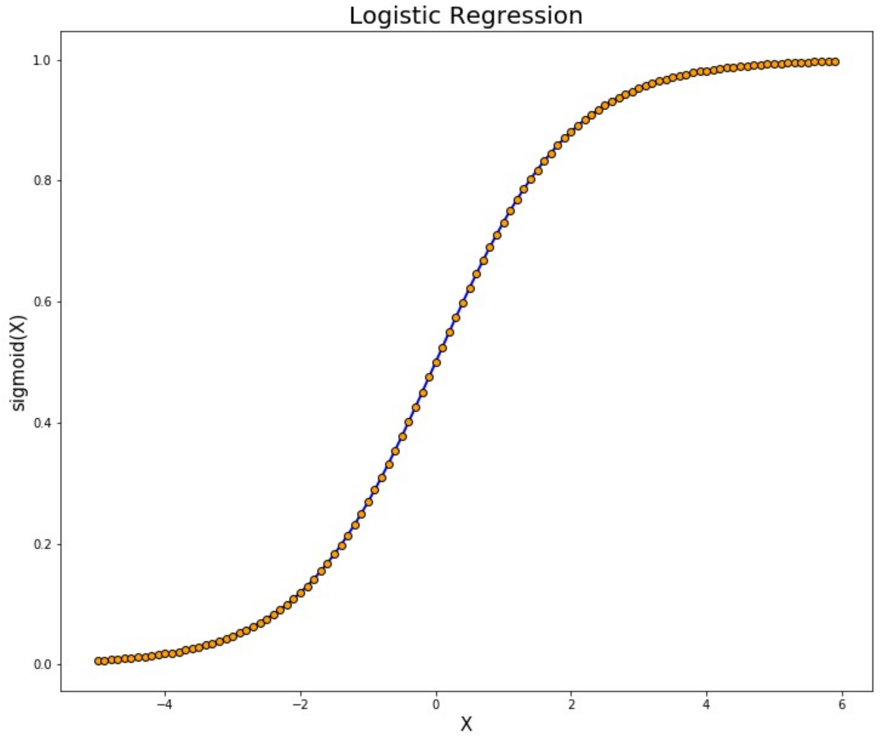
- Logistic Regression

Non-Classical:

- DistilBERT
- DeBERTa
- RoBERTa

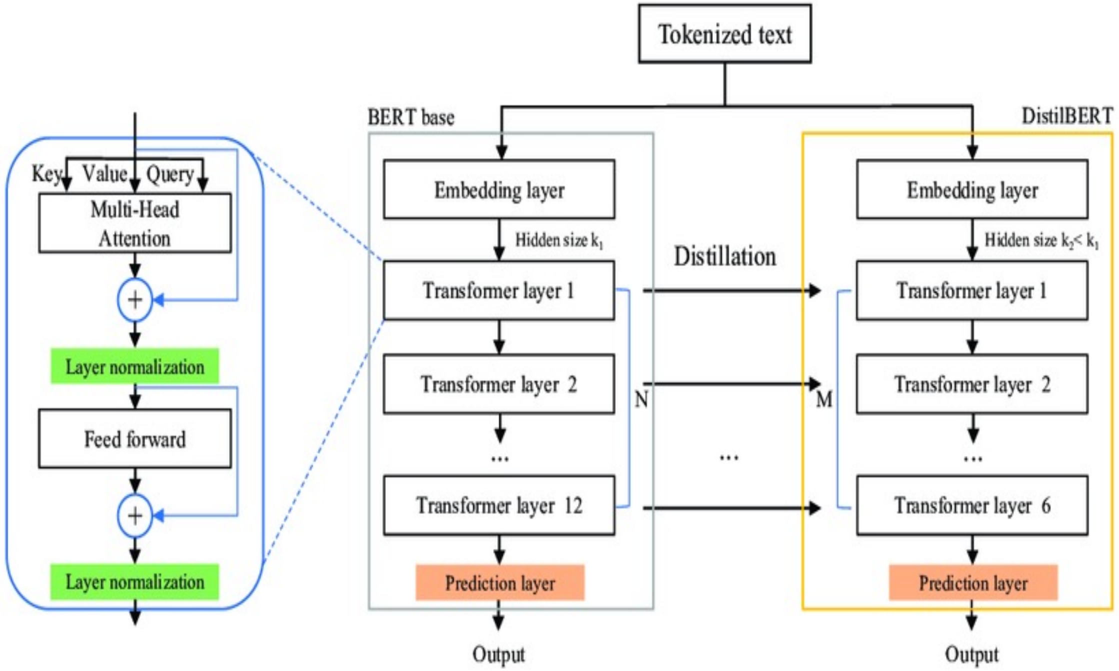
Modeling: Logistic Regression

In statistics, the (binary) logistic model (or logit model) is a statistical model that models the probability of one event (out of two alternatives) taking place by having the log-odds (the logarithm of the odds) for the event be a linear combination of one or more independent variables ("predictors").



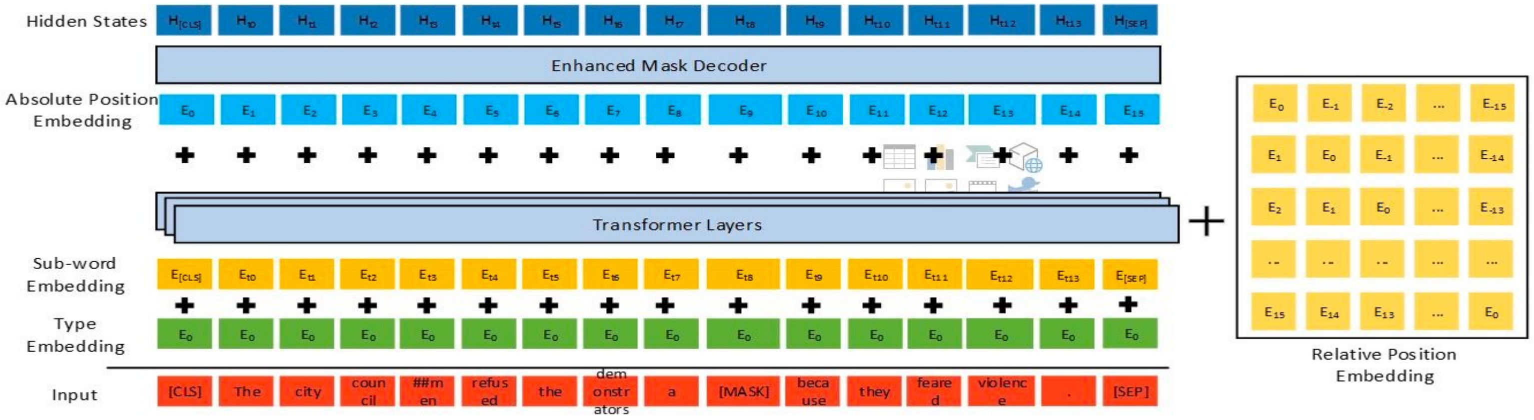
Modeling: DistilBERT

DistilBERT is a technique for pre-training a smaller general-purpose language representation model that can then be fine-tuned to perform effectively on a range of tasks, similar to its larger counterparts. While most previous research focused on utilizing knowledge distillation to develop task-specific models, we can lower the size of a BERT model by 40% while preserving 97 percent of its language understanding skills and being 60% faster by employing knowledge distillation during the pre-training phase. Additionally, it provides a triple loss that combines language modeling, distillation, and cosine-distance losses to take use of the inductive biases learned by larger models throughout the training process.



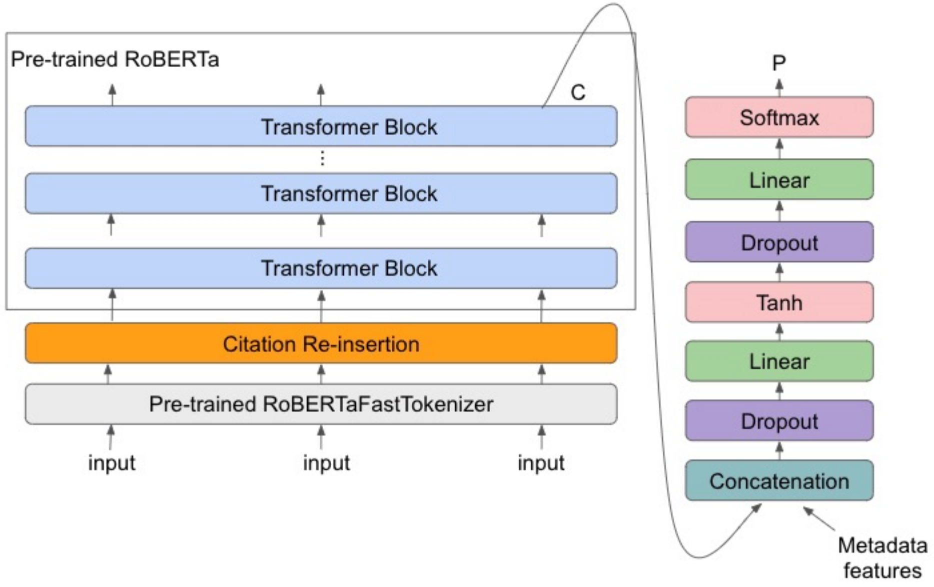
Modeling: DeBERTa

- DeBERTa (Decoding-enhanced BERT with Disentangled Attention) uses two unique strategies to improve the BERT and RoBERTa models.
- The first is the disentangled attention mechanism, in which each word is represented by two vectors that convey its content and position, and attention weights between words are calculated using disentangled matrices on their contents and relative positions.
- Second, instead of using the output SoftMax layer to predict the masked tokens for model pretraining, an upgraded mask decoder is employed. We show that these two strategies improve model pre training efficiency and downstream task performance significantly.



Modeling: RoBERTa

RoBERTa builds on BERT's language masking method, which teaches the system to predict purposely hidden content within otherwise unannotated language instances. RoBERTa modifies critical hyperparameters in BERT, such as deleting BERT's next-sentence pretraining target and training with considerably bigger mini-batches and learning rates, which was implemented in PyTorch. As a result, RoBERTa outperforms BERT on the masked language modeling objective, resulting in improved downstream task performance. We also look into training RoBERTa on an order of magnitude more data and over a longer period of time than BERT.



Model Results

Classical Model Results:

model	precision		recall		f1-score		accuracy
	0's	1's	0's	1's	0's	1's	
Logistic Regression	0.99	0.99	0.99	0.99	0.99	0.99	0.9875

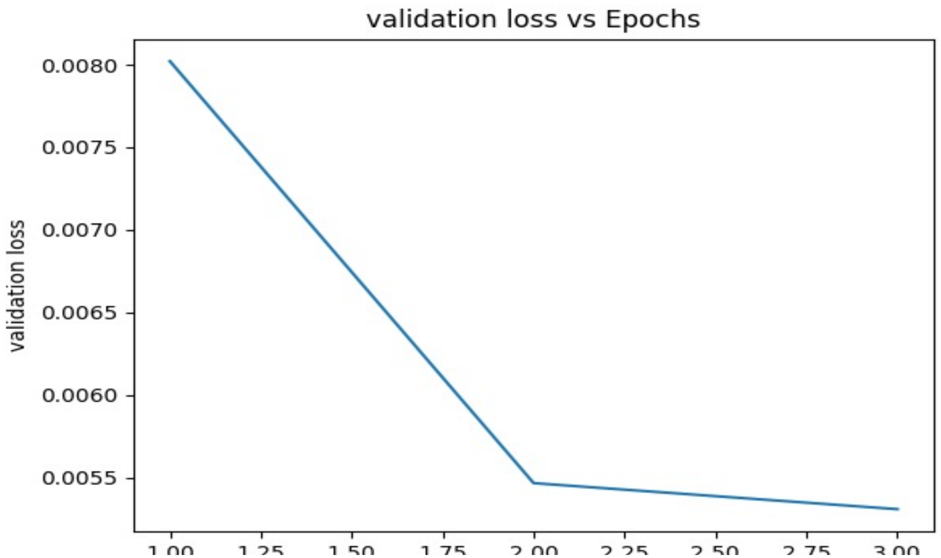
Model Results

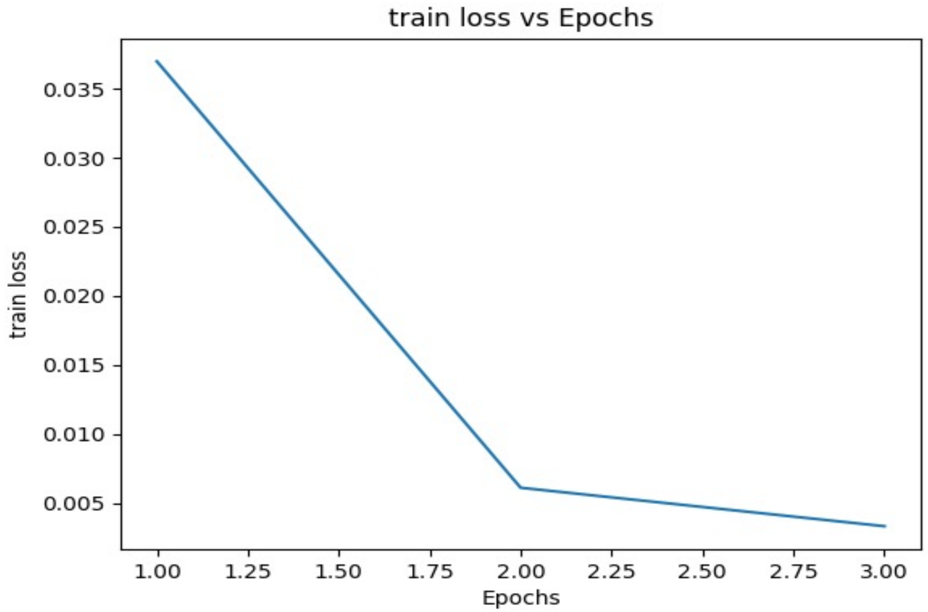
Non-Classical Model Results:

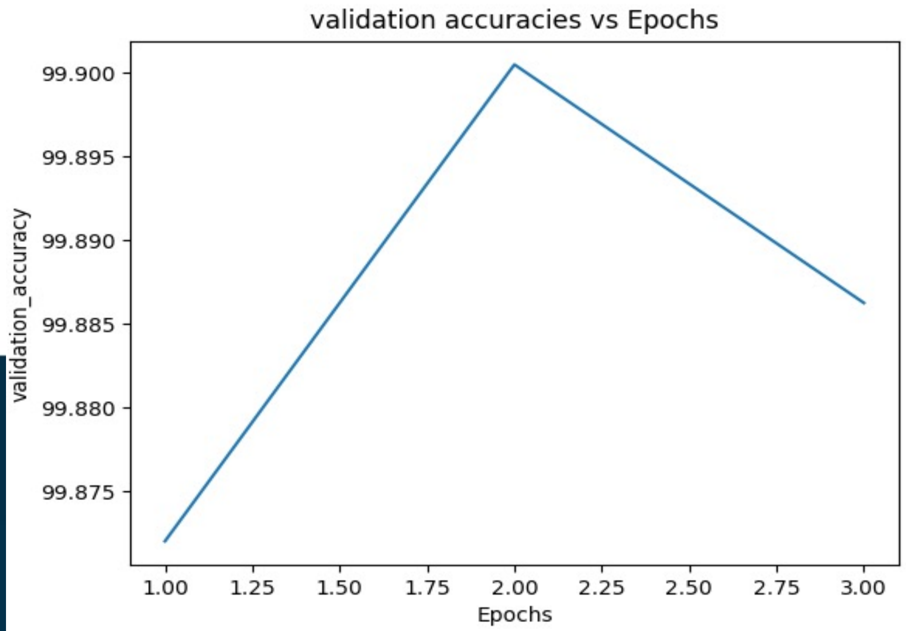
Model	Optimizer	Epoch	Batch SIze	Max_Len	Learning Rate	Train Accuracy	Validation Accuracy
DistilBERT	AdamW	4	3	30	0.001	52.3	52.29
DistilBERT	AdamW	2	32	256	0.01	52.28	52.36
DistilBERT	AdamW	3	32	256	1.00E-05	99.93	99.89
DeBERTa	AdamW	4	3	30	0.001	52.13	52.29
DeBERTa	AdamW	3	4	256	1.00E-05	99.95	99.91
RoBERTa	AdamW	3	32	256	1.00E-05	99.84	99.89
RoBERTa	AdamW	3	16	256	1.00E-05	99.93	99.9

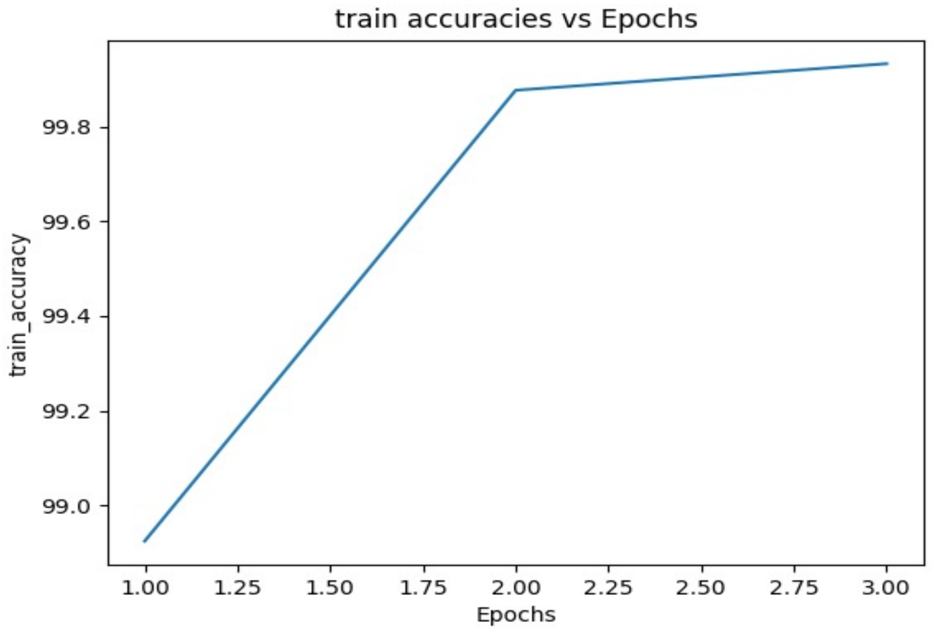
Model	Precision		recall		f1-score		Total F1 Score
	0's	1's	0's	1's	0's	1's	
DistilBERT	0.52	0	1	0	0.69	0	
DistilBERT							
DistilBERT	0.98	1	1	0.98	0.99	0.99	0.98951
DeBERTa	0.52	0	1	0	0.69	0	
DeBERTa	0.98	1	1	0.98	0.99	0.99	0.98845
RoBERTa	0.98	1	1	0.98	0.99	0.99	0.988457
RoBERTa	0.98	1	1	0.98	0.99	0.99	0.991631

DistilBERT Results:

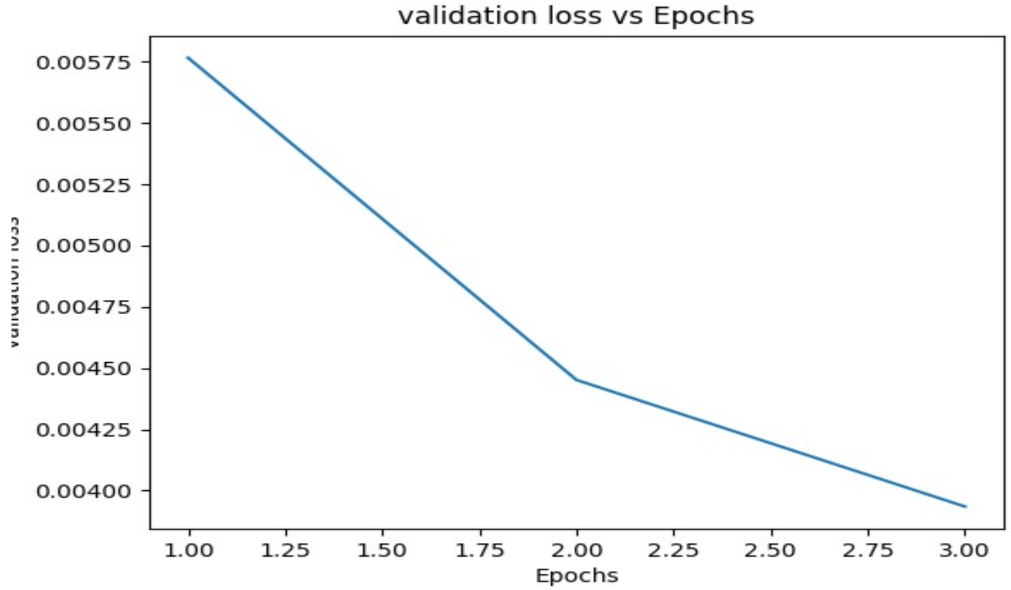


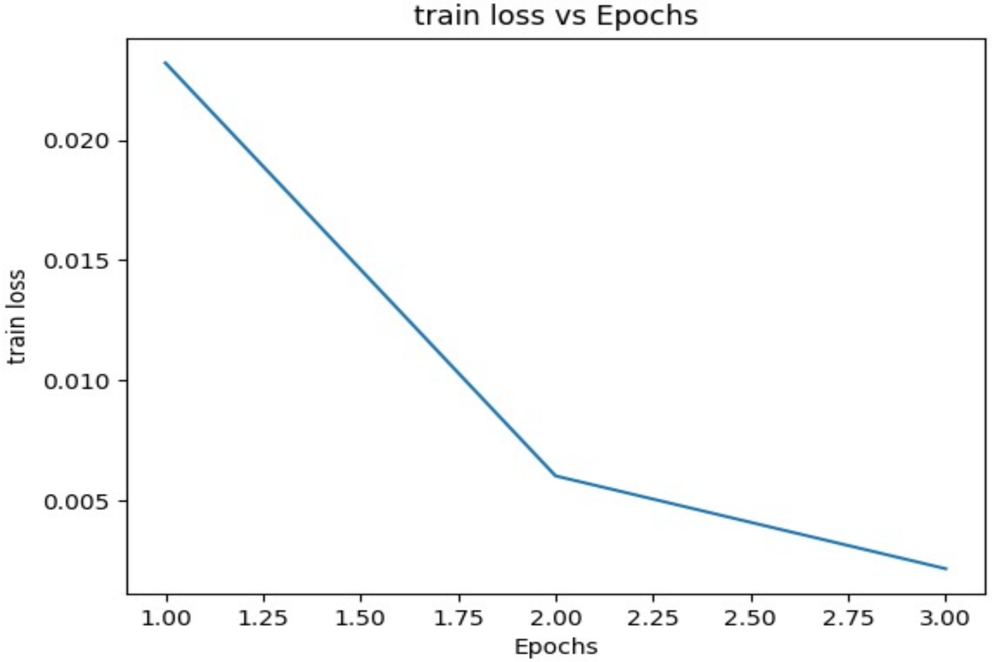


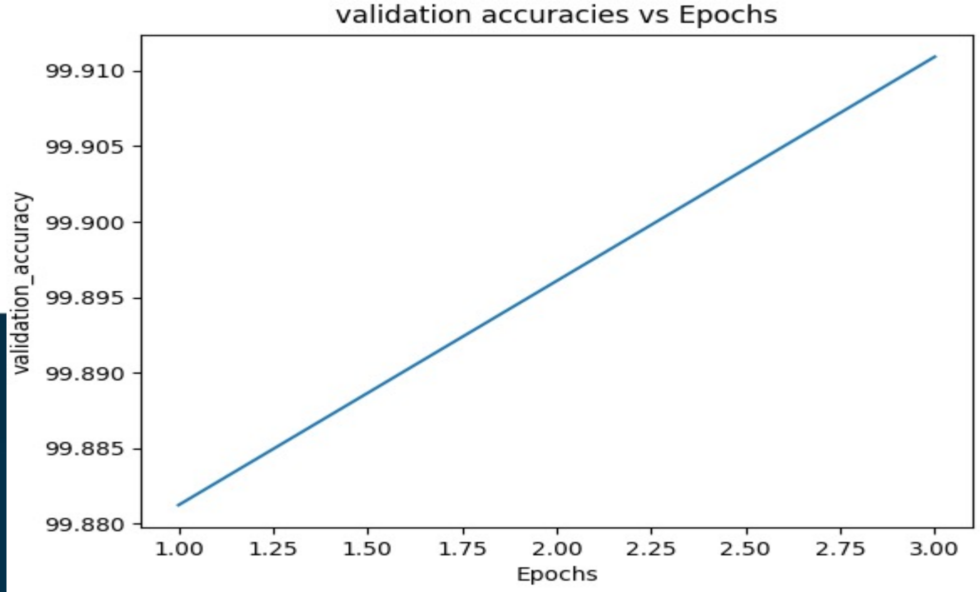


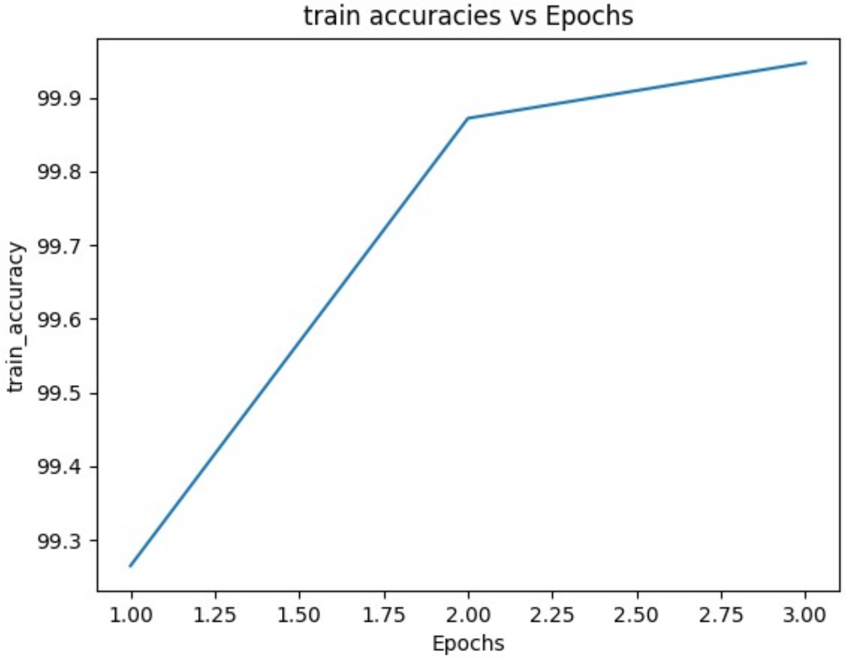


DeBERTa Results:



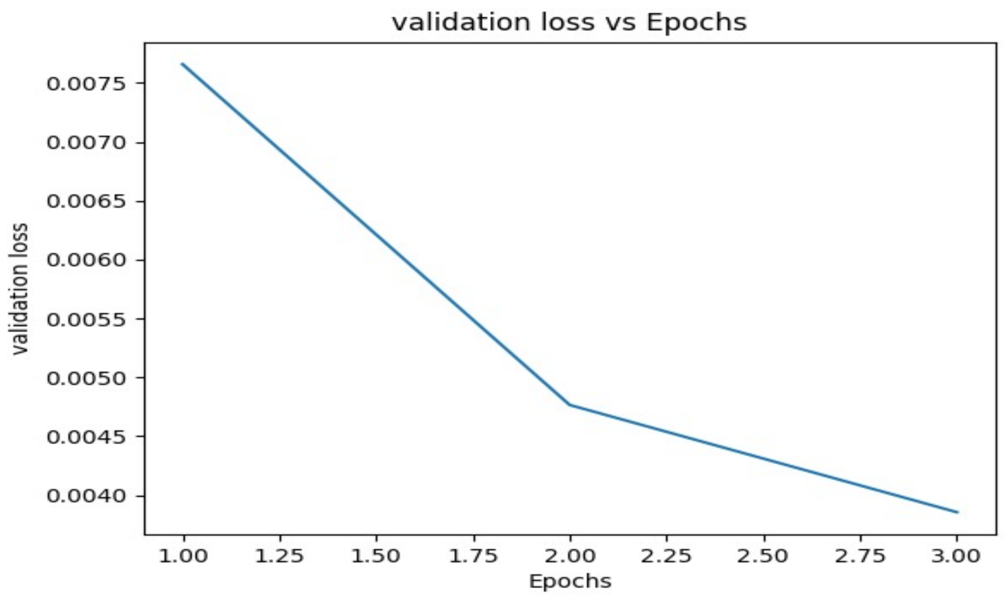


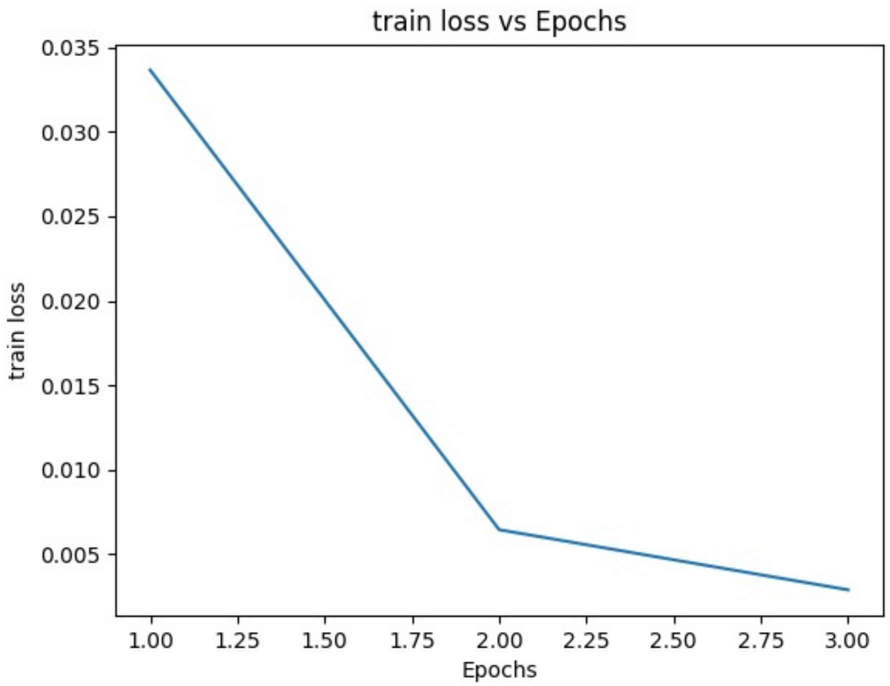


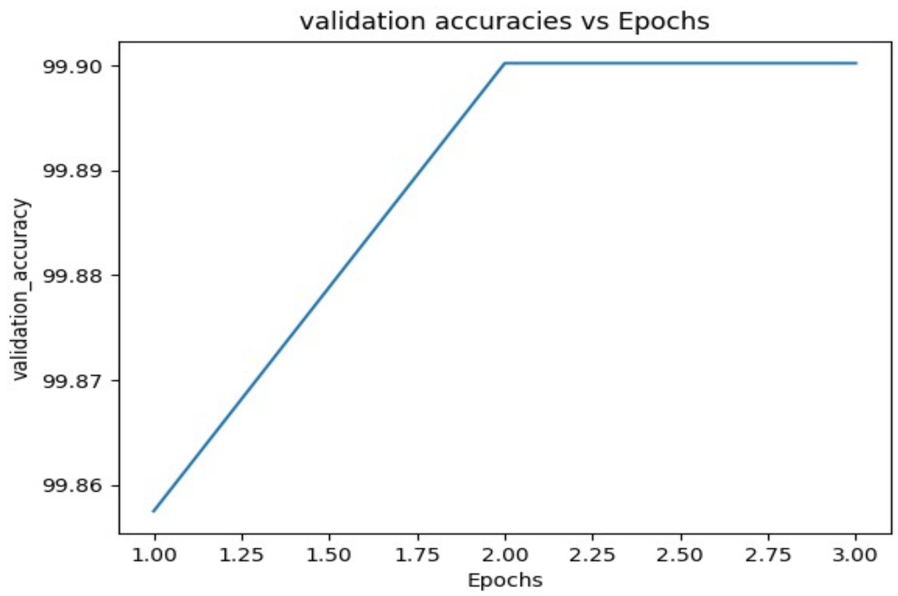


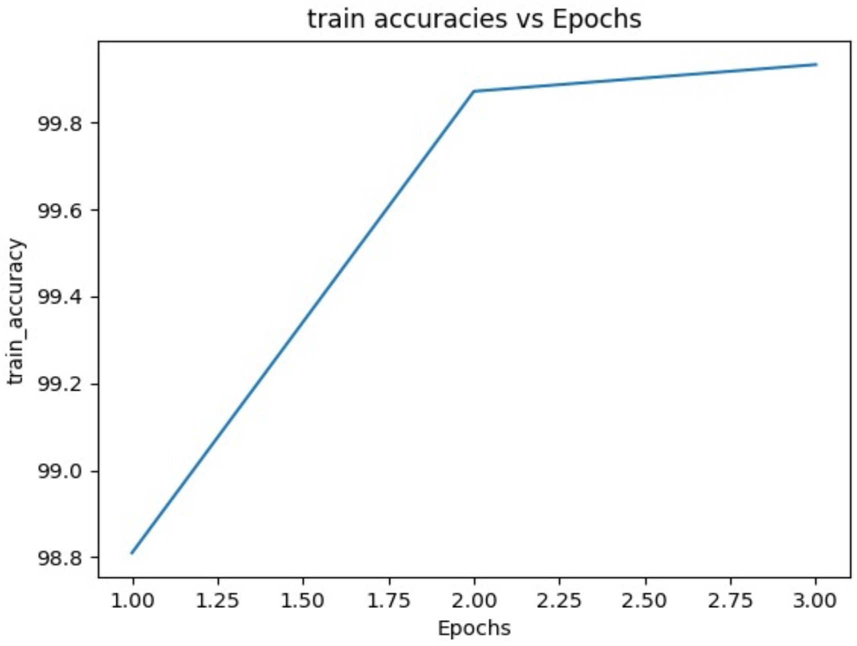
THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

RoBERTa Results:









THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Conclusion

- By comparing the results of our experiments, we found out that the non-classical models outperformed the classical model(Logistic Regression).
- We found that the RoBERTa model is the best model of the transformers with f1-score = 0.99163.
- In the future work, we should try to improve the architecture to give the results faster. We can also use higher computing power so that we can experiment with more hyperparameters such as batch size, epochs, optimizers, learning rate and find the model with higher accuracy.

Model	Trainable Parameters	f1 Score
Logistic Regression	-	0.9868544600938968
DeBERTa	139,193,858	0.9884575026232949
RoBERTa	124,647,170	0.9916317991631799
DistilBERT	66,955,010	0.989517819706499

References

1. Ahmed H, Traore I, Saad S. “Detecting opinion spams and fake news using text classification”, Journal of Security and Privacy, Volume 1, Issue 1, Wiley, January/February 2018.

2. Ahmed H, Traore I, Saad S. (2017) “Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, vol 10618. Springer, Cham (pp. 127- 138).

3. https://www.analyticsvidhya.com/blog/2021/07/detecting-fake-news-with-natural-language-processing/

4. https://ai.facebook.com/blog/roberta-an-optimized-method-for-pretraining-self-supervised-nlp-systems/

Any Questions?

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC