

TUGAS 05 MATA KULIAH AI – SUPERVISED LEARNING - NAIVE BAYES

Anggota kelompok:

1. Andhika Rizki Pratama
2. Syamsuddin

Kasus yang diambil:

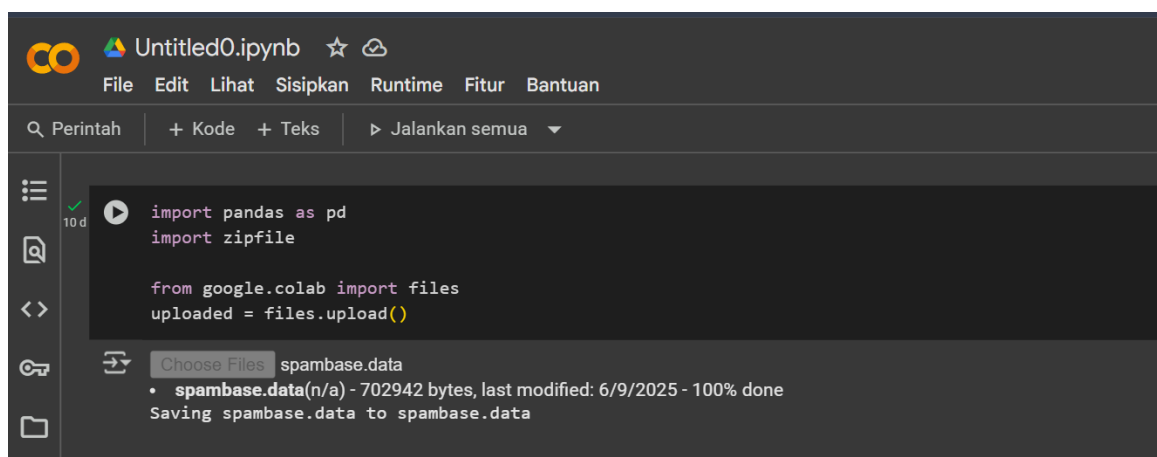
Masalah yang dicari solusinya adalah mengklasifikasikan apakah sebuah email merupakan spam atau bukan, berdasarkan fitur yang dapat terdapat dari isi email tersebut. Tujuan akhirnya adalah agar sistem email bisa secara otomatis memfilter email yang mencurigakan (spam) dan mencegah gangguan pada pengguna.

Jenis masalah dan isi data:

Jenis masalahnya yaitu klasifikasi, dan isi datanya yaitu memiliki 57 fitur numerik yang mewakili frekuensi kata/symbol tertentu dan satu label kelas, 1: Spam 0: Bukan spam.

Tahapan menyelesaikan masalah:

1. Persiapan data, seperti mengimpor dataset dan load dataset

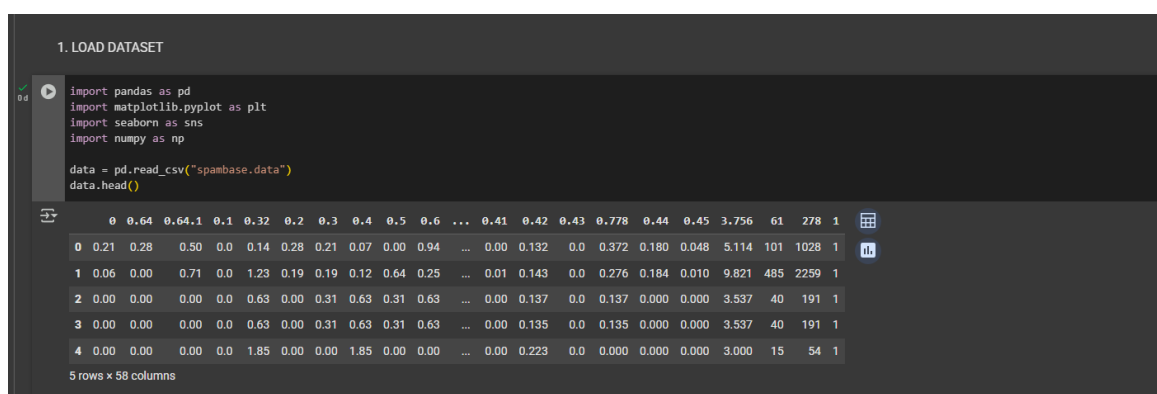


The screenshot shows the Google Colab interface. At the top, there's a menu bar with 'File', 'Edit', 'Lihat', 'Sisipkan', 'Runtime', 'Fitur', and 'Bantuan'. Below it is a toolbar with 'Perintah', '+ Kode', '+ Teks', and '▶ Jalankan semua'. The main area shows a code cell with the following code:

```
import pandas as pd
import zipfile

from google.colab import files
uploaded = files.upload()
```

Below the code cell, there's a file upload section. It shows a 'Choose Files' button and a list of files. The file 'spambase.data' is selected, with details: 'spambase.data(n/a) - 702942 bytes, last modified: 6/9/2025 - 100% done'. Below this, it says 'Saving spambase.data to spambase.data'.



The screenshot shows the Google Colab interface with the title '1. LOAD DATASET'. The code cell contains the following code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

data = pd.read_csv("spambase.data")
data.head()
```

Below the code cell, there's a preview of the data. It shows a table with 5 rows and 58 columns. The first row is:

	0	0.64	0.64.1	0.1	0.32	0.2	0.3	0.4	0.5	0.6	...	0.41	0.42	0.43	0.778	0.44	0.45	3.756	61	278	1
0	0.21	0.28	0.50	0.0	0.14	0.28	0.21	0.07	0.00	0.94	...	0.00	0.132	0.0	0.372	0.180	0.048	5.114	101	1028	1
1	0.06	0.00	0.71	0.0	1.23	0.19	0.19	0.12	0.64	0.25	...	0.01	0.143	0.0	0.276	0.184	0.010	9.821	485	2259	1
2	0.00	0.00	0.00	0.0	0.63	0.00	0.31	0.63	0.31	0.63	...	0.00	0.137	0.0	0.137	0.000	0.000	3.537	40	191	1
3	0.00	0.00	0.00	0.0	0.63	0.00	0.31	0.63	0.31	0.63	...	0.00	0.135	0.0	0.135	0.000	0.000	3.537	40	191	1
4	0.00	0.00	0.00	0.0	1.85	0.00	0.00	1.85	0.00	0.00	...	0.00	0.223	0.0	0.000	0.000	0.000	3.000	15	54	1

Below the table, it says '5 rows x 58 columns'.

2. Pra-pemrosesan, seperti memisahkan fitur dan label, dan membagi data

```
Memisahkan fitur (X) dan label (Y)

from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import recall_score, f1_score

X = data.iloc[:, :-1]
y = data.iloc[:, -1]

Membagi data menjadi training dan testing

[18] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

3. Modeling, melatih model Naive Bayes

```
Melatih model Naive Bayes

[20] model = GaussianNB()
      model.fit(X_train, y_train)

      y_pred = model.predict(X_test)
```

4. Evaluasi, menghitung dan menampilkan metrik evaluasi, serta memvisualisasikan hasil menggunakan confusion matriks

```
Menghitung dan menampilkan metrik evaluasi (Recall)

[24] recall = recall_score(y_test, y_pred)
      f1 = f1_score(y_test, y_pred)

      recall, f1

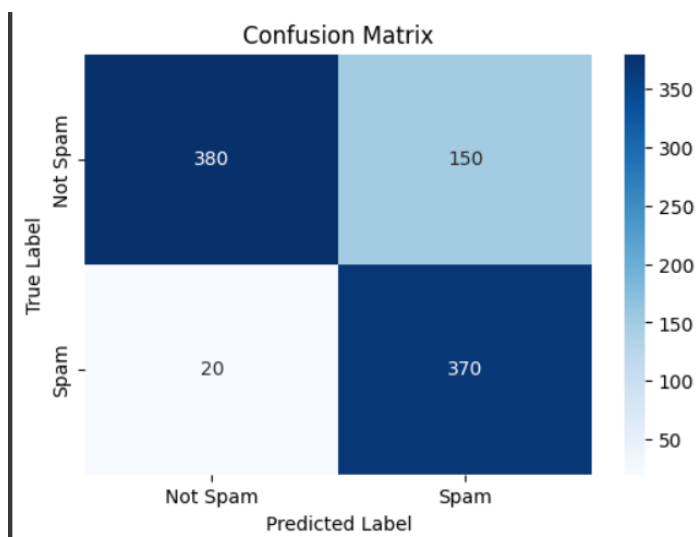
(0.9487179487179487, 0.8131868131868132)

Memvisualisasikan

import seaborn as sns
from sklearn.metrics import confusion_matrix

# Buat confusion matrix
cm = confusion_matrix(y_test, y_pred)

# Visualisasikan
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=["Not Spam", "Spam"], yticklabels=["Not Spam", "Spam"])
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix")
plt.show()
```



Error metric yang digunakan dan alasan memilih:

Recall = Mengurangi risiko pengguna terganggu atau tertipu oleh spam/phishing dan cocok untuk kasus di mana False Negative lebih berbahaya (spam tidak terdeteksi).

F1 score = Kombinasi presisi dan recall, sangat berguna saat data tidak seimbang, menghindari model yang "hanya mengejar recall" tapi mengorbankan precision terlalu besar (yaitu asal menandai banyak email sebagai spam agar recall tinggi), dan memberikan gambaran realistis terhadap performa keseluruhan model.

Hasil kinerja model dan rekomendasi:

- Model memiliki akurasi yang cukup tinggi secara umum.
- Recall sangat tinggi, artinya model sangat baik dalam menangkap spam (positif).
- F1 Score, yang merupakan keseimbangan antara precision dan recall, menunjukkan performa model yang solid untuk tugas ini.
- Jika fokus utama adalah melindungi pengguna dari spam, model ini cukup bagus.
- Tapi kalau banyak email penting dianggap spam, maka perlu perbaikan atau tuning model (misalnya lewat pengaturan threshold, fitur tambahan, atau hybrid model).

Link referensi:

Dataset: <https://archive.ics.uci.edu/dataset/94/spambase>

Code: ChatGPT

Teori: https://scikit-learn.org/stable/modules/naive_bayes.html

Link Github: