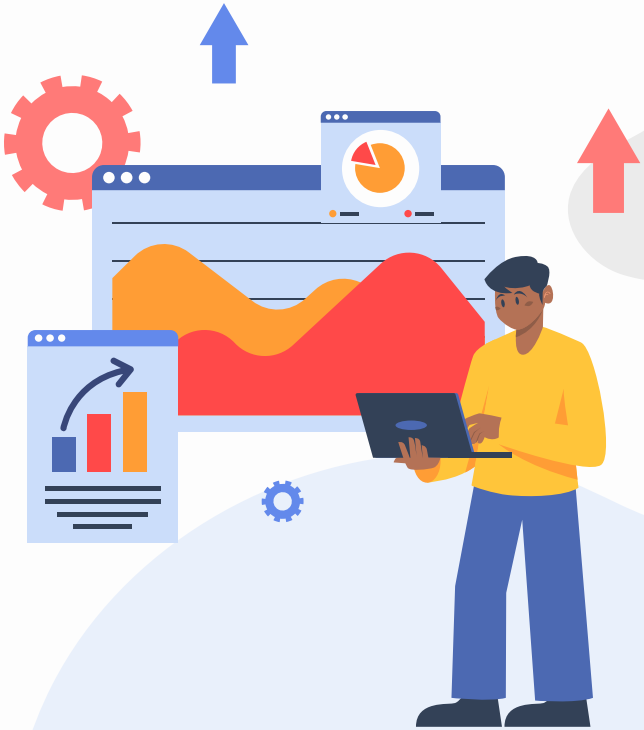
Analisis Sentimen pada Teks Bahasa Indonesia Menggunakan NN dan LSTM



Kelompok 1

(Ardhini, Imaduddin, Natasya)
Binar Academy Data Science Bootcamp

01
Pendahuluan

Latar Belakang

Di era digital saat ini, data memiliki peran yang sangat vital dalam berbagai bidang, termasuk analisis sentimen. Analisis sentimen adalah proses memahami, mengekstraksi, dan mengolah teks untuk mendapatkan informasi sentimen yang terkandung di dalamnya. Salah satu penerapan dari analisis sentimen ini adalah perusahaan dapat lebih mudah memahami opini publik terhadap produk atau layanan mereka. Dataset yang berisi teks-teks dengan label sentimen sangat penting dalam melatih model machine learning untuk tugas ini.

Dalam penelitian ini, kami akan menggunakan model Long Short-Term Memory (LSTM) dan Neural Network (Multilayer Perceptron Classifier, MLPClassifier) untuk analisis sentimen pada dataset yang didapat dari IndoNLU..

Rumusan Masalah

Bagaimana Mempersiapkan Data ?

Bagaimana cara **mengolah dan mempersiapkan data** teks agar siap digunakan dalam **model LSTM dan MLPClassifier** untuk **analisis sentimen**?

Bagaimana Mengukur Performa Model ?

Bagaimana mengukur **performa model LSTM dan MLPClassifier** yang telah dilatih dengan dataset ini?

Model Mana yang Paling Baik?

Model manakah yang memiliki **kinerja lebih baik** dalam analisis sentimen?

Tujuan Penelitian

Pre-processing Data

Mengolah dan **mempersiapkan data** teks agar siap digunakan model untuk **analisis sentimen** serta **evaluasi** perbandingan model.

Model Training

Melatih model **LSTM** dan **MLPClassifier** untuk analisis sentimen dan **mengukur performanya** untuk memastikan akurasi yang tinggi dalam prediksi sentimen.

Model Comparison

Menentukan model mana yang memiliki **kinerja lebih baik** dalam analisis sentimen

Batasan Masalah

- Dataset hanya mencakup **teks-teks dalam bahasa Indonesia** dengan **sentimen** yang dianalisis terbatas pada kategori **positif, negatif, dan netral.**
- Pre-processing data hanya menggunakan ***cleansing*** (sesuai dataset), ***stop word removal***, dan ***feature extraction*.**
- Model yang digunakan adalah **LSTM** dan **MLPClassifier.**
- Performa model diukur menggunakan metrik ***accuracy, precission, recall,*** dan ***F1-score***.

02
Metode Penelitian

Flow Process

Pre-processing Data

Cleansing, Remove Duplicates, Remove Stopwords, and Split Data

Feature Extraction

Using Bag of Words & Tokenizing and Padding

Modelling and Train

MLPClassifier and LSTM

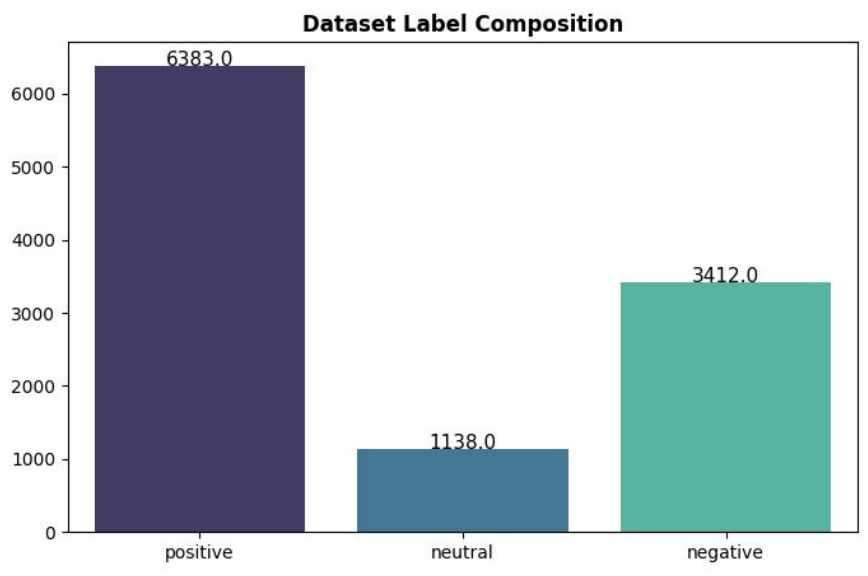
Evaluation

Confusion Matrix

API Predict

Deployment model to FlaskAPI

About Data



Data Sekunder

Didapat dari IndoNLU.

11.000 rows

Data berjumlah **11.000** rows dengan 2 kolom

Kolom Text

Berisi text dengan berbagai macam bahasan

Kolom Label

Label mengandung 3 sentimen sesuai dengan text.

Pre-processing Data

01 Drop Duplicates

Terdapat **67** data duplicates, maka total data menjadi **10933**.

03 Data Cleansing

- Lower Case
- Remove non-alphanumeric
- Remove Stopwords
- Remove White Space

02 Stopwords

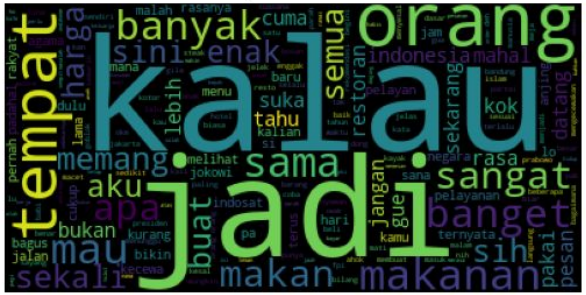
Terdapat **125** kata stopword dari library Sastrawi, ditambah kata **“nya”**. Total **126** Stopwords.

Word Count

Dengan menggunakan *wordcloud*, kita dapat mengetahui kata apa saja yang **paling banyak digunakan** pada **data**, maupun pada tiap **sentimen**.

Words

Negative Words

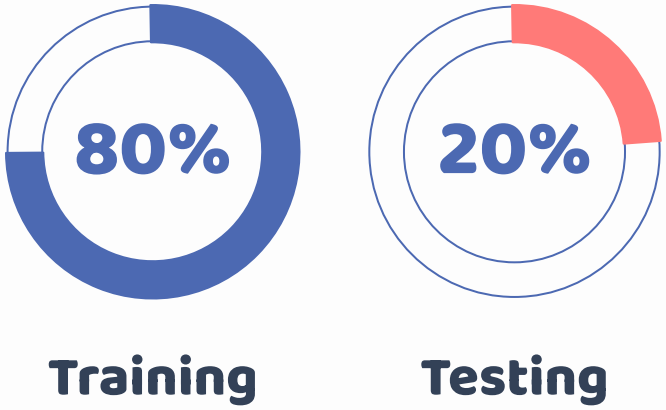


Neutral Words

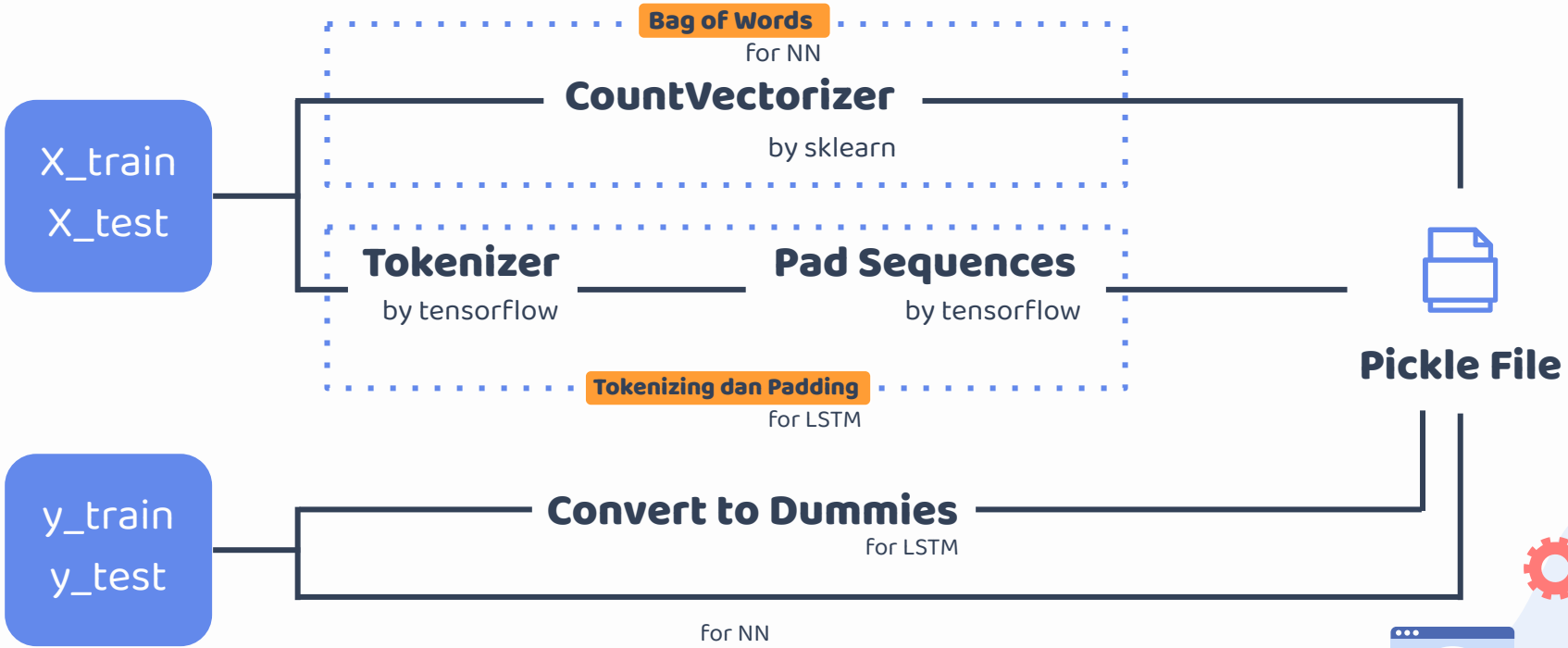
Positive Words

Data Split

Setelah pre-processing data. Data dibagi menjadi 80% untuk ***training*** dan 20% untuk ***testing***. Pembagian dilakukan sebelum proses ***Feature Extraction*** agar data yang akan dilatih antara kedua model memiliki persebaran yang sama.



Feature Extraction



Count Vectorizer

Data = ['The', 'quick', 'brown', 'fox', 'jumps', 'over', ' the', 'lazy', 'dog']

	The	quick	brown	fox	jumps	over	lazy	dog
Data	2	1	1	1	1	1	1	1

Tokenizer dan Pad Sequences

Tokenizing

Kalimat 1
I Love My Dog
1 2 3 4

Kalimat 2
I Go To School
1 5 6 7

Kalimat 3
School
7

Pad Sequences

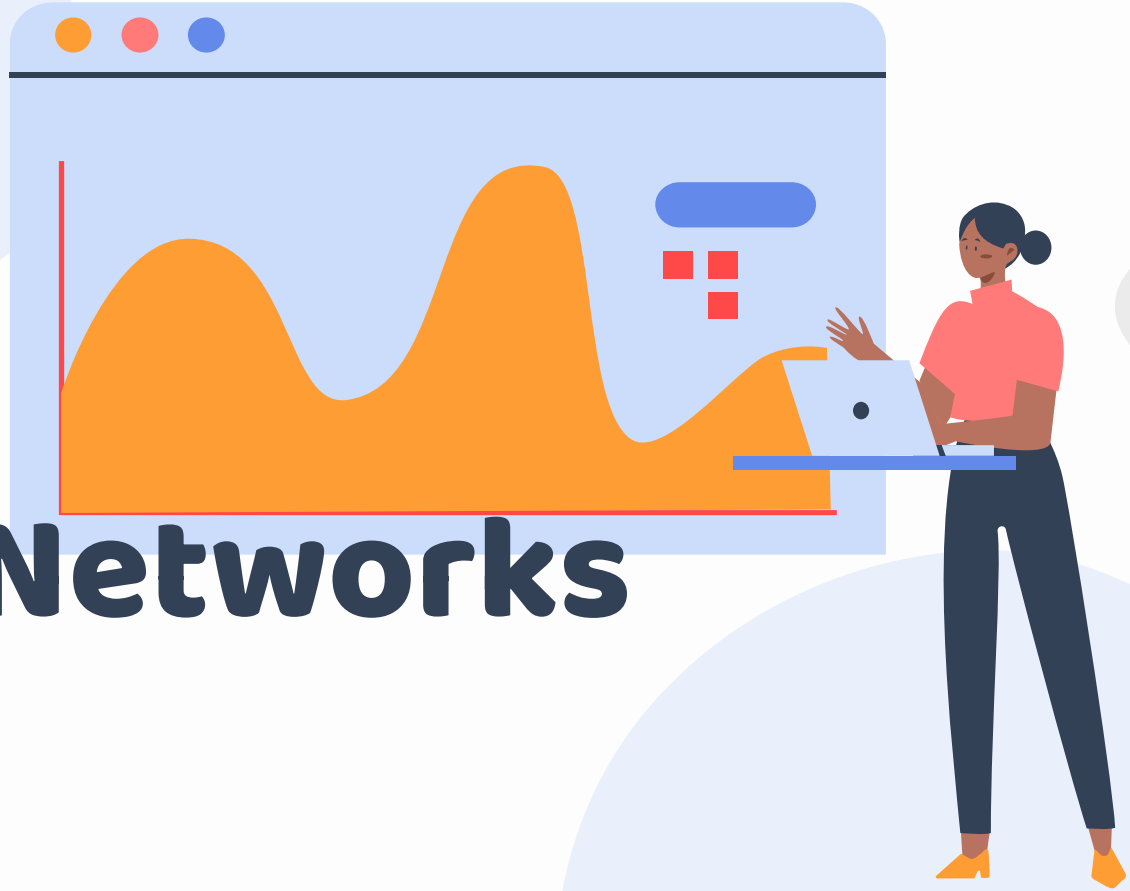
Total rows

[1, 2, 3, 4]
[1, 5, 6, 7]
[0, 0, 0, 7]

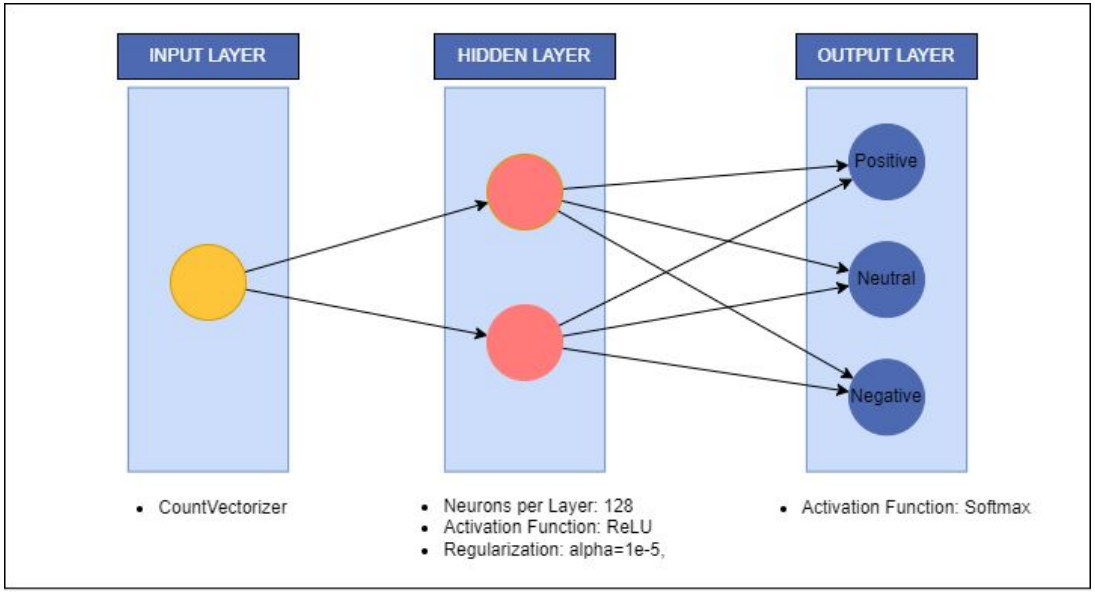
Maxlen

03
Hasil Penelitian

Neural Networks



NN Layer/Algorithm



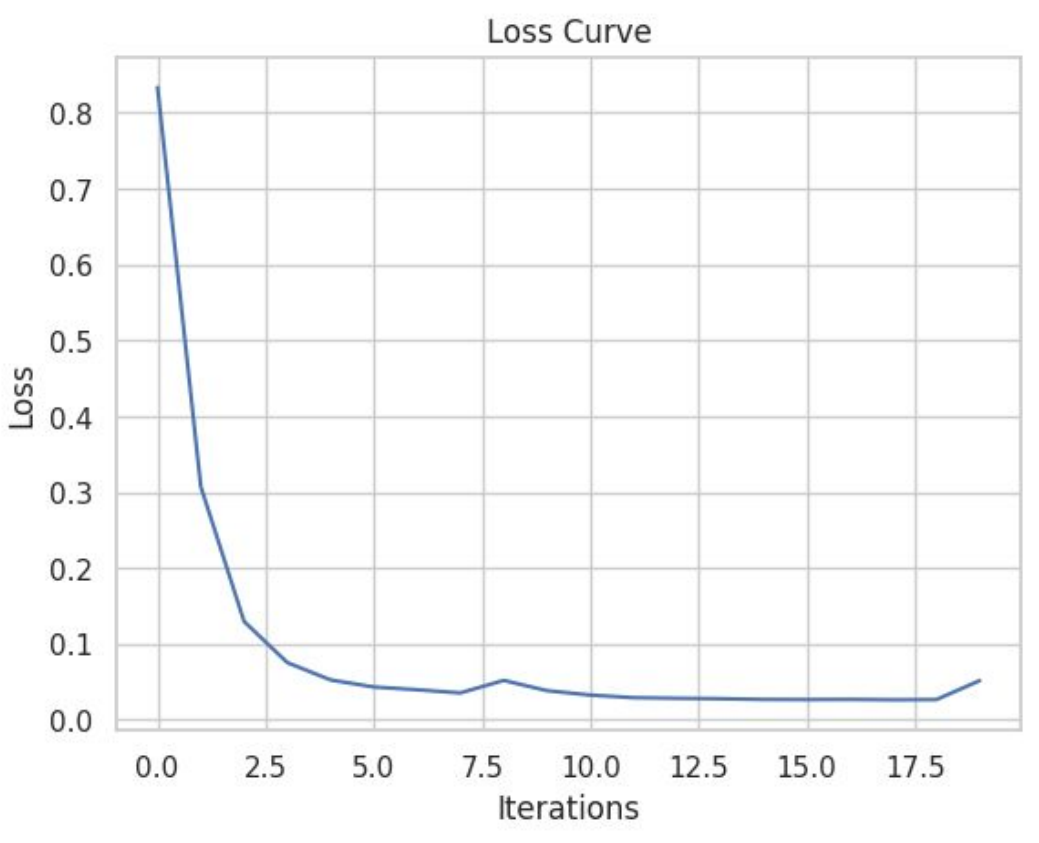
NN Parameter & Hyperparameter

- Solver: Adam
- Batch Size: 16
- Max Iterations: 20
- Random State: 42

Neural Networks Metrics

	Precision	Recall	F1-Score	Support
0	0.74	0.77	0.75	677
1	0.81	0.59	0.69	234
2	0.88	0.90	0.89	1276
Accuracy			0.83	2187
Macro Avg	0.81	0.75	0.78	2187
Weighted Avg	0.83	0.83	0.82	2187

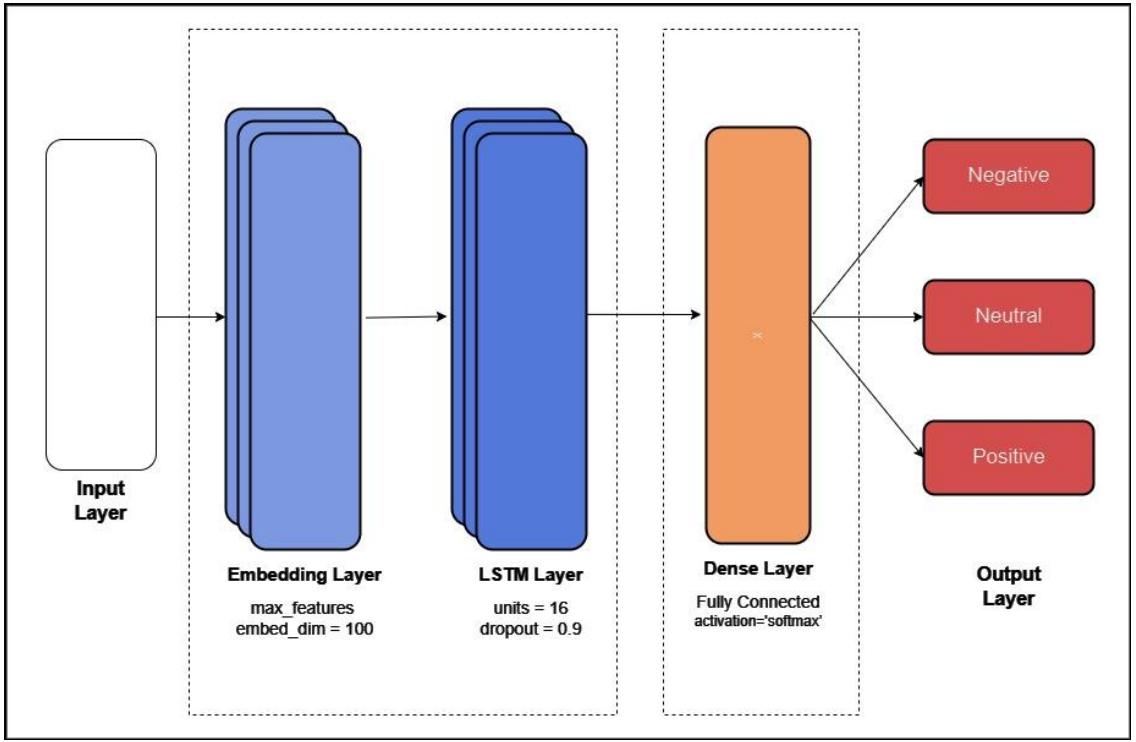
Neural Networks Loss Graph



LSTM

Long Short-Term Memory

LSTM Layer/Algorithm



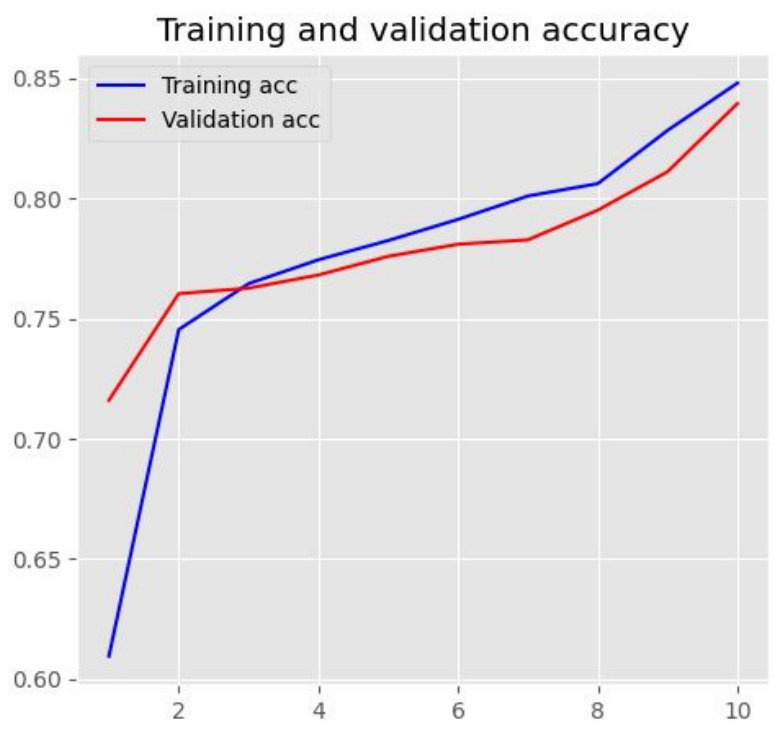
LSTM Parameter & Hyperparameter

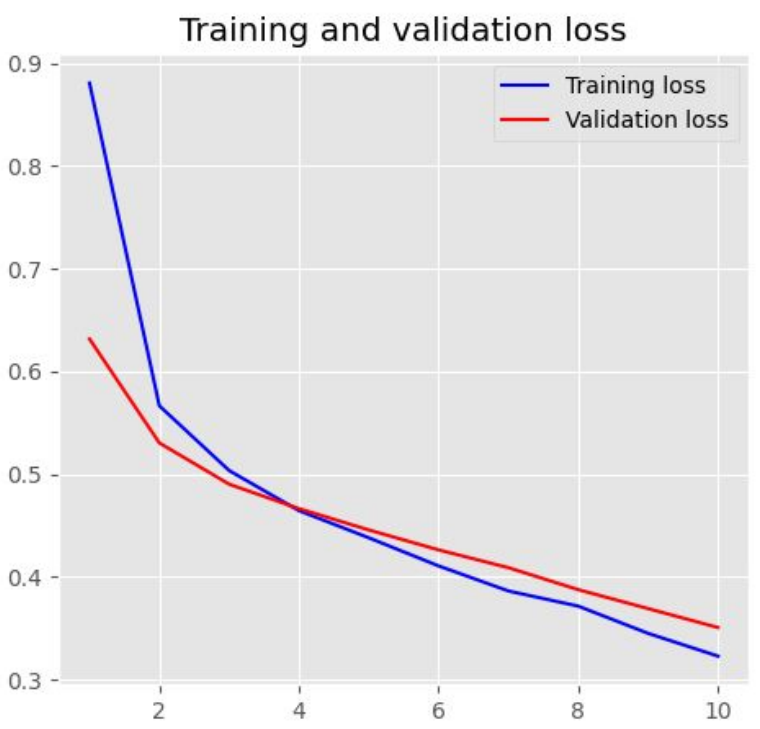
- Optimizer = Adam
- loss = 'categorical_crossentropy
- metrics = ['accuracy']

LSTM Metrics

	Precision	Recall	F1-Score	Support
0	0.77	0.79	0.78	706
1	0.81	0.52	0.63	223
2	0.88	0.92	0.90	1258
Accuracy			0.84	2187
Macro Avg	0.82	0.74	0.77	2187
Weighted Avg	0.84	0.84	0.84	2187

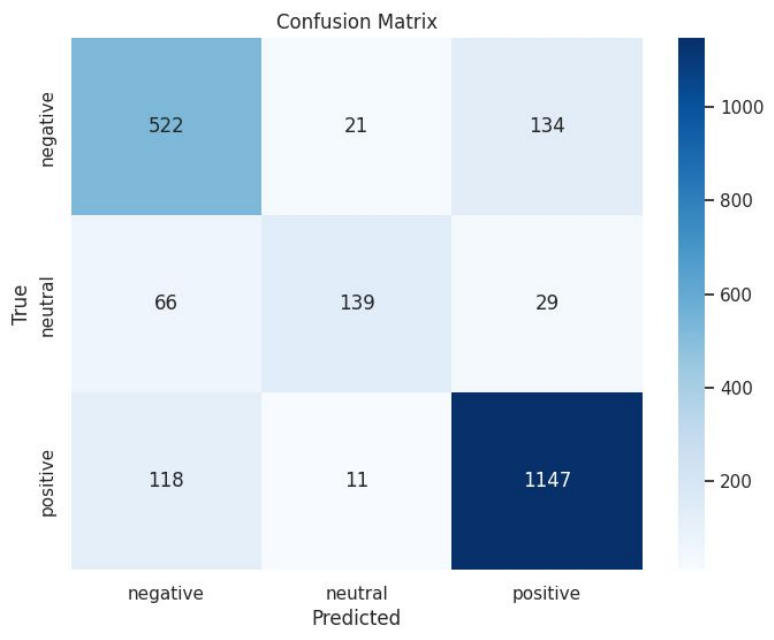
Accuracy and Loss Graph LSTM



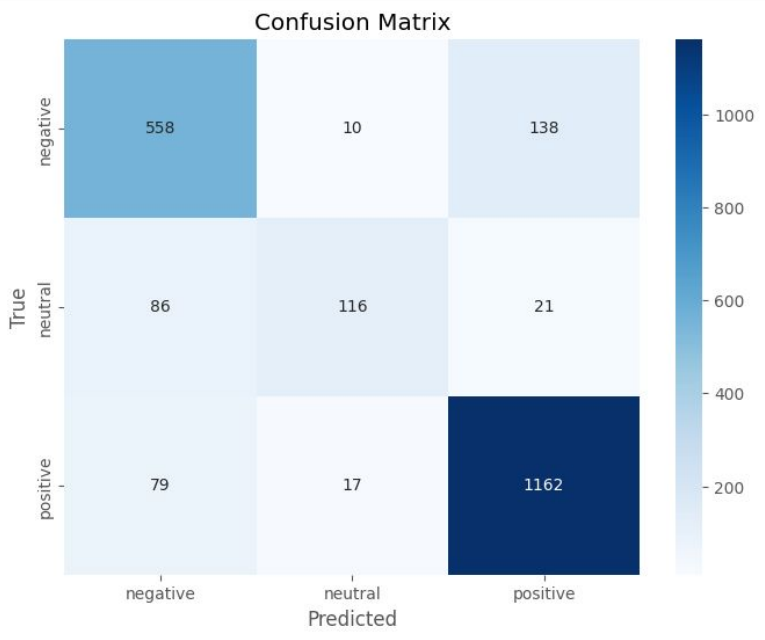


Confusion Matrix

Neural Networks



LSTM



Demo API

Predict Text

POST /predict_text

Parameters

Name	Description
text * required string *(formData)*	Barangnya Bagus banget... aku suka
model * required string *(query)*	Pilih model yang diinginkan NN

Response body

```
{
  "Prediksi": "positive",
  "Teks": "Barangnya Bagus banget... aku suka",
  "description": "Hasil predict",
  "status_code": 200
}
```

Text Prediction File

POST /text_predict_file

Parameters

Name	Description
file * required file *(formData)*	File to upload Choose file test1 - Sheet1.csv
model * required string *(query)*	Pilih model yang diinginkan LSTM

Execute

Response body

```
{
  "description": "Teks yang sudah diproses",
  "hasil": [
    [
      "text",
      "negative"
    ],
    [
      "hujan turun hari ini",
      "negative"
    ],
    [
      "\"/.,././.,sepi banget sih\"",
      "negative"
    ],
    [
      "\"/.,. kok dia marah ya?\"",
      "negative"
    ]
  ],
  "status_code": 200
}
```

Kesimpulan dan Saran

Berdasarkan penelitian, dapat diambil kesimpulan bahwa:

1. Pre-processing data meliputi menghapus data **duplikat**, ***cleansing***, ***split data*** train dan test **sebesar 8:2**, dan **Feature Extraction** menggunkan **BoW** dan ***Tokenizing & Padding***.
2. **Akurasi** yang didapat dari model **NN** sebesar **83%**, sedangkan akurasi dari model **LSTM** sebesar **84%** dan terindikasi ***good fit***.
3. Dari kedua model yang digunakan, model **LSTM** memiliki **akurasi** dan juga **f1-score** dari tiap sentimen **lebih baik** dibandingkan dengan model MLPClassifier.

Saran

Diperlukan pengecekan lebih lanjut untuk mengoptimalkan model dalam mengatasi isu seperti overfitting dan underfitting, serta menangani data yang tidak optimal. Penanganan ini akan memungkinkan hasil prediksi yang lebih akurat, sejalan dengan tujuan awal penelitian untuk menentukan model dengan kinerja terbaik dalam kondisi yang ideal.

Thanks!

Do you have any questions?