FAJAR | FIKI | REGINA

GROUP 2

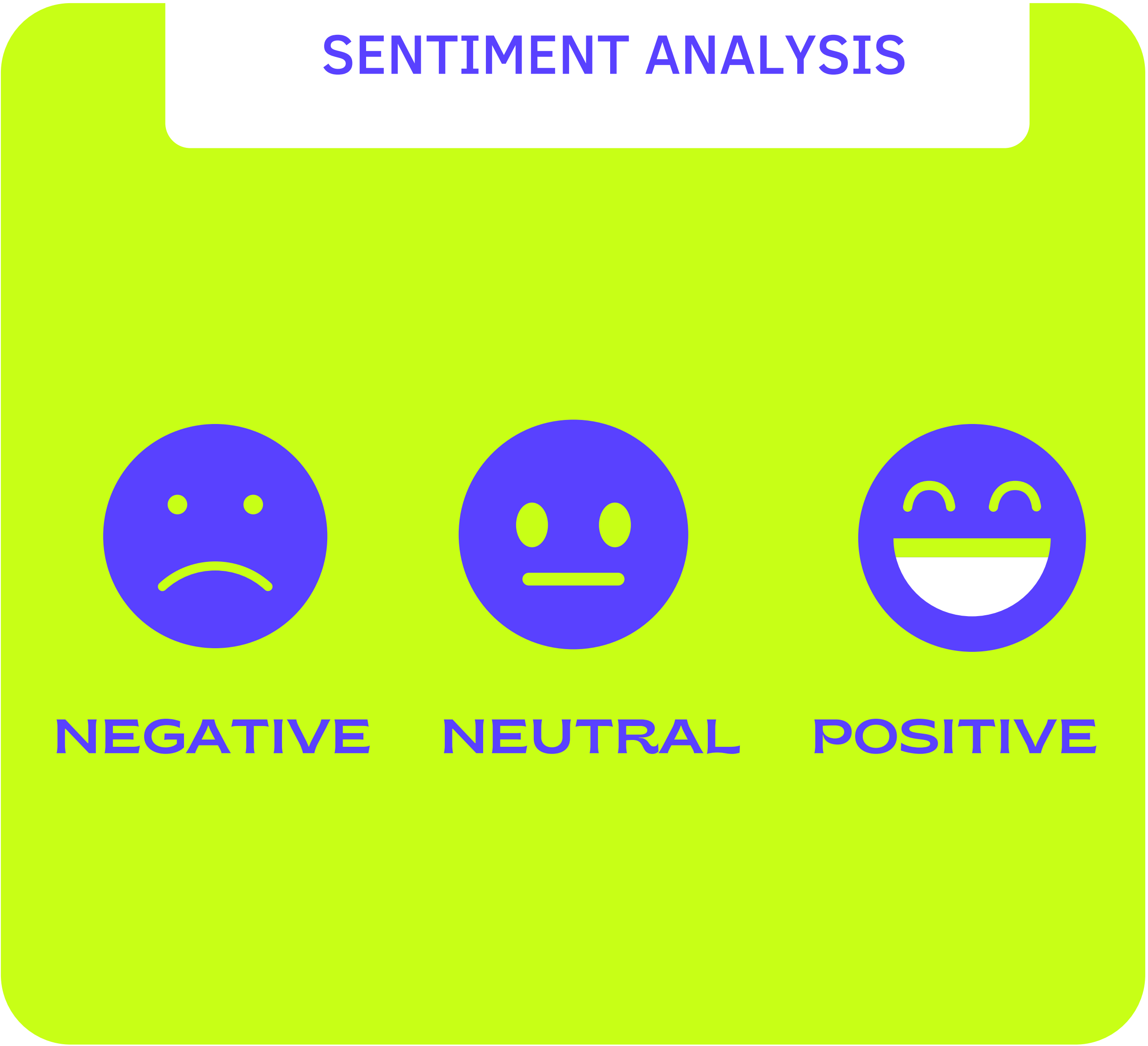
FINAL CHALLENGE

MODEL FOR SENTIMENT PREDICTION USING NEURAL NETWORK & LSTM

BINAR DATA SCIENCE BOOTCAMP WAVE 1

INTRODUCTION

- Nowadays, the flow of information, social media, and other digital channel are massively abundant and extremely easy for us to access
- In the other hand, the information in Digital era, has a tendency effect to people's behavior, based on the Journal of Research by NIKIJULUW entitled The Behavior of Society in the Digital Age.
- The impact of information, social media and other digital channel also reinforced by the thesis of Fatkhul Muin entitled Behavioral Change due to online social media usage study case in a village in central java. He concluded that there is an impact from online social media usage towards behavior or even culture.
- The need for a technological solution that could help suppress the propagation of sentiment through digital media especially negative sentiment



SENTIMENT ANALYSIS

NEGATIVE NEUTRAL POSITIVE

BINAR ACADEMY
DATA SCIENCE WAVE 1

PROBLEMS STATEMENT

What is technology capable of analyzing and predicting a sentiment in the form of text or text files?

What is suitable model's approach regarding problems stated?

How the way for data preprocessing into analysis of Sentiment?

What is coherent Interface to run the models?

RESEARCH GOALS

1 **1st Goal**

Create Technological Solutions that are able to predict sentiment in data or information through Text or Text Files

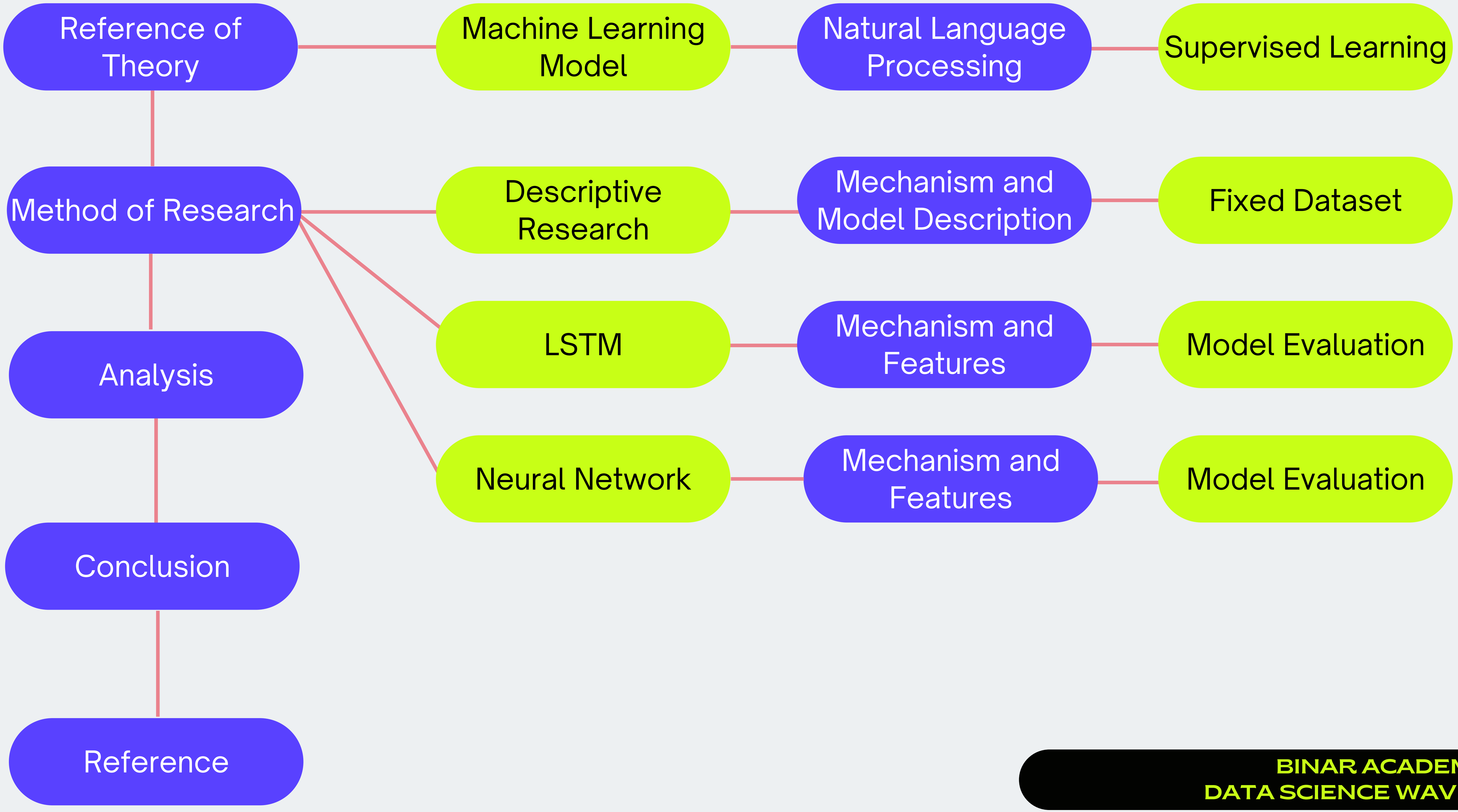
2 **2nd Goal**

Provide education and references to the public in receiving and processing data of information

3 **3rd Goal**

Suppress the propagation of negative sentiment in social media users

PRESENTATION OVERVIEW



BINAR ACADEMY
DATA SCIENCE WAVE 1

BINAR ACADEMY DATA SCINCE WAVE 1

THEORY REFERENCE

Machine Learning Definition

"the field of study that gives computers the ability to learn without explicitly being programmed." by Arthur Samuel

Why we need Machine Learning ???

67% of companies worldwide use machine learning to increase company effectiveness and efficiency. especially in Language Processing, Image Analysis & Object Detection, Fraud detection, etc.

Point to take

Referring to the problem we raised earlier, that machine learning is a suitable technological solution for analyzing and predicting sentiments, based on the Language Processing function.

NATURAL LANGUAGE PROCESSING

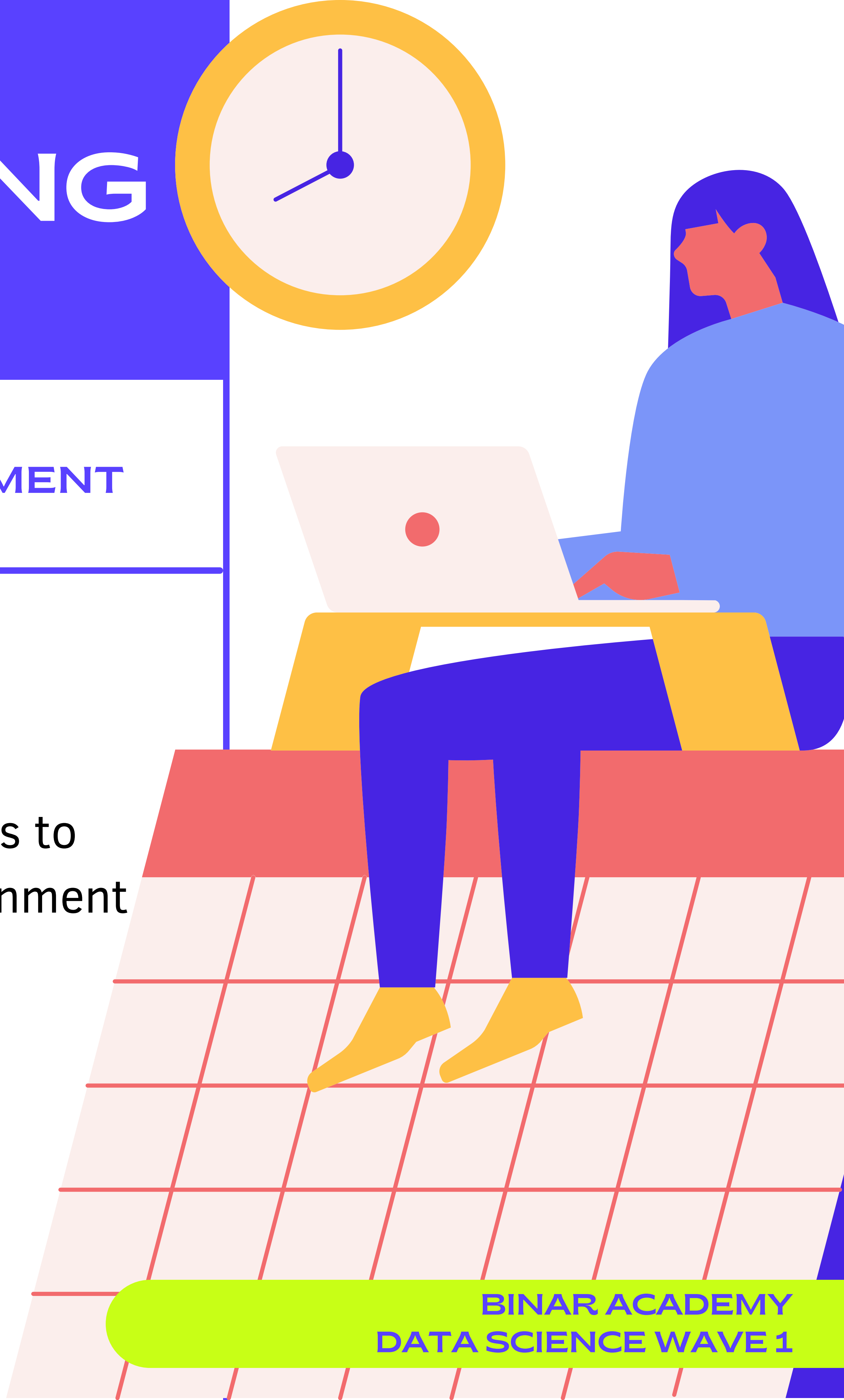
NLP is one of field in Machine Learning, which focuses on understanding human language more accurately.

NLP used for »»

- Chatbot
- Sentiment Analysis
- Google translate
- Voice Recognition
- Etc

TYPE OF MACHINE LEARNING

SUPERVISED	UNSUPERVISED	REINFORCEMENT
Task driven (Regression/Classification)	Data Driven (Clustering)	Algorithm learns to react to an environment



BINAR ACADEMY
DATA SCIENCE WAVE 1

METHOD OF RESEARCH

1

Descriptive Research

This research aims to describe a reality phenomenon with a descriptive statistical approach

2

Method for Sentiment Analysis (2 Models Approach)

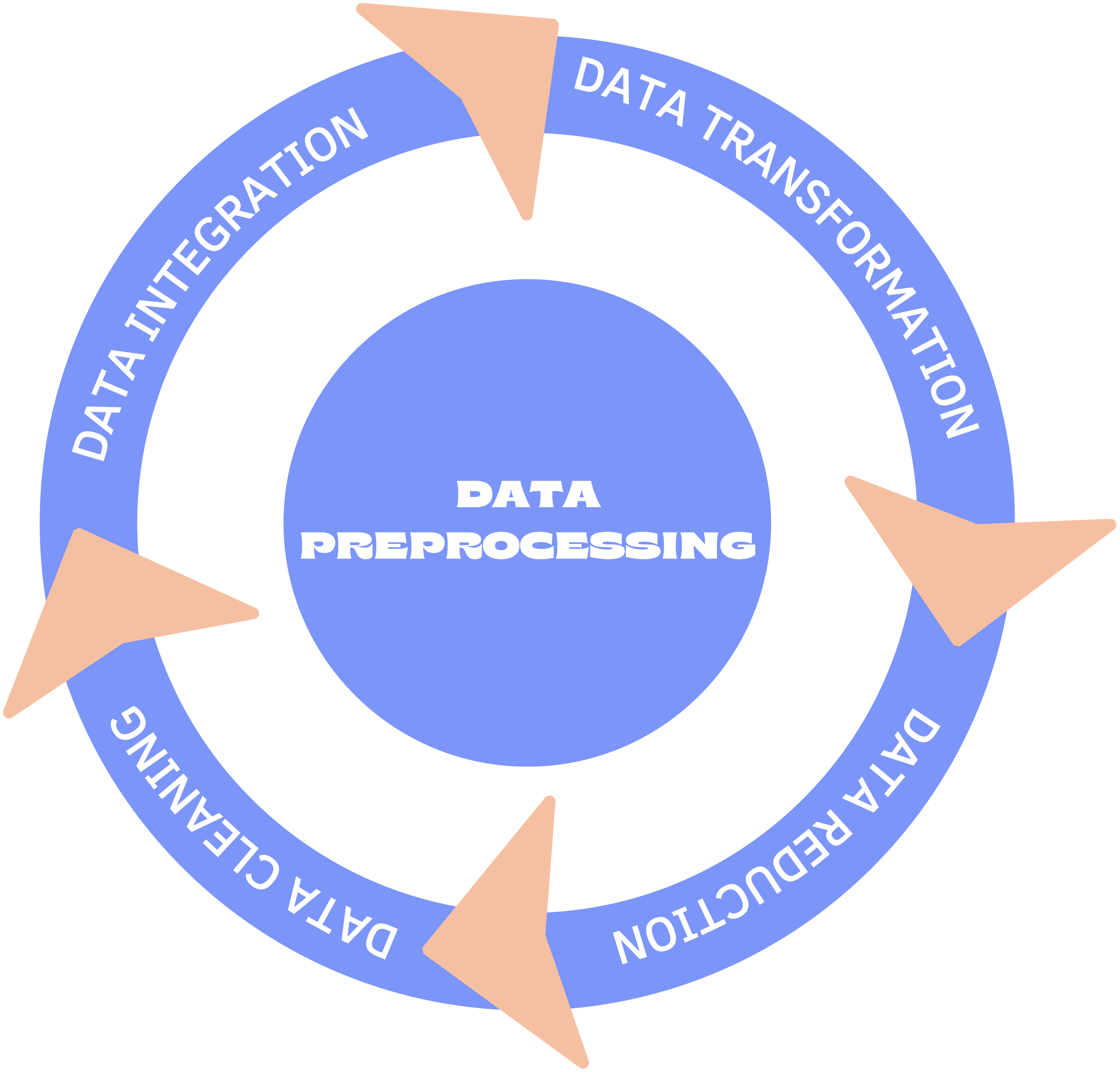
- LSTM based on Tensorflow
- Neural Network based on Sklearn

3

Data

The use of data for experiments on the model is fixed data which has 3 sentiment classifications (Positive, Negative, Neutral)

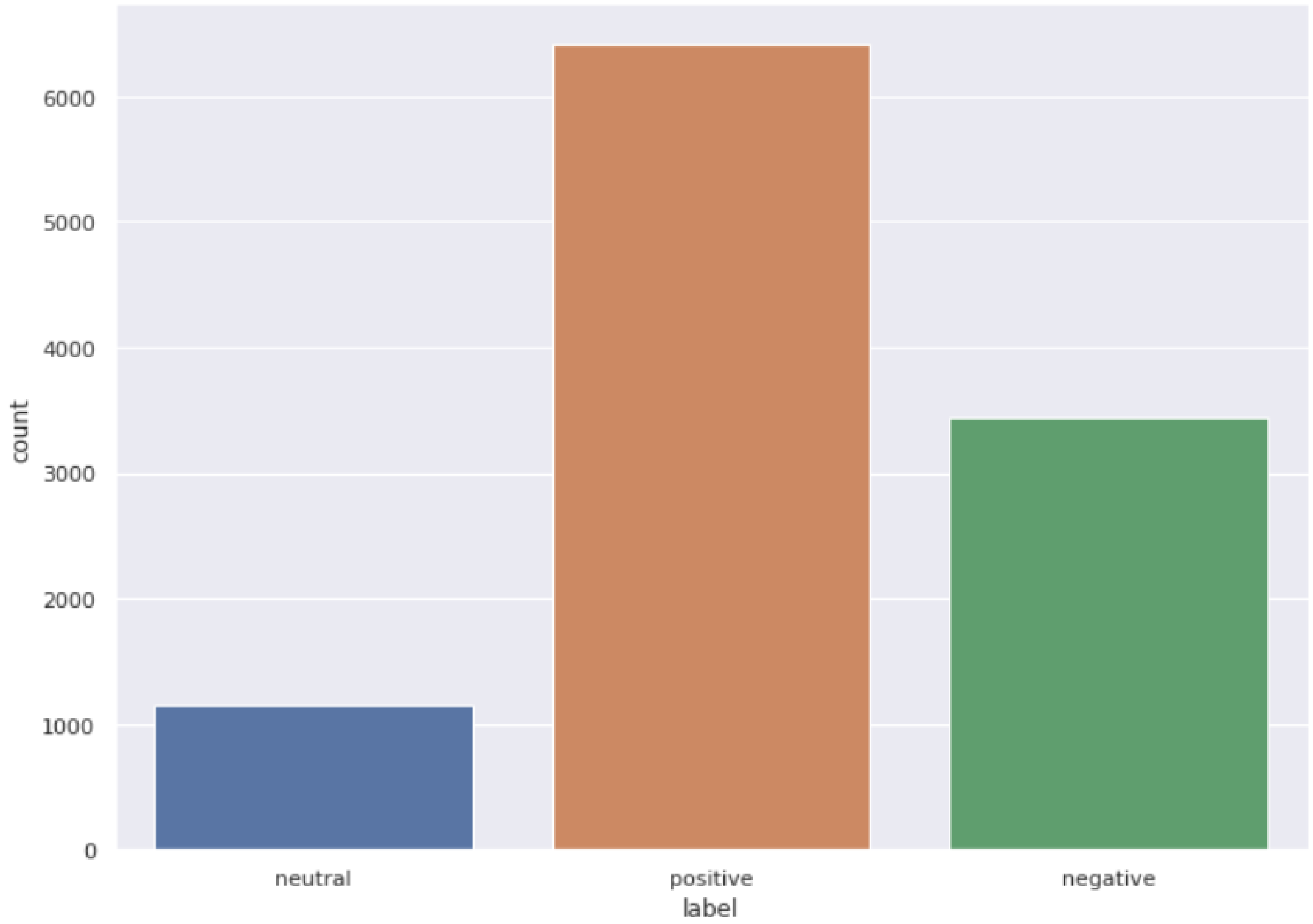
PREPROCESSING



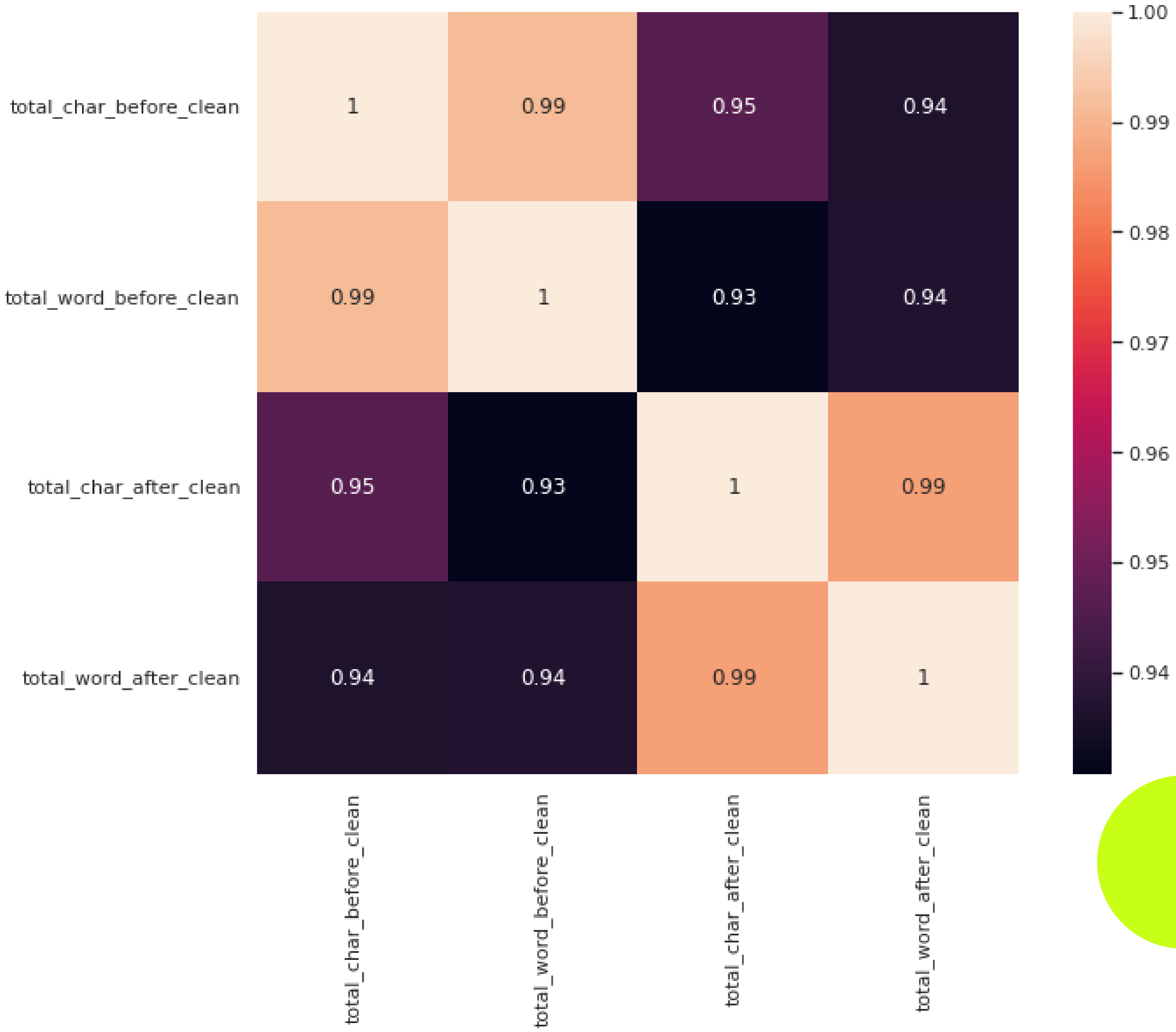
ANALYSIS

BINAR ACADEMY DATA SCIENCE WAVE 1

DATA DISTRIBUTION



Based on the dataset that we applied, it can be seen that there is indeed an imbalance in the distribution of sentiment data. Much more positive sentiment data with numbers above 6000 rows



Visualization by Heatmap

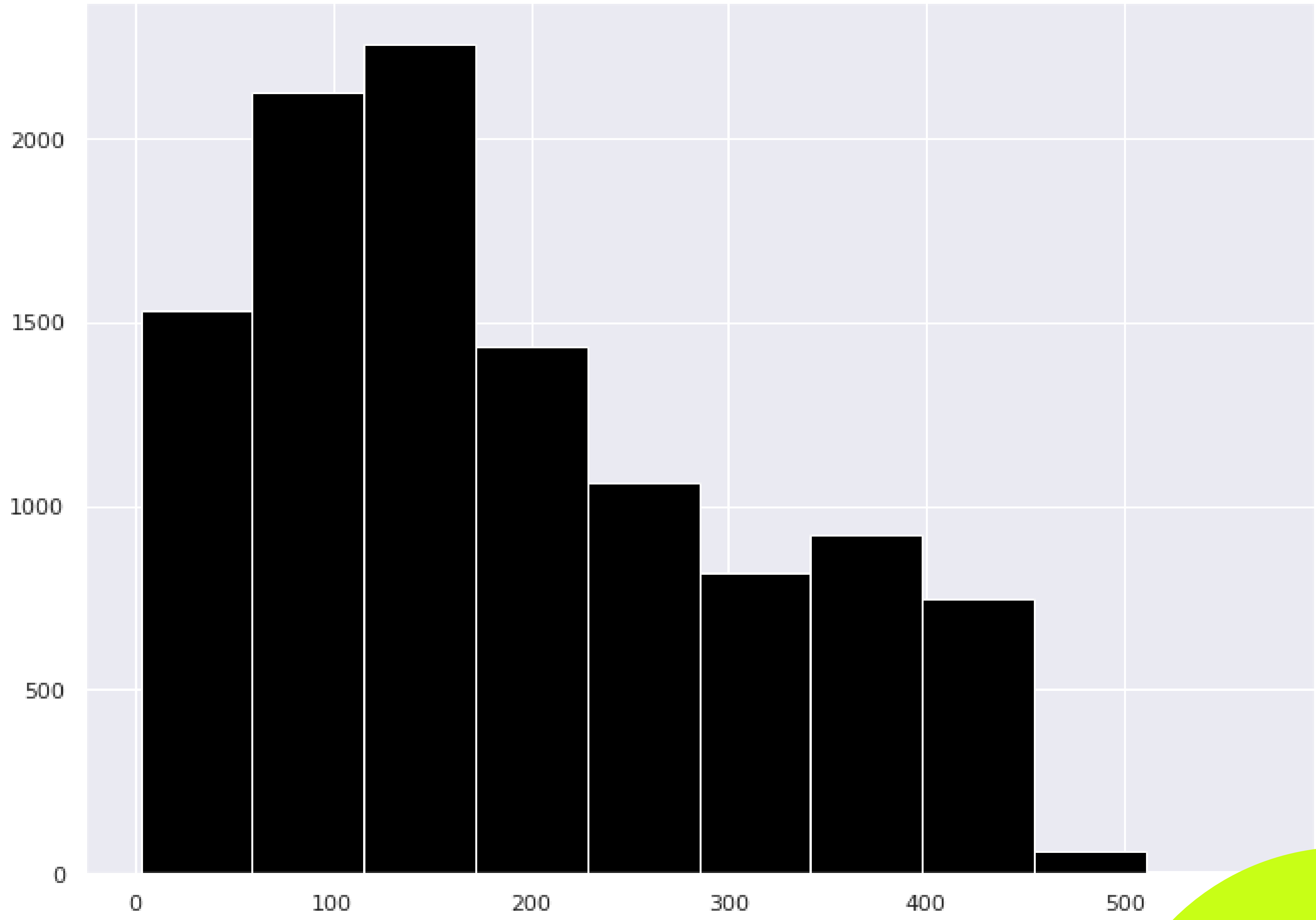
We can see the comparison of the value of the character before being cleaned in terms of value is 1, while after being cleaned it is 0.95

The total word value before being cleaned was 0.99 to 0.94 the total word value after being cleaned

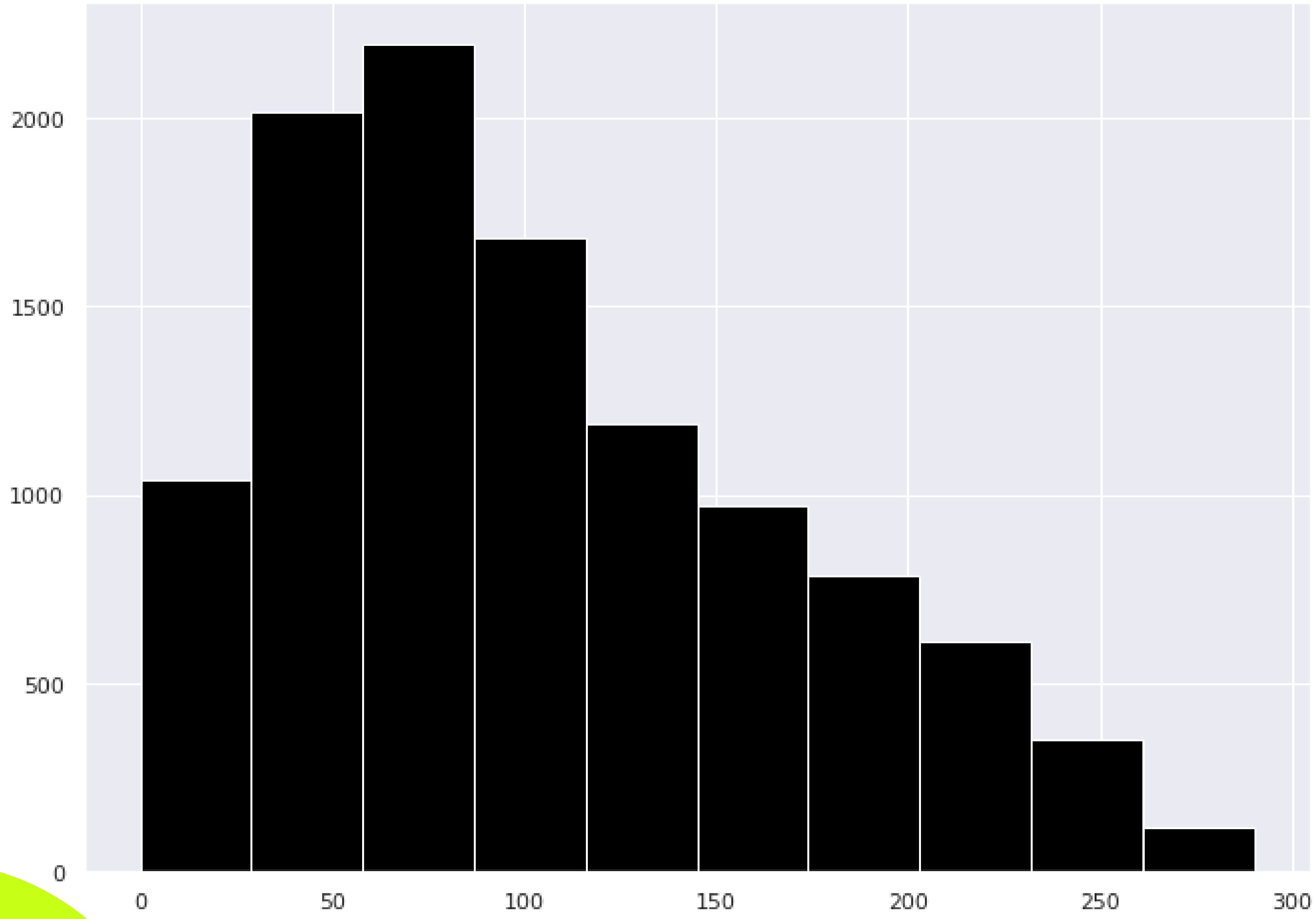
BINAR ACADEMY
DATA SCIENCE WAVE 1

HISTOGRAM

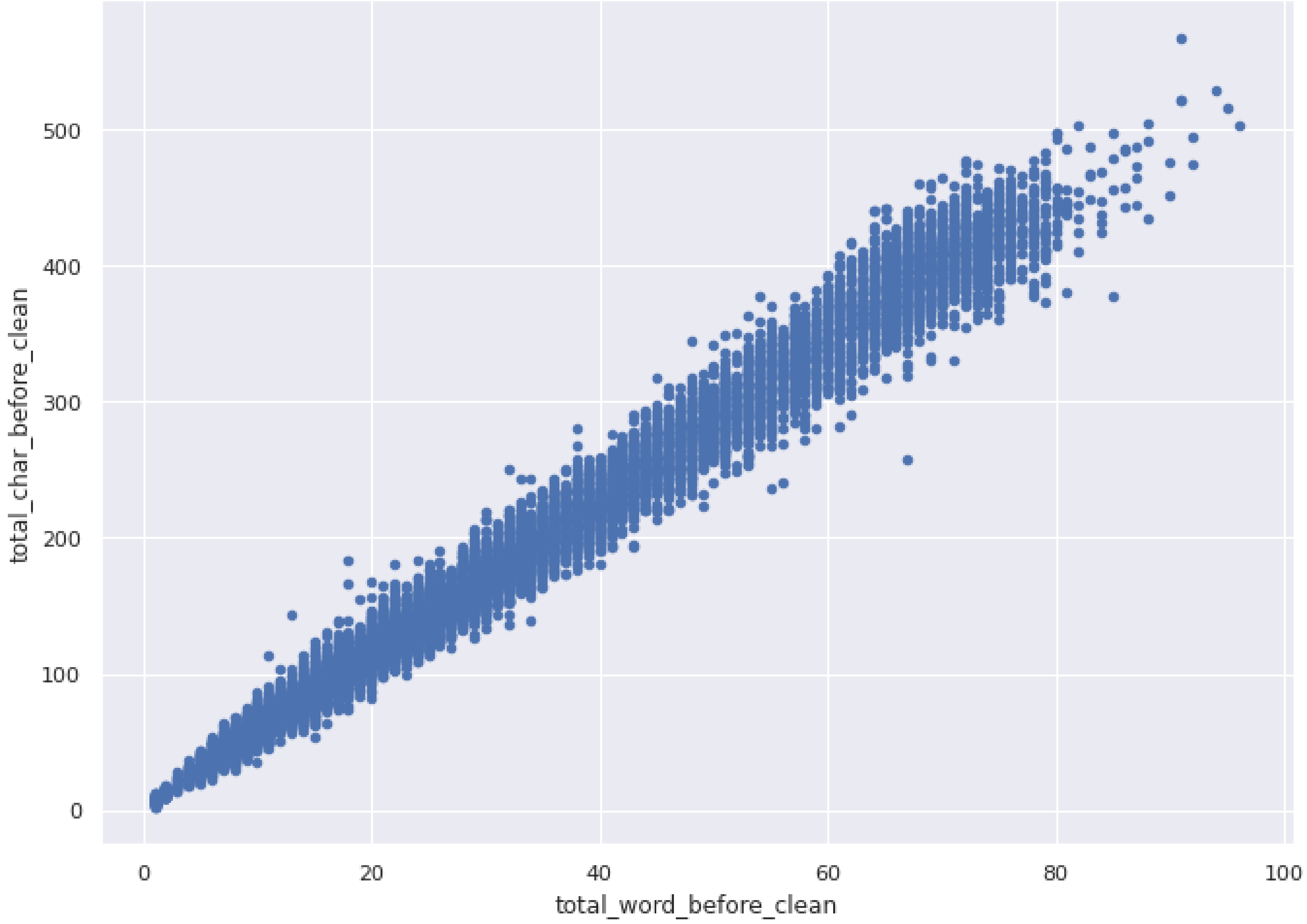
Total Character Before Clean

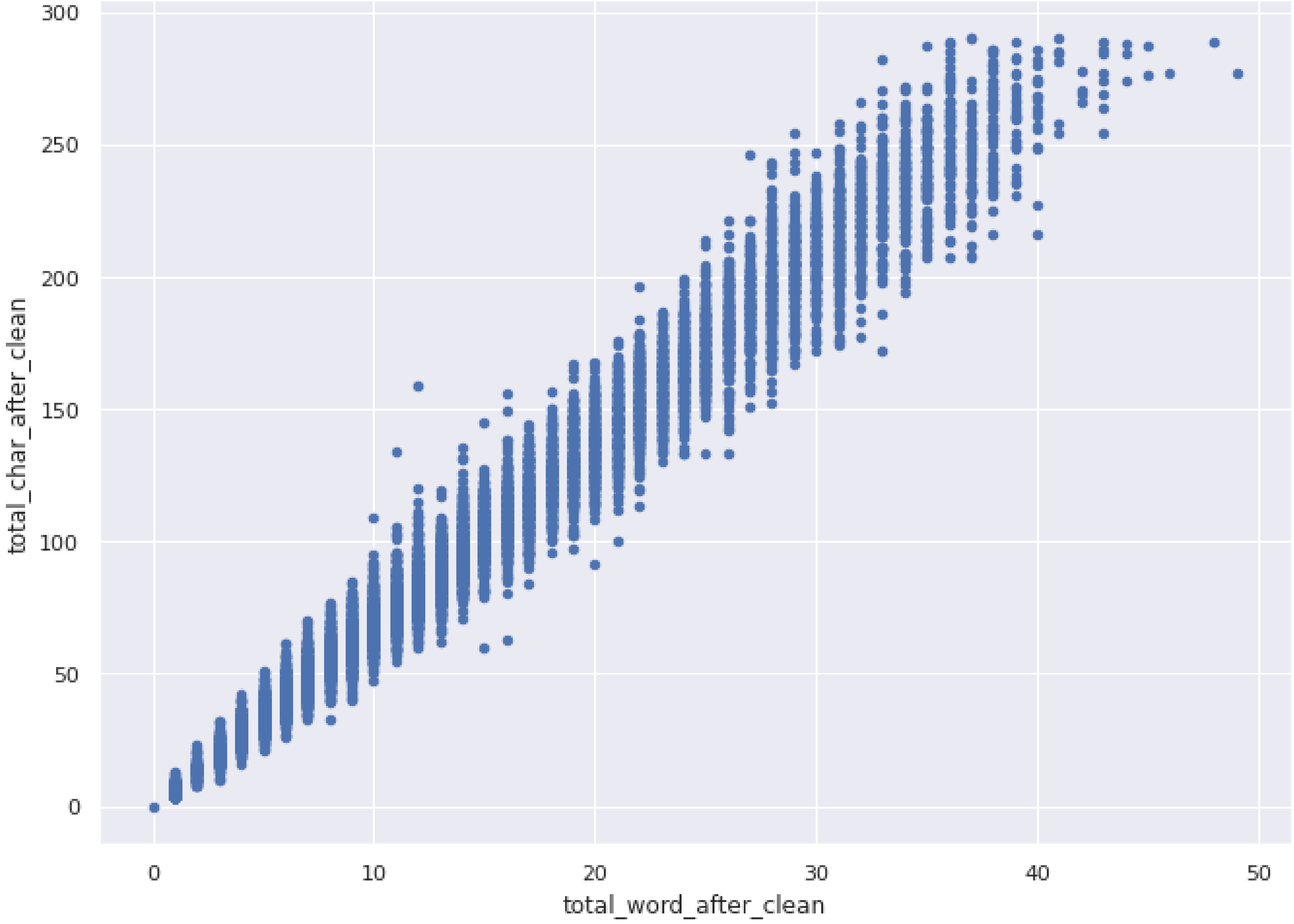


Total Character After Clean



VISUALIZATION SCATTER PLOT



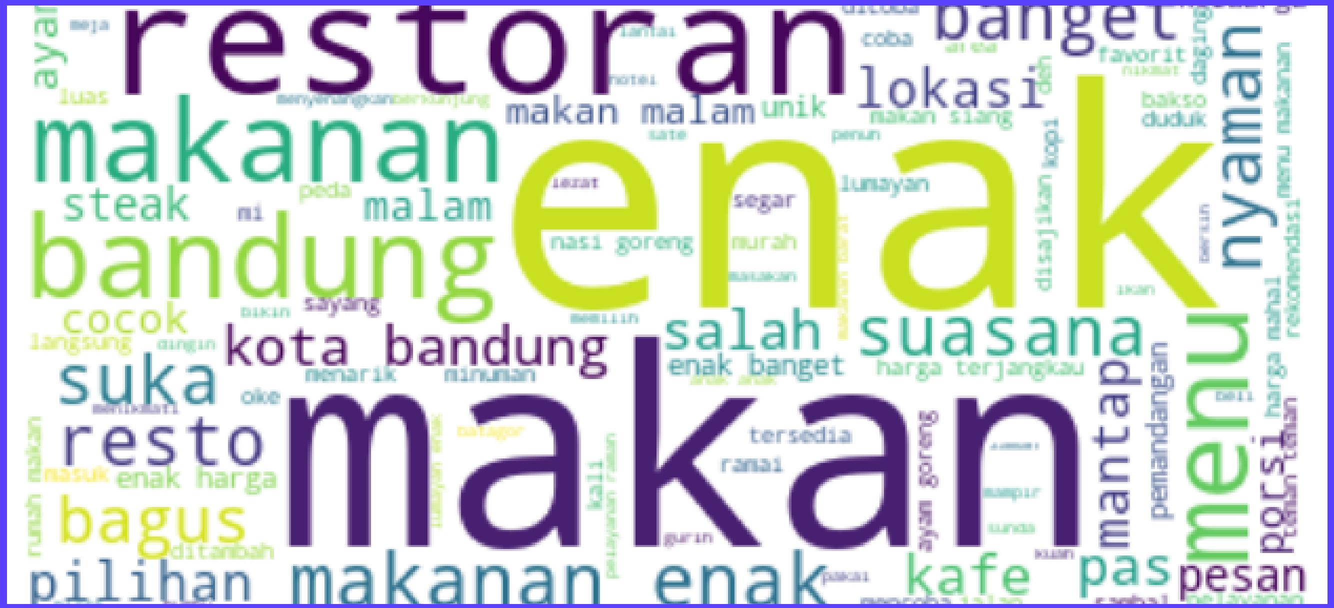


Based on this Scatter Plot visualization, we can see the density level between before and after cleansing

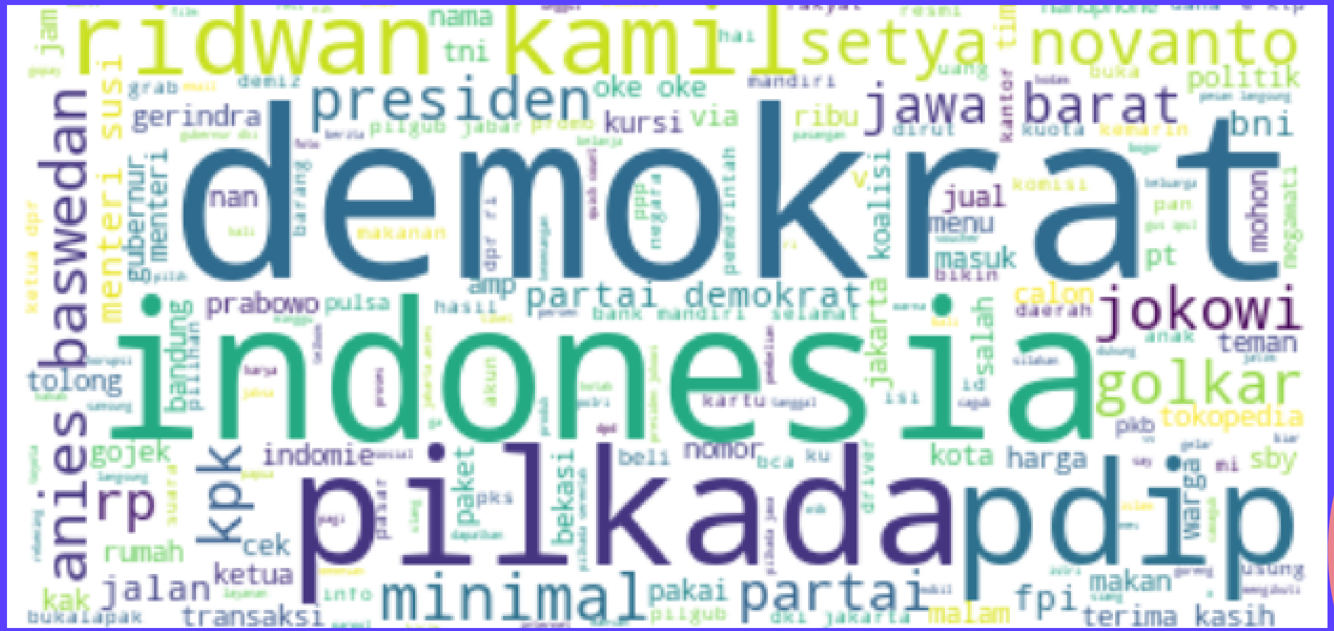
THE MOST FREQUENT WORDS

BINAR ACADEMY DATA SCIENCE WAVE 1

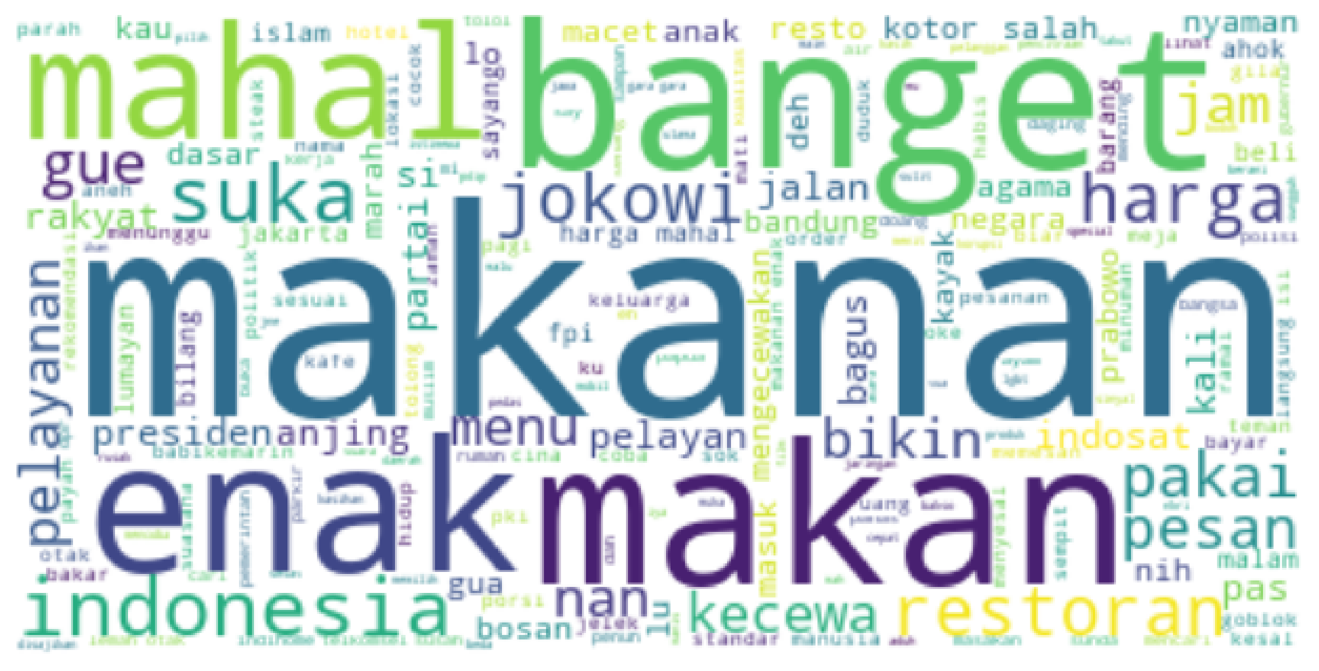
POSITIVE WORDS 😁



NEUTRAL WORDS 😐



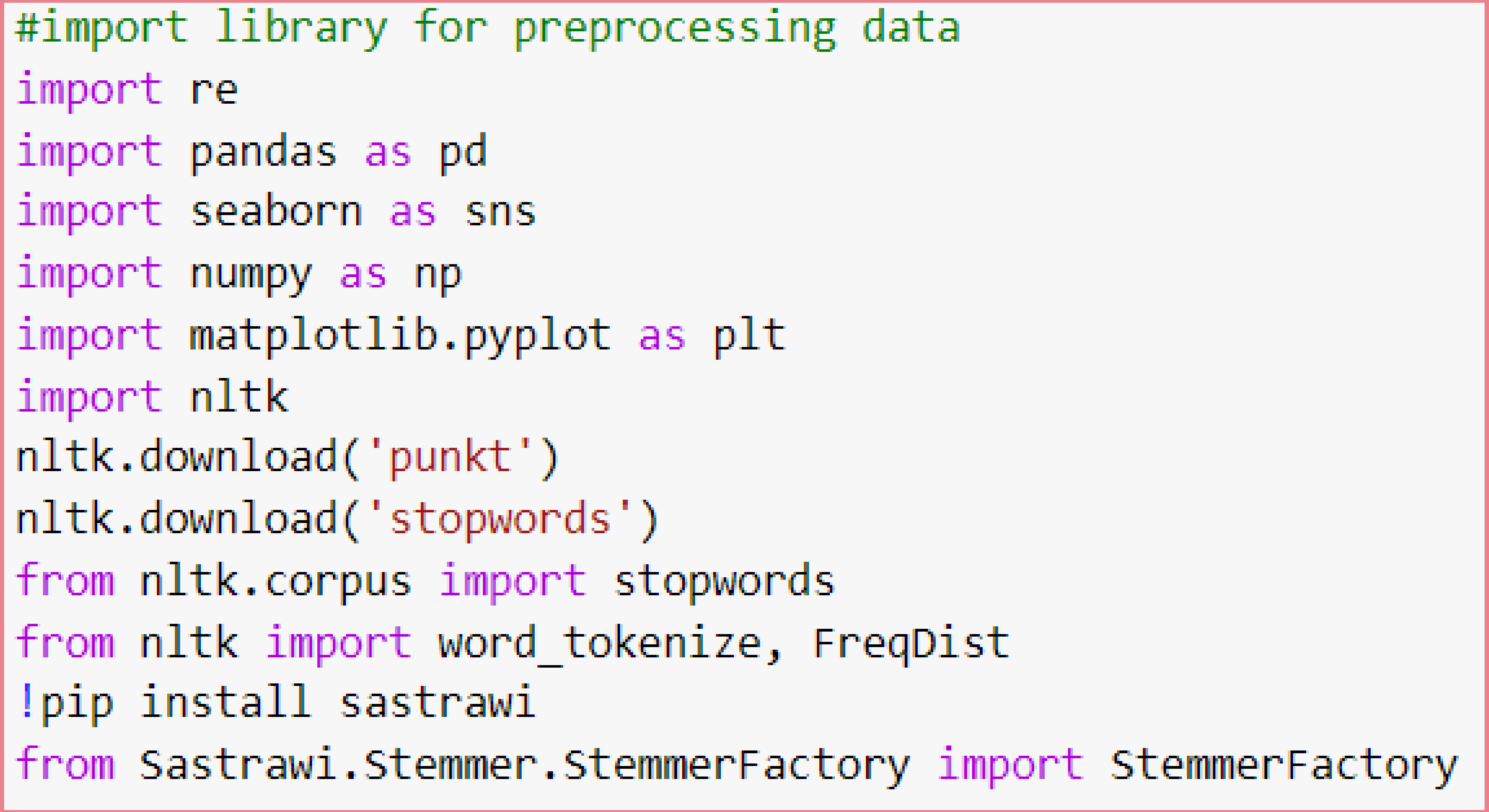
NEGATIVE WORDS ☹️



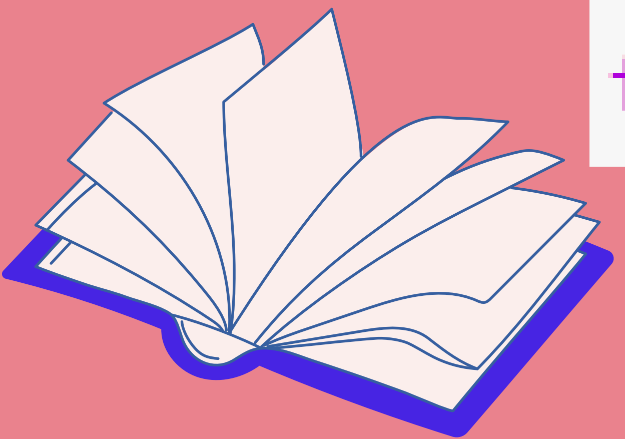
MODEL IMPLEMENTATION

BINAR ACADEMY DATA SCIENCE WAVE 1

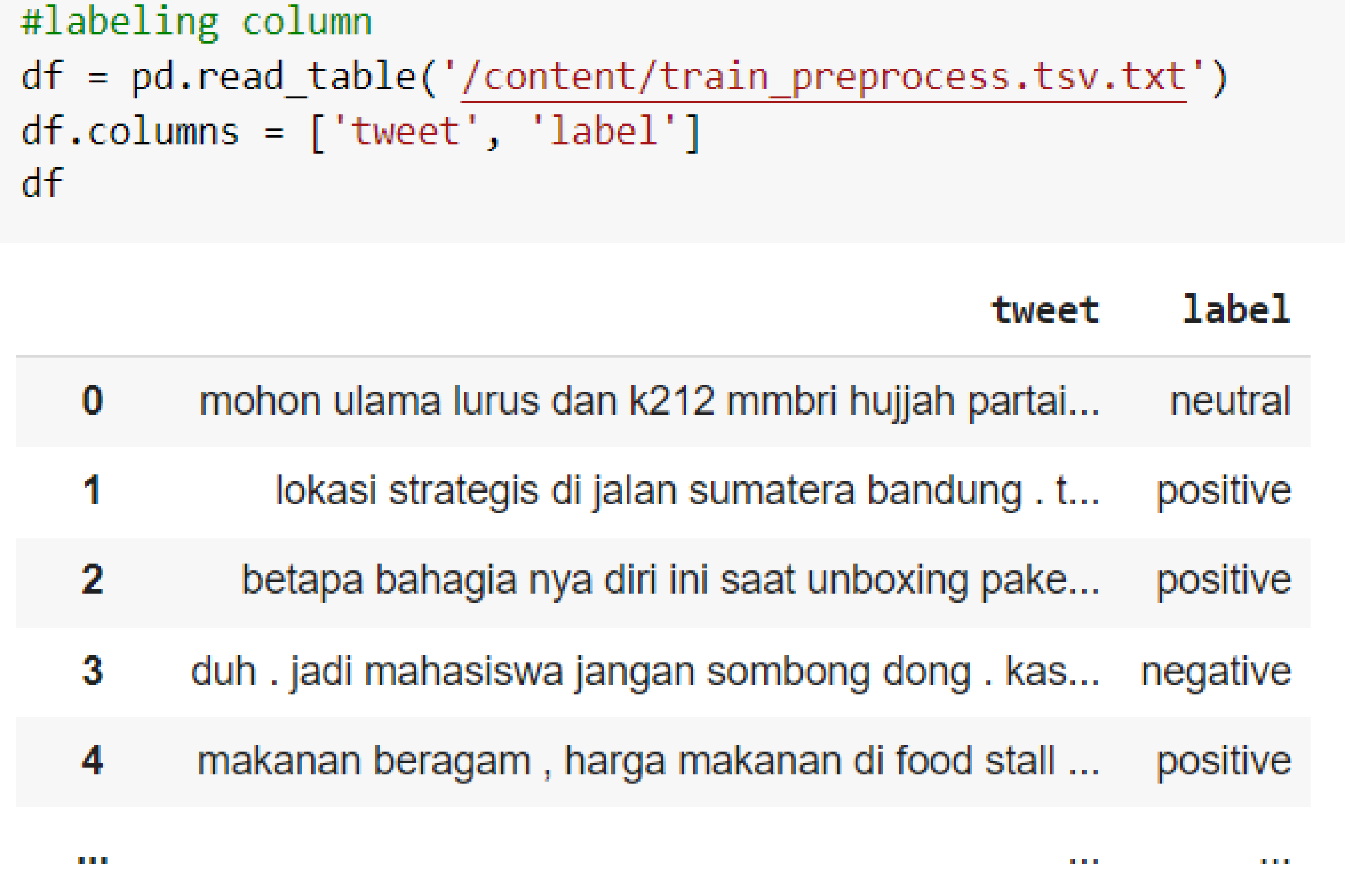
MODULE OF LIBRARIES



```
#import library for preprocessing data
import re
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import nltk
nltk.download('punkt')
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk import word_tokenize, FreqDist
!pip install sastrawi
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
```



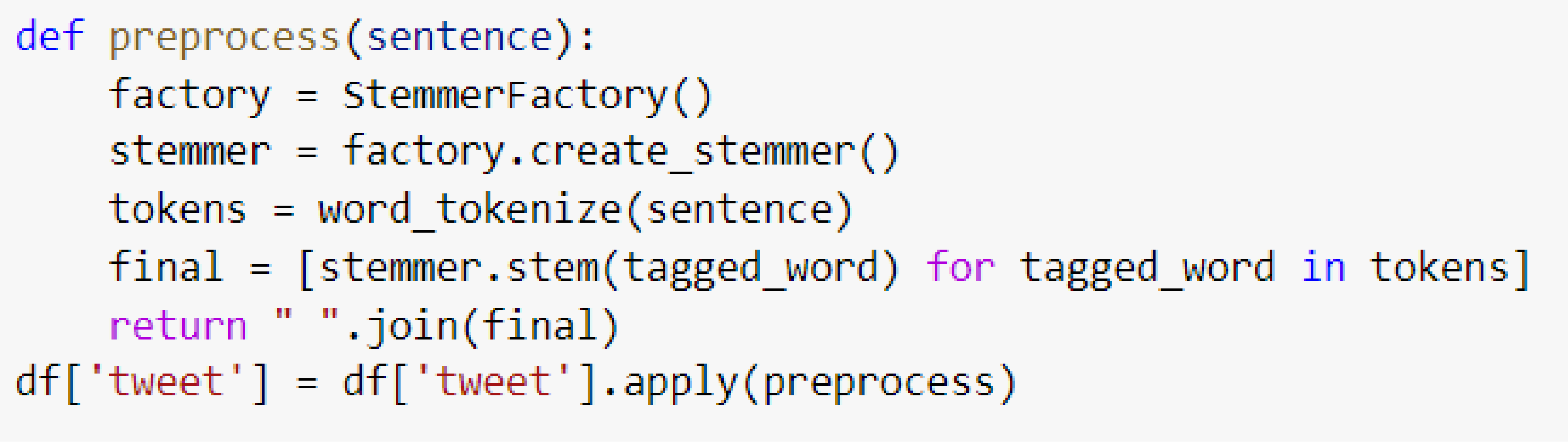
PREPROCESSING DATA



```
#labeling column
df = pd.read_table('/content/train_preprocess.tsv.txt')
df.columns = ['tweet', 'label']
df
```

	tweet	label
0	mohon ulama lurus dan k212 mmbri hujjah partai...	neutral
1	lokasi strategis di jalan sumatera bandung . t...	positive
2	betapa bahagia nya diri ini saat unboxing pake...	positive
3	duh . jadi mahasiswa jangan sombong dong . kas...	negative
4	makanan beragam , harga makanan di food stall ...	positive
...	...	...

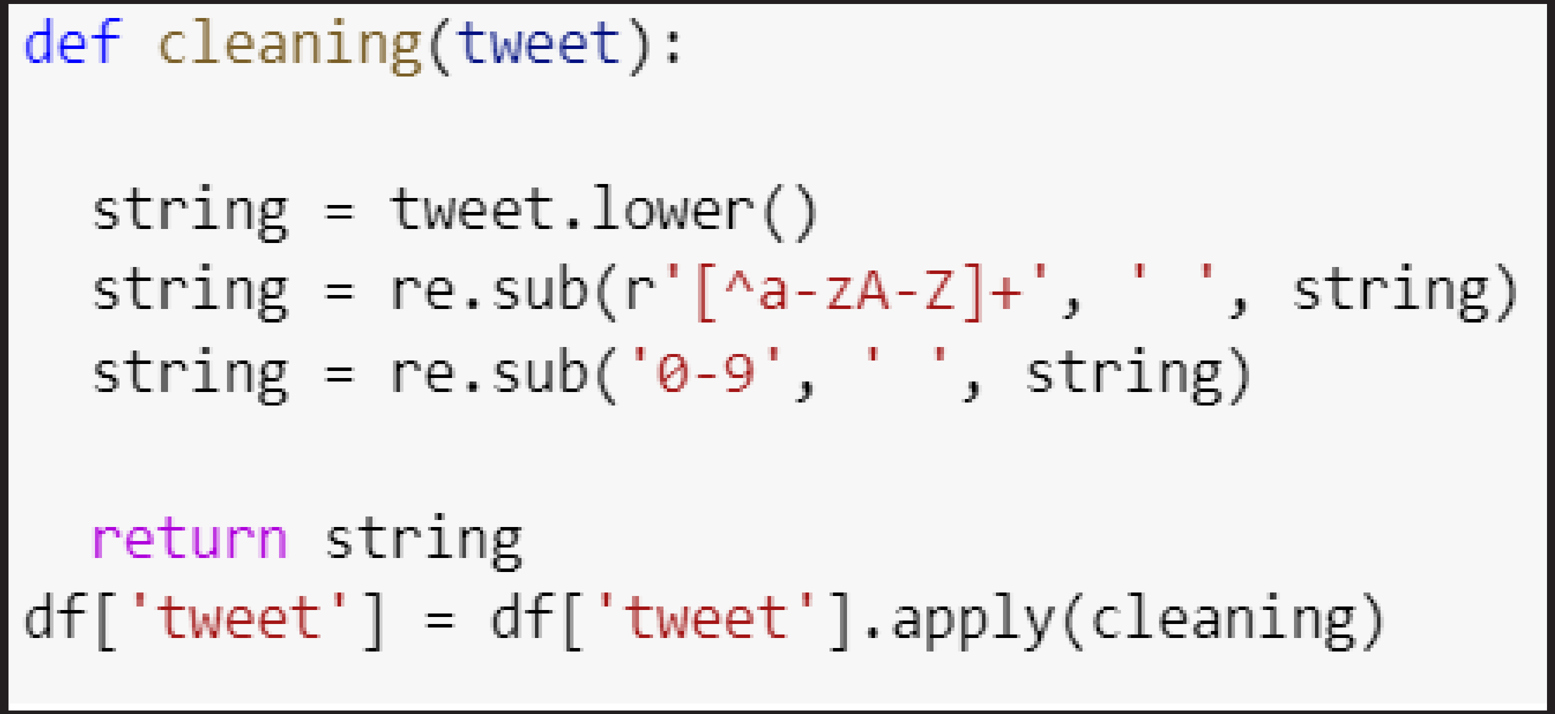
Using Pandas to import files into a dataframe



```
def preprocess(sentence):
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()
    tokens = word_tokenize(sentence)
    final = [stemmer.stem(tagged_word) for tagged_word in tokens]
    return " ".join(final)
df['tweet'] = df['tweet'].apply(preprocess)
```

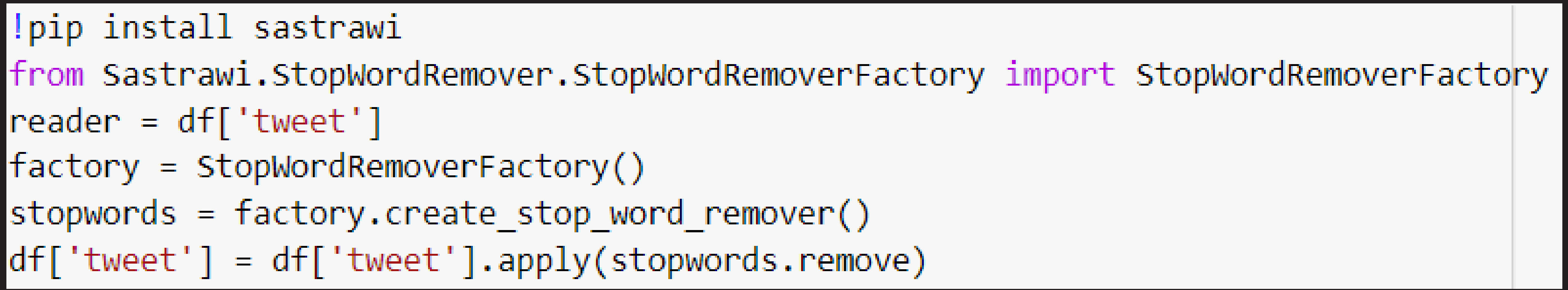
Stemming and Tokenization Process Stages

PREPROCESSING DATA

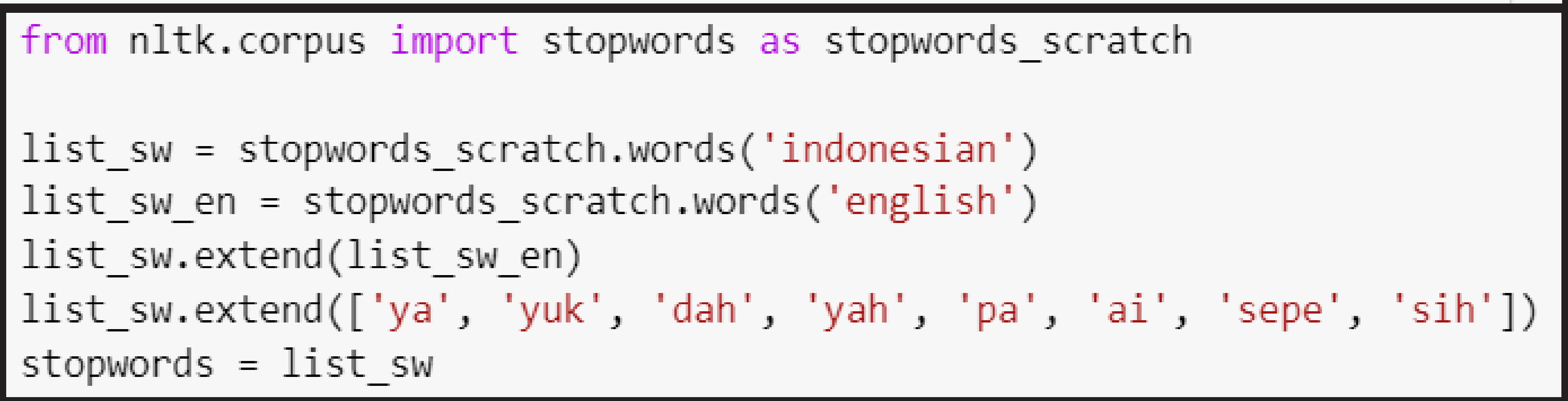


```python
def cleaning(tweet):

  string = tweet.lower()
  string = re.sub(r'[^a-zA-Z]+', ' ', string)
  string = re.sub('0-9', ' ', string)

  return string
df['tweet'] = df['tweet'].apply(cleaning)
```

Cleaning function for data cleansing using regex



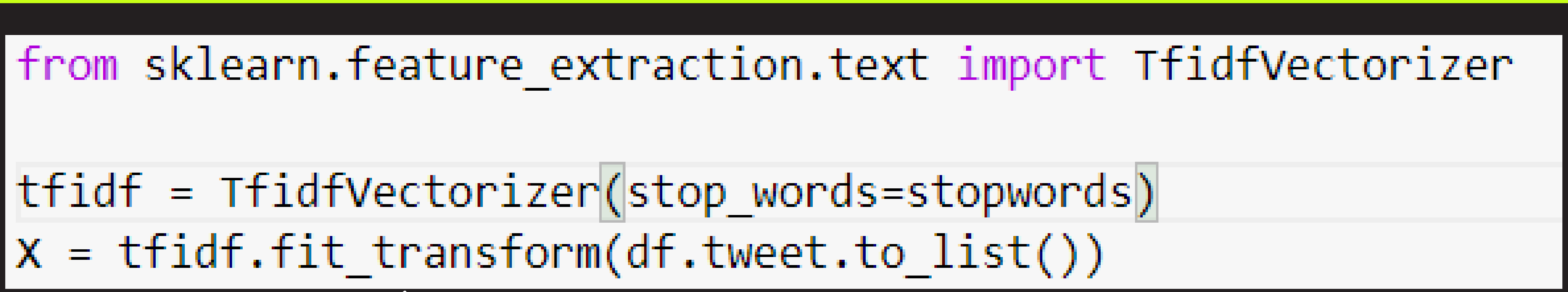
```python
!pip install sastrawi
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
reader = df['tweet']
factory = StopWordRemoverFactory()
stopwords = factory.create_stop_word_remover()
df['tweet'] = df['tweet'].apply(stopwords.remove)
```



```python
from nltk.corpus import stopwords as stopwords_scratch

list_sw = stopwords_scratch.words('indonesian')
list_sw_en = stopwords_scratch.words('english')
list_sw.extend(list_sw_en)
list_sw.extend(['ya', 'yuk', 'dah', 'yah', 'pa', 'ai', 'sepe', 'sih'])
stopwords = list_sw
```

Applying stopwords for words that have no meaning

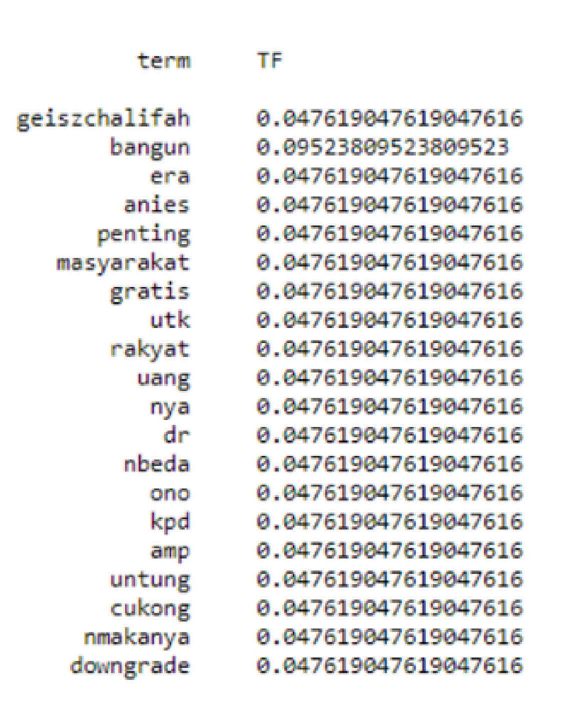
PREPROCESSING DATA



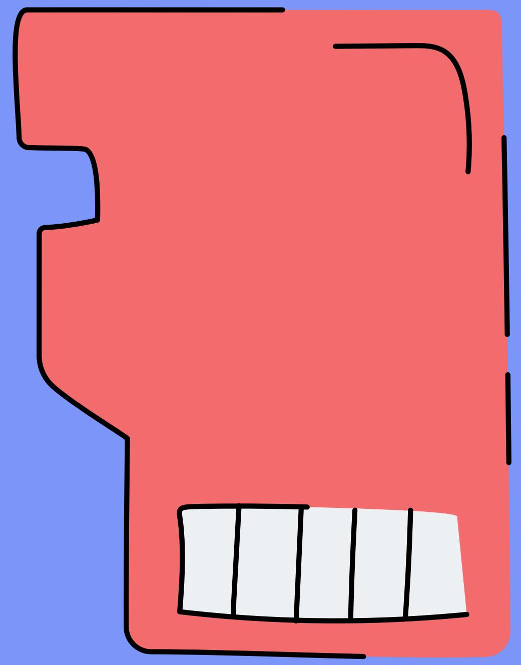
```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(stop_words=stopwords)
X = tfidf.fit_transform(df.tweet.to_list())
```

We use TfidfVectorizer from Sklearn

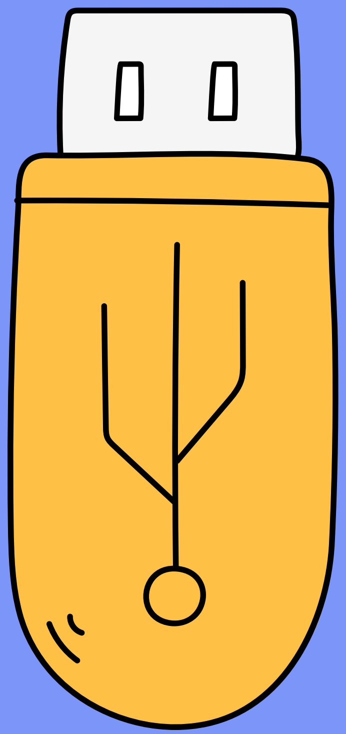
The final stage of data preprocessing is to carry out Term weighting or weighting of objects, because this will later be used as the basis for calculating algorithms in predicting object values in generalizing the model.



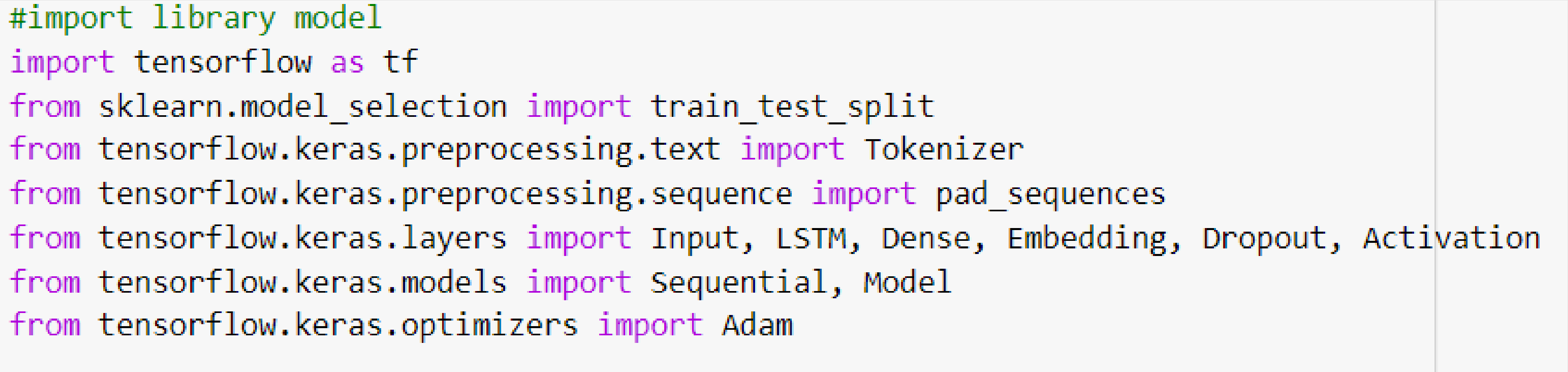
```
         term       TF

geiszchalifah       0.047619047619047616
       bangun       0.09523809523809523
          era       0.047619047619047616
        anies       0.047619047619047616
      penting       0.047619047619047616
   masyarakat       0.047619047619047616
       gratis       0.047619047619047616
          utk       0.047619047619047616
       rakyat       0.047619047619047616
         uang       0.047619047619047616
          nya       0.047619047619047616
           dr       0.047619047619047616
        nbeda       0.047619047619047616
          ono       0.047619047619047616
          kpd       0.047619047619047616
          amp       0.047619047619047616
       untung       0.047619047619047616
       cukong       0.047619047619047616
     nmakanya       0.047619047619047616
    downgrade       0.047619047619047616
```



LONG SHORT -TERM MEMORY MODEL

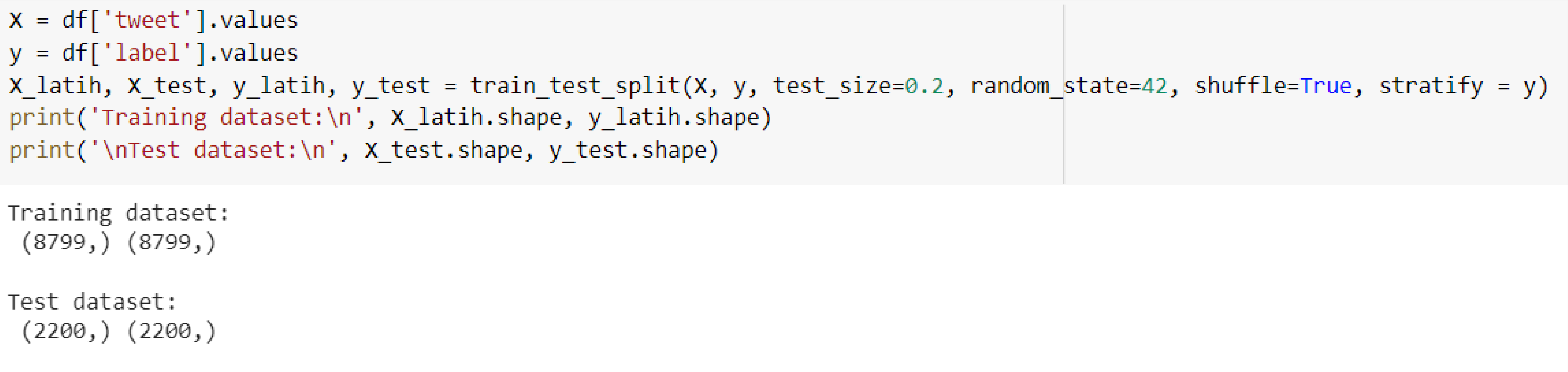


BINAR ACADEMY DATA SCIENCE WAVE 1



```
#import library model
import tensorflow as tf
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.layers import Input, LSTM, Dense, Embedding, Dropout, Activation
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.optimizers import Adam
```

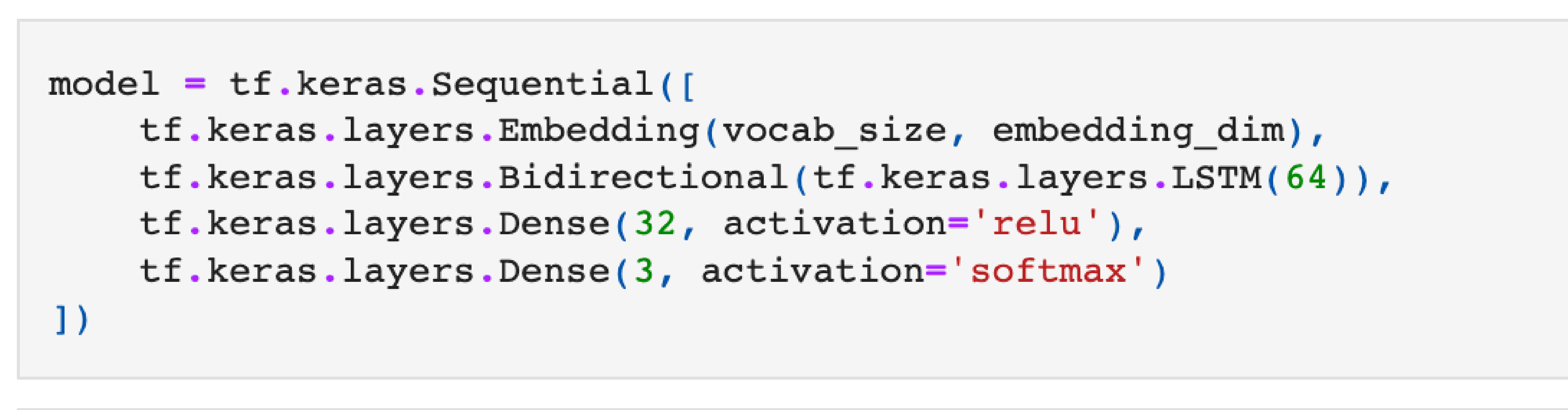
Libraries to implement LSTM Model



```
X = df['tweet'].values
y = df['label'].values
X_latih, X_test, y_latih, y_test = train_test_split(X, y, test_size=0.2, random_state=42, shuffle=True, stratify = y)
print('Training dataset:\n', X_latih.shape, y_latih.shape)
print('\nTest dataset:\n', X_test.shape, y_test.shape)
```

```
Training dataset:
 (8799,) (8799,)

Test dataset:
 (2200,) (2200,)
```

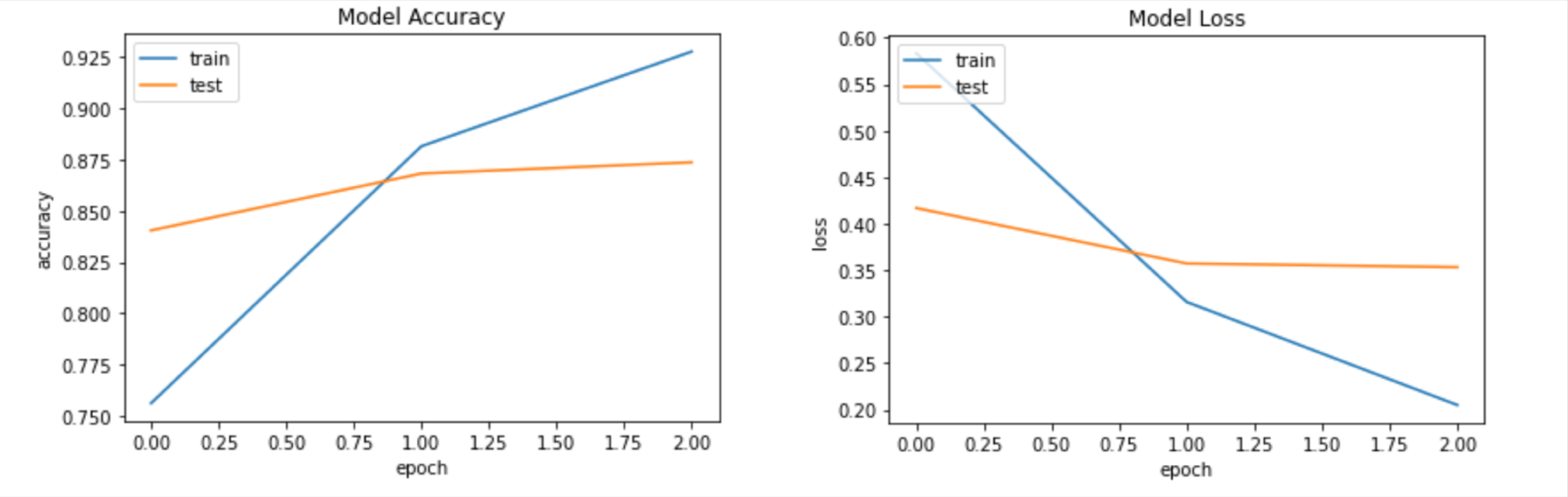
dividing the variables in the model, before the training and test process is carried out



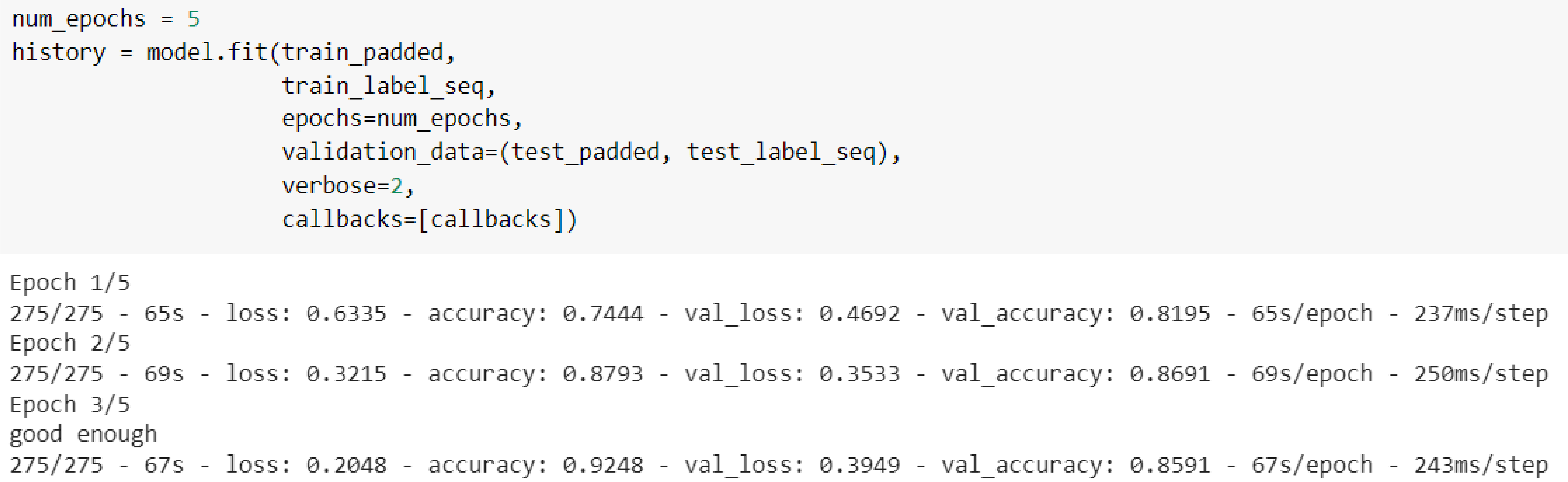
```
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(vocab_size, embedding_dim),
    tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64)),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dense(3, activation='softmax')
])
```

Model Layering

LSTM MODEL LOSS & ACCURACY (TRAIN,TEST)



MODEL FITTING

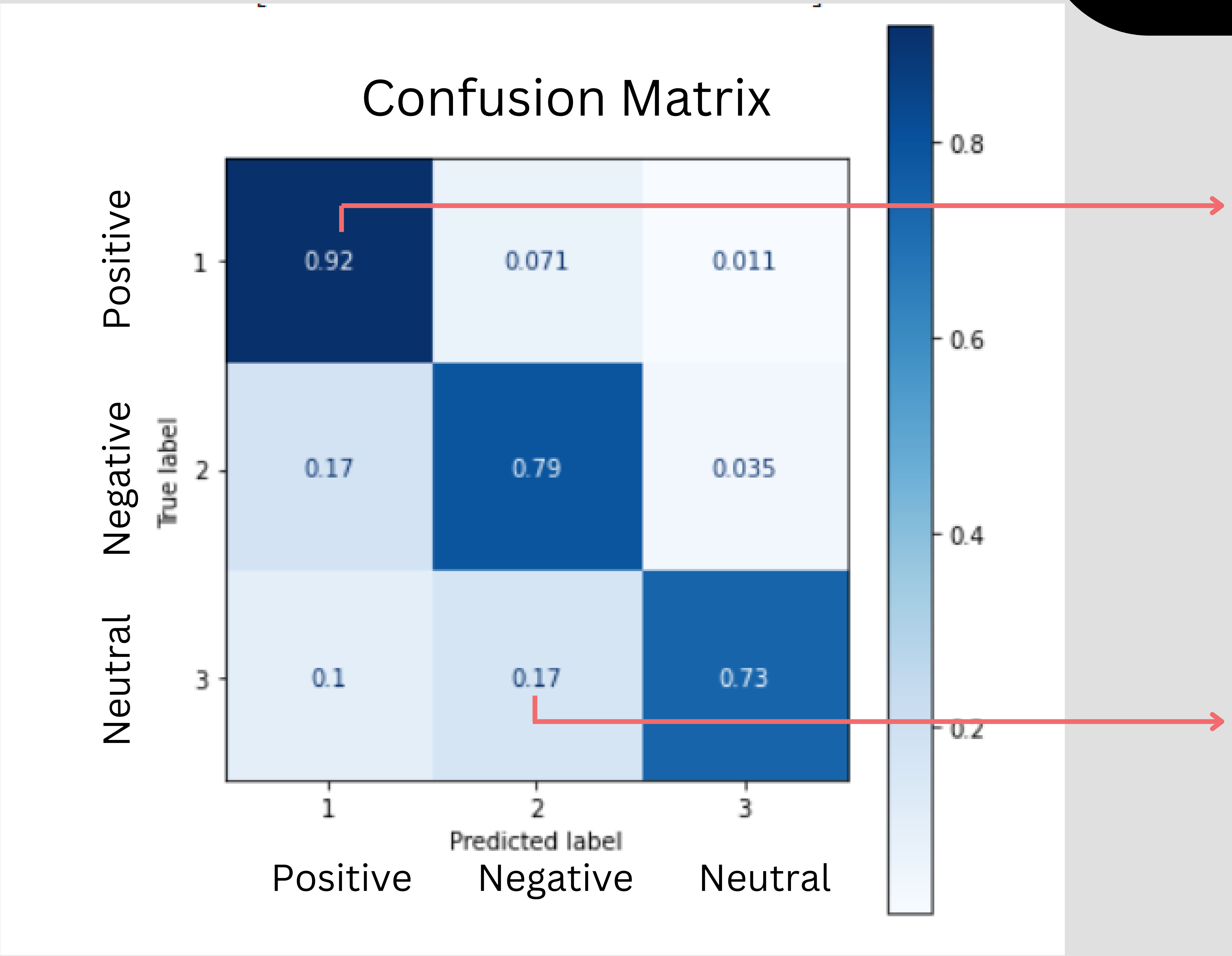


```
num_epochs = 5
history = model.fit(train_padded,
                    train_label_seq,
                    epochs=num_epochs,
                    validation_data=(test_padded, test_label_seq),
                    verbose=2,
                    callbacks=[callbacks])
```

```
Epoch 1/5
275/275 - 65s - loss: 0.6335 - accuracy: 0.7444 - val_loss: 0.4692 - val_accuracy: 0.8195 - 65s/epoch - 237ms/step
Epoch 2/5
275/275 - 69s - loss: 0.3215 - accuracy: 0.8793 - val_loss: 0.3533 - val_accuracy: 0.8691 - 69s/epoch - 250ms/step
Epoch 3/5
good enough
275/275 - 67s - loss: 0.2048 - accuracy: 0.9248 - val_loss: 0.3949 - val_accuracy: 0.8591 - 67s/epoch - 243ms/step
```

The Source code above shows the epoch 3/5 model has
loss value = 20.48% , accuray value = 92.48% with comparison on
val_loss value = 39.49%, val_accuracy value = 85.91%
in conclusion the fitting model is in a good fit enough position

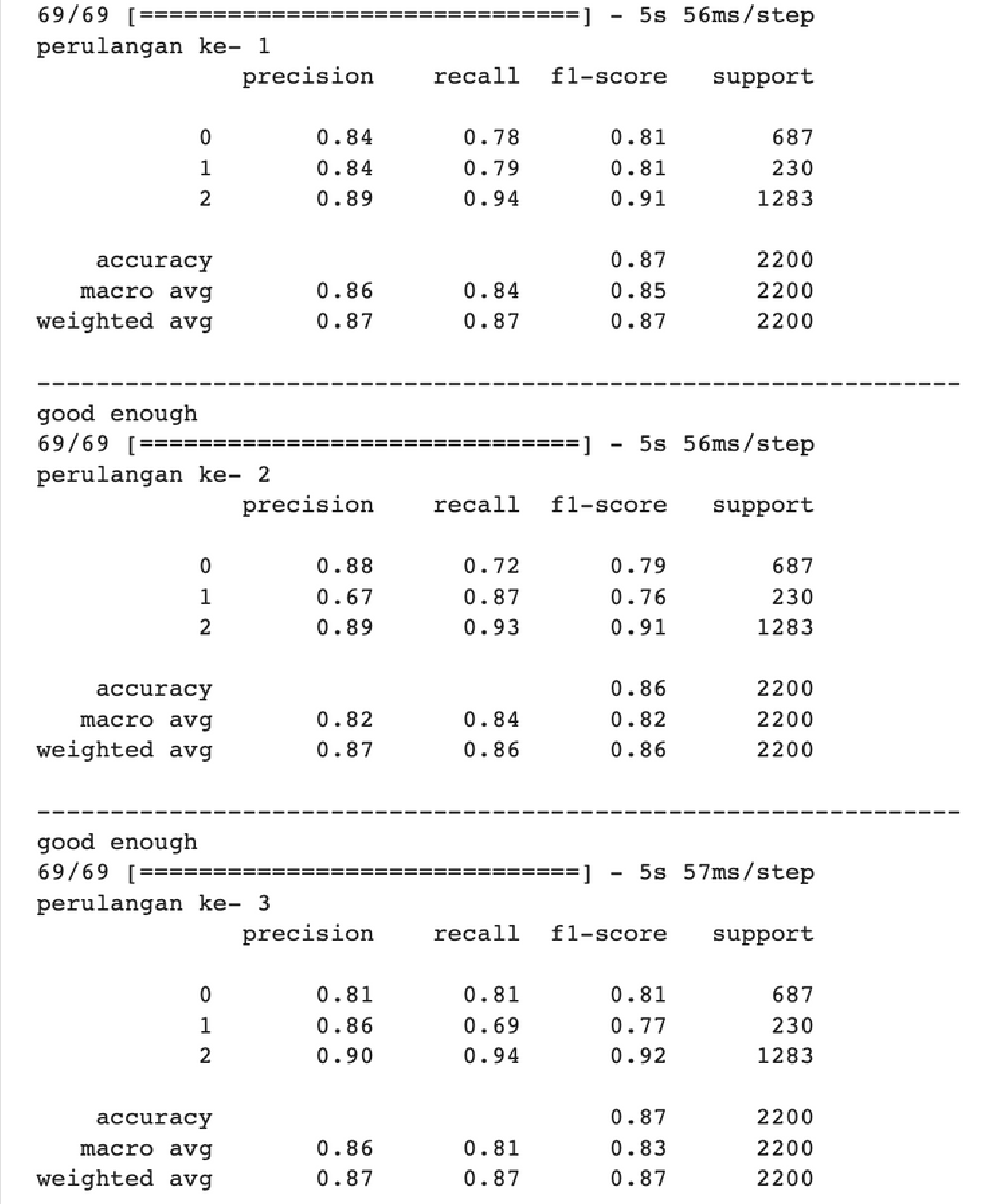
LSTM Model Evaluation



The True Positive position is at 0.92, where out of 10 predictions 9 are in accordance with the facts

False Negative position at 0.17, the opposite of TP or true positive

EVALUATION BASED ON CLASSIFICATION REPORT



```
69/69 [==============================] - 5s 56ms/step
perulangan ke- 1
              precision    recall  f1-score   support

           0       0.84      0.78      0.81       687
           1       0.84      0.79      0.81       230
           2       0.89      0.94      0.91      1283

    accuracy                           0.87      2200
   macro avg       0.86      0.84      0.85      2200
weighted avg       0.87      0.87      0.87      2200

---------------------------------------------------------------
good enough
69/69 [==============================] - 5s 56ms/step
perulangan ke- 2
              precision    recall  f1-score   support

           0       0.88      0.72      0.79       687
           1       0.67      0.87      0.76       230
           2       0.89      0.93      0.91      1283

    accuracy                           0.86      2200
   macro avg       0.82      0.84      0.82      2200
weighted avg       0.87      0.86      0.86      2200

---------------------------------------------------------------
good enough
69/69 [==============================] - 5s 57ms/step
perulangan ke- 3
              precision    recall  f1-score   support

           0       0.81      0.81      0.81       687
           1       0.86      0.69      0.77       230
           2       0.90      0.94      0.92      1283

    accuracy                           0.87      2200
   macro avg       0.86      0.81      0.83      2200
weighted avg       0.87      0.87      0.87      2200
```

NOTES :

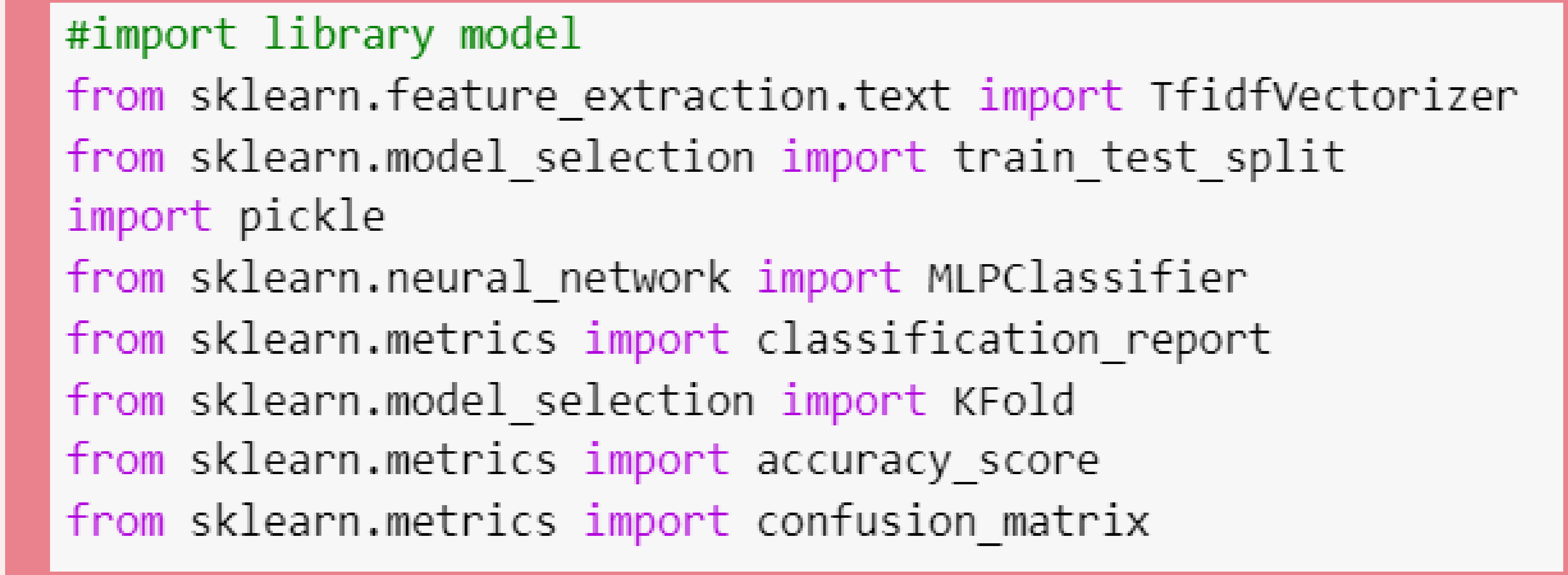
0 : negative class

1 : neutral class

2 : positive class

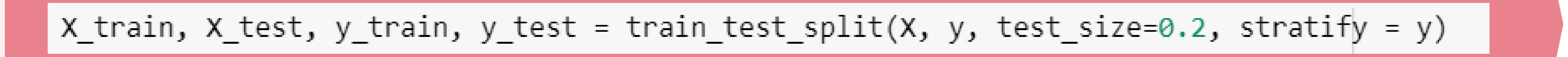
NEURAL NETWORK

BINAR ACADEMY DATA SCIENCE WAVE 1



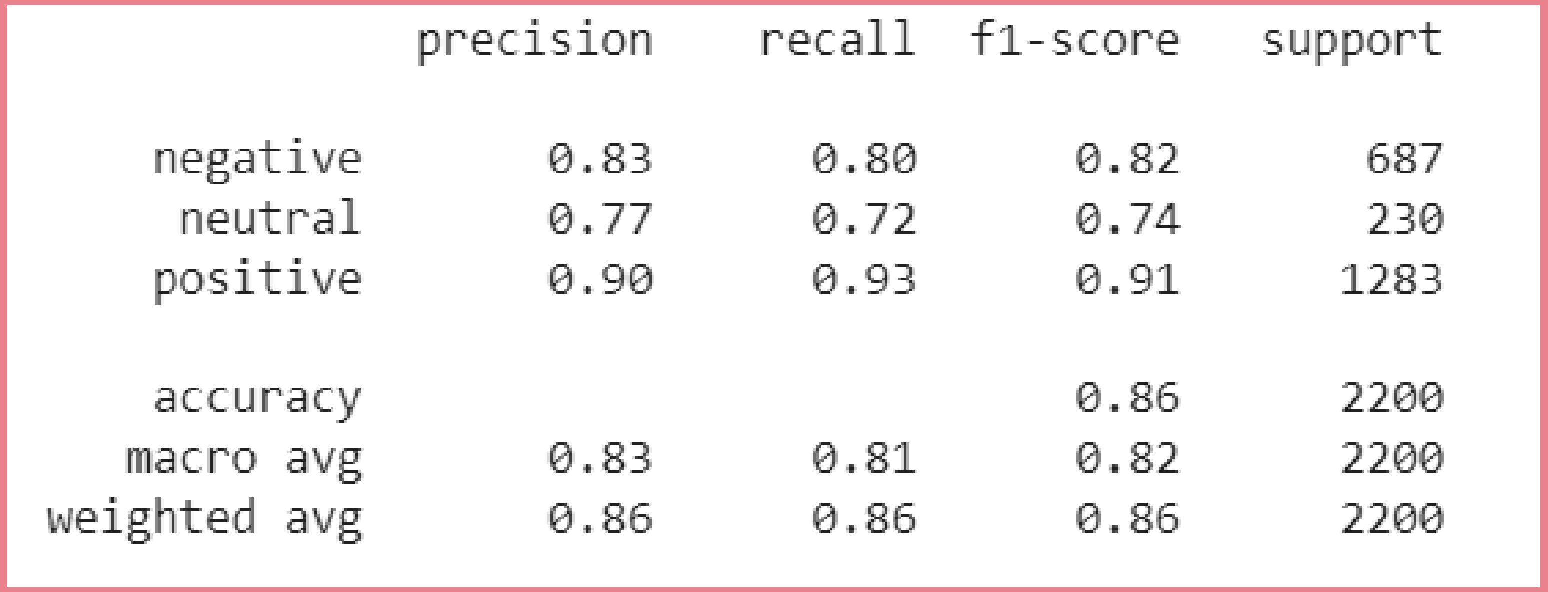
```
#import library model
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
import pickle
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import classification_report
from sklearn.model_selection import KFold
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

Libraries used for Neural Network model applications



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify = y)
```

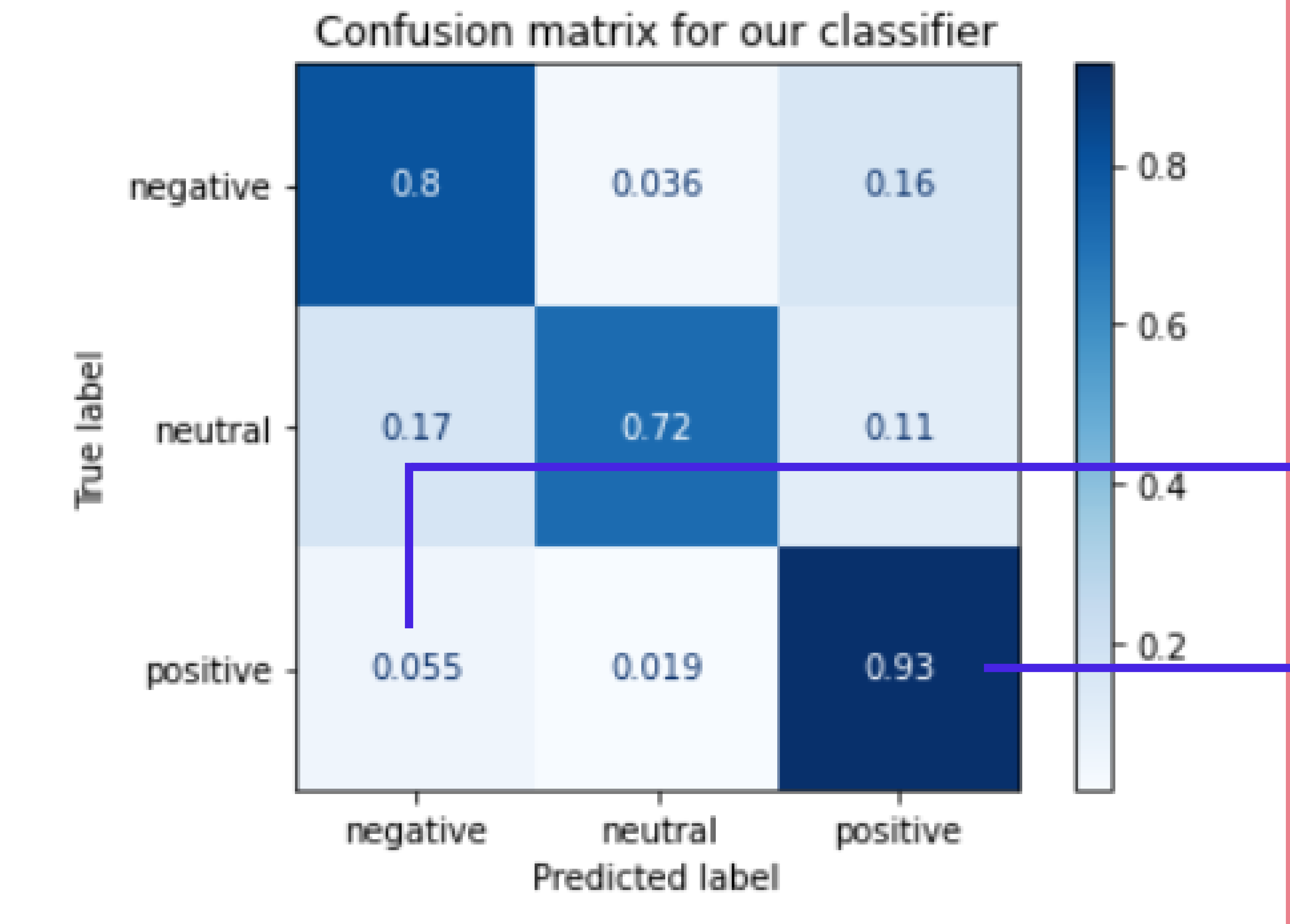
Splitting for training and test processes



```
              precision    recall  f1-score   support

    negative       0.83      0.80      0.82       687
     neutral       0.77      0.72      0.74       230
    positive       0.90      0.93      0.91      1283

    accuracy                           0.86      2200
   macro avg       0.83      0.81      0.82      2200
weighted avg       0.86      0.86      0.86      2200
```

Classification Report on Neural Network and test accuracy value is 80%

NEURAL NETWORK



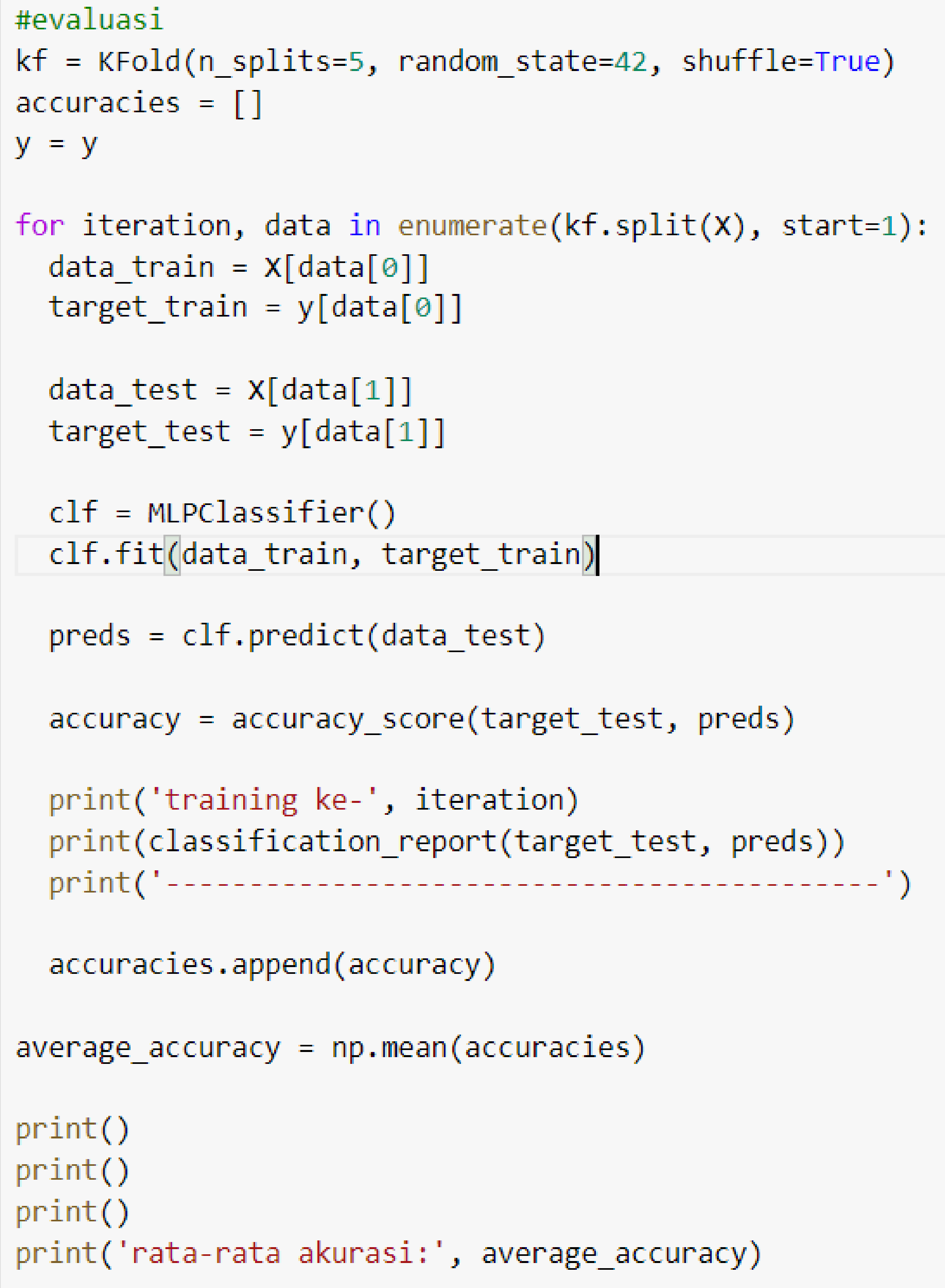
Confusion matrix for our classifier

True label \ Predicted label	negative	neutral	positive
negative	0.8	0.036	0.16
neutral	0.17	0.72	0.11
positive	0.055	0.019	0.93

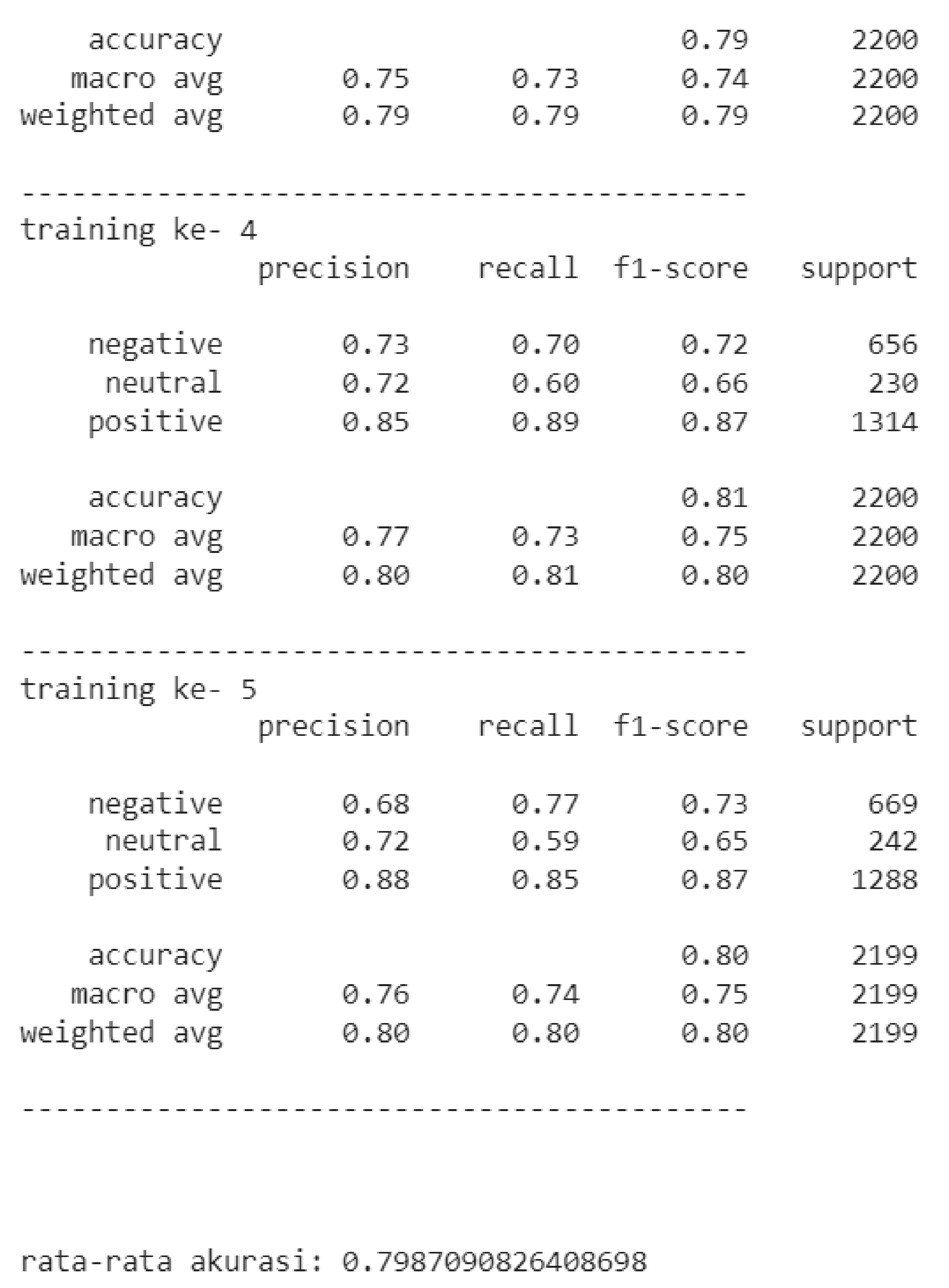
False Negative values are in the range of 0.055

While the True Positive Value is in the range of 0.93, where out of 10 predictions 9 are in accordance with the facts

Average Accuracy Evaluation

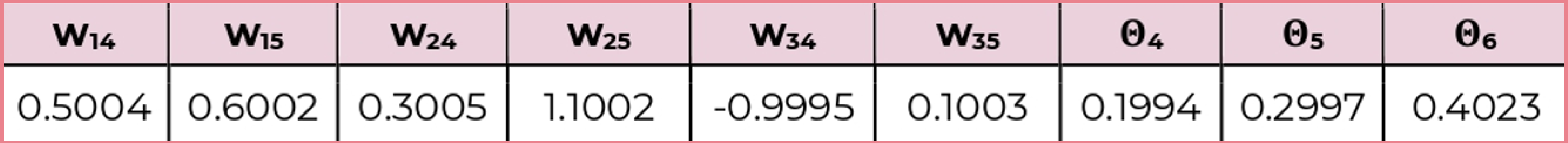


```
#evaluasi
kf = KFold(n_splits=5, random_state=42, shuffle=True)
accuracies = []
y = y

for iteration, data in enumerate(kf.split(X), start=1):
  data_train = X[data[0]]
  target_train = y[data[0]]

  data_test = X[data[1]]
  target_test = y[data[1]]

  clf = MLPClassifier()
  clf.fit(data_train, target_train)

  preds = clf.predict(data_test)

  accuracy = accuracy_score(target_test, preds)

  print('training ke-', iteration)
  print(classification_report(target_test, preds))
  print('-------------------------------------------')

  accuracies.append(accuracy)

average_accuracy = np.mean(accuracies)

print()
print()
print()
print('rata-rata akurasi:', average_accuracy)
```



```
    accuracy                           0.79      2200
   macro avg       0.75      0.73      0.74      2200
weighted avg       0.79      0.79      0.79      2200

-------------------------------------------
training ke- 4
              precision    recall  f1-score   support

    negative       0.73      0.70      0.72       656
     neutral       0.72      0.60      0.66       230
    positive       0.85      0.89      0.87      1314

    accuracy                           0.81      2200
   macro avg       0.77      0.73      0.75      2200
weighted avg       0.80      0.81      0.80      2200

-------------------------------------------
training ke- 5
              precision    recall  f1-score   support

    negative       0.68      0.77      0.73       669
     neutral       0.72      0.59      0.65       242
    positive       0.88      0.85      0.87      1288

    accuracy                           0.80      2199
   macro avg       0.76      0.74      0.75      2199
weighted avg       0.80      0.80      0.80      2199

-------------------------------------------



rata-rata akurasi: 0.7987090826408698
```

SENTIMENT ANALYSIS CALCULATION NEURAL NETWORK MODEL



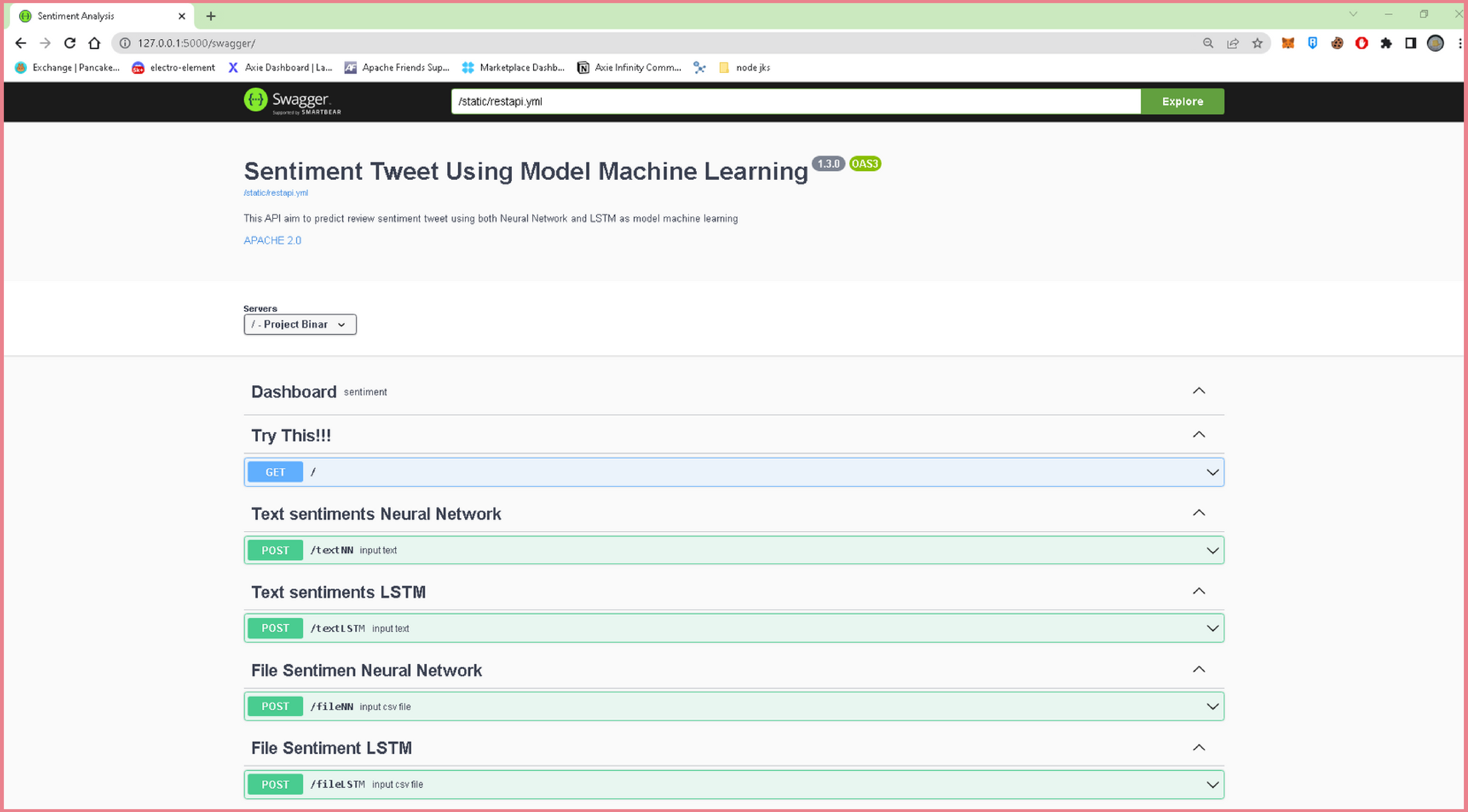
W_{14}	W_{15}	W_{24}	W_{25}	W_{34}	W_{35}	θ_4	θ_5	θ_6
0.5004	0.6002	0.3005	1.1002	-0.9995	0.1003	0.1994	0.2997	0.4023

Based on Calculations made from Initial Values and initial random from the model, in the table above is conclusion of the calculation of weight and θ (Theta).

API

BINAR ACADEMY DATA SCIENCE WAVE 1

API VISUALIZATION



Sentiment Tweet Using Model Machine Learning 1.3.0 OAS3

/static/restapi.yml

This API aim to predict review sentiment tweet using both Neural Network and LSTM as model machine learning

APACHE 2.0

Servers

/ - Project Binar

Dashboard sentiment

Try This!!!

GET /

Text sentiments Neural Network

POST /textNN input text

Text sentiments LSTM

POST /textLSTM input text

File Sentimen Neural Network

POST /fileNN input csv file

File Sentiment LSTM

POST /fileLSTM input csv file

CONCLUSION

The technology for analyzing sentiment is to use a Machine Learning Model with the following process :

1. Dataset utilization regarding sentiment
2. Preprocessing data before being processed by models
3. Domain of NLP using 2 types of supervised learning model (LSTM dan Neural Network)
4. Create an Algorithm to carry out the classification process on 3 sentiments (Positive, Neutral, Negative)
5. Use of the Flask API to interface and receive input text and files for sentiment prediction

RECOMMENDATION

- Data mining and scrapping might be necessary to collect more data to get excellent result
- Simple machine learning might be the answer to complicated problem
- Balance sample for every category in train and test also need to be notice
- for LSTM model, accuracy and loss should not be the only thing to consider if the model is good. Evaluation on F1, recall and precision in confusion matrix are important

ARTICLE REFERENCE

01 https://deepai.org/machine-learning-glossary-and-terms/weight-artificial-neural-network#:~:text=What%20is%20Weight%20(Artificial%20Neural,weight%2C%20and%20a%20bias%20value.

02 https://medium.com/techcrush/how-to-deploy-your-ml-model-in-jupyter-notebook-to-your-flask-app-d1c4933b29b5

03 https://datascience.stackexchange.com/questions/42599/what-is-the-relationship-between-the-accuracy-and-the-loss-in-deep-learning

04 https://towardsdatascience.com/random-initialization-for-neural-networks-a-thing-of-the-past-bfcdd806bf9e

05 https://machinelearningmastery.com/a-gentle-introduction-to-sigmoid-function/

06 https://github.com/arashy76/Multi-class-persian-text-classification-using-LSTM/blob/main/NLP981_Phase2.ipynb

RESEARCHS AND BOOKS REFERENCES

01 Muin, Fatkhul, 2019, PERUBAHAN PERILAKU REMAJA AKIBAT PENGGUNAAN
MEDIA SOSIAL ONLINE DI DESA KARANGMANGU, Surabaya, UINSA

02 Huyen, Chip, 2022, Designing MachineLearning Systems, Boston, O'Reilly

03 Geron, Aurelien, 2019, Hands-On Machine Learning with Scikit-Learn, Keras, and
TensorFlow, Boston, O'Reilly

04 Wiley, 2020, Storytelling with data, New Jersey. cole nussbaumer knaflic

05 Widodo, Ayuningtyas, Hermawan. 2022. NEXT WORD PREDICTION USING LSTM Jakarta. JOURNAL OF INFORMATION TECHNOLOGY AND ITS UTILIZATION, VOLUME 5, ISSUE 1, JUNE-2022.

06 Saputra. 2020. Classification Using Artifical Neural Network Method in Protecting Credit Fitness. Jakarta. Indonesian Journal of Artificial Intelligence and Data Mining

THANK YOU

GROUP 2

FAJAR | FIKI | REGINA