Data Preprocessing Bagian - II

04 April 2024

Materi Hari Ini

Kita akan mempelajari antara lain:

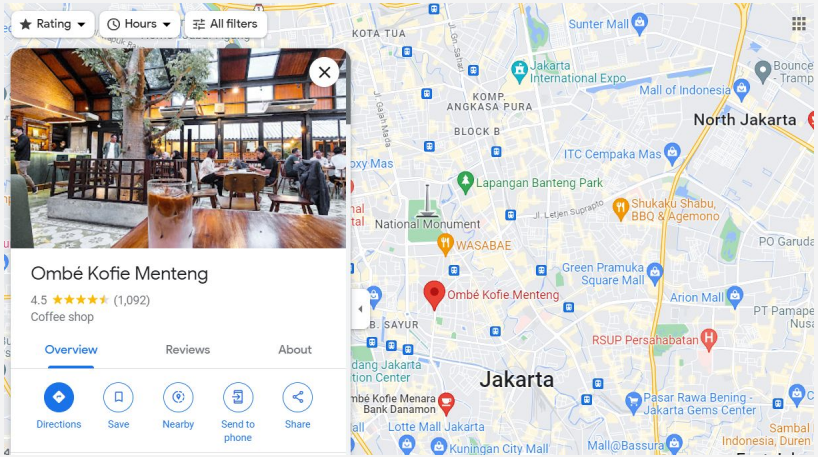
1. Pra-proses untuk Data Teks
2. Vektorisasi pada Data Teks
3. Penanganan Data dengan Proporsi Tak Seimbang (Imbalance Data)

Pra-proses Data Teks (Data Text Preprocessing)

Apa dan Bagaimana

Data Tanpa Struktur: Teks

"Came here during lunch time on a public holiday and the place was packed. Not so much that it disturbs the conversation, but packed enough that it feels like you're surrounded by a bee swarm. Their service crew was very attentive and quick to assist."

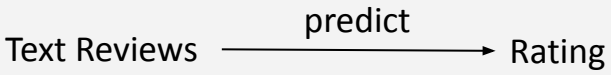


"This place is like a secret place at the center of Jakarta. It's nice to visited here. Fine place fine food and also the coffee is great! They had good taste of coffee. If you consider to find a place for meeting or chit chat, you should go here!"

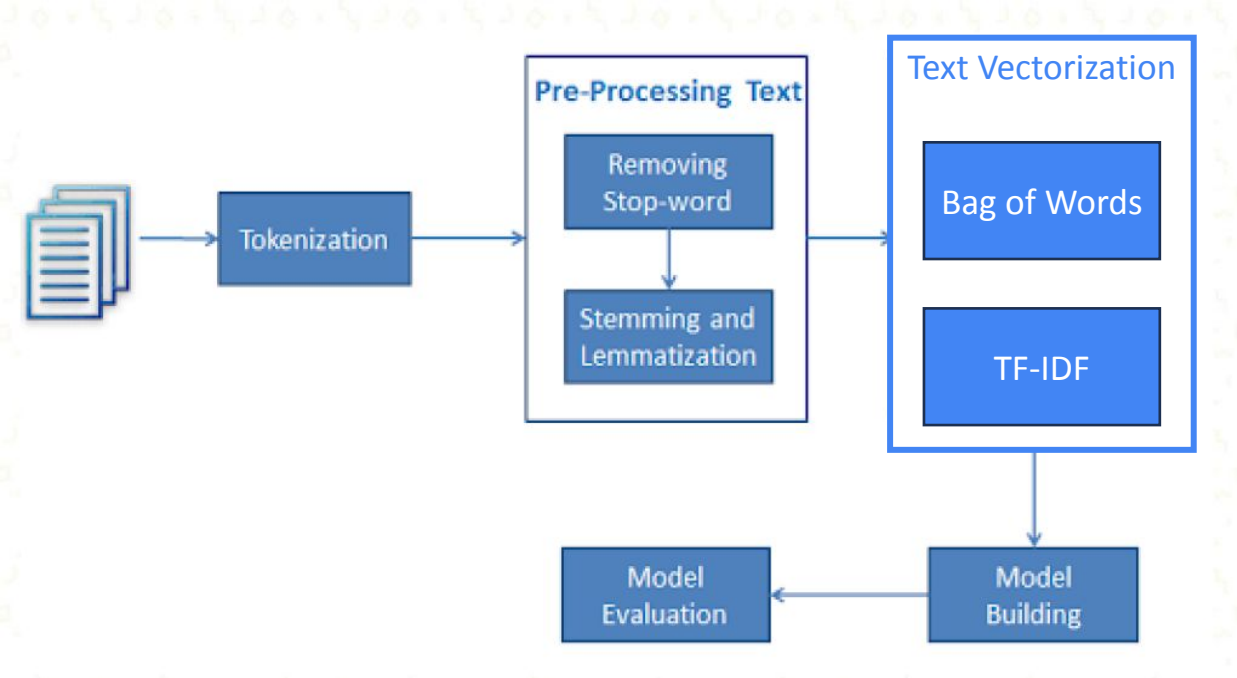
"Horrible experience here. Waited 1 hour and a half for scrambled eggs and they messed up my order twice. This place is NOT vegetarian friendly. I used to be a regular, but now I found that the quality of food has decreased dramatically and the price now is way too expensive for their small portion. Waiter did NOT take responsibility for their actions. I do not recommend eating here."

Data Tanpa Struktur: Teks

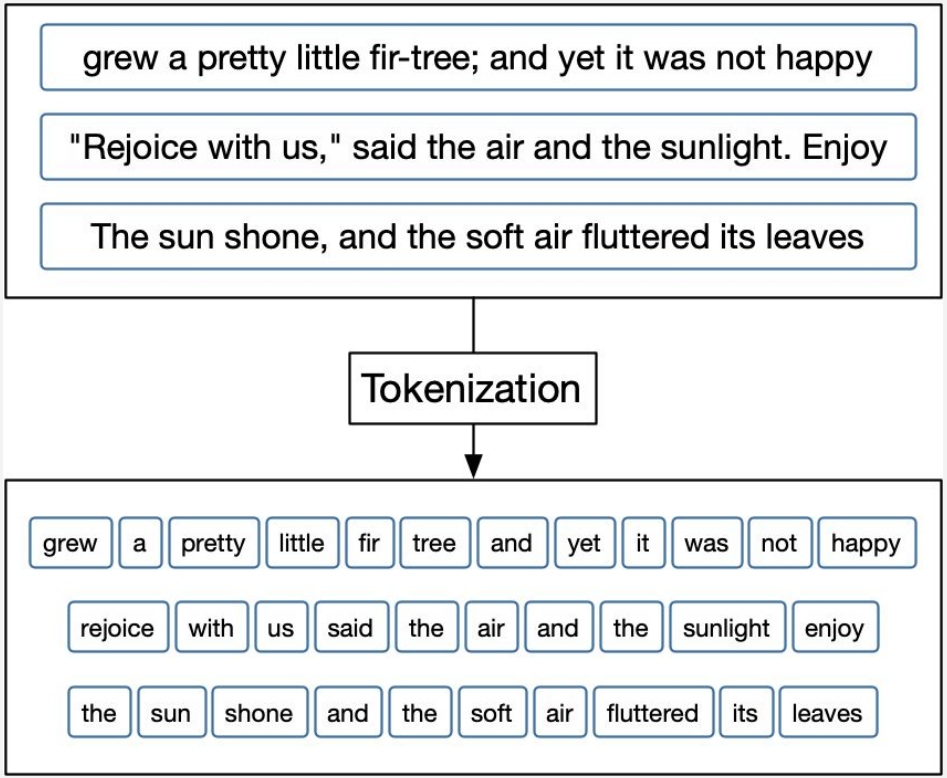
Text	Rating
“Came here during lunch time on a public holiday and the place was packed. Not so much that it disturbs the conversation, but packed enough that it feels like you’re surrounded by a bee swarm. Their service crew was very attentive and quick to assist.”	5
“This place is like a secret place at the center of Jakarta. It's nice to visited here. Fine place fine food and also the coffee is great! They had good taste of coffee. If you consider to find a place for meeting or chit chat, you should go here!”	4
…	…
“Horrible experience here. Waited 1 hour and a half for scrambled eggs and they messed up my order twice. This place is NOT vegetarian friendly. I used to be a regular, but now I found that the quality of food has decreased dramatically and the price now is way too expensive for their small portion. Waiter did NOT take responsibility for their actions. I do not recommend eating here.”	1



Menangani Data Teks



Pra-proses Teks: Tokenisasi (1)



Pra-proses Teks: Case Folding (2)

- **Case Folding** menggambarkan proses konsolidasi beberapa ejaan dari satu kata yang hanya berbeda dalam kapitalisasi.
- **Teknik normalisasi** ini juga dikenal sebagai **case normalization.**
- Case Folding adalah salah satu cara untuk mengurangi ukuran kosakata dan memungkinkan generalisasi yang lebih baik untuk NLP Pipeline.
- Namun, normalisasi kasus **dapat mengakibatkan hilangnya informasi berharga dan penting**.
- Ambil contoh “**Turki**” dan “turkey” (kalkun). Yang pertama adalah negara. Yang kedua adalah burung. Dua makna yang sangat berbeda.

Pra-proses Teks: Penghapusan Stopwords (3)

- Stop words adalah sekumpulan kata yang umum digunakan dalam setiap bahasa.
- Sebagai contoh, dalam bahasa Inggris, "the", "is", dan "and" dengan mudah dapat dikategorikan sebagai kata-kata penghubung.
- Dalam NLP, kata-kata penghubung digunakan untuk menghilangkan kata-kata yang tidak penting, memungkinkan aplikasi untuk fokus pada kata-kata penting saja.

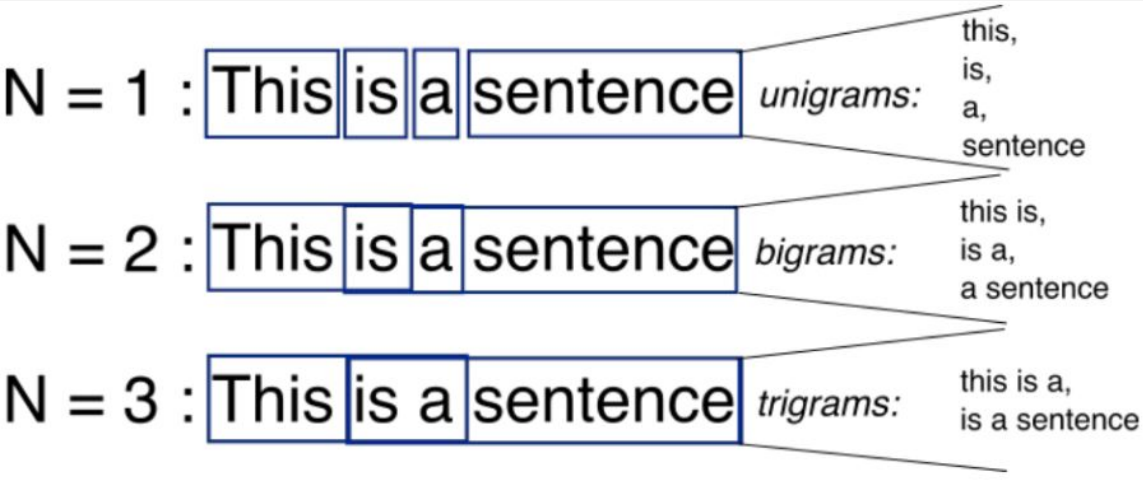
WITH STOP WORDS	WITHOUT STOP WORDS
/growing-up-with-hearing-loss/	/growing-hearing-loss/
/coming-to-terms-with-hearing-loss/	/coming-terms-hearing-loss/
/standing-in-the-line-of-fire/	/standing-line-fire/
/against-all-odds/	/odds/
/what-is-hearing-loss/	/what-hearing-loss/
/cant-start-a-fire/	/start-fire/
/living-with-pain/	/living-pain/

Pra-proses Teks: Stemming & Lemmatization (4)

Stemming	Lemmatization
Stemming adalah proses yang menghilangkan beberapa karakter terakhir dari sebuah kata, sering kali menghasilkan makna dan ejaan yang salah.	**Lemmatization** mempertimbangkan konteks dan mengonversi kata ke bentuk dasarnya yang memiliki makna, yang disebut Lemma.
Sebagai contoh, proses stemming pada kata 'Caring' akan mengembalikan 'Car'.	Sebagai contoh, memproses lemmatization pada kata 'Caring' akan mengembalikan 'Care’
Stemming digunakan dalam kasus dataset besar di mana performa menjadi isu utama.	Lemmatization membutuhkan komputasi yang mahal karena melibatkan tabel pencarian.

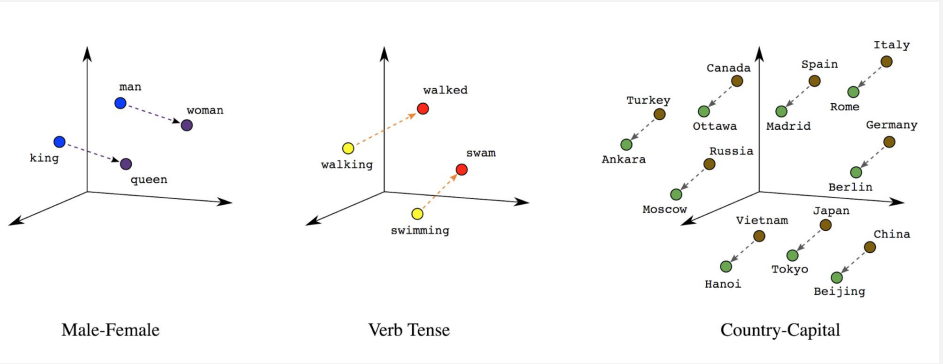
Pra-proses Teks:
N-gram (Optional)

- **N-gram** adalah langkah awal dalam analisis teks dimana teks dibagi menjadi potongan berurutan yang terdiri dari n kata atau karakter. n-gram digunakan untuk menangkap konteks lokal dalam teks dengan mempertimbangkan hubungan antara kata-kata atau karakter-karakter yang berdekatan



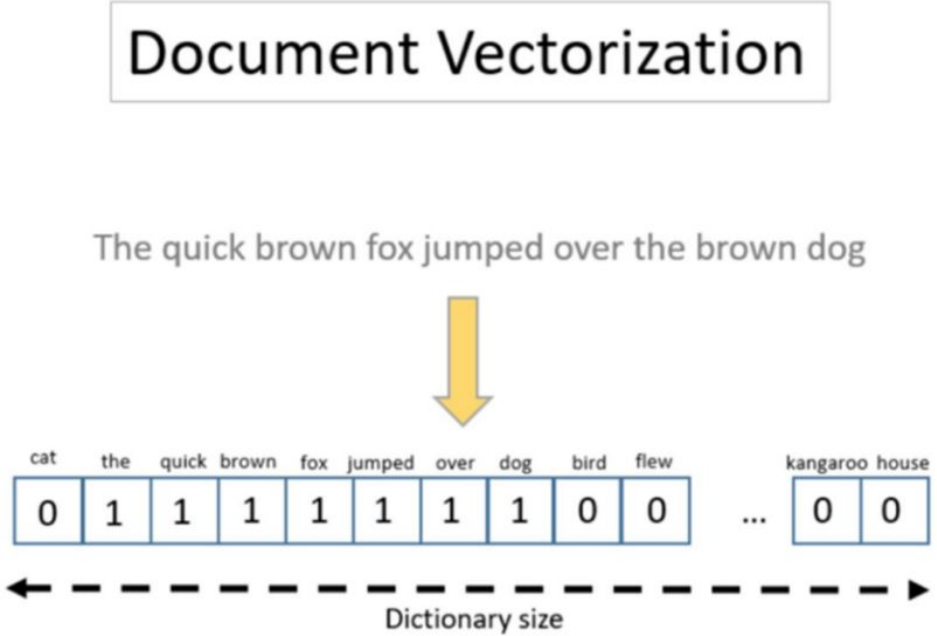
Vektorisasi pada Data Teks (Data Text Vectorization)

Apa dan Bagaimana



Vektorisasi Teks

- **Vektorisasi Teks** adalah proses mengubah teks menjadi representasi numerik
- **Teknik-teknik** Vektorisasi Teks: Bag of Words (BoW), TF-IDF, Word Embedding (Word2Vec, BERT, etc.)



Teknik Bag Of Words (BoW)

	1 This	2 movie	3 is	4 very	5 scary	6 and	7 long	8 not	9 slow	10 spooky	11 good	Length of the review(in words)
Review 1	1	1	1	1	1	1	1	0	0	0	0	7
Review 2	1	1	2	0	0	1	1	0	1	0	0	8
Review 3	1	1	1	0	0	0	1	0	0	1	1	6

Review 1: This movie is very scary and long
Review 2: This movie is not scary and is slow
Review 3: This movie is spooky and good

Teknik TF-IDF

- **TF-IDF** merupakan kepanjangan dari Term Frequency-Inverse Document Frequency. metode yang digunakan untuk menilai tingkat pentingnya suatu kata atau frasa dalam suatu dokumen atau kumpulan dokumen.
- **Term Frequency (TF)** menghitung seberapa sering kata tertentu muncul dalam sebuah dokumen. Jika kata tersebut muncul lebih sering, maka semakin tinggi nilai TF-nya.
- **Inverse Document Frequency (IDF)** menghitung seberapa umum kata tersebut di seluruh dokumen yang tersedia. Jika kata tersebut muncul di banyak dokumen, maka nilai IDF-nya akan menurun, karena kata tersebut dianggap kurang penting untuk mengidentifikasi dokumen tersebut

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

Term	Review 1	Review 2	Review 3	IDF	TF-IDF (Review 1)	TF-IDF (Review 2)	TF-IDF (Review 3)
This	1	1	1	0.00	0.000	0.000	0.000
movie	1	1	1	0.00	0.000	0.000	0.000
is	1	2	1	0.00	0.000	0.000	0.000
very	1	0	0	0.48	0.068	0.000	0.000
scary	1	1	0	0.18	0.025	0.022	0.000
and	1	1	1	0.00	0.000	0.000	0.000
long	1	0	0	0.48	0.068	0.000	0.000
not	0	1	0	0.48	0.000	0.060	0.000
slow	0	1	0	0.48	0.000	0.060	0.000
spooky	0	0	1	0.48	0.000	0.000	0.080
good	0	0	1	0.48	0.000	0.000	0.080

Word Embedding: Representasi Makna dalam Vektor

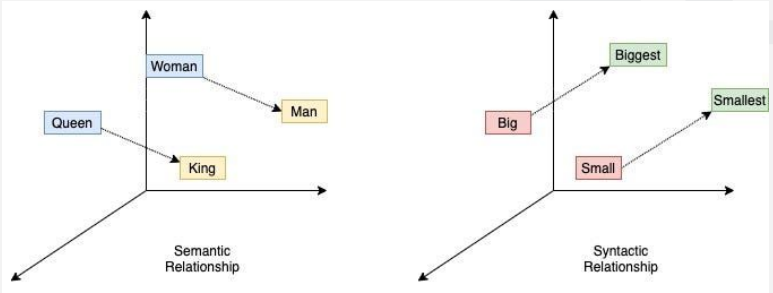
- Word Embedding adalah teknik dalam pemrosesan bahasa alami (NLP) yang mengonversi kata-kata menjadi representasi numerik (vektor) dalam ruang dimensi yang kontinu.

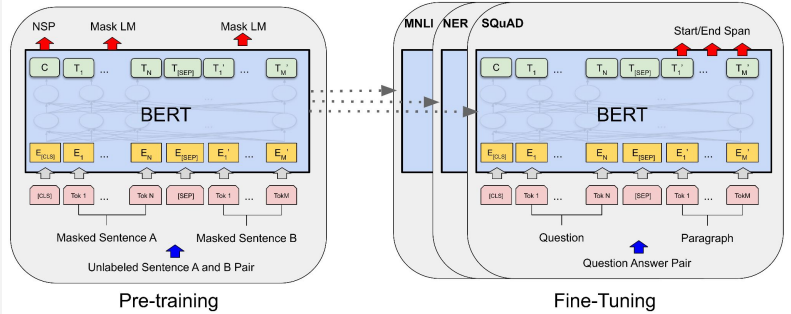
- Aplikasi Utama:
  - **Word2Vec**: Membuat vektor representasi untuk kata-kata dengan memperhitungkan konteks di sekitarnya. Berguna untuk memahami hubungan semantik antara kata-kata dalam teks.

  - **BERT** (Bidirectional Encoder Representations from Transformers): Menggunakan model transformer yang besar untuk mempelajari representasi kontekstual kata-kata dalam teks. Mampu memahami makna kata berdasarkan konteks globalnya.

- Keunggulan:
  - Memungkinkan komputer memahami dan memproses bahasa manusia.

  - Meningkatkan kinerja dalam berbagai tugas pemrosesan bahasa alami seperti klasifikasi teks, pemahaman bahasa, dan generasi teks.





Rangkuman dari Vektorisasi Teks

- Bag of Words hanya membuat kumpulan vektor yang berisi jumlah kemunculan kata dalam dokumen (ulasan), sedangkan model TF-IDF mengandung informasi tentang kata-kata yang lebih penting dan yang kurang penting juga.

- Vektorisasi Bag of Words mudah untuk diinterpretasikan. Namun, TF-IDF seringkali memberikan performa yang lebih baik dalam model pembelajaran mesin.

- Vektorisasi teks Word Embedding seperti Word2Vec atau BERT menggunakan pendekatan yang lebih canggih dalam mengubah teks menjadi representasi numerik. Word2Vec mempelajari representasi vektor untuk setiap kata dalam teks dengan memperhitungkan konteks di sekitarnya.

- BERT menggunakan model transformer yang sangat besar untuk mempelajari representasi kontekstual kata-kata dalam teks. Ini memungkinkan BERT untuk memahami makna kata berdasarkan konteks globalnya, serta mengatasi masalah ambiguitas kata dalam teks.

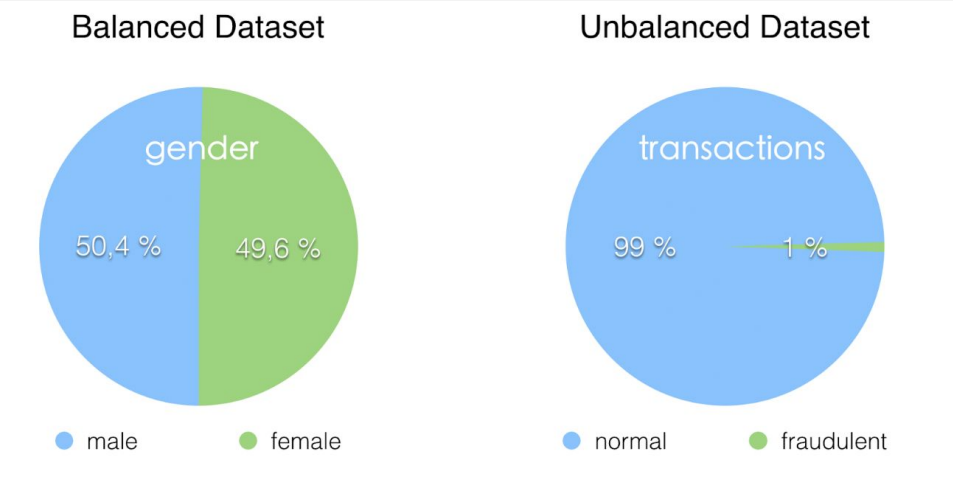
Penanganan Data dengan Proporsi Tak Seimbang (Data Imbalance Handling)

Kenapa dan Bagaimana

Penanganan Data Imbalance

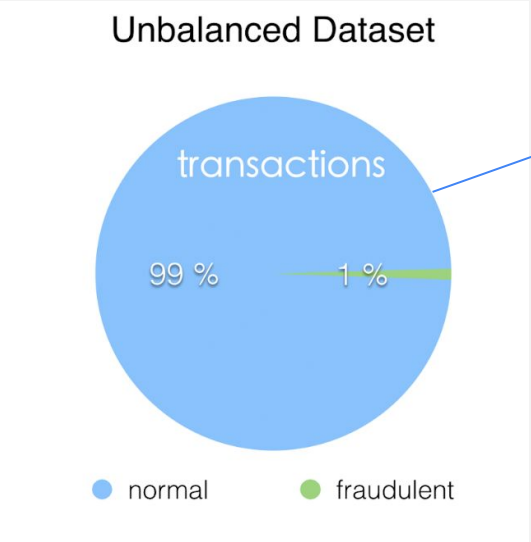
- **Data Imbalance** biasanya mencerminkan distribusi yang tidak seimbang dari kelas-kelas dalam sebuah dataset.

- Sebagai contoh, dalam dataset kartu kredit fraud detection, **sebagian besar transaksi kartu kredit bukan penipuan** dan hanya sedikit transaksi yang merupakan penipuan.



Kenapa Data Imbalance Perlu Ditangani?

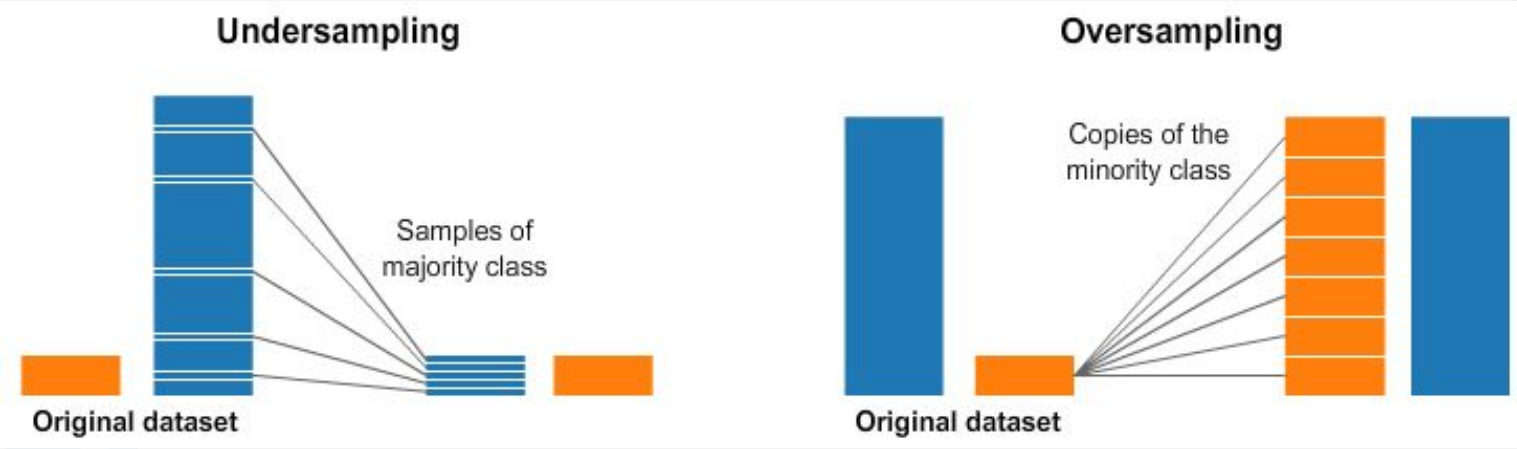
- Imbalanced data akan **membuat algoritma cenderung menghasilkan hasil yang baik dengan mengutamakan (bias) mayoritas**. Hal ini akan menjadi masalah jika Anda lebih tertarik pada pendeteksian kelas yang minoritas.



Just predict “Normal” 100% all the time and we will get 99% accuracy

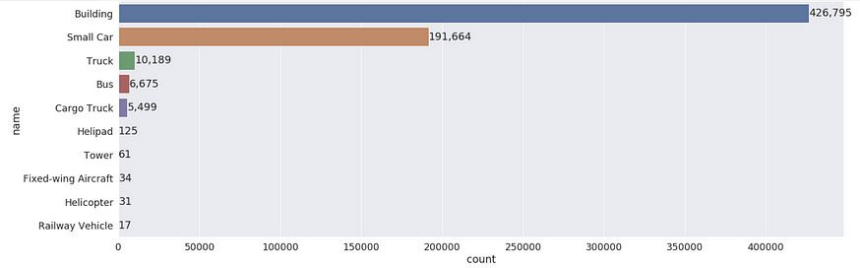
Penanganan Data Imbalance

- Secara umum, ada 2 cara untuk menangani dataset yang tidak seimbang, yaitu undersampling dan oversampling.



Oversampling using Data Augmentation

- Kerugian dari oversampling secara naif adalah menciptakan banyak titik data duplikat.
- Daripada membuat duplikat yang sama persis, tambahkan perubahan kecil pada titik data yang disalin.
- Pada Object Detection, Anda dapat dengan mudah memutar, membalik, dan meregangkan gambar masukan untuk mendapatkan gambar yang mirip namun berbeda.
- Pada data tabular, Anda bisa mempertimbangkan untuk menambahkan sedikit noise acak pada nilai-nilai sehingga sedikit berbeda dari nilai aslinya.



Original image

SMOTE

- SMOTE (Synthetic Minority Oversampling Technique) adalah algoritma yang melakukan augmentasi data dengan membuat titik data sintetis berdasarkan titik data asli.

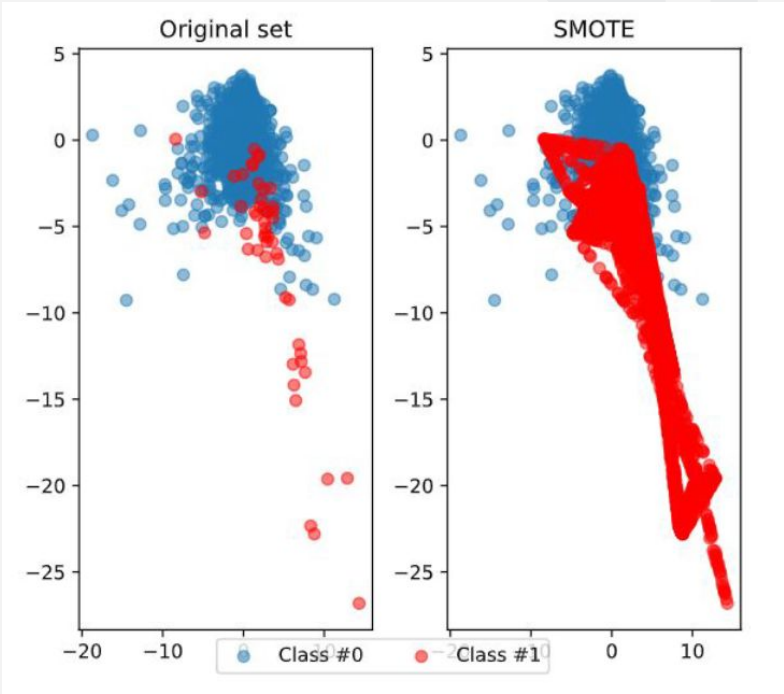
How SMOTE Algorithm Works?

1. You draw a random sample from the minority class.
2. For the observations in this sample, you will identify the k nearest neighbors.
3. You will then take one of those neighbors and identify the vector between the current data point and the selected neighbor.
4. You multiply the vector by a random number between 0 and 1.
5. To obtain the synthetic data point, you add this to the current data point.

- Operasi ini sebenarnya mirip dengan memindahkan sedikit titik data ke arah tetangganya.
- Dengan cara ini, memastikan bahwa titik data sintetis yang dihasilkan bukanlah salinan persis dari titik data yang ada, sekaligus memastikan bahwa juga tidak terlalu berbeda dari observasi yang diketahui dalam kelas minoritas yang ada.

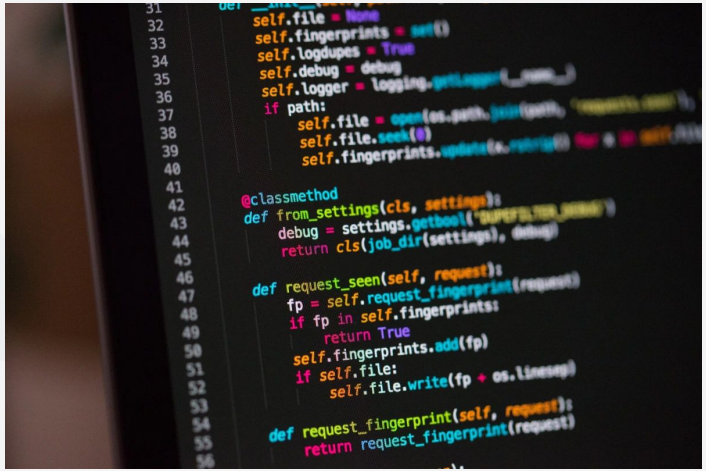
SMOTE

```
from imblearn.over_sampling import SMOTE

method = SMOTE()

# Create the resampled feature set
X_resampled, y_resampled = method.fit_resample(X, y)
plot_data(X_resampled, y_resampled)
```



Implementasi

- EDA, Preprocessing, API Testing
- Hands-on Coding in Google Colab



Thank You