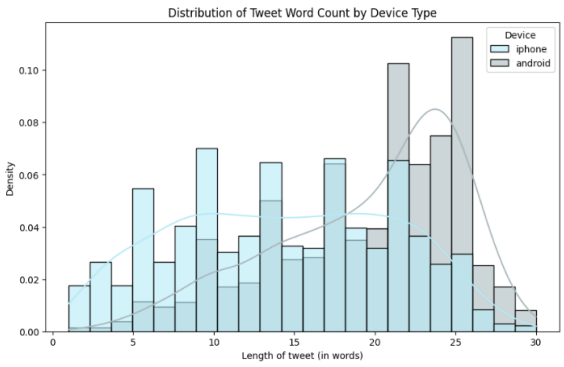
NLP Assignment 3 -Text Classification and Authorship Attribution

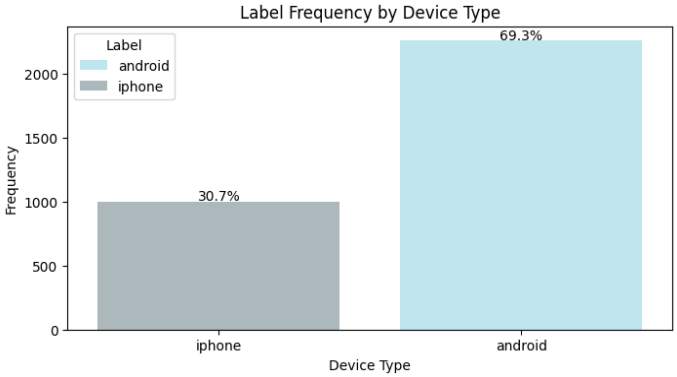
In modern politics, social media is crucial for communication. Politicians often rely on staff to manage their social media, but Donald Trump, the 45th President of the United States, is known for personally managing his unfiltered Twitter account. During his campaign, it was speculated that Trump's access to Twitter was restricted to prevent PR issues. By analyzing tweets using data from the Twitter API this task aims to use machine learning classifiers to validate the hypothesis about Trump's tweeting habits.

1. Data

A small dataset of 3,528 tweets from Trump's account, posted between 2015 and 2017, was used for this analysis. The data, already cleaned and filtered, included tweet ID, user handle, tweet text, timestamp, and device information. It is known that Trump used an Android phone, while his staffers likely used iPhones. The dataset is imbalanced, with more tweets originating from Android devices.

The EDA revealed an imbalanced dataset with more tweets originating from Android devices. Additionally, there were differences in tweet lengths based on the device type.





2. Preprocess

We converted the "device" field into the dependent variable "NotTrump". Tweets from Android were labeled 0 , and tweets from iPhone were labeled 1, with all other device types filtered out. This transformation allowed us to distinguish between tweets likely posted by Trump (0) and those by his staffers (1). We equalized the dataset to address the imbalance, resulting in 994 tweets from each class. For the classification task, feature-based classification was employed for all models, except DistilBERT. The feature vector captures the unique characteristics of each tweet and includes the following:

- **Time Features:** Extracted features include the hour, day, month, and year of the tweet's timestamp, providing temporal context for when the tweet was posted.
- **Text Features:** Features such as word count, capital letters count, hashtag count, retweet mentions, exclamation marks count, URL count, and pronoun count were extracted to analyse the structural and content aspects of each tweet.
- **Sentiment Features:** Sentiment analysis was performed to calculate polarity and subjectivity scores, along with counts of positive and negative words, capturing the emotional tone of the tweets.
- **Emotion Features**: Emotion detection was performed using the 'distil-Roberta' model to classify tweets into emotions (anger, disgust, fear, joy, neutral, sadness, and surprise) providing a detailed emotional profile for each tweet.

As for the DistilBERT model, we leveraged its capability for sequential memory by utilizing word embeddings for the tweet text.

3. Models

We employed five models from different model families for the classification task to find the best fit. Each model was trained with 5-fold cross-validation, except for the distil-BERT model, which was trained on an 80-20 train-test split due to the time-intensive process and limited Colab resources. The hyperparameters were chosen using a grid search. The models used with their respective parameters are as follows:

- **Logistic Regression:** A linear model for binary classification tasks. Parameters: {'C': 0.1, 'penalty': 'l2', 'solver': 'liblinear'}, with a threshold of 0.5.
- **SVM**: A powerful classification model that can use different kernel functions. Both linear and nonlinear kernels were experimented with. The best result was achieved using a linear kernel with {'C': 10}, and a threshold of 0.55.
- **FFNN**: A neural network model with multiple layers for capturing more complex patterns. The chosen architecture includes 2 hidden layers with dropout and ReLU activation, and a sigmoid activation in the output layer. Parameters: {'optimizer': Adam, 'epochs': 100, 'patience': 10}, }, and a threshold of 0.45.
- **XGBoost**: An efficient and scalable implementation of gradient boosting. Parameters: {'colsample_bytree': 0.9, 'gamma': 0.1, 'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 200, 'subsample': 0.9}, with a threshold of 0.5.
- **DistilBERT**: A lighter version of the BERT model designed for efficiency while retaining performance. We used word embeddings for the tweet text. Parameters: {'epochs': 4}.

We selected XGBoost for its efficiency and strong performance in classification tasks. DistilBERT was chosen for its high accuracy and computational efficiency, making it well-suited for NLP tasks and processing sequential tweet text data.

4. Evaluation

To evaluate the performance of our classification models, we used several key metrics to gain a comprehensive understanding of their effectiveness. We assessed accuracy, precision, recall, F1, and ROC-AUC scores.

5. Results

	Precision	Recall	F1 Score	AUC	Accuracy
LR	0.861	0.700	0.771	0.873	0.794
SVM	0.788	0.750	0.768	0.868	0.774
FFNN	0.854	0.805	0.827	0.910	0.834
XGBoost	0.860	0.835	0.845	0.931	0.848
DistilBERT	**0.903**	**0.905**	**0.904**	**0.962**	**0.905**

The results show that DistilBERT outperformed all other models, achieving the highest scores in precision (0.903), recall (0.905), F1 score (0.904), AUC (0.962), and accuracy (0.905). XGBoost and FFNN also performed well but did not match DistilBERT's effectiveness. Logistic Regression and SVM yielded the least satisfactory results. These results highlight the advantage of using models that capture advanced patterns and are specifically adapted for sequential memory and language processing.

6. Conclusions

The models that aren't adjusted for sequential memory, such as Logistic Regression and SVM, achieved less satisfactory results. While the feature vector we created captures many dimensions of the tweet, it does not reach the same level of accuracy as the DistilBERT model. DistilBERT, specifically adapted to tasks of language processing, sequence memory, and dealing with embeddings and word contexts, demonstrated superior performance. For this task, the results indicate that the most suitable algorithm is DistilBERT. Based on its characteristics, it effectively distinguishes between tweets authors, revealing patterns that traditional machine learning models find more challenging to detect.

*This assignment was refined using Grammarly and ChatGPT for polishing and rephrasing.

**Link to the Colab Notebook: https://colab.research.google.com/drive/1ioT9-Jr5bpE4R84__pdfepzBVx9aHdNb?usp=sharing