# 2019 Presidential Election **Public Opinion Tweet Sentiment Analysis**

Using LSTM & Random Forest Algorithm

2023 - Apollo Team - Indonesia AI NLP Batch 2

# Table of Content

# Project Brief

Given the 2019 presidential election's **tweet data** of public opinion, this project :

- **Experiment** with varieties of preprocessing and vectorization technique
- **Test** the Random Forest & LSTM algorithm
- **Optimized** model (Hyperparameter Tuning)
- **Evaluate** and **conclude** which algorithms is the **best performing**

**Methodology**

**EDA, Data Exploration & Understanding**

- Data Quantity
- Data Distribution
- EDA Wordcloud

**Data Cleaning & Preprocessing**

- **Text Cleaning**
  *(e.g. Stopwords, Punctuation, Case Folding)*
- **Text Preprocessing**
  - Stopwords removal
  - Word Tokenization
  - Sentence Tokenization
  - Word2Vec (CBOW)

**Feature Extraction**

- TF-IDF Vectorizer
- Split Data Train & Test (80:20)

**Data Modeling**

- Test 1 : Random Forest Algorithm
- Test 2 : LSTM Algorithm

**Model Evaluation**

- Precision
- Recall
- F1-score

# Data Exploration & Understanding

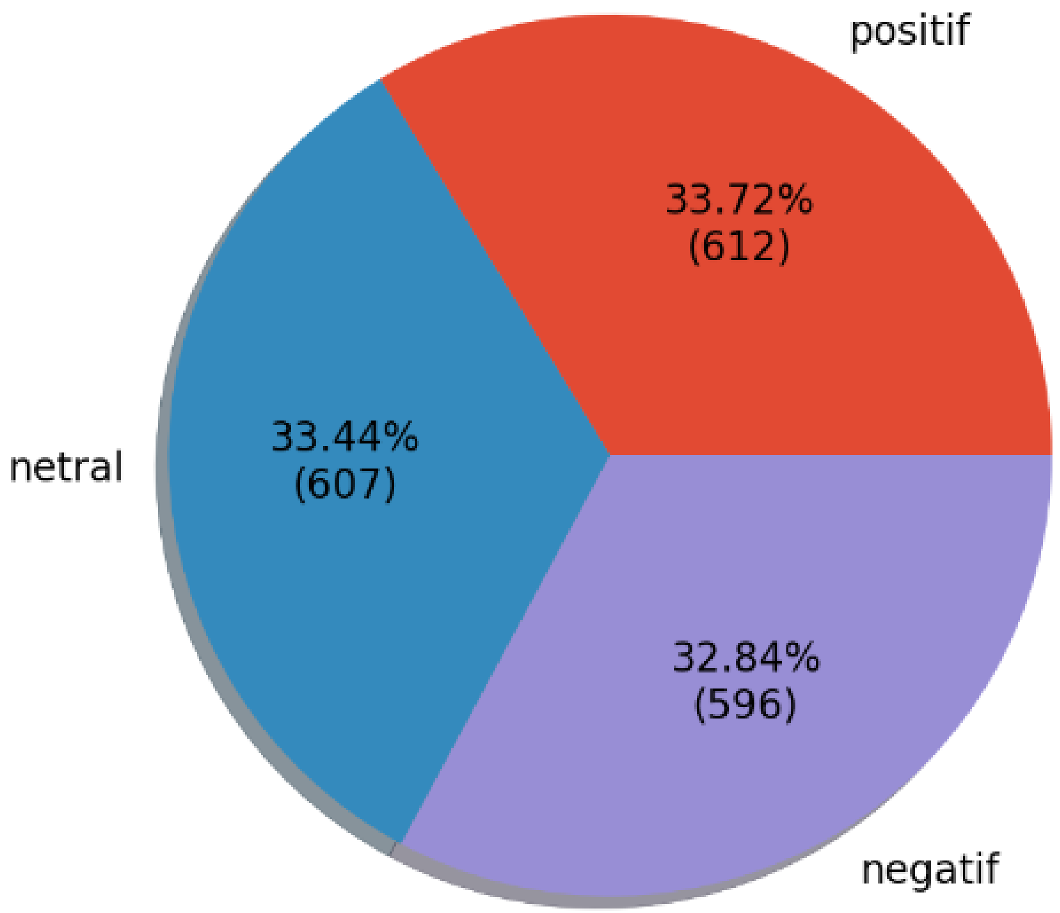
There are **1,814 tweet data** related to 2019 presidential election
which **distributed almost evenly (~33%)** among 3 sentiments (Positive, Negative, Neutral)

## Data Info

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1815 entries, 0 to 1814
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Unnamed: 0  1815 non-null   int64
 1   sentimen    1815 non-null   object
 2   tweet       1815 non-null   object
dtypes: int64(1), object(2)
memory usage: 42.7+ KB
```

## Sentiment Data Distribution



# Data Cleaning & Preprocessing

## Data Final

```
data.head()
```

| | Unnamed: 0 | sentimen | tweet | tweet_clean | tweet_sw | tweet_token_words | tweet_token_sentences | tweet_w2v_model |
|---|---|---|---|---|---|---|---|---|
| **0** | 0 | negatif | Kata @prabowo Indonesia tidak dihargai bangsa ... | kata indonesia dihargai bangsa ase berita past... | kata indonesia dihargai bangsa asing berita pa... | [kata, indonesia, dihargai, bangsa, ase, berit... | [kata indonesia dihargai bangsa ase berita pas... | [-0.031047378, 0.108620666, 0.06571932, 0.0231... |
| **1** | 1 | netral | Batuan Langka, Tasbih Jokowi Hadiah dari Habib... | batuan langka tasbih jokowi hadiah habib luthf... | batuan langka tasbih jokowi hadiah habib luthf... | [batuan, langka, tasbih, jokowi, hadiah, habib... | [batuan langka tasbih jokowi hadiah habib luth... | [0.0012729826, 0.003569824, -0.003325973, 0.00... |
| **2** | 2 | netral | Di era Jokowi, ekonomi Indonesia semakin baik.... | era jokowi ekonomi indonesia semakin baik indo... | era jokowi ekonomi indonesia semakin baik indo... | [era, jokowi, ekonomi, indonesia, semakin, bai... | [era jokowi ekonomi indonesia semakin baik ind... | [-0.032167733, 0.10593001, 0.0747535, 0.035609... |
| **3** | 3 | positif | Bagi Sumatera Selatan, Asian Games berdampak p... | sumatera selatan asian game berdampak pd ekono... | sumatera selatan asian games berdampak pd ekon... | [sumatera, selatan, asian, game, berdampak, pd... | [sumatera selatan asian game berdampak pd ekon... | [-0.0008489212, 0.019022403, 0.020040477, -0.0... |

**Text Cleaning**
(e.g. Stopwords, Punctuation, Case Folding)

**Text Preprocessing**
Stopwords removal
Word Tokenization
Sentence Tokenization
Word2Vec (CBOW)

# Exploratory Data Analysis **(EDA)**

**"Ekonomi"** is a popular topic opiniated by public related to 2019 presidential elections as it's **consistently topN in all sentiment category.**

On candidates side : **"Jokowi"** were topN in all sentiments, occured slightly more compared to **"Prabowo"** in positive and neutral sentiments. However, **"Jokowi"** surpassed **"Ekonomi"** as the top1 word in negative sentiment tweets while **"Prabowo"** were mentioned way less in comparison

**Positive** Sentiment Word Cloud

Positive Reviews Word Cloud


**Neutral** Sentiment Word Cloud

Neutral Reviews Word Cloud


**Negative** Sentiment Word Cloud

Negative Reviews Word Cloud

# Random Forest & LSTM Model Results

## Random Forest Baseline

1. Model Evaluation on **train set**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| negatif | 1.00 | 1.00 | 1.00 | 479 |
| netral | 1.00 | 1.00 | 1.00 | 480 |
| positif | 1.00 | 1.00 | 1.00 | 493 |
| accuracy | | | 1.00 | 1452 |
| macro avg | 1.00 | 1.00 | 1.00 | 1452 |
| weighted avg | 1.00 | 1.00 | 1.00 | 1452 |

2. Model Evaluation on **test set**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| negatif | 0.58 | 0.74 | 0.65 | 117 |
| netral | 0.61 | 0.61 | 0.61 | 127 |
| positif | 0.60 | 0.43 | 0.50 | 119 |
| accuracy | | | 0.59 | 363 |
| macro avg | 0.59 | 0.59 | 0.59 | 363 |
| weighted avg | 0.59 | 0.59 | 0.59 | 363 |

## LSTM Baseline

1. Model Evaluation on **train set**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| negatif | 0.98 | 0.98 | 0.98 | 479 |
| netral | 0.96 | 0.96 | 0.96 | 480 |
| positif | 0.96 | 0.96 | 0.96 | 493 |
| accuracy | | | 0.97 | 1452 |
| macro avg | 0.97 | 0.97 | 0.97 | 1452 |
| weighted avg | 0.97 | 0.97 | 0.97 | 1452 |

2. Model Evaluation on **test set**

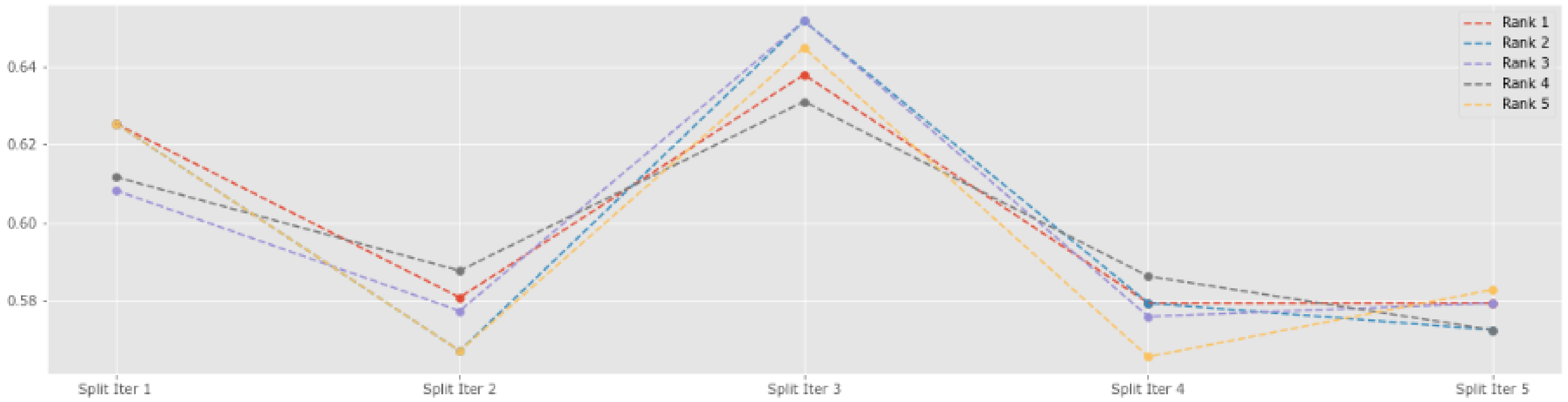
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| negatif | 0.98 | 0.98 | 0.98 | 134 |
| netral | 0.97 | 0.95 | 0.96 | 107 |
| positif | 0.96 | 0.98 | 0.97 | 122 |
| accuracy | | | 0.97 | 363 |
| macro avg | 0.97 | 0.97 | 0.97 | 363 |
| weighted avg | 0.97 | 0.97 | 0.97 | 363 |

LSTM Evaluation Metrics (precision, recall, F1-Score) were around ~97%,

**LSTM evaluation were significantly better** compared to random forest (~59%)

# Random Forest Parameter Tuning

## Top 5 Metrics Parameter Tuning




```
### Get Best Parameters
grid_search.best_params_
```

```
{'max_depth': 15, 'max_features': 'auto', 'n_estimators': 500, 'n_jobs': -1}
```

```
# Define random forest model
model = RandomForestClassifier(max_depth = 15, max_features = "auto", n_estimators = 500, n_jobs = -1)
```

## Random Forest Post - Tuning

1. Model Evaluation on **train set**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| negatif | 0.94 | 0.90 | 0.92 | 479 |
| netral | 0.82 | 0.94 | 0.88 | 480 |
| positif | 0.97 | 0.87 | 0.92 | 493 |
| accuracy | | | 0.90 | 1452 |
| macro avg | 0.91 | 0.90 | 0.90 | 1452 |
| weighted avg | 0.91 | 0.90 | 0.90 | 1452 |

2. Model Evaluation on **test set**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| negatif | 0.59 | 0.72 | 0.65 | 117 |
| netral | 0.62 | 0.56 | 0.59 | 127 |
| positif | 0.60 | 0.53 | 0.56 | 119 |
| accuracy | | | 0.60 | 363 |
| macro avg | 0.60 | 0.60 | 0.60 | 363 |
| weighted avg | 0.60 | 0.60 | 0.60 | 363 |

# Sentiment Analysis **Model Reliability Test**

```
# Panggil fungsi predict dengan teks yang ingin Anda prediksi
predict("di era jokowi ekonomi indonesia semakin baik indonesiamaju jokowilagi jokowimenang total debat") # Aktual Positif
```

```
1/1 [==============================] - 0s 163ms/step
Sentimen: Negative
Waktu prediksi: 0.3933317661285400 4 detik
```

```
# Panggil fungsi predict dengan teks yang ingin Anda prediksi
predict("kata indonesia dihargai bangsa asing berita pasti hoax buatan penguasa ") # Aktual Negatif
```

```
1/1 [==============================] - 0s 35ms/step
Sentimen: Negative
Waktu prediksi: 0.09423494338989258 detik
```

```
# Panggil fungsi predict dengan teks yang ingin Anda prediksi
predict("negarawan sejati sll bangga mengedepankan harga diri bangsanya berdaulat gantipresiden") # Aktual Netral
```

```
1/1 [==============================] - 0s 144ms/step
Sentimen: Negative
Waktu prediksi: 0.36240100860595703 detik
```

```
# Panggil fungsi predict dengan teks yang ingin Anda prediksi
predict("bangun bangsa mendukung perekonomian negara bersama pak jokowi ayo kerja") # Aktual Netral
```

```
1/1 [==============================] - 0s 98ms/step
Sentimen: Negative
Waktu prediksi: 0.23909282684326172 detik
```

Upon testing the model to a text, the prediction were heavily resulted as "negative sentiment" instead of its actual sentiment (4 out of 4 were flagged negative)

**Further checks & test might need to be done** to find and address the cause

# Conclusion

- **Word Tokenization** along with **wordcloud** can help to visualize the top word for analysis

- Based on the test, **LSTM algorithm's evaluation metrics were significantly better** compared to Random Forest

- **Parameter tuning on Random Forest** does improve the evaluation metrics performance, however the **improvement was not significant (~1%)** and still low compared to LSTM evaluation

- **Sample test** is good to be conducted **to ensure the model reliability**

# Thank You!

2023 - Apollo Team - Indonesia AI NLP Batch 2