Analisis Hasil yang didapatkan dari NLP klasifikasi dengan Algoritma Naive Bayes

Pada pengaplikasian algoritma Naive Bayes terhadap data teks, diperlukan beberapa tahapan sebelum melakukan eksekusi final dan prediksi.

Tahap-tahap yang dilalui antara lain :

1. preprocessing data

tahap ini merupakan step yang penting, karena sebelum data bisa dieksekusi dengan Naive Bayes, data harus sudah bersih dari missing values serta output yang mulanya kategorical harus ditransformasi menjadi data numerical

Sehingga 2 hal pada tahap pertama yang dilakukan pada notebook ini adalah mencari missing_value dan mengatasinya (dalam hal ini dilakukan penghapusan (drop)). metode drop dipilih karena jumlah missing value yang ditemukan adalah 77/1070 raw dan ditambah lagi data merupakan text, sehingga metode terbaik yang dipilih adalah didrop (dibandingkan dg mengganti dengan nilai mean, median, atau modus). Tahap yang dilakukan selanjutnya adalah mentransformasi data output yang semula kategorikal (sentimen) menjadi numerik. hal dilakukan agar dataset yang dimiliki dapat diproses lebih lanjut menggunakan algoritma machine learning (Naive Bayes). Mengingat bahwa, library machine learning python hanya bisa memproses data numerik.

Adapun data-data yang dilakukan transformasi :

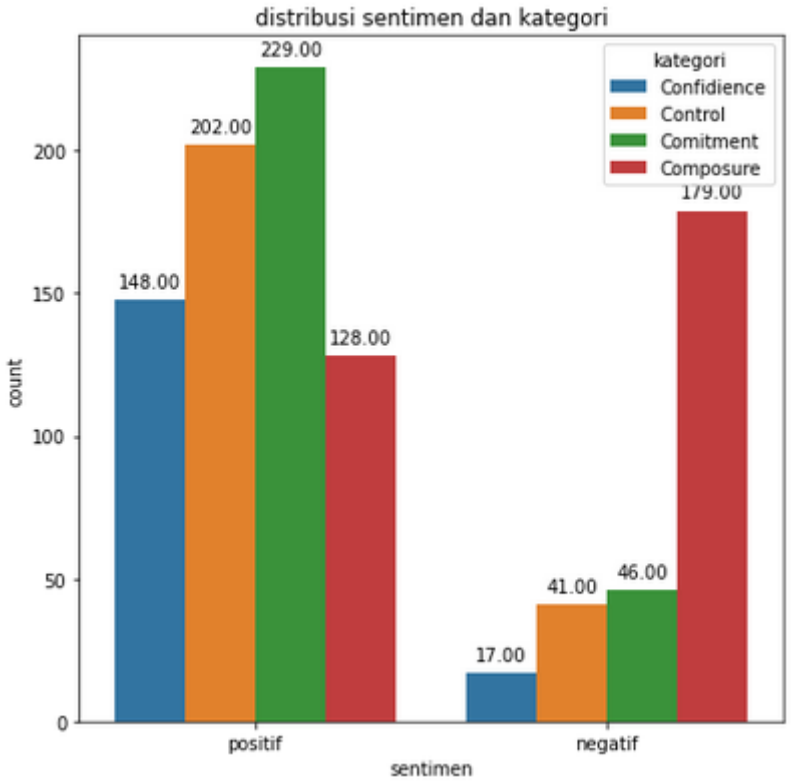
- sentimen (positif =1, negatif=0)
- kategori (0= commitment, 1= composure, 2= confidence, 3= control)

Setelah melakukan transformasi dari kategorikal ke numerikal mari kita lihat korelasi antara kedua kelompok kelas tersebut pada gambar di bawah.

	sentimen	kategori
sentimen	1.000000	0.107575
kategori	0.107575	1.000000

Dari gambar di atas tampak bahwa, antara kelua kelompok kelas (kelas kategori dan sentimen tidak memiliki korelasi / berkorelasi sangat lemah (mendekati nol). Hal ini menujukkan bahwa yang mempengaruhi nilai positif/negatif serta kategori (0= commitment, 1= composure, 2= confidence, 3= control) dari suatu kalimat adalah isi dan intensi dari kalimat itu sendiri, yang dapat kita nilai dari *text processing* dan algoritma Naive bayes yang kita aplikasikan.

Pada project ini kedua kelas diatas akan dipisah prediksi klasifikasinya, sehingga di notebook pertama merupakan klasifikasi kelas kategori (4 kelas) dan pada notebook kedua merupakan klasifikasi kelas sentimen (2 kelas). Pada gambar di bawah ini ditampilkan grafik frekuensi perbandingan antara kelas sentimen dan kelas kategori jika keduanya berada pada output yang sama.



Setelah semua missing values telah diatasi, dan kategorikal data ditransformasi menjadi data numerik, maka selanjutnya hal yang dilakukan adalah *text preprocessing.* Pada bahasa pemrograman Python text processing yang dilakukan adalah penghilangan tanda baca (kutip, koma, dsb), mengecilkan huruf, serta transformasi text manjadi vektor, agar bisa dilakukan pengolahan data dengan algoritma machine learning (Naive Bayes).

2. Tahap pembagian dataset menjadi data latih dan data test

perbandingan antara data latih dan data testing pada project ini adalah 20% data testing dan 80% data latih, seperti yang ditampilkan cuplikan *code* python pada gambar berikut.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_, y, test_size = 0.20, random_state = 0)
```

Secara teori, perbandingan data latih dan data testing dapat berkisar antara 80:20, 70:30, 75:25.

3. Eksekusi dataset yang telah dilatih dengan menggunkan library machine learning (*sklearn*) untuk algoritma Naive Bayes

   Algoritma Naive Bayes merupakan metode yang cocok untuk klasifikasi biner dan multikelas. Ia menerapkan teknik *supervised learning* untuk klasifikasi objek di masa depan dengan menetapkan label kelas ke *instance*/catatan menggunakan probabilitas bersyarat.

   Pada bahasa pemrograman python, sudah tersedia library untuk mengeksekusi algoritma Naive Bayes, yang bernama library *sklearn*. Seperti yang tampak pada gambar di bawah :

```
#mengimport modul untuk eksekusi algoritma Naive Bayes
from sklearn.naive_bayes import GaussianNB
classifier= GaussianNB()
classifier.fit(X_train, y_train) #melatih model dari data training yang ada
```

   setelah menyocokkan antara data latih input (x_train) dengan data latih keluaran (y_train), maka dilakukan prediksi pada data testing yang telah diambil dari 20% data. Dimana, hasil prediksi dari data testing merupakan hasil yang didapatkan dari model Naive Bayes yang telah dilatih. Berikut adalah potongan contoh hasil prediksi dan nilai sebenarnya dari model Naive Bayes yang telah dibuat :

	nilai_sebenarnya	hasil_prediksi
278	1	1
1278	0	0
464	0	1
152	1	1
202	1	1
...	...	...
1140	0	0
1045	0	0
1158	0	0
390	1	1
1256	0	0

nilai sebenarnya (data testing) dan hasil predisksi pada 2 kelas

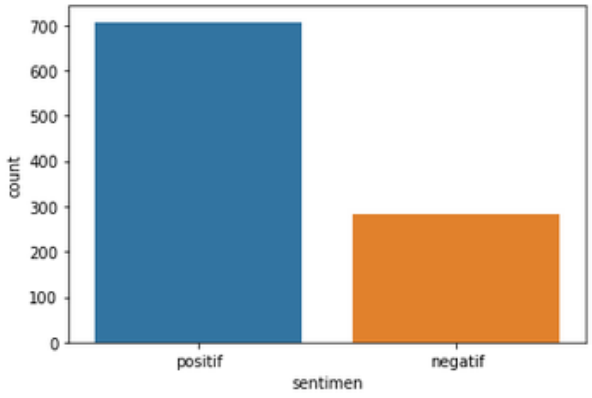
	nilai_sebenarnya	hasil_prediksi
18	0	3
546	3	0
268	1	1
1221	3	0
874	2	2
...	...	...
945	1	0
283	3	1
349	1	0
333	1	2
589	1	3

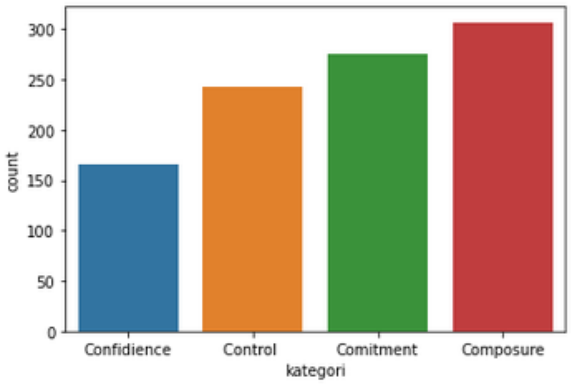
nilai sebenarnya (data testing) dan hasil predisksi pada 4 kelas

Untuk membuktikan bahwa model Naive Bayes yang kita latih memberikan hasil yang akurat, maka kita melakukan perhitungan akurasi terhadap data latih dan data testing.

Namun sebelumnya pada project ini, dilakukan teknik tambahan yaitu teknik resampling. Sehingga data akurasi yang ditampilkan adalah 2 data nilai akurasi antara model Naive Bayes yang datanya telah diresmpaling dan model tanpa resampling data.

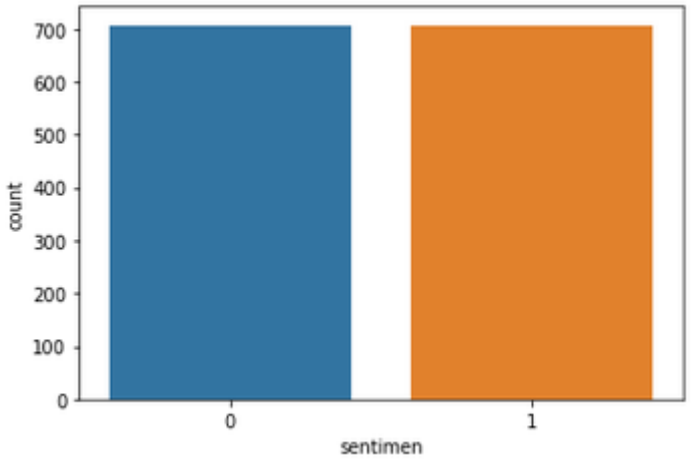
Gambar di bawah ini menampilkan distribusi kelas dari sentimen (positif dan negatif) serta kategori (kategori (0= commitment, 1= composure, 2= confidence, 3= control)

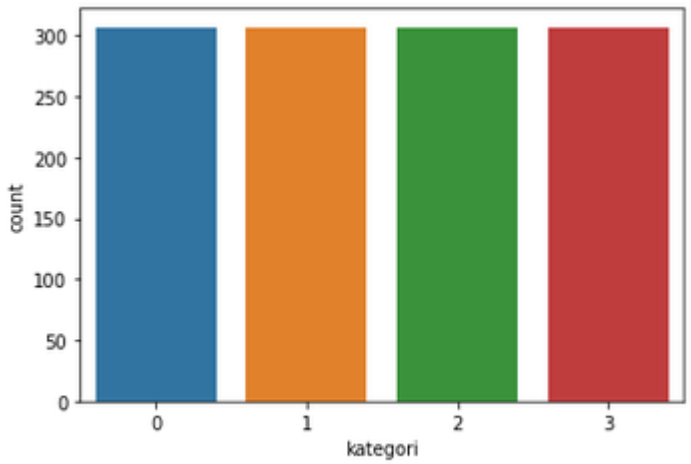




Dari 2 gambar di atas tampak bahwa, kedua kelompok memiliki masalah ketidakseimbangan distribusi data (*imbalance dataset*). permasalahan ini dapat menyebabkan ketidakakurasian dalam prediksi. Sehingga pada project ini permasalahan *imbalance dataset* ini diatasi dengan melakukan teknik *resampling*. teknik *resampling* ini menduplikasi secara acak data dari kelas minoritas maupun kelas mayoritas, tergantung teknik *resampling* yang dipilih. Dalam hal ini, tekniki *resampling* yang dipilih adalah ROS (*Random Over Sampling*), maka data acak yang dipilih adalah dari kelas mayorita, lalu disebar secara merata ke seluruh data. teknik ROS dipilih karena data yang kita miliki lumayan sedikit yakni hanya 990 raws data (setelah dilakukan pembersihan).

Berikut ditampilkan data setelah dilakukan resampling. (data yang ditampilkan dalam bentuk numerik yang telah ditransformasi dari data kategorikal)





Sehingga setelah mengaplikasikan algoritma Naive Bayes, mari kita analisis tingkat akurasinya (prediksi benar per total prediksi) berdasarkan diaplikasikan/tidaknya teknik resampling

1. klasifikasi 4 kategori (0= commitment, 1= composure, 2= confidence, 3= control)

Dengan Resampling	Tanpa Resampling
0,52	0,34

2. klasifikasi 2 kategori (0= negatif, 1= positif)

Dengan Resampling	Tanpa Resampling
0.75	0,61

Dari hasil tingkat akurasi di atas tampak bahwa teknik resampling memberikan pengaruh signifikan terhadap tingkat akurasi data prediksi.

Selain tingkat akurasi, terdapat 3 metrik lain yang bisa kita gunakan untuk menghitung performa dari model kita yaitu *precission*, *recall*, dan f1-score.

Dimana :

- precision adalah jumlah prediksi benar kelas positif per jumlah prediksi kelas positif
- recall adalah jumlah prediksi benar kelas positif per total data untuk kelas positif
- f1-score adalah keseimbangan dari precision dan recall

dimana, semakin mendekati 1 nilai-nilai metrik di atas, semakin baik pula model yang kita pakai bekerja.

1. untuk 4 kategori (sebelum resampling)

kategori	precision	recall	f1-score
0	0,30	0,29	0,29
1	0,40	0,27	0,32
2	0,23	0,31	0,26
3	0,42	0,50	0,46

2. untuk 4 kategori (setelah resampling)

kategori	precision	recall	f1-score
0	0,39	0,33	0,36
1	0,37	0,28	0,32
2	0,68	0,63	0,65
3	0,49	0,71	0,58

Nilai tiap parameter menujukkan kualitas model untuk memprediksi tiap kelas. Dari kedua tabel di atas, tampak bahwa, nilai tiap parameter mengalami peningkatan setelah dilakukan resampling pada dataset yang digunakan. Hal ini menunjukkan bahwa pada kelas yang tidak seimbang sangat dibutuhkan teknik resampling untuk mengatasi ketidak akuratan serta ketidakpresisisan data. Mari kita lihat untuk data pada kelas sentimen (positif dan negatif)

1. Data pada data sebelum diresampling

kategori	precision	recall	f1-score
0	0,36	0,53	0,43
1	0,78	0,64	0,70

2. Data pada data setelah diresampling

kategori	precision	recall	f1-score
0	0,79	0,63	0,70
1	0,70	0,84	0,76

Kembali kita lihat dari dua tabel terakhir, bahwa teknik resampling memberikan pengaruh signifikan dalam peningkatan metrik pengukuran performa model Naive Bayes.
Sehingga untuk dataset sejenis, sangat dianjurkan untuk melakukan preprocesing berupa resampling sebelum mengaplikasikan algoritma Naive Bayes.

Beberapa temuan penting :

- dua kelompok kelas sentimen dan kategori tidak berkorelasi satu sama lain. kelas dipengaruhi oleh input teks.
- Resampling memberikan pengaruh signifikan dalam menungkatkan akurasi, presisi dan recall dari model Naive Bayes.
- metode Naive Bayes belum memberikan akurasi yang baik pada dataset 4 kelas terutama pada data yang belum diresampling, akurasi meningkat pada data yang telah diresampling namun, akurasi yang diberikan masih 0.5 (sedang)
- sedangkan Naive Bayes menunjukkan performa yang baik (akurasi 0,7) pada dataset dengan 2 kelas, terutama setelah dilakukan resampling data, sehingga untuk dataset sejenis, bisa direkomendasikan untuk memakai Naive Bayes untuk prediksi.