

Tugas Text Mining SMS Spam dataset

Meliania Endang Nyimas Lisna

2110631250050





Tujuan

- Membuat model untuk mengklasifikasikan SMS menjadi Normal, Penipuan, atau Promo.
- Mengukur kinerja algoritma SVM dalam mengolah data teks SMS.



Dataset

Dataset merupakan Dataset SMS Spam
(Bahasa Indonesia) Rahmi, F. and
Wibisono, Y. (2016)

Jumlah Data

(1143, 2)

Terdapat 2 kolom yaitu: Teks dan
Label

Keterangan label (dataset_sms_spam_v1.csv):

0: sms normal

1: fraud atau penipuan

2: promo

	Teks	label
0	[PROMO] Beli paket Flash mulai 1GB di MY TELKO...	2
1	2.5 GB/30 hari hanya Rp 35 Ribu Spesial buat A...	2
2	2016-07-08 11:47:11.Plg Yth, sisa kuota Flash ...	2
3	2016-08-07 11:29:47.Plg Yth, sisa kuota Flash ...	2
4	4.5GB/30 hari hanya Rp 55 Ribu Spesial buat an...	2
5	5 HARI LAGI ! EKSTRA Pulsa 50rb dg beli paket ...	2
6	Ada iRing dgn tarif Rp. 0,1/7hr (perpanjangan ...	2
7	Akhir bulan harus tetap eksis loh! Internetan ...	2
8	Aktifkan iRing Coboy Jr - Terhebat. Tekan *808...	2
9	Ambil bonus harianmu di *600# (Bebas Pulsa). D...	2

Tahapan

Import Library &
Read Dataset



Text
Preprocessing



Text Cleaning



Modelling



```
[15] !pip install nltk  
!pip install Sastrawi  
  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
#for text pre-processing  
import re, string  
import nltk  
from nltk.tokenize import word_tokenize  
from nltk.corpus import stopwords  
from nltk.stem import WordNetLemmatizer  
  
nltk.download('punkt')  
nltk.download('averaged_perceptron_tagger')  
nltk.download('wordnet')  
nltk.download('stopwords')  
nltk.download('punkt_tab')  
  
#for modelling  
from sklearn.model_selection import train_test_split  
from sklearn.feature_extraction.text import TfidfVectorizer  
from sklearn.svm import SVC  
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score  
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score  
  
from sklearn.model_selection import train_test_split  
from sklearn.feature_extraction.text import TfidfVectorizer  
from sklearn.naive_bayes import MultinomialNB  
from sklearn.metrics import accuracy_score, classification_report
```

Import Library & Read Dataset



```
[17] data=pd.read_csv('dataset_sms_spam_v1.csv')  
print(data.shape)
```

Text Cleaning



Cek dan Hapus Duplikat Data

```
[20] # Mengecek duplikat
duplicat = data[data.duplicated()]
print(duplicat)
print(f"Jumlah duplikat: {duplicat.shape[0]}")

Teks    label
679  bebas nama1, terus nanti kalau ada tgl libur, ...      0
Jumlah duplikat: 1

[21] data = data.drop_duplicates()

# Mengecek duplikat
duplicat = data[data.duplicated()]
print(f"Jumlah duplikat: {duplicat.shape[0]}")

Jumlah duplikat: 0
```

Cek Missing Value

```
0d  #Missing values
data.isna().sum()

0
Teks 0
label 0
dtype: int64
```

Membersihkan karakter, tanda baca, emoticon yang tidak bermakna.

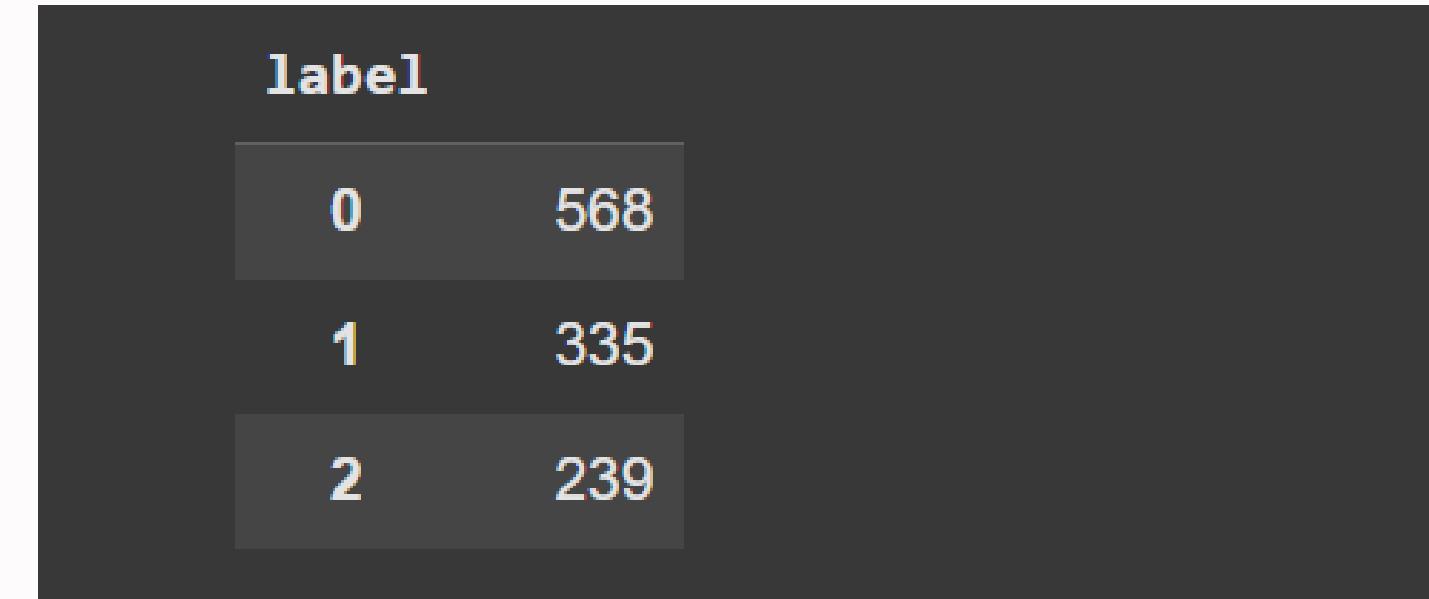
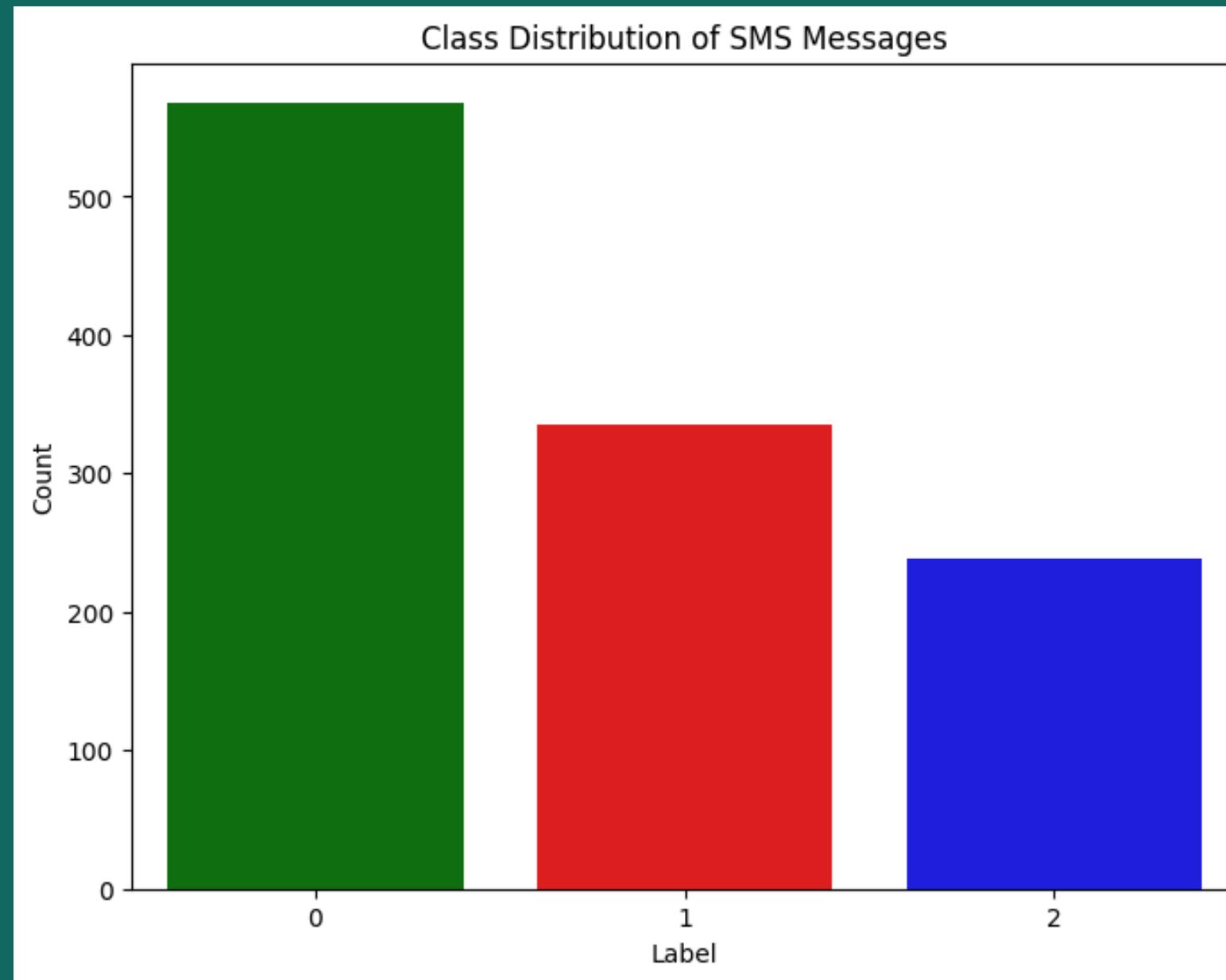
```
def clean_text(text):
    text = re.sub(r'[^a-zA-Z0-9\s]', '', text) # Remove special characters
    text = re.sub(r'\d+', '', text)          # Remove numbers
    text = text.lower() # Convert to lowercase
    text = re.sub(r"https?:\/\/.*[\r\n]*", "", text, flags=re.MULTILINE)
    text = re.sub(r'\b\w{1,2}\b', '', text) # Remove single characters
    text = re.sub(r'\s+', ' ', text) # Remove extra spaces
    text = re.sub(r'\b[a-zA-Z]\b', "", str(text)) # Remove single letters

    return text

data['clean_text'] = data['Teks'].apply(clean_text)
display(data[['Teks', 'clean_text']].style)
```

Distribusi Data

(1142 Data)



Pesan normal (bar berwarna hijau) mendominasi dataset dengan jumlah 568, diikuti oleh pesan penipuan (bar berwarna merah) 335, dan pesan promosi (bar berwarna biru) yang paling sedikit, kurang dari 239. Distribusi ini menunjukkan adanya ketidakseimbangan data (*imbalance data*), di mana pesan normal jauh lebih banyak dibandingkan dua kategori lainnya.

Text Preprocessing



• Tokenizing

Berfungsi untuk memecah teks menjadi unit-unit kecil seperti kata (word tokenization) atau kalimat (sentence tokenization).

kesempatan menjadi agen pulsaall operator mkios harga murah ketik gabung kirim	['kesempatan', 'menjadi', 'agen', 'pulsaall', 'operator', 'mkios', 'harga', 'murah', 'ketik', 'gabung', 'kirim']
kirim bank bca rek andrianto djunaedi usahakan hari ini trms	['kirim', 'bank', 'bca', 'rek', 'andrianto', 'djunaedi', 'usahakan', 'hari', 'ini', 'trms']
kirim mandiri aja arya rek klo sudah kabarin	['kirim', 'mandiri', 'aja', 'arya', 'rek', 'klo', 'sudah', 'kabarin']

• Normalisasi

Bertujuan untuk mengubah kata tidak baku, singkatan (seperti dpt menjadi dapat) ke bentuk standar

Silahkan klo pd mau besok gpp, saya nyusul aja paling laporan ke bapaknya. Udh ngehubungin bapak jd.	['silahkan', 'kalau', 'mau', 'besok', 'tidak apa apa', 'saya', 'nyusul', 'saja', 'paling', 'laporan', 'bapaknya', 'sudah', 'ngehubungin', 'bapak']
--	--

Text Preprocessing



• Stopwords

Berfungsi menghapus kata-kata umum yang tidak signifikan (tidak penting)

nggk masalah toh model bisa diload dicontroler mana aja kan	['tidak', 'masalah', 'model', 'diload', 'dicontroler']
yah saya dikostan temen dil hihi	['yah', 'dikostan', 'temen', 'dil', 'hihi']
yahh masih lama urgent ini wkwk	['yahh', 'urgent', 'wkwk']
yang ada waktu luang besok pada futsal jam sampoerna jangan ngaret	['luang', 'besok', 'futsal', 'jam', 'sampoerna', 'ngaret']

• Stemming

Mengembalikan kata ke bentuk dasar untuk menyamakan kata dengan makna serupa.

sel akan segera dikirimkan masukkan pin jika diminta terimakasih	['setting', 'ponsel', 'kirim', 'ma']
n ponsel anda ikuti petunjuk yang diberikan untuk menyelesaikan setting internet dan kirim fotmms	['setting', 'kirim', 'ponsel', 'ikut', 'tunjuk', 'selesai', 'setting', 'i']

Algoritma Support Vector Machine (SVM)



Modelling

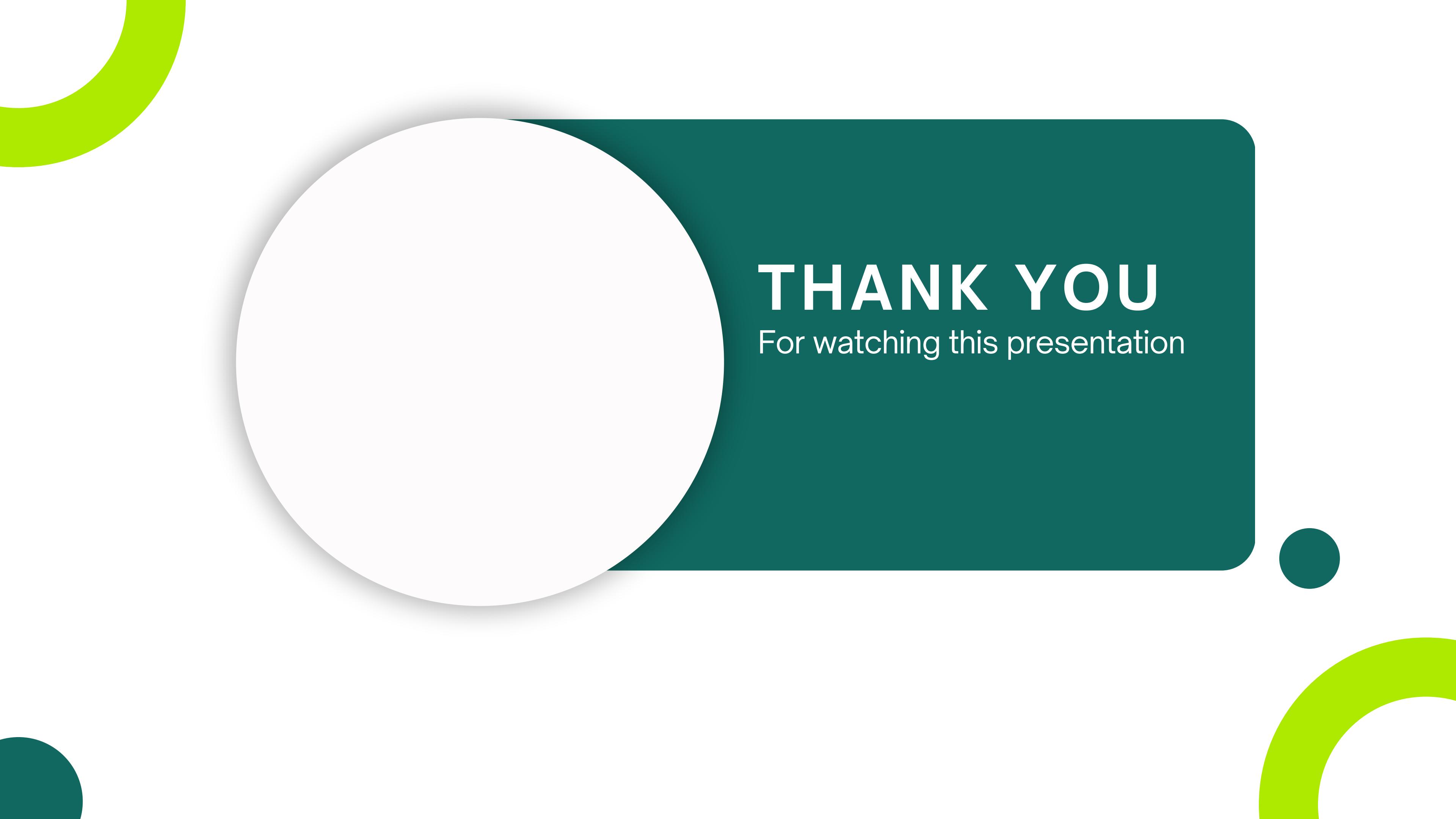
- Memisahkan data menjadi fitur (X) yang berisi teks yang telah dibersihkan dan label (y) yang sesuai. Selanjutnya, dataset dibagi menjadi dua bagian: 80% untuk data pelatihan (X_train, y_train) dan 20% untuk data pengujian (X_test, y_test)
- Menggunakan TfidfVectorizer untuk mengubah teks menjadi representasi numerik yang dapat digunakan oleh model SVM
- Membuat model SVM dengan kernel linear
- Setelah model dilatih, kita melakukan prediksi pada data pengujian
- Menghitung berbagai metrik evaluasi untuk menilai kinerja model. accuracy_score menghitung akurasi model, sedangkan precision_score, recall_score, dan f1_score menghitung presisi, recall, dan skor F1, masing-masing, dengan menggunakan rata-rata berbobot untuk menangani kelas yang tidak seimbang.

Output:

	precision	recall	f1-score	support
0	0.92	0.90	0.91	99
1	0.92	0.89	0.91	82
2	0.79	0.88	0.83	48
accuracy			0.89	229
macro avg	0.88	0.89	0.88	229
weighted avg	0.89	0.89	0.89	229
Confusion Matrix:	[[89 4 6] [4 73 5] [4 2 42]]			
AUC:	0.9753020411501652			

Kesimpulan

1. Model SVM dengan kernel linear berhasil mencapai akurasi sebesar 90%
2. Kinerja Klasifikasi Berdasarkan Kelas:
 - a. Kelas 0 (SMS Normal) : Presisi 91%, recall 92%, dan F1-score 91%
 - b. Kelas 1 (SMS Fraud): Presisi sebesar 93% dan recall 87% menunjukkan bahwa model memiliki sedikit kesulitan dalam mendeteksi beberapa kasus fraud dan F1-score 90%.
 - c. Kelas 2 (SMS Promo): Presisi 81%, recall 90%, dan F1-score 85%, menunjukkan bahwa beberapa pesan promo sulit diklasifikasikan dengan benar.
3. Confusion Matrix:
Sebagian besar data diklasifikasikan dengan benar, tetapi terdapat beberapa kesalahan klasifikasi.
4. Nilai AUC sebesar 97% menunjukkan bahwa model memiliki kemampuan yang sangat baik untuk membedakan antara kelas-kelas yang ada dalam data.



THANK YOU

For watching this presentation