LAPORAN PROYEK

SENTIMENT ANALYSIS PADA

STREAMING DATA TWITTER RESESI EKONOMI GLOBAL

12S4058 - PENGOLAHAN DATA BESAR

SISTEM INFORMASI

FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO

INSTITUT TEKNOLOGI DEL

2023

DAFTAR ISI

DAFTAR ISI **2**

DAFTAR GAMBAR **4**

DAFTAR KODE PROGRAM **5**

BAB I PENDAHULUAN **6**

1.1 Latar Belakang 6

1.2 Rumusan Masalah 7

1.3 Tujuan 7

1.4 Scope 7

BAB II DESIGN 8

2.1 Arsitektur Sistem 8

2.1.1 Streaming data twitter 9

2.1.2 Data Storage 9

2.1.3 Streaming Data Processing 9

2.1.4 Evaluate Model 9

2.2 Machine Learning Pipeline 9

2.2.1 Twitter Data Stream 10

2.2.2 Data Understanding 10

2.2.3 Data Preparation 11

2.2.4 Sentiment Analysis using TextBlob 11

2.2.5 Data Visualization 11

2.2.6 Train Model 12

2.2.7 Evaluate Model 12

BAB III IMPLEMENTASI 13

3.1 Data Retrieval 13

3.2.1 Akses API Twitter 13

3.2.2 Crawling Data 14
3.2 Data Understanding 15
3.2.2 Data Describe 17
3.2.3 Data Validation 18
3.3 Data Preparation 19
3.3.1 Data Cleaning 19
3.4 Text Processing 19
3.4.1 Filtering 19
3.4.2 Cleaning Text 20
3.4.3 Remove Stopwords 20
3.4.4 Stemming **21**
BAB 4 HASIL DAN PEMBAHASAN 23
4.1 Preprocess Data 23
4.2 Data Labelling 24
4.3 Classification Using Pipeline in Apache Spark 25
4.4 Word Cloud Visualization 27
4.4.1 Positive Sentiment 28
4.4.2 Negative Sentiment 28
4.5 Evaluating the Model 29

DAFTAR GAMBAR

Gambar 2. 1 Arsitektur Sistem....8
Gambar 2. 2 Machine Learning Pipeline....10
Gambar 3. 1 Akses API Twitter....13
Gambar 3. 2 Import Library Tweepy....14
Gambar 3. 3 Tampilan data collection....17
Gambar 3. 4 Data deskriptif dengan metode statistik....18
Gambar 3. 5 Hasil Filtering....20
Gambar 3. 6 Hasil *Cleaning Text*....20
Gambar 3. 6 Hasil Stopword Removal....21
Gambar 3. 7 Hasil Stemming....22
Gambar 4. 1 Tampilan data setelah text preprocessing....23
Gambar 4. 2 Data labelling menggunakan libraryTextBlob....25
Gambar 4. 5 Tampilan data konversi ke pandas dataframe....27
Gambar 4. 3 Positive Sentiment Word Cloud....28
Gambar 4. 4 Negative Sentiment Word Cloud....29
Gambar 4. 6 Training areaUnderROC....30
Gambar 4. 7 Testing areaUnderROC....30
Gambar 4. 8 Metric Evaluation....30
Gambar 4. 9 Confusion Matrix....31

DAFTAR KODE PROGRAM

Kode Program 3. 1 Crawling Data 15
Kode program 3. 2 Pseudocode Filtering 19
Kode program 3. 3 Pseudocode Cleaning Text 20
Kode program 3. 3 Pseudocode Removing topwords 21
Kode program 3. 4 Stemming 22
Kode program 4. 1 Cuplikan code preprocess data 23

BAB I
PENDAHULUAN

Berikut ini akan dijelaskan latar belakang, tujuan, dan ruang lingkup proyek yang akan dikembangkan.

1.1 Latar Belakang

Dalam era digital, data yang dihasilkan setiap harinya semakin banyak, terutama dengan adanya media sosial seperti Twitter yang memungkinkan pengguna untuk dengan mudah dan cepat membagikan pemikiran dan pandangan mereka tentang suatu hal. Oleh karena itu, analisis sentimen pada data Twitter menjadi penting untuk mengidentifikasi perasaan, pandangan, atau pendapat seseorang terhadap suatu hal dengan menggunakan teks sebagai data input. Teknik analisis sentimen dapat diterapkan pada berbagai jenis data teks, termasuk tweet, ulasan hotel, ulasan pelanggan, dan lain-lain.

Dalam konteks saat ini, analisis sentimen pada data Twitter mengenai resesi global menjadi semakin penting karena dapat mempengaruhi berbagai aspek kehidupan, termasuk keuangan pribadi, bisnis, dan ekonomi secara keseluruhan. Melalui twitter maka akan diketahui bagaimana bahwasannya perasaan masyarakat maupun pandangannya terhadap resesi secara real-time. Resesi dapat mempengaruhi perekonomian dan kesejahteraan sosial. Dengan teknologi dan metode analisis sentimen, data tweet terkait resesi ekonomi dapat diambil dan dianalisis dengan cepat, sehingga para peneliti dan pengambil keputusan dapat membuat kebijakan yang tepat dan efektif. Oleh karena itu, proyek pengolahan data besar ini diharapkan dapat memberikan manfaat dalam mengolah data dari Twitter dan meningkatkan akurasi algoritma dalam melakukan proses analisis sentimen terkait resesi.

Proyek pengolahan data besar ini mencakup klasifikasi analisis sentimen pada streaming data twitter dengan menggunakan layanan API Stream yang disediakan oleh Twitter. Tahap preprocessing diperlukan sebelum analisis sentimen, seperti folding case, penghapusan simbol, tokenisasi, konversi slang word, dan penghapusan stopword. Penelitian ini diharapkan dapat memberikan manfaat dalam mengolah data besar dari Twitter dan meningkatkan akurasi algoritma dalam melakukan proses analisis sentimen.

1.2 Rumusan Masalah

Bagaimana melakukan analisis sentimen pada data live streaming Twitter dengan menggunakan keyword "resesi" untuk memahami pandangan dan opini pengguna Twitter terhadap fenomena resesi ekonomi global?

1.3 Tujuan

Tujuan dari proyek ini adalah untuk melakukan analisis sentimen pada dataset live streaming Twitter dengan keyword "resesi" agar dapat memahami pandangan dan opini pengguna Twitter terhadap fenomena resesi ekonomi global.

1.4 Scope

Berikut ini dijabarkan batasan dari pengerjaan proyek sentimen analisis pada dataset live streaming twitter sebagai berikut.

1. Penggunaan algoritma Logistic Regression dalam melakukan analisis sentimen pada dataset live streaming Twitter dengan keyword "resesi" dalam bahasa inggris.
2. Pengumpulan data dilakukan dengan memanfaatkan layanan API Stream yang disediakan oleh Twitter.
3. Data yang digunakan adalah data tweet yang mengandung keyword "resesi" dalam bahasa Inggris yang diposting dalam rentang waktu tertentu.
4. Analisis sentimen akan fokus pada polaritas sentimen positif dan negatif dalam konteks pandangan dan opini pengguna Twitter terhadap fenomena resesi ekonomi global.

BAB II
DESIGN

Bab ini memaparkan sebuah arsitektur sistem yang dapat digunakan sebagai *design* solusi dalam melakukan analisis sentimen Live Streaming Data Twitter menggunakan keyword "resesi" untuk mengetahui pandangan ataupun perasaan masyarakat melalui diskusi pada twitter terhadap resesi ekonomi global. Analisis sentimen dilakukan untuk mengekstrak sentimen dari teks dalam bahasa Inggris pada tweet yang terkait dengan fenomena resesi dari data streaming Twitter. Hasil analisis sentimen akan memberikan informasi tentang apakah tweet tersebut memiliki sentimen positif dan negatif. Metode yang digunakan dalam analisis sentimen adalah VanderSentiment. Arsitektur Big Data yang digunakan dalam penelitian ini mencakup struktur keseluruhan dari sistem *logical* dan *physical* yang digunakan untuk menyimpan, mengakses, mengakses dan memproses big data.

2.1 Arsitektur Sistem

Berikut ini merupakan arsitektur sistem pada analisis sentimen pada streaming data twitter untuk resesi.

Gambar 2. 1 Arsitektur Sistem

Pada gambar di atas dijelaskan bagaimana alur arsitektur sistem yang akan digunakan dalam melakukan analisis sentimen pada Live Streaming Twitter dengan menggunakan kata kunci "resesi". Arsitektur sistem dibentuk agar dapat mengolah data yang terstruktur dan tidak terstruktur. Berikut adalah tahapan yang dilakukan pada arsitektur sistem analisis sistem.

2.1.1 Streaming data twitter

Tahapan yang pertama dilakukan adalah Streaming data twitter. Pada tahap ini, data streaming Twitter diambil menggunakan Tweepy dan difilter menggunakan fungsi Tweepy untuk mengambil data yang sesuai dengan kata kunci “resesi”. Data yang telah diambil kemudian akan disimpan dalam format yang dapat diolah dan diproses.

2.1.2 Data Storage

Data streaming yang diterima dapat disimpan dalam sistem penyimpanan data yang sesuai, seperti dalam format CSV (Comma-Separated Values) merupakan salah satu pilihan yang umum digunakan.

2.1.3 Streaming Data Processing

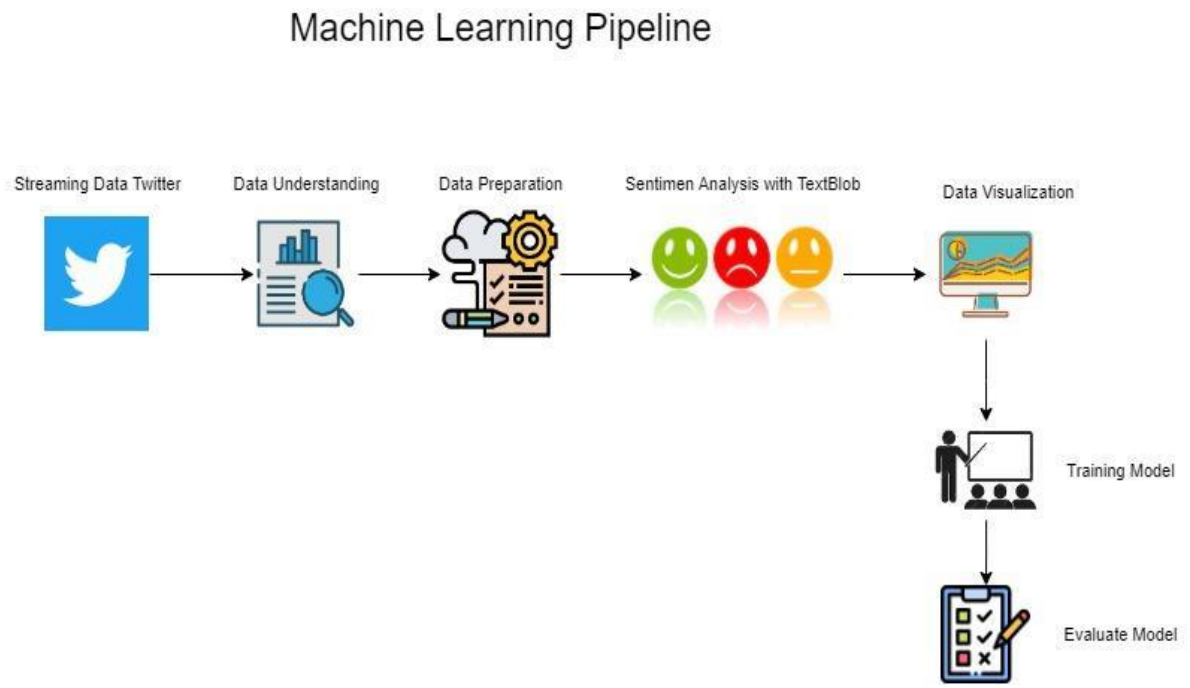
Setelah menyimpan data streaming dari Twitter dalam format CSV, langkah selanjutnya adalah melakukan streaming data processing menggunakan structured data streaming. Dalam structured data streaming Twitter, data streaming Twitter yang disimpan. Output dari pemrosesan data streaming dapat disimpan dalam format yang sesuai dan mencakup wawasan yang mendalam tentang data streaming Twitter. Structured data streaming Twitter memfasilitasi pemrosesan yang terstruktur dan efisien, mendukung analisis mendalam, dan pengambilan keputusan yang lebih baik dalam konteks Twitter.

2.1.4 Evaluate Model

Setelah memproses data streaming dengan struktur data streaming, langkah selanjutnya adalah mengevaluasi model yang digunakan. Evaluasi model bertujuan untuk mengukur sejauh mana kinerja model dalam menghasilkan prediksi akurat. Hasil evaluasi meliputi metrik-metrik seperti akurasi, presisi, recall, F1-score. Selain itu, klasifikasi report memberikan informasi rinci tentang performa model pada setiap kelas. Evaluasi model ini penting untuk memahami efektivitas model dalam pemrosesan data streaming dan mendukung pengambilan keputusan.

2.2 Machine Learning Pipeline

Machine learning pipeline merupakan rangkaian proses yang terintegrasi yang dirancang untuk mengolah dan mempersiapkan data sebelum digunakan dalam pelatihan model machine learning. Tahapan yang ada dalam Machine Learning Pipeline dapat dilihat pada gambar berikut ini.



Gambar 2. 2 Machine Learning Pipeline

2.2.1 Twitter Data Stream

Pada tahap ini adalah mengumpulkan data dari Twitter secara real-time menggunakan library Tweepy dan memfilternya untuk mendapatkan data yang relevan dengan topik yang ingin dianalisis. Pertama, diperlukan akses ke Twitter API untuk dapat mengumpulkan data dari Twitter secara real-time. Untuk mengakses Twitter API, perlu mendaftarkan aplikasi di Twitter Developer Platform dan memperoleh credential untuk mengakses API tersebut. Setelah mendapatkan credential untuk mengakses Twitter API, langkah selanjutnya adalah menggunakan library Tweepy untuk mengakses API tersebut. Setelah meng-import Tweepy, langkah selanjutnya adalah membuat sebuah streaming object dengan menggunakan class Stream dan API key yang telah didapatkan sebelumnya. Streaming object ini akan digunakan untuk mengakses data dari Twitter secara real-time. Dalam pembuatan streaming object, dapat menambahkan filter untuk memperoleh data yang relevan dengan topik yang ingin dianalisis.

2.2.2 Data Understanding

Tahapan selanjutnya dilakukan untuk pemrosesan data. Pertama, data dikumpulkan dan ditampilkan. Kemudian, dilakukan deskripsi statistik data dengan menampilkan ringkasan statistik. Informasi mengenai kolom-kolom yang ada dalam dataset juga diambil. Selanjutnya, jumlah baris dan kolom dalam dataset dicetak. Skema dan tipe data dari kolom-kolom dalam dataset juga dicetak. Selanjutnya, validasi data dilakukan dengan

memeriksa nilai null dalam setiap kolom dataset, dan juga dengan menghitung jumlah nilai null pada kolom "Date" atau "Tweet".

2.2.3 Data Preparation

Setelah pemrosesan data Twitter, langkah selanjutnya adalah persiapan data dengan Data Cleaning, Text Processing, dan Labeling Data. Data Cleaning melibatkan penghapusan data yang tidak relevan seperti tautan, tanda baca, kata-kata tidak penting, dan penggabungan kata menjadi bentuk dasarnya. Tujuannya adalah membersihkan data dari elemen yang tidak diperlukan. Text Processing dilakukan untuk mempersiapkan data sebelum analisis lebih lanjut. Ini melibatkan penghapusan stopwords, tanda baca, tautan, serta penggabungan kata-kata. Tujuannya adalah memudahkan pemahaman dan analisis. Labeling Data melibatkan penambahan label pada data untuk mengklasifikasikan sentimen menjadi positif atau negatif. Dalam hal ini, algoritma TextBlob digunakan untuk menganalisis sentimen berdasarkan teks. Dengan melakukan tahapan-tahapan di atas, data Twitter telah siap untuk analisis lebih lanjut, terutama dalam analisis sentimen. Tahapan ini penting untuk memastikan data yang digunakan memiliki kualitas baik dan siap untuk pemodelan dan visualisasi.

2.2.4 Sentiment Analysis using TextBlob

Sentiment Analysis menggunakan TextBlob adalah salah satu metode yang dapat digunakan untuk menganalisis sentimen pada data teks, termasuk dalam proses analisis sentimen pada data streaming Twitter terkait dengan topik resesi.

TextBlob merupakan sebuah library Python yang digunakan untuk memproses data teks, termasuk di dalamnya sentiment analysis. Library ini menyediakan metode-metode untuk melakukan ekstraksi fitur dari teks, seperti ekstraksi kata-kata kunci, pemberian nilai pada kata-kata kunci tersebut, dan kemudian pengelompokkan teks berdasarkan nilai-nilai tersebut.

2.2.5 Data Visualization

Setelah dilakukan analisis sentimen, hasilnya akan divisualisasikan untuk memberikan pemahaman yang lebih baik tentang pandangan dan opini pengguna Twitter terhadap fenomena resesi. Pada tahap ini, data yang telah diproses akan divisualisasikan dalam bentuk grafik atau diagram untuk mempermudah dalam membaca dan memahami data.

2.2.6 Train Model

Setelah melakukan visualisasi data, langkah berikutnya adalah melatih model menggunakan PySpark dan menggunakan pipeline. Dalam pipeline, terdapat beberapa tahap seperti tokenisasi, penghitungan frekuensi kata, perhitungan bobot IDF, dan pelatihan model regresi logistik. Model tersebut akan dilatih menggunakan data train dan kemudian dapat digunakan untuk melakukan prediksi pada data baru yang belum pernah dilihat sebelumnya.

2.2.7 Evaluate Model

Setelah melakukan pelatihan pada model, langkah berikutnya adalah mengevaluasi model untuk memahami sejauh mana kemampuan model dalam menjalankan tugas yang diberikan. Untuk melakukan evaluasi ini, BinaryClassificationEvaluator digunakan untuk mengukur performa model dalam melakukan klasifikasi biner dengan menggunakan metrik area di bawah kurva ROC. Hasil evaluasi ini kemudian dicetak untuk memberikan informasi mengenai kualitas model tersebut. Evaluasi model bertujuan untuk mengukur performa model dan memberikan pemahaman mengenai keakuratan dan keandalan model dalam melakukan klasifikasi.

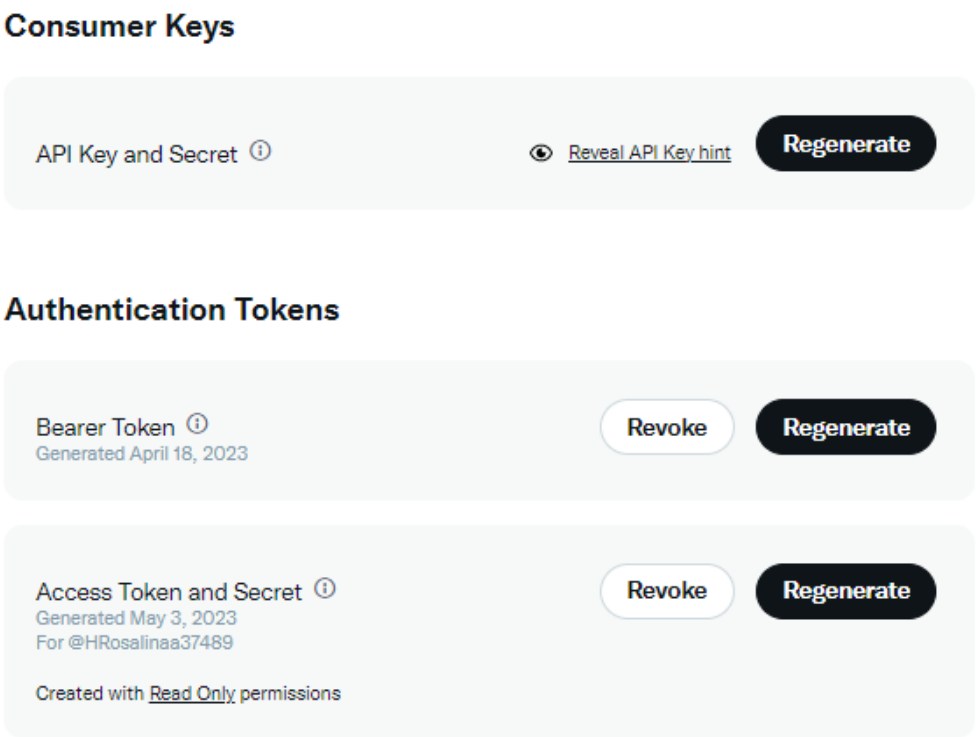
BAB III
IMPLEMENTASI

3.1 Data Retrieval

Twitter menghasilkan aliran data kontinu yang mana data tersebut dihasilkan oleh aktivitas pengguna di media sosial tersebut. Data ini mencakup berbagai jenis informasi seperti tweet, retweet, mention, like, dan juga metadata seperti lokasi, tanggal, dan waktu. Twitter menyediakan API (Application Programming Interface) yang memungkinkan pengembang untuk mengakses dan mengumpulkan data ini untuk berbagai tujuan analisis dan pengolahan. Menggunakan stream data Twitter, pengguna dapat memantau dan menganalisis topik, tren, dan sentimen yang sedang populer dalam waktu nyata. Berikut ini beberapa langkah dalam mengolah stream data Twitter.

3.2.1 Akses API Twitter

Untuk mengakses data streaming Twitter, pengguna perlu mendaftar dan membuat aplikasi di situs pengembang Twitter. Setelah aplikasi dibuat, pengguna akan menerima kredensial API seperti API key, API secret key, access token, dan access token secret.



Gambar 3. 1 Akses API Twitter

Selanjutnya token dan key akan dikonfigurasikan sesuai dengan autentikasi koneksi API yang digunakan pada baris kode python dengan melakukan import library tweepy terlebih dahulu seperti berikut.

```
import tweepy
from tweepy import OAuthHandler
import pandas as pd
import re
```

```
#Twitter API credentials
consumer_key = '5OBZDcObonT3Pdf9du705G5KP'
consumer_secret = '24qgOlLKmywVLeID6FV0IHNscyPt1mTcWeXA5M8tDDV6bngvJt'
access_key = '1266772148875489282-AhXazdXEk0gMfT5V9tRtGSy6pLGIP5'
access_secret = 'IziL7kVEUNF8KwgQVVAMWxhPLA4xNaZN1lbKDkVG3PE7W'
```

```
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_key, access_secret)
api = tweepy.API(auth,wait_on_rate_limit=True)
```

Gambar 3. 2 Import Library Tweepy

3.2.2 Crawling Data

Setelah melakukan konfigurasi API, selanjutnya akan dilakukan penarikan data dengan kata kunci “recession”. Pada saat penarikan tweets, data yang diambil maksimal sebanyak 100 data, karena sesuai dengan ketentuan penggunaan pengguna yang berlaku. Kemudian data akan dibagi menjadi 2 kolom, yakni “Date” dan “Tweet” . Data resesi yang diambil dispesifikasikan dengan mengambil kata kunci yang memiliki bahasa inggris, disamping itu kata yang mengandung tautan atau *link* serta memiliki kata yang sama persis akan di filter, hal ini bertujuan untuk menghindari redudansi data. Penerapan ini dilandasi oleh adanya penggunaan data yang berulang akibat adanya penggunaan BOT sehingga data yang digunakan cenderung menghasilkan data yang duplikat sehingga dianggap tidak sesuai dengan fokus tujuan dari penelitian ini, yakni sentimen analisis.

```
def crawltweets(search_tweet, n_tweets=100):
    data_tweets = pd.DataFrame(columns=['Date', 'Tweet'])
    tweets = tweepy.Cursor(api.search_tweets, q=search_tweet,
lang="en", tweet_mode='extended').items(n_tweets)
    url_pattern = re.compile(r'http\S+|www\S+')

    for tweet in tweets:
```

```
        Date = tweet.created_at
        try:
            Tweet = tweet.retweeted_status.full_text
        except AttributeError:
            Tweet = tweet.full_text
        if not bool(url_pattern.search(Tweet)):
            ith_tweet = [Date, Tweet]
            data_tweets.loc[len(data_tweets)] = ith_tweet
    data_tweets = data_tweets.drop_duplicates(subset='Tweet',
keep='first')
    for index, tweet in enumerate(data_tweets['Tweet']):
      if index == 20:
          break
      print(tweet)
    print('Crawling is Done =', len(data_tweets))
    namafile = 'recession.csv'
    data_tweets.to_csv(namafile, index=False)


search_tweet = "recession"
crawltweets(search_tweet)
```

Kode Program 3. 1 Crawling Data

Setelah data duplikat di filter, maka tahapan berikutnya adalah menampilkan 20 sampel data yang diambil pada hasil keluaran yang bertujuan untuk menampilkan bagaimana contoh data tweets yang telah diambil. Selanjutnya apabila data telah mencapai jumlah penarikan data yang telah ditetapkan, maka akan menampilkan *output* Crawling is Done dengan menampilkan jumlah spesifik data yang diambil. Pada bagian terakhir, data akan disimpan dalam format csv dengan nama recession.csv.

3.2 Data Understanding

Dalam proyek ini, tahapan Data Understanding menjadi langkah awal yang krusial untuk mendapatkan pemahaman yang komprehensif mengenai data yang akan digunakan dalam analisis. Tahap ini bertujuan untuk memperoleh informasi yang lebih dalam mengenai dataset yang relevan dan mempersiapkannya sebelum melanjutkan ke tahap analisis yang lebih mendalam.

3.2.1 Data Collection

Pada tahap ini, fokus utama adalah mengumpulkan data dari berbagai sumber yang terkait dengan proyek ini. Sumber data adalah API. Proses pengumpulan data ini dilakukan dengan hati-hati dan mengacu pada kriteria yang telah ditentukan sebelumnya.

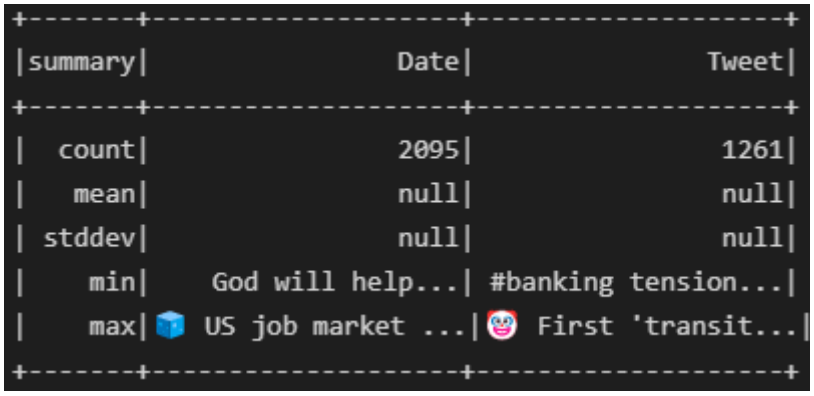
Berikut ini adalah tampilan data yang telah dikumpulkan.

Date	Tweet
2023-05-03 06:25:...	@PRSundar64 US is...
2023-05-03 06:25:...	"@saxena_puru But...
2023-05-03 06:25:...	@_swimfish @stats...
IndiaIN has rep...	null
2023-05-03 06:24:...	@TaxpayersParty @...
2023-05-03 06:24:...	After losing at H...
Recession probabi...	2023:
India- 0%	null
Pakistan- 40%	null
Bangladesg- 33%	null
Big blow to Modi'...	null
2023-05-03 06:24:...	@TheKouk You have...
2023-05-03 06:24:...	Republicans' posi...
- They ARE willin...	kills millions o...
- They are NOT wi...	null
2023-05-03 06:23:...	The Treasury Dept...
MAGA Republicans ...	null
Their demands?	null
Either let them d...	000 jobs & ca...
2023-05-03 06:23:...	@CWeston_Indo Dri...

Gambar 3. 3 Tampilan data collection

3.2.2 Data Describe

Data Describe adalah tahap dalam analisis data yang bertujuan untuk memberikan pemahaman yang lebih dalam tentang dataset yang telah dikumpulkan. Pada tahap ini, data dianalisis secara deskriptif menggunakan metode statistik dan teknik eksplorasi data untuk menggambarkan karakteristik data tersebut. Berikut ini adalah tampilan data yang telah dianalisis secara deskriptif menggunakan metode statistik.



summary	Date	Tweet
count	2095	1261
mean	null	null
stddev	null	null
min	God will help...	#banking tension...
max	🧊 US job market ...	🤡 First 'transit...

Gambar 3. 4 Data deskriptif dengan metode statistik

Berdasarkan Gambar 3.4, tahapan data describe melibatkan beberapa statistik ringkasan, antara lain:

- Jumlah Data (Count): Statistik "count" menunjukkan jumlah entri yang tersedia untuk setiap kolom dalam dataset. Dalam contoh yang diberikan, terdapat 2095 entri untuk kolom "Date" dan 1261 entri untuk kolom "Tweet". Informasi ini memberikan gambaran tentang volume data yang telah dikumpulkan.
- Rata-rata (Mean): Pada output yang diberikan, nilai "null" ditampilkan pada kolom "Date" dan "Tweet". Hal ini menunjukkan bahwa rata-rata tidak dapat dihitung untuk kedua kolom tersebut karena jenis data yang ada di dalamnya.
- Simpangan Standar (Standard Deviation): Seperti pada nilai rata-rata, nilai "null" ditampilkan pada kolom "Date" dan "Tweet". Hal ini mengindikasikan bahwa simpangan standar tidak dapat dihitung untuk kedua kolom tersebut karena jenis data yang ada di dalamnya.
- Nilai Terendah (Minimum): Pada kolom "Date", nilai terendah yang ditemukan adalah "God will help...". Sedangkan pada kolom "Tweet", nilai terendah adalah "#banking tension...". Ini memberikan contoh teks dengan nilai terendah dalam dataset yang telah dikumpulkan.
- Nilai Tertinggi (Maximum): Pada kolom "Date", nilai tertinggi yang ditemukan adalah "🧊 US job market...". Sedangkan pada kolom "Tweet", nilai tertinggi adalah "🤡 First 'transit...". Ini memberikan contoh teks dengan nilai tertinggi dalam dataset yang telah dikumpulkan

3.2.3 Data Validation

Data validation adalah proses memastikan bahwa data yang ada dalam dataset memenuhi kriteria atau persyaratan yang telah ditetapkan sebelumnya. Data validation adalah proses memeriksa dan memastikan bahwa data dalam dataset memenuhi kriteria atau persyaratan yang telah ditetapkan. Dalam dataset yang diberikan, terdapat 2095 entri non-null pada kolom "Date" dan 1261 entri non-null pada kolom "Tweet". Selain itu, terdapat 834 nilai null secara keseluruhan dalam dataframe tersebut.

3.3 Data Preparation

Data preparation bertujuan untuk mengubah, membersihkan, mengintegrasikan, dan mengorganisir data sehingga dapat digunakan secara efektif dalam analisis.Data preparation melibatkan serangkaian langkah yang meliputi:

3.3.1 Data Cleaning

Tahap ini melibatkan identifikasi dan penanganan nilai yang hilang (missing values) dalam dataset, penanganan nilai yang tidak valid atau tidak masuk akal, dan penanganan duplikasi data.

3.4 Text Processing

Text processing, atau pengolahan teks, adalah proses manipulasi dan analisis teks dalam bentuk data. Tujuan dari text processing adalah untuk mengolah dan memahami teks secara komputasional, sehingga memungkinkan mesin untuk memahami, menganalisis, dan mengekstrak informasi dari teks secara otomatis.

Text processing melibatkan serangkaian langkah dan teknik yang digunakan untuk memanipulasi dan menganalisis teks. Beberapa langkah umum dalam text processing meliputi:

3.4.1 Filtering

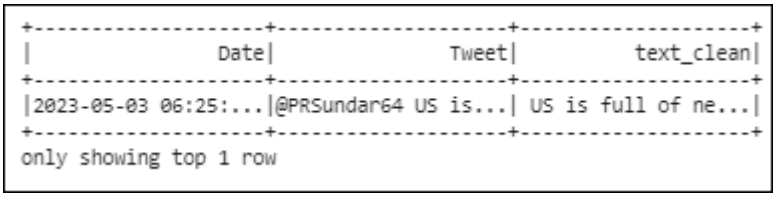
Langkah ini melibatkan penggunaan ekspresi reguler untuk menghapus elemen-elemen yang tidak diinginkan dari teks. Beberapa elemen yang dihapus termasuk link web, @username, #tagger, tanda baca, spasi berlebih, karakter \n, karakter khusus, dan emoji. Tujuannya adalah untuk membersihkan teks dan membuang elemen yang tidak relevan

Berikut ini akan cuplikan kode yang digunakan untuk proses filtering :

```
from pyspark.sql.functions import udf
from pyspark.sql.types import StringType

# Membuat UDF dari filtering
filtering_udf = udf(filtering, StringType())

# Mengaplikasikan UDF pada kolom "Tweet" untuk membuat kolom baru
"text_clean"
df = df.withColumn("text_clean", filtering_udf(df["Tweet"]))

# Menampilkan semua kolom pada dataframe
spark.conf.set("spark.sql.repl.eagerEval.enabled", True)
df.show(1)
```

Kode program 3. 2 Pseudocode *Filtering*

Berikut adalah hasil dari proses hasil *Filtering data* sebagai berikut.



```
+--------------------+--------------------+--------------------+
|                Date|               Tweet|          text_clean|
+--------------------+--------------------+--------------------+
|2023-05-03 06:25:...|@PRSundar64 US is...| US is full of ne...|
+--------------------+--------------------+--------------------+
only showing top 1 row
```

Gambar 3. 5 Hasil Filtering

3.4.2 Cleaning Text

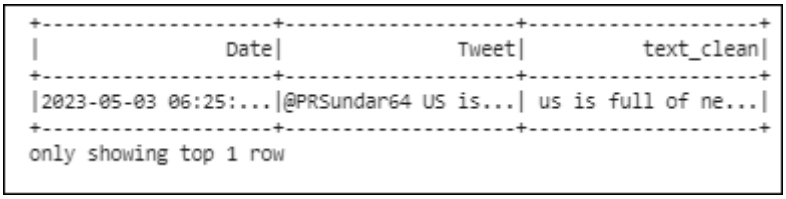
Fungsi clean_text melakukan beberapa tahap pembersihan pada teks. Tahapan-tahapan tersebut meliputi mengubah teks menjadi huruf kecil, menghapus teks di dalam tanda kurung siku, menghapus link web, menghapus tanda baca, menghapus kata yang mengandung angka, menghapus angka, menghapus karakter @ yang diikuti oleh non-alphanumeric, menghapus karakter # yang diikuti oleh non-alphanumeric, menghapus karakter tanda baca tertentu, menghapus spasi berlebih, mengganti karakter \n dengan spasi, mengganti karakter \ ■ dengan spasi, dan menghapus karakter & yang diikuti oleh non-space.

Berikut ini akan cuplikan kode yang digunakan untuk proses *Cleaning text* :

```python
from pyspark.sql.functions import udf
from pyspark.sql.types import StringType

# Membuat UDF dari clean_text
clean_text_udf = udf(clean_text, StringType())

# Mengaplikasikan UDF pada kolom "Tweet" untuk membuat kolom baru "text_clean"
df = df.withColumn("text_clean", clean_text_udf(df["Tweet"]))
df.show(1)
```

Kode program 3. 3 Pseudocode *Cleaning Text*

Berikut adalah hasil dari proses hasil *Cleaning Text* sebagai berikut.



```
+--------------------+--------------------+--------------------+
|                Date|               Tweet|          text_clean|
+--------------------+--------------------+--------------------+
|2023-05-03 06:25:...|@PRSundar64 US is...| us is full of ne...|
+--------------------+--------------------+--------------------+
only showing top 1 row
```

Gambar 3. 6 Hasil *Cleaning Text*

3.4.3 Remove Stopwords

Langkah ini melibatkan penghapusan kata-kata stopwords (kata-kata umum yang tidak memberikan makna penting) dari teks. Stopwords yang digunakan adalah stopwords dalam bahasa Inggris, ditambah dengan beberapa stopwords tambahan.

Berikut ini akan cuplikan kode yang digunakan untuk menghapus stopword data:

```
from nltk.corpus import stopwords
stop_words = stopwords.words('english')
more_stopwords = ['u', 'im', 'c']
stop_words = stop_words + more_stopwords

def remove_stopwords(text):
    text = ' '.join(word for word in text.split(' ') if word not in
stop_words)
    return text

# Membuat UDF dari remove_stopwords
remove_stopwords_udf = udf(remove_stopwords, StringType())

# Mengaplikasikan UDF pada kolom "Tweet" untuk membuat kolom baru
"text_clean"
df = df.withColumn("text_clean",
remove_stopwords_udf(df["text_clean"]))
```

Kode program 3. 4 Pseudocode Removing topwords

Berikut adalah hasil dari proses hasil *stopword removal* sebagai berikut.

```
+-------------------+--------------------+--------------------+
|               Date|               Tweet|          text_clean|
+-------------------+--------------------+--------------------+
|2023-05-03 06:25:...|@PRSundar64 US is...| us full news deb...|
|2023-05-03 06:25:...|"@saxena_puru But...|saxenapuru know c...|
|2023-05-03 06:25:...|@_swimfish @stats...|swimfish statsfee...|
|2023-05-03 06:24:...|@TaxpayersParty @...|taxpayersparty av...|
|2023-05-03 06:24:...|After losing at H...|losing happiness ...|
+-------------------+--------------------+--------------------+
only showing top 5 rows
```

Gambar 3. 7 Hasil Stopword Removal

3.4.4 Stemming

Proses stemming menggunakan algoritma SnowballStemmer dari library nltk. Langkah ini mengubah kata-kata dalam teks menjadi bentuk dasarnya (stem) untuk mengurangi variasi kata yang sama.

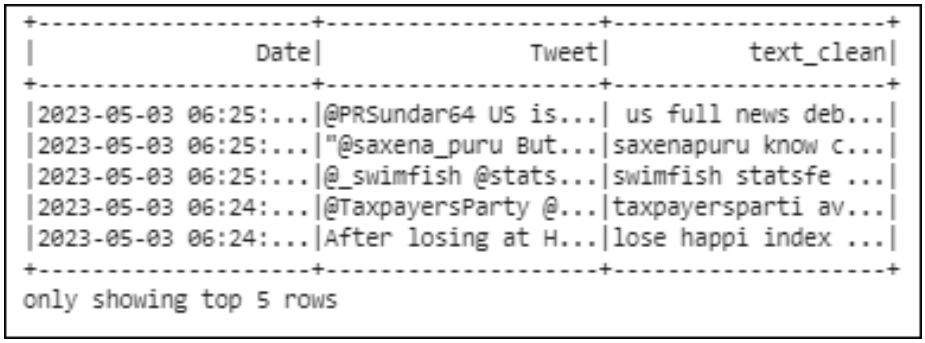
Berikut ini akan cuplikan kode proses stemming:

```
stemmer = nltk.SnowballStemmer("english")

def stemm_text(text):
    text = ' '.join(stemmer.stem(word) for word in text.split(' '))
    return text

# Membuat UDF dari stemm_text
```

```
stemm_text_udf = udf(stemm_text, StringType())

# Mengaplikasikan UDF pada kolom "Tweet" untuk membuat kolom baru
"text_clean"
df = df.withColumn("text_clean", stemm_text_udf(df["text_clean"]))
df.show(5)
```

Kode program 3. 5 Stemming

Berikut adalah hasil dari proses hasil *Stemming* sebagai berikut.



```
+--------------------+--------------------+--------------------+
|                Date|               Tweet|          text_clean|
+--------------------+--------------------+--------------------+
|2023-05-03 06:25:...|@PRSundar64 US is...| us full news deb...|
|2023-05-03 06:25:...|"@saxena_puru But...|saxenapuru know c...|
|2023-05-03 06:25:...|@_swimfish @stats...|swimfish statsfe ...|
|2023-05-03 06:24:...|@TaxpayersParty @...|taxpayersparti av...|
|2023-05-03 06:24:...|After losing at H...|lose happi index ...|
+--------------------+--------------------+--------------------+
only showing top 5 rows
```

Gambar 3. 8 Hasil *Stemming*

BAB 4
HASIL DAN PEMBAHASAN

4.1 Preprocess Data

Fungsi preprocess_data menggabungkan beberapa tahapan sebelumnya, yaitu clean_text, remove_stopwords, dan stemming. Tujuannya adalah untuk membersihkan, menghapus stopwords, dan melakukan stemming pada teks.

Setelah setiap langkah, kolom "text_clean" diperbarui dengan teks yang telah diproses menggunakan UDF yang sesuai. Secara keseluruhan, langkah-langkah ini bertujuan untuk membersihkan, menghapus elemen yang tidak relevan, mengurangi variasi kata, dan mempersiapkan teks sehingga dapat digunakan dalam analisis atau pemodelan lebih lanjut.

Berikut ini akan cuplikan kode yang digunakan untuk preprocess data:

```
# Membuat UDF dari preprocess_data
preprocess_data_udf = udf(preprocess_data, StringType())

# Mengaplikasikan UDF pada kolom "Tweet" untuk membuat kolom baru
"text_clean"
df = df.withColumn("text_clean",
preprocess_data_udf(df["text_clean"]))
df.show(5)
```

Kode program 4. 1 Cuplikan code preprocess data

Berikut merupakan tampilan data setelah dilakukan text processing.

Date	Tweet	text_clean
2023-05-03 06:25:...	@PRSundar64 US is...	us full news deb...
2023-05-03 06:25:...	"@saxena_puru But...	saxenapuru know c...
2023-05-03 06:25:...	@_swimfish @stats...	swimfish statsf s...
2023-05-03 06:24:...	@TaxpayersParty @...	taxpayersparti av...
2023-05-03 06:24:...	After losing at H...	lose happi index ...

Gambar 4. 1 Tampilan data setelah text preprocessing

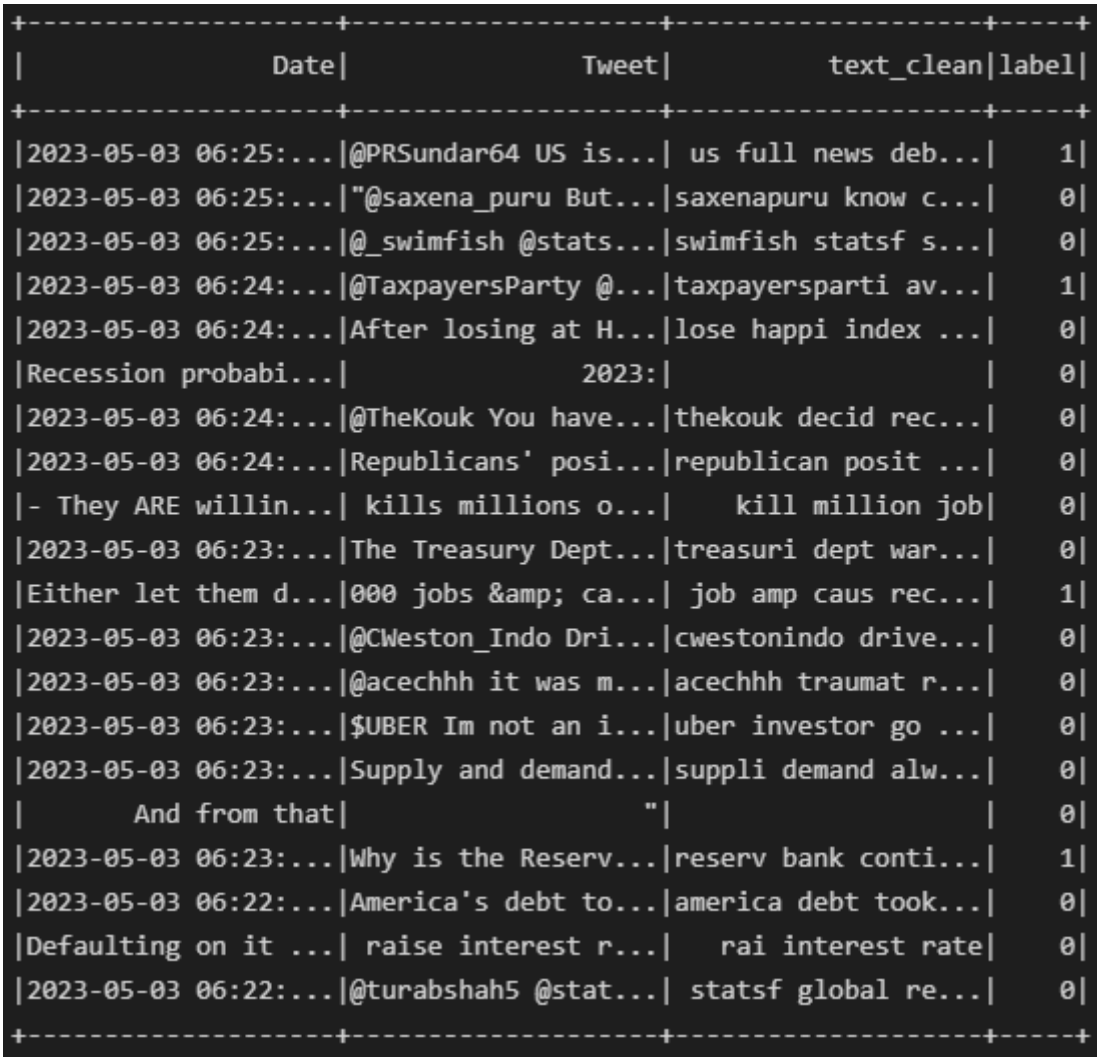
4.2 Data Labelling

Data labelling, juga dikenal sebagai anotasi data, merupakan proses memberikan label atau kategori kepada setiap dataset. Tujuan utama dari data labelling adalah untuk memberikan informasi yang spesifik dan bermakna pada setiap contoh data, sehingga dapat digunakan dalam melatih dan menguji model pembelajaran mesin. Dalam konteks proyek yang sedang dibahas, contoh data labelling yang digunakan adalah anotasi sentimen.

Dalam kode program yang digunakan, anotasi sentimen dilakukan dengan memanfaatkan library TextBlob. TextBlob merupakan sebuah library pemrosesan bahasa alami (NLP) yang menyediakan berbagai fungsi dan metode untuk melakukan analisis sentimen. Sentimen pada teks dinyatakan dalam nilai polaritas yang berada dalam rentang -1.0 hingga 1.0. Nilai 0 mengindikasikan sentimen positif, nilai 1 menunjukkan sentimen negatif. Pada tahap selanjutnya, nilai polaritas sentimen yang dihasilkan oleh TextBlob diubah menjadi label positif atau negatif. Jika nilai polaritas lebih besar dari 0, maka label yang diberikan adalah 1, yang menunjukkan sentimen positif. Jika nilai polaritas kurang dari atau sama dengan 0, maka label yang diberikan adalah 0, yang menunjukkan sentimen negatif.

Dengan menggunakan proses data labelling ini, tujuannya adalah untuk memberikan pemahaman yang jelas dan terstruktur tentang sentimen yang terkandung dalam teks pada setiap contoh data. Hal ini memungkinkan adanya analisis sentimen berdasarkan isi teks dalam dataset yang digunakan.

Berikut adalah data yang telah dilabelling menggunakan library TextBlob.



Date	Tweet	text_clean	label
2023-05-03 06:25:...	@PRSundar64 US is...	us full news deb...	1
2023-05-03 06:25:...	"@saxena_puru But...	saxenapuru know c...	0
2023-05-03 06:25:...	@_swimfish @stats...	swimfish statsf s...	0
2023-05-03 06:24:...	@TaxpayersParty @...	taxpayersparti av...	1
2023-05-03 06:24:...	After losing at H...	lose happi index ...	0
Recession probabi...	2023:		0
2023-05-03 06:24:...	@TheKouk You have...	thekouk decid rec...	0
2023-05-03 06:24:...	Republicans' posi...	republican posit ...	0
- They ARE willin...	kills millions o...	kill million job	0
2023-05-03 06:23:...	The Treasury Dept...	treasuri dept war...	0
Either let them d...	000 jobs & ca...	job amp caus rec...	1
2023-05-03 06:23:...	@CWeston_Indo Dri...	cwestonindo drive...	0
2023-05-03 06:23:...	@acechhh it was m...	acechhh traumat r...	0
2023-05-03 06:23:...	$UBER Im not an i...	uber investor go ...	0
2023-05-03 06:23:...	Supply and demand...	suppli demand alw...	0
And from that	"		0
2023-05-03 06:23:...	Why is the Reserv...	reserv bank conti...	1
2023-05-03 06:22:...	America's debt to...	america debt took...	0
Defaulting on it ...	raise interest r...	rai interest rate	0
2023-05-03 06:22:...	@turabshah5 @stat...	statsf global re...	0

Gambar 4. 2 Data labelling menggunakan libraryTextBlob

4.3 Classification Using Pipeline in Apache Spark

Klasifikasi menggunakan Pipeline dalam Apache Spark adalah proses membangun alur kerja pembelajaran mesin yang melibatkan beberapa tahap untuk pra-pemrosesan teks dan pelatihan model. Pipeline memastikan aliran data dan operasi yang sistematis, sehingga memudahkan penanganan tugas pemrosesan teks yang kompleks dan pelatihan model klasifikasi dengan efisien.

Berikut merupakan langkah-langkah klasifikasi menggunakan pipeline dalam apache spark:

1. Menghapus Kolom yang Tidak Diperlukan:

   Kolom "Date" dan "Tweet" dihapus dari dataframe "df" menggunakan metode "drop". Tujuan dari langkah ini adalah melakukan persiapan data sebelum dilakukan pemrosesan lebih lanjut. Variabel "drop_cols" digunakan untuk menyimpan daftar

kolom yang akan dihapus. Kemudian, dilakukan operasi "drop(*drop_cols)" pada dataframe "df" untuk menghapus kolom-kolom tersebut.

2. Definisi Stages untuk Pipeline:

   Tahap-tahap (stages) dalam pipeline didefinisikan, termasuk Tokenizer, CountVectorizer, IDF, dan LogisticRegression. Tujuan dari langkah ini adalah untuk mengatur langkah-langkah pemrosesan teks dan pelatihan model klasifikasi dalam pipeline. Untuk melakukan itu, kita mengimpor kelas-kelas yang diperlukan dari modul "pyspark.ml.feature" dan "pyspark.ml.classification". Kemudian, objek tokenizer, countVectorizer, idf, dan logisticRegression didefinisikan sebagai tahap-tahap dalam pipeline.

3. Membuat Pipeline:

   Pipeline dibuat dengan menggabungkan semua tahap yang telah didefinisikan sebelumnya. Pipeline ini bertujuan untuk mengatur alur kerja pemrosesan teks dan pelatihan model secara otomatis. Untuk membuat pipeline, kita menggunakan objek-objek tahap yang telah didefinisikan sebelumnya dan menggunakan kelas "Pipeline" dari modul "pyspark.ml". Hasilnya, pipeline disimpan dalam variabel "pipeline".

4. Pembagian Data menjadi Train Set dan Test Set:

   Data dibagi menjadi set pelatihan (train set) dan set pengujian (test set) menggunakan metode "randomSplit". Tujuan dari langkah ini adalah untuk membagi data dengan proporsi tertentu agar dapat digunakan untuk melatih dan menguji model klasifikasi. Proporsi pembagian data train dan test ditentukan dengan menggunakan metode "randomSplit". Hasil pembagian data train dan test disimpan dalam variabel "trainDF" dan "testDF".

5. Melatih Model dengan Data Train menggunakan Pipeline:

   Model klasifikasi dilatih menggunakan dataset pelatihan (trainDF) menggunakan pipeline yang telah dibuat sebelumnya. Pipeline akan menjalankan langkah-langkah pemrosesan teks dan pelatihan model secara berurutan. Untuk melatih model, menggunakan metode "fit" pada objek pipeline dengan menggunakan data train (trainDF). Hasil model yang telah dilatih disimpan dalam variabel "model".

6. Prediksi pada Data Test:

   Setelah model dilatih, dilakukan prediksi pada dataset pengujian (testDF). Hasil prediksi disimpan dalam dataframe "predictions". Prediksi dilakukan menggunakan metode "transform".

7. Konversi ke Pandas DataFrame:

Dataframe "predictions" dikonversi menjadi Pandas DataFrame dengan nama "pred" untuk memudahkan analisis dan visualisasi data. Hal ini dilakukan dengan menggunakan metode "toPandas()" pada dataframe "predictions" dan hasilnya disimpan dalam variabel "pred". Untuk melihat beberapa baris pertama dari DataFrame "pred", menggunakan metode "head()".

Berikut adalah tampilan data untuk melakukan konversi ke pandas dataframe.

	text_clean	label	words_tokens	rawFeatures	features	rawPrediction	probability	prediction
0		0	[]	(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[5.861769995862287, -5.861769995862287]	[0.9971618764146486, 0.002838123585351404]	0.0
1		0	[]	(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	[5.861769995862287, -5.861769995862287]	[0.9971618764146486, 0.002838123585351404]	0.0
2	alreadi seen mani pune especi local builder s...	0	[, alreadi, seen, mani, pune, especi, local, b...	(1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...	(0.3497545010941586, 1.2361198296722717, 0.0, ...	[2.305795152101517, -2.305795152101517]	[0.9093558550758548, 0.09064414492414519]	0.0
3	anupampkh narendramodi plea studi basic econo...	0	[, anupampkh, narendramodi, plea, studi, basic...	(2.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, ...	(0.6995090021883172, 1.2361198296722717, 0.0, ...	[-1.1569202397994092, 1.1569202397994092]	[0.23922734395393663, 0.7607726560460634]	1.0
4	economi danger	0	[, economi, danger]	(0.0, 1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, ...	(0.0, 1.2361198296722717, 0.0, 0.0, 0.0, 2.701...	[3.0245654257513306, -3.0245654257513306]	[0.9536716535636671, 0.04632834643633288]	0.0

Gambar 4. 3 Tampilan data konversi ke pandas dataframe

8. Langkah Tambahan:

   Setelah pelatihan model, terdapat tahap tambahan yang melibatkan proses tokenisasi, penghitungan fitur kata menggunakan CountVectorizer, penghitungan bobot fitur menggunakan IDF, dan menampilkan data yang telah diproses pada set pelatihan (trainDF). Tahap-tahap tambahan ini berguna untuk melihat hasil pemrosesan teks secara lebih detail dan melakukan analisis tambahan pada data pelatihan (trainDF). Pada tahap-tahap ini, objek-objek seperti tokenizer, countVectorizer, idf, dan rescaledData digunakan untuk melihat hasil pemrosesan teks dan analisis fitur.

Tujuan dari tahap ini adalah mempersiapkan data, melatih model klasifikasi dengan menggunakan pipeline yang mencakup langkah-langkah pemrosesan teks dan pembuatan fitur, serta melakukan prediksi pada data pengujian. Hal ini bertujuan untuk memperoleh model klasifikasi yang dapat digunakan untuk memprediksi sentimen atau klasifikasi lainnya berdasarkan teks yang diberikan

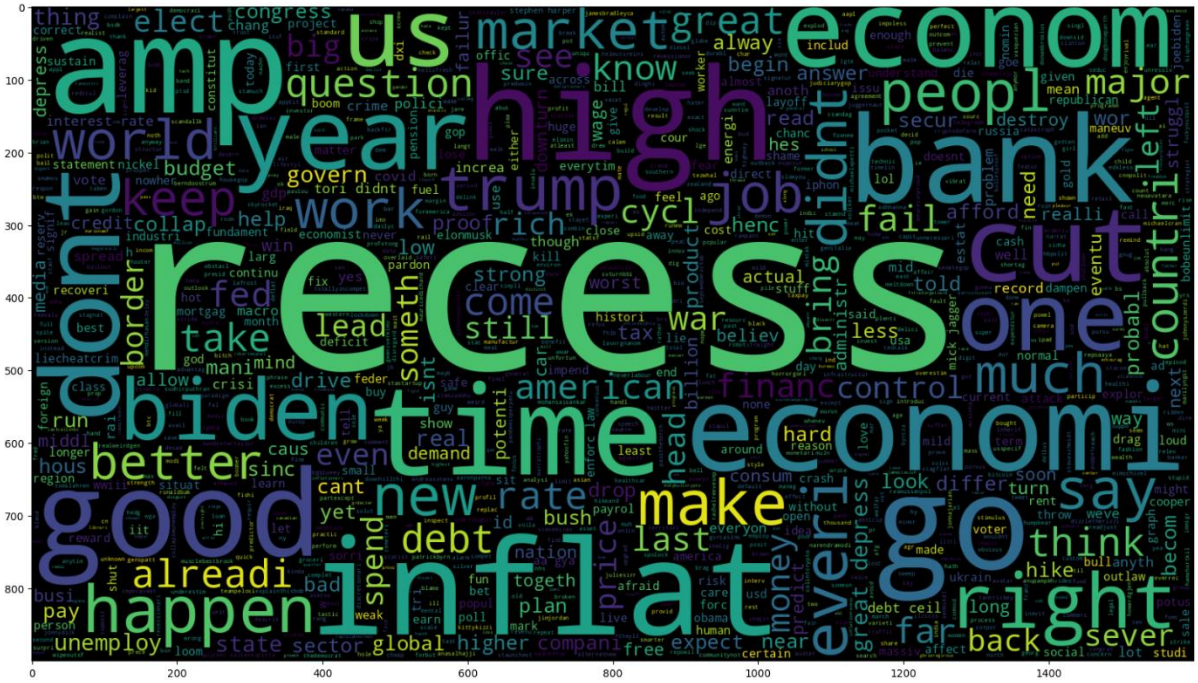
4.4 Word Cloud Visualization

Visualisasi Word Cloud adalah teknik yang digunakan untuk menampilkan kata-kata yang paling sering muncul dalam suatu kumpulan teks. Ini memberikan representasi visual mengenai kepentingan atau frekuensi relatif kata-kata, di mana ukuran sebuah kata dalam visualisasi sesuai dengan kejadian atau signifikansinya dalam teks.

4.4.1 Positive Sentiment

Positive Sentiment Word Cloud mengacu pada Word Cloud yang menampilkan kata-kata yang paling sering muncul dalam teks dengan sentimen positif. Dalam visualisasi ini, kata-kata yang berhubungan dengan hal-hal seperti kebahagiaan, kepuasan, kesuksesan, dan semangat akan muncul lebih besar dan menonjol dalam Word Cloud. Hal ini memberikan gambaran visual tentang kata-kata yang secara umum dikaitkan dengan sentimen positif dalam teks yang dianalisis.

Berikut merupakan positive sentiment word cloud.

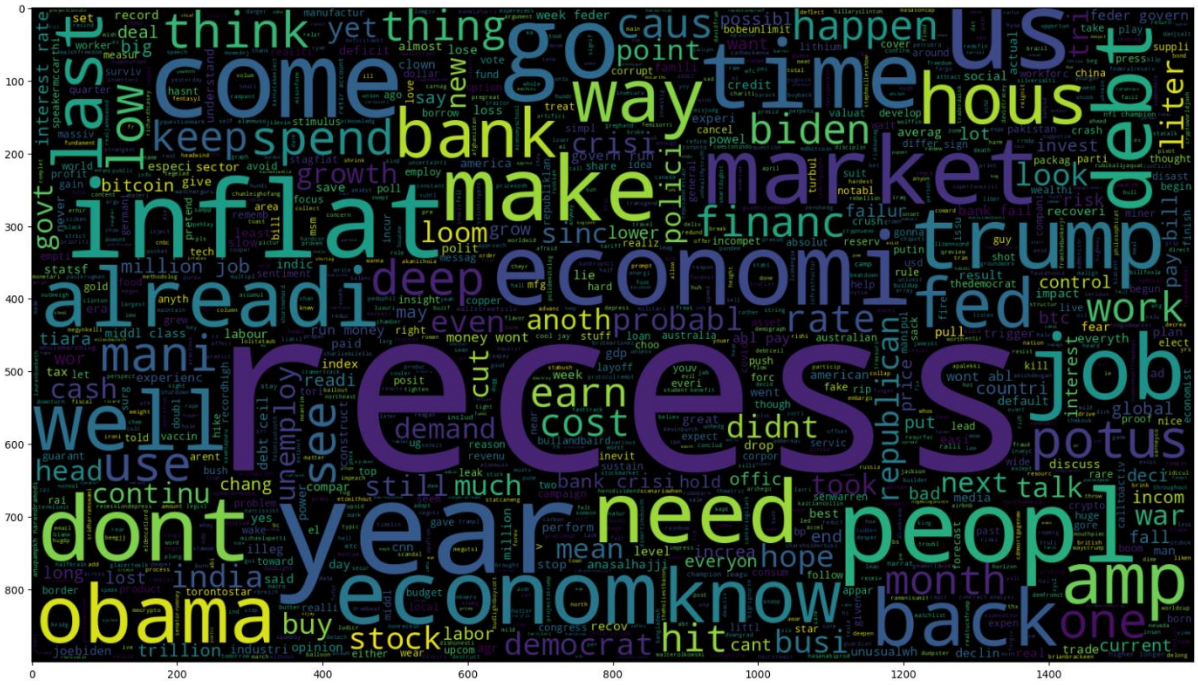


Gambar 4. 4 Positive Sentiment Word Cloud

4.4.2 Negative Sentiment

Negative Sentiment Word Cloud adalah Word Cloud yang menampilkan kata-kata yang paling sering muncul dalam teks dengan sentimen negatif. Dalam visualisasi ini, kata-kata yang berkaitan dengan hal-hal seperti kesedihan, kekecewaan, kemarahan, dan ketidakpuasan akan muncul lebih besar dan menonjol dalam Word Cloud. Ini memberikan gambaran visual tentang kata-kata yang secara umum dikaitkan dengan sentimen negatif dalam teks yang dianalisis.

Berikut merupakan negative sentiment word cloud.



Gambar 4. 5 Negative Sentiment Word Cloud

Dengan menggunakan Positive Sentiment Word Cloud dan Negative Sentiment Word Cloud, kita dapat dengan cepat mengidentifikasi kata-kata kunci yang mendukung atau mencerminkan sentimen positif atau negatif dalam teks. Ini membantu dalam pemahaman yang lebih baik tentang sentimen yang terkandung dalam teks, serta memberikan wawasan yang lebih mendalam tentang bagaimana kata-kata tertentu berkontribusi terhadap penilaian umum atau evaluasi yang positif atau negatif.

4.5 Evaluating the Model

Evaluasi model adalah tahap penting dalam proses pemodelan. Di tahap ini, kita memeriksa seberapa baik model yang telah dilatih dalam memprediksi data yang belum pernah dilihat sebelumnya, seperti data pengujian atau validasi.

Salah satu metrik evaluasi yang kita gunakan adalah area di bawah kurva ROC (areaUnderROC), yang memberikan gambaran tentang seberapa baik model dapat membedakan antara kelas positif dan negatif.

Pada bagian berikut ini akan ditampilkan hasil evaluasi areaUnderROC pada bagian *training*.

```
[ ]  # Menghasilkan prediksi pada data pelatihan
     trainPredictions = model.transform(trainDF)

     # Menghitung area under ROC pada data pelatihan
     trainAreaUnderROCTest = evaluator.evaluate(trainPredictions)
     print(f"The training areaUnderROC of our Logistic Regression model is: {trainAreaUnderROCTest}")
```

```
The training areaUnderROC of our Logistic Regression model is: 1.0
```

Gambar 4. 6 Training areaUnderROC

Pada bagian berikut ini akan ditampilkan hasil evaluasi areaUnderROC pada bagian *testing*.

```
[ ]  areaUnderROCTest = evaluator.evaluate(predictions)
     print(f"The testing areaUnderROC of our Logistic Regression model is: {areaUnderROCTest}")
```

```
The testing areaUnderROC of our Logistic Regression model is: 0.8219194002963651
```

Gambar 4. 7 Testing areaUnderROC

Selain itu, kita juga melihat metrik evaluasi lainnya seperti presisi, recall, dan f1-score untuk setiap kelas. Presisi mengukur seberapa sering model benar ketika memprediksi kelas positif, sementara recall mengukur seberapa baik model dalam menemukan semua contoh positif dalam data. F1-score adalah gabungan dari presisi dan recall yang memberikan gambaran tentang seimbangnya model dalam mengukur kedua metrik tersebut.

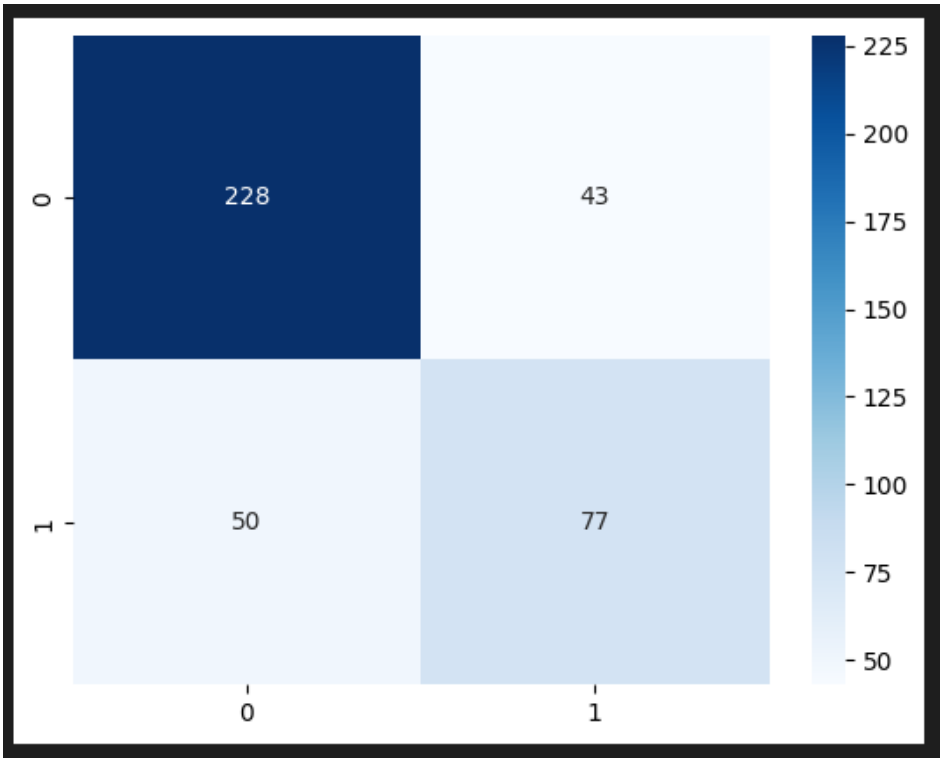
	precision	recall	f1-score	support
0.0	0.82	0.84	0.83	271
1.0	0.64	0.61	0.62	127
accuracy			0.77	398
macro avg	0.73	0.72	0.73	398
weighted avg	0.76	0.77	0.76	398

Gambar 4. 8 Metric Evaluation

Kita juga mempertimbangkan jumlah data (support) untuk setiap kelas, yang menunjukkan seberapa banyak contoh yang kita miliki untuk kelas tersebut. Ini penting untuk memahami apakah model kita mungkin bias terhadap kelas tertentu.

Model kita memiliki akurasi 77%, yang berarti model memprediksi dengan benar dalam 77% dari total data pengujian. Rata-rata metrik evaluasi juga disajikan untuk semua kelas, dengan Macro avg memberikan rata-rata metrik tanpa mempertimbangkan ketidakseimbangan data pada setiap kelas, sedangkan weighted avg memperhitungkan ketidakseimbangan tersebut.

Confusion matrix memberikan cara lain yang berguna untuk memahami kinerja model kita. Matriks ini menunjukkan empat kemungkinan hasil.



Gambar 4. 9 Confusion Matrix

Ada 228 contoh di mana model dengan benar memprediksi kelas negatif, dan 43 contoh di mana model salah memprediksi kelas positif. Sedangkan untuk kelas positif, ada 50 contoh yang diprediksi dengan benar oleh model, dan 77 contoh di mana model salah memprediksi sebagai kelas negatif. Melihat ini, kita bisa melihat bahwa model kita cenderung membuat lebih banyak kesalahan dalam memprediksi kelas positif sebagai negatif.

Secara keseluruhan, informasi ini memberikan gambaran yang mendalam tentang bagaimana model kita bekerja, dan memungkinkan kita untuk mengidentifikasi area di mana model mungkin memerlukan perbaikan.