# DATS-6312 NLP for DataScience

## *Fake or Real News Classification*

Prof. Amir Jafari

**Final Term Project**

Aasish Kumar, Greeshmanjali Bandlamudi, Pranay Bhakthula

# Table of Contents

# 1.Introduction:

The goal of our project is to classify real and fake news on the internet.As the scourge of “fake news” continues to plague our information environment, attention has turned toward devising automated solutions for detecting problematic online content. But, in order to build reliable algorithms for flagging “fake news,” we will need to go beyond broad definitions of the concept and identify distinguishing features that are specific enough for machine learning.We consume news through several mediums throughout the day in our daily routine, but sometimes it becomes difficult to decide which one is fake and which one is authentic.

Do you trust all the news you consume from online media?Every news that we consume is not real. If you listen to fake news it means you are collecting the wrong information from the world which can affect society because a person’s views or thoughts can change after consuming fake news which the user perceives to be true.Since all the news we encounter in our day-to-day life is not authentic, how do we categorize if the news is fake or real?

A sort of sensationalist reporting, counterfeit news embodies bits of information that might be lies and is, for the most part, spread through web-based media and other online media.This is regularly done to further or force certain kinds of thoughts or for false promotion of products and is frequently accomplished with political plans.Such news things may contain bogus and additionally misrepresented cases and may wind up being virtualized by calculations, and clients may wind up in a channel bubble.Our project aims to train a model on fake and real news dataset, and predict.

# 2.Dataset Description:

There are two sorts of articles in the dataset: bogus and true news. The true articles were retrieved via crawling articles from Reuters.com; the dataset was compiled from real-world sources (News website). The phony news pieces were gathered from a variety of sources. 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 the dataset.

The first file, "True.csv," contains more than 12,600 reuter.com articles. The second file, "False.csv," has almost 12,600 items culled from various fake news sources. The following information is included in each article: the title, the text, the type, and the date the article was published. We focused on gathering articles from 2016 to 2017 to match the false news data acquired for kaggle.com. The information gathered was cleaned and processed, but the punctuation and errors found in the fake news were left in the text. The real news contains 21417 articles and we have two types in that one is world-news with article size of 10145 and the other is political news with 11272. The fake news contains 23481 articles and we have government-news with article size 1570, middle-east with size 778, US news with 783 size, left-news has 4459 size, politics has 6841, news has 9050 article size.

## *3.Data Preprocessing:*

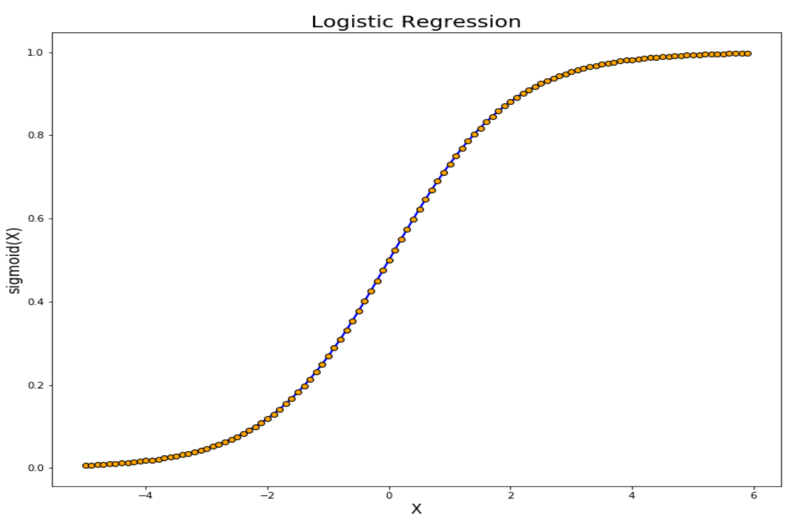
We performed the data cleaning steps to remove the unwanted words and simplify the text. Firstly, we removed the stopwords, followed by punctuations and lemmatizing the data. This helped us reduce the number of unique tokens in the dataset, thus reducing the computational load. First we have cleared the urls and lowered the text and then removed contractions and punctuations and also made sure to remove special characters i.e., characters other than alphabets and now removed the stopwords at the end we applied stemming and lemmatization. The dataset was splitted to train, validation and test with a ratio of 0.8 :0.15:0.05.

# *4.Network and Models:*

The Logistic Regression model and DeBERTa, RoBERTa and DistilBERT transformer models were chosen to be tested.

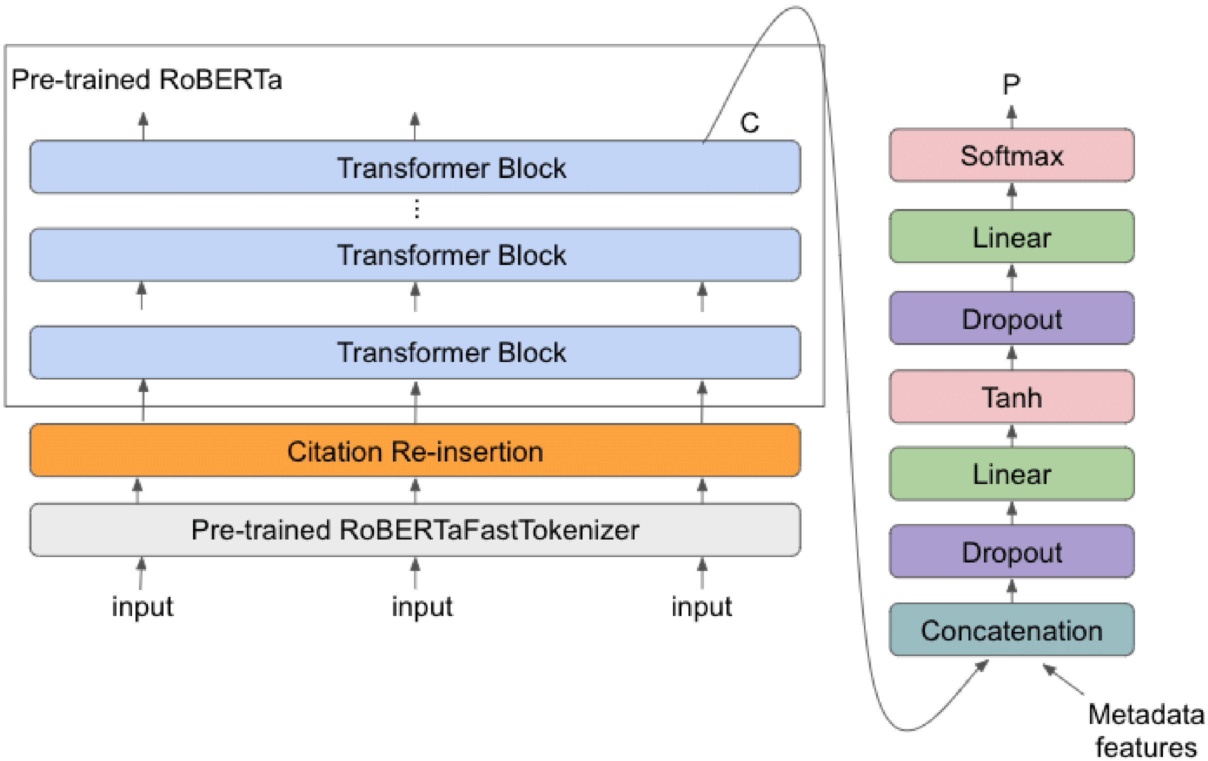
*Logistic Regression:*

This sort of statistical analysis (also known as a logit model) is commonly used for predictive analytics and modeling, as well as machine learning applications. The dependent variable in this analytics approach is either finite or categorical: either A or B (binary regression) or a range of finite possibilities A, B, C, or D (multiple regression) (multinomial regression). By estimating probabilities using a logistic regression equation, it is employed in statistical software to comprehend the relationship between the dependent variable and one or more independent variables.This form of analysis can assist you in predicting the chances of an occurrence or a decision occurring.



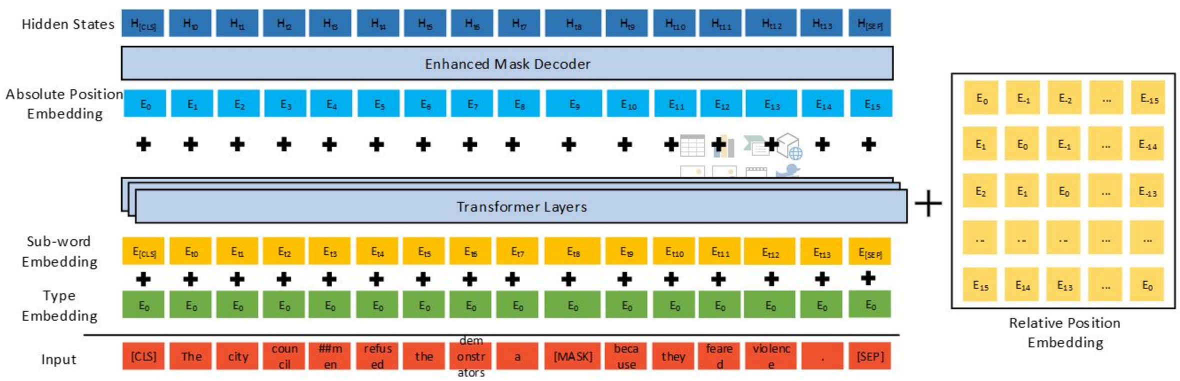

*RoBERTa:*

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.



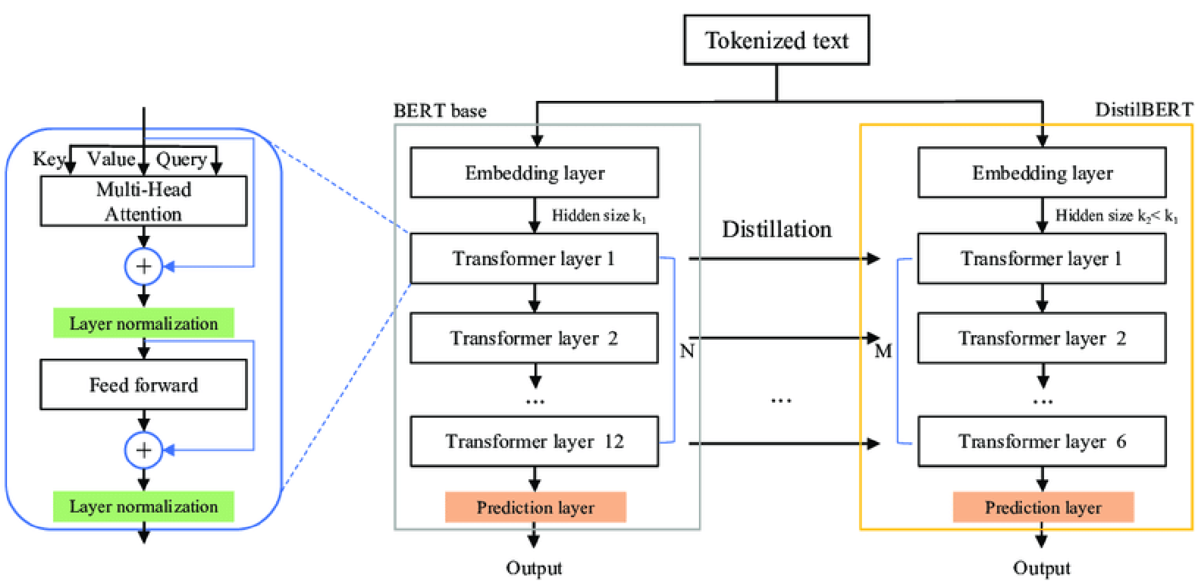

*DeBERTa:*

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.




*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.



# 5.Model Metrics:

We looked at the following metrics in deciding how effective our models were:

*F1-score:*

F1-score weights precision and recall [blog.floydhub.com].

$$F1 - Score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

*Accuracy:*

Accuracy is simply what percentage of observations were classified correctly [blog.floydhub.com].

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative}$$

# 6. Experimental Setup:

We first started off by doing data preprocessing like removing stop words, punctuations, contractions, stemming, lemmatization, then we moved on to building machine learning models by changing various hyperparameters.

We experimented with the Logistic Regression model and RoBERTa, DeBERTa, DistilBERT transformer models to see which would perform best on the dataset. When testing, we changed both epoch count, batch size, learning rate, maximum length in an attempt to improve model performance. We used Logistic Regression as a classical/base model in order to compare the results with the transformer model.

We came into memory and efficiency concerns and had to adjust the epoch and batch size to allow our models to process the data. Unfortunately, because of time limits and processing resource limitations, we were unable to evaluate other hyper factors such as optimizer and learning rate.We have used the AdamW optimizer for all three models.

For Logistic regression we performed Tfidf vectorizer, for the RoBERTa model we used the RoBERTa tokenizer, for the DeBERTa model we used the DeBERTa tokenizer and for the DistilBERT model we used the DistilBERT tokenizer.

# *7. Results:*

<u>*Classical Model Results*</u>

| 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 |

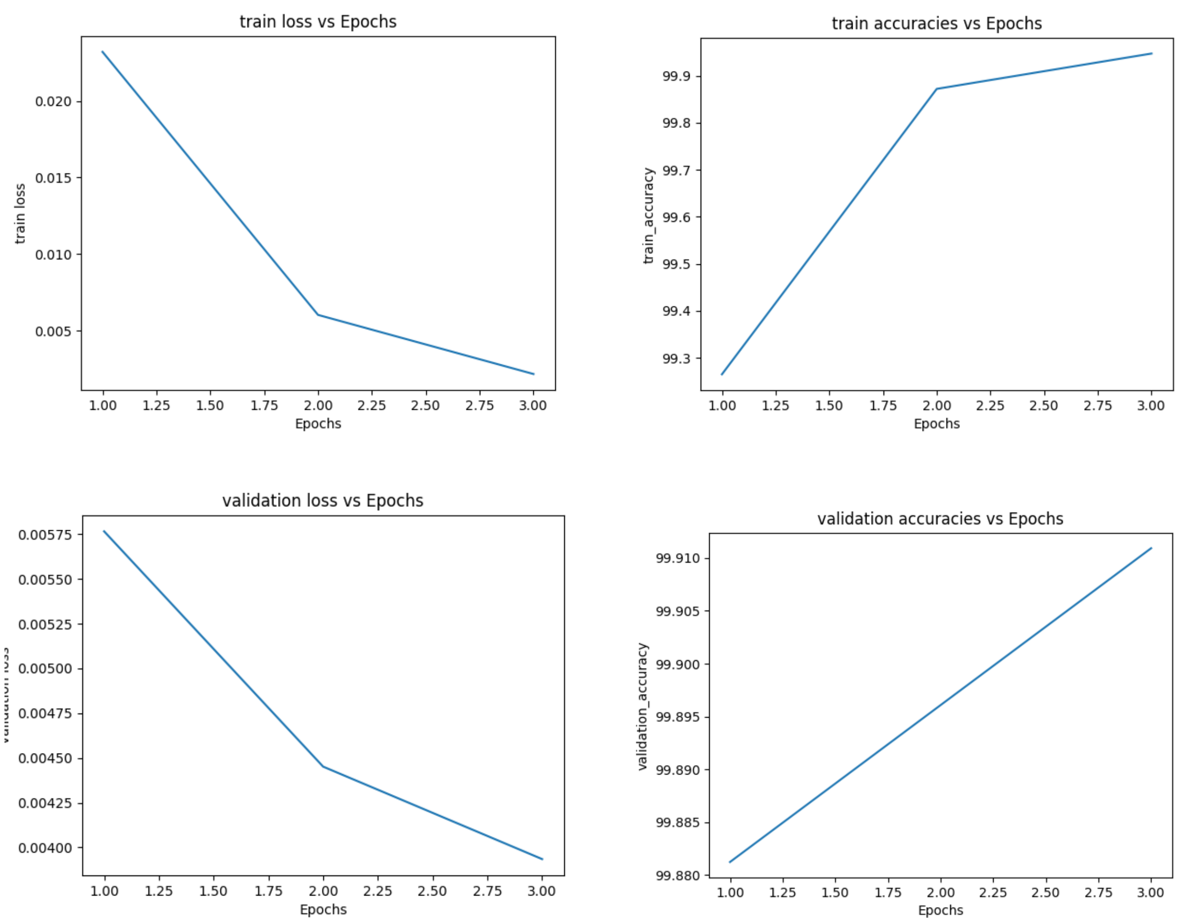
<u>*Non-Classical Model Results*</u>

| 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 |

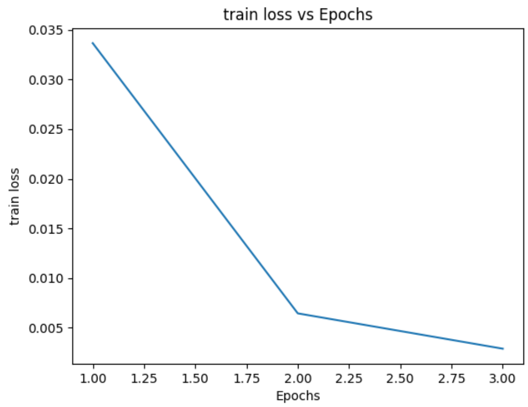
| Model | Precision | | recall | | f1-score | | Total F1 Score |
|---|---|---|---|---|---|---|---|
| | 0's | 1's | 0's | 1's | 0's | 1's | |
| DistilBERT | 0.52 | 0 | 1 | 0 | 0.69 | 0 | |
| DistilBERT | | | | | | | |
| DistilBERT | 0.98 | 1 | 1 | 0.98 | 0.99 | 0.99 | 0.98951 |
| DeBERTa | 0.52 | 0 |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

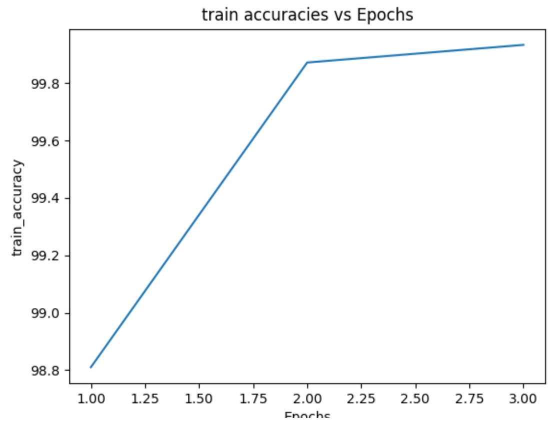
**Logistic Regression** final model f1-score=0.9868544600938968
**DeBERTa** final model f1-score = 0.9884575026232949
**RoBERTa** final model f1-score = 0.9916317991631799
**DistilBERT** final model f1-score =0.989517819706499

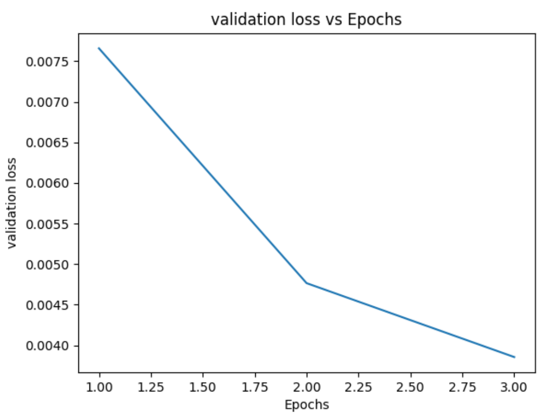
*DeBERTa final model*

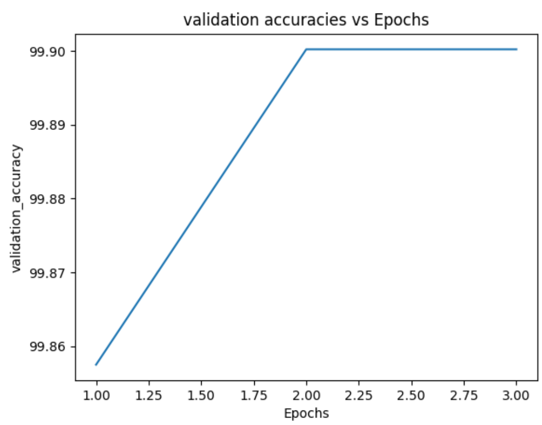


## *RoBERTa final model*

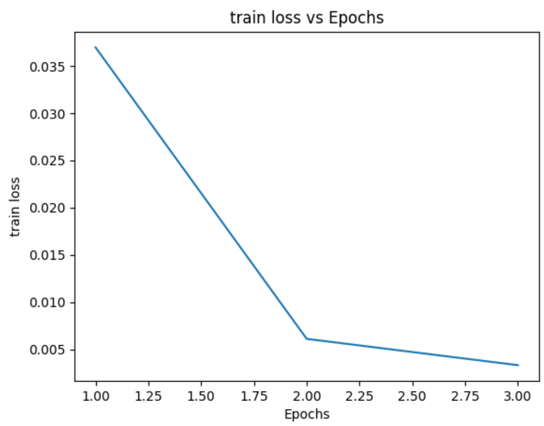


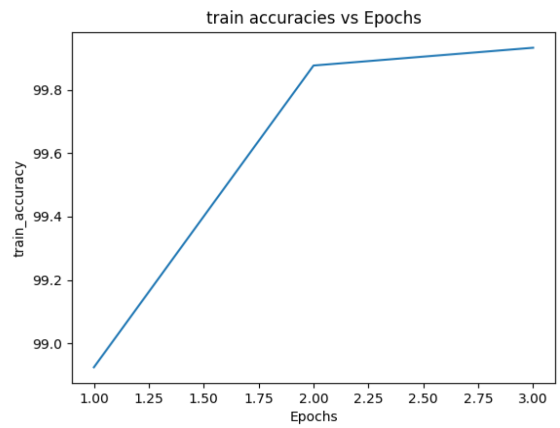



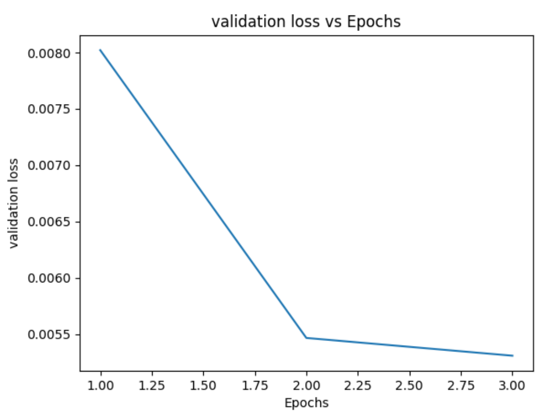



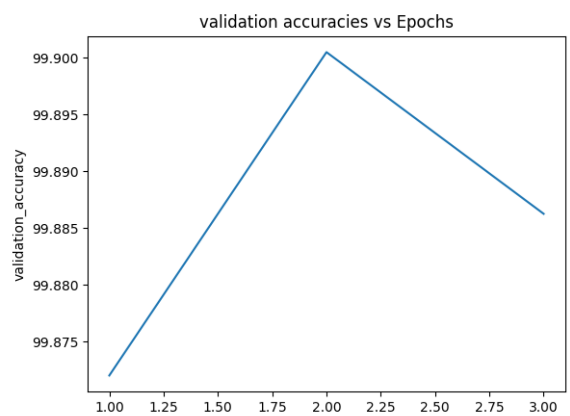



## *DistilBERT final model*



# *8. Conclusions:*

By comparing the results of our experiments we found out that the non-classical models (DeBERTa,RoBERTa,DistilBERT) outperformed the classical model(Logistic Regression). We found that the RoBERTa model is the best model of the transformers with f1-score = 0.9916.

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.

# *9. 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/*