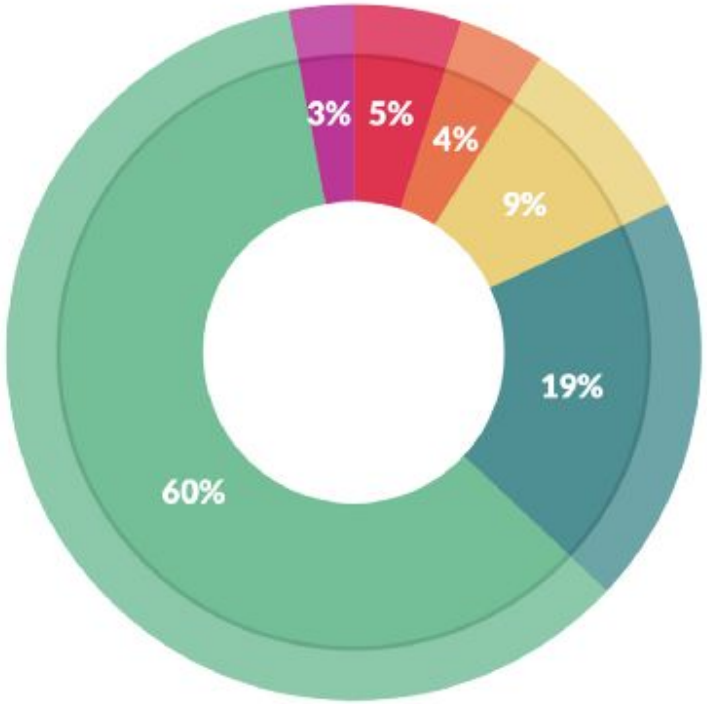
DigitalSkola

Data Preprocessing II

Data Science Time Spent

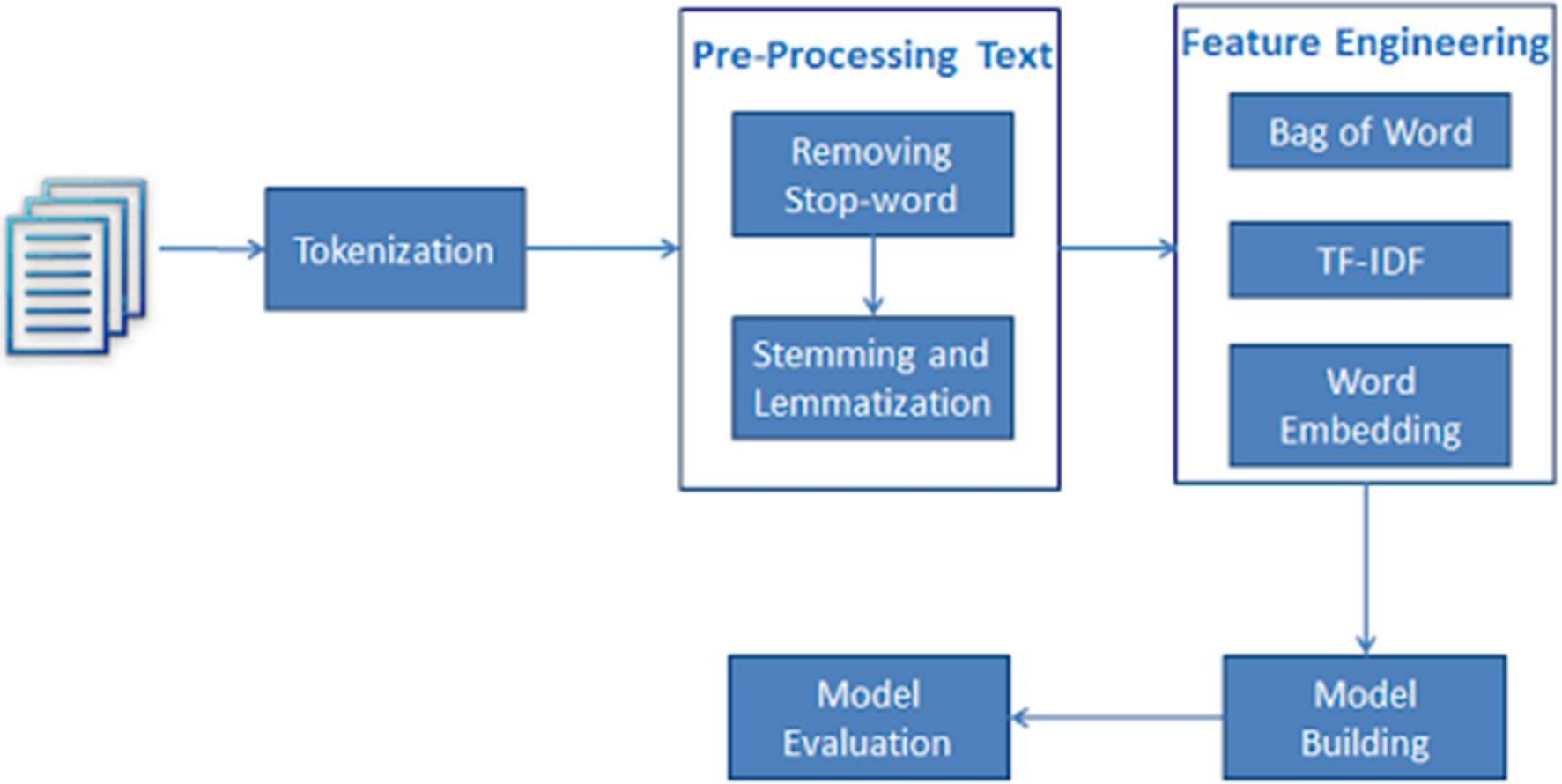


What data scientists spend the most time doing

- *Building training sets: 3%*
- *Cleaning and organizing data: 60%*
- *Collecting data sets; 19%*
- *Mining data for patterns: 9%*
- *Refining algorithms: 4%*
- *Other: 5%*

Text Data

Text Data Handling



Cleansing & Tokenization

1. Text Cleansing (Case Folding)

- ✓ Membersihkan format text dari special character (,.&^%$#?, dll.)
- ✓ Bisa dibantu dengan ***Regular Expression*** (kita tidak akan focus ke sini)

```
text = re.sub(r"[-()\"#/@;:<>{}-=~|.?,]", "", text)
```

2. Text Tokenization

- ✓ Menjadikan text menjadi potongan-potongan kata dalam bentuk list python.

```
from nltk.tokenize import word_tokenize

tokenized_word=word_tokenize(text)

print(tokenized_word)
```

```
['Hello', 'Mr.', 'Smith', ',', 'how', 'are', 'you', 'doing', 'today', '?', 'The',
```

Removing Stopwords

```
from nltk.corpus import stopwords
stop_words=set(stopwords.words("english"))
print(stop_words)
```

```
{'their', 'then', 'not', 'ma', 'here', 'other', 'won', 'up', 'weren', 'being', 'we',
```

```
filtered_sent=[]
for w in tokenized_sent:
    if w not in stop_words:
        filtered_sent.append(w)
print("Tokenized Sentence:",tokenized_sent)
print("Filterd Sentence:",filtered_sent)
```

```
Tokenized Sentence: ['Hello', 'Mr.', 'Smith', ',', 'how', 'are', 'you', 'doing', 'today', '?']
Filterd Sentence: ['Hello', 'Mr.', 'Smith', ',', 'today', '?']
```

3.**Removing Stopwords**

Kata sambung dan kata repetitive perlu kita buang agar tidak mendominasi

Stemming

```
# Stemming
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize


ps = PorterStemmer()


stemmed_words=[]
for w in filtered_sent:
    stemmed_words.append(ps.stem(w))


print("Filtered Sentence:",filtered_sent)
print("Stemmed Sentence:",stemmed_words)
```

4.**Stemming**

proses normalisasi linguistik, yang mereduksi kata menjadi kata dasar atau memotong imbuhan

Misalnya, Connect, Connecting, Connected, Connection direduksi menjadi kata umum “connect".

Lemmatization

```
from nltk.stem.wordnet import WordNetLemmatizer

lem = WordNetLemmatizer()


from nltk.stem.porter import PorterStemmer

stem = PorterStemmer()


word = "flying"

print("Lemmatized Word:",lem.lemmatize(word,"v"))

print("Stemmed Word:",stem.stem(word))
```

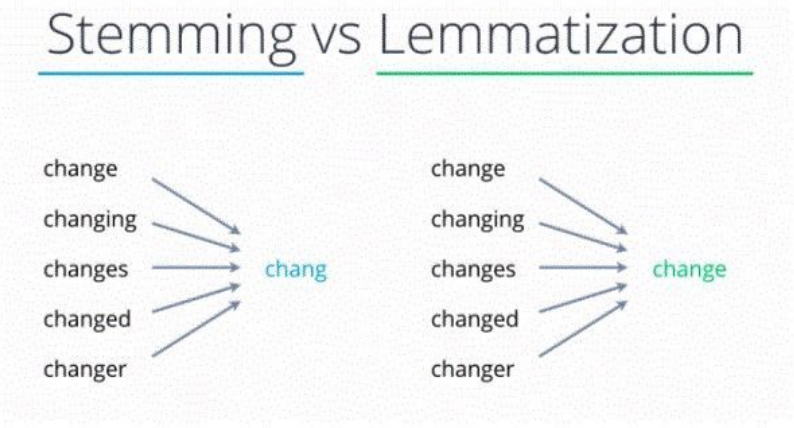
```
Lemmatized Word: fly

Stemmed Word: fli
```

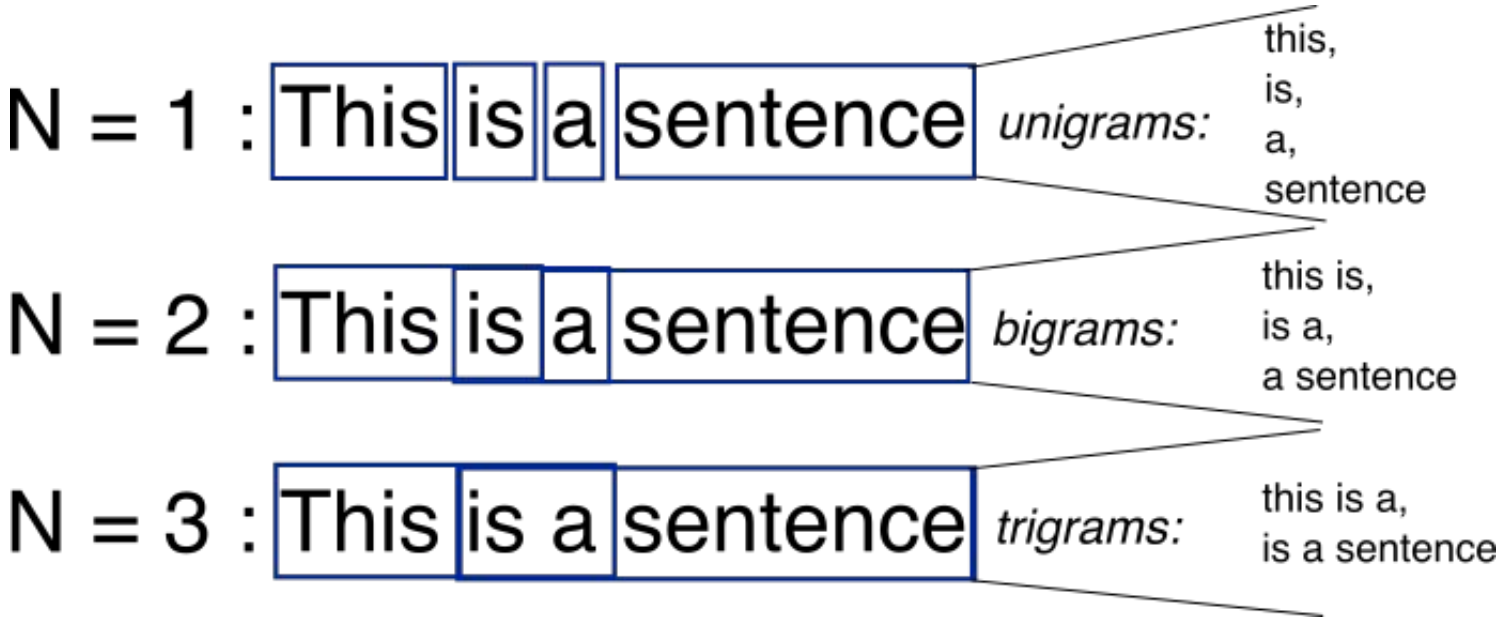
5.**Lemmatization**

mereduksi kata menjadi kata dasarnya, yaitu lemma yang benar secara linguistik. Ini mengubah kata dasar dengan penggunaan kosakata dan analisis morfologis.

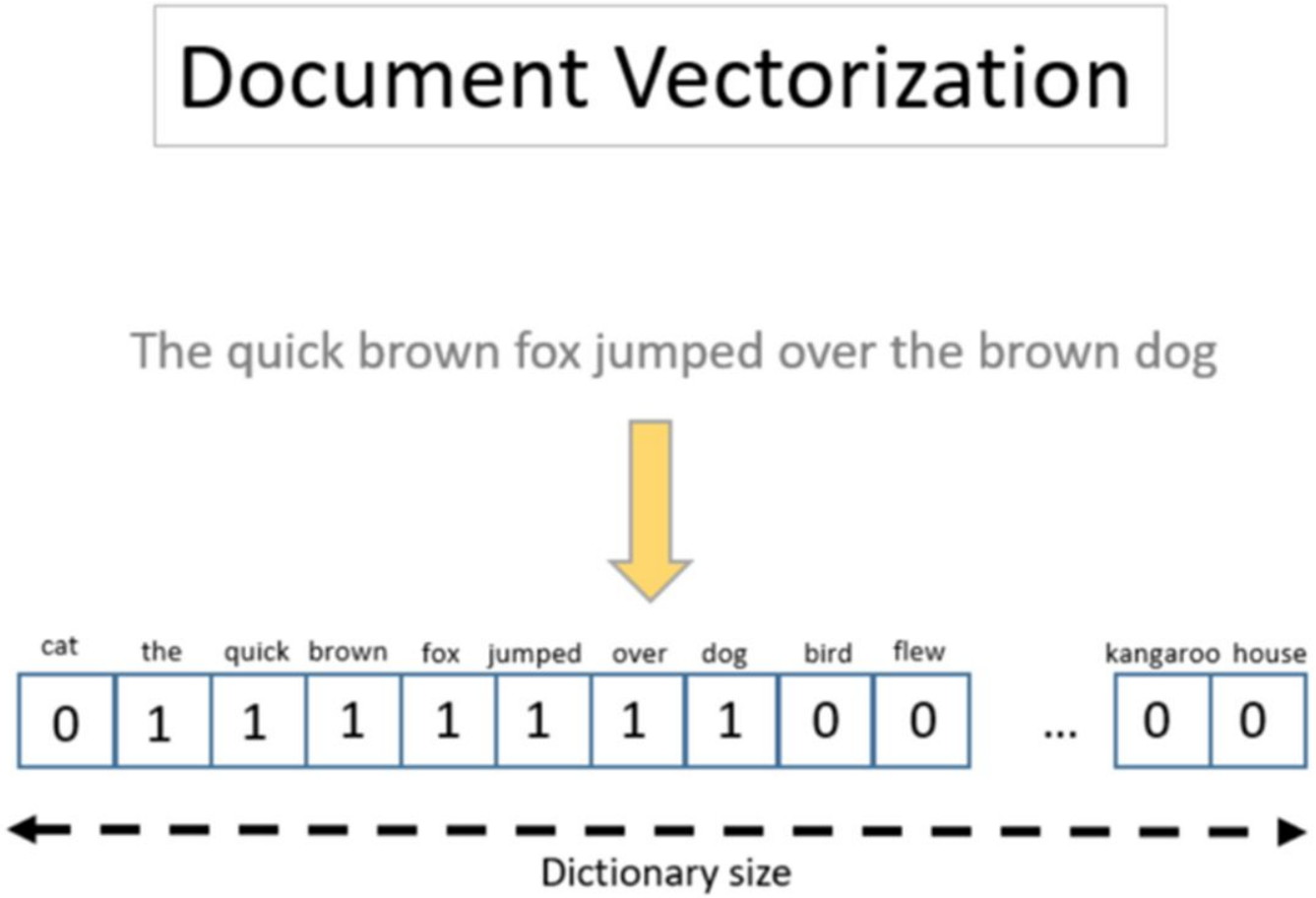
Contoh Better >> Good, Worst >> Bad



N-Gram



Vectorization



Bag of Words

Bag of Words (BoW)

Cara paling sederhana untuk mengekstrak fitur dari teks.
BoW mengubah teks menjadi matriks kemunculan kata-kata dalam dokumen.
Model ini memperhatikan apakah kata-kata yang diberikan muncul atau tidak dalam dokumen.

Example: There are three documents:
Doc 1: I love dogs. Doc 2: I hate dogs and knitting. Doc 3: Knitting is my hobby and my passion

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	1	1	1							
Doc 2	1		1	1	1	1				
Doc 3					1	1	1	2	1	1

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

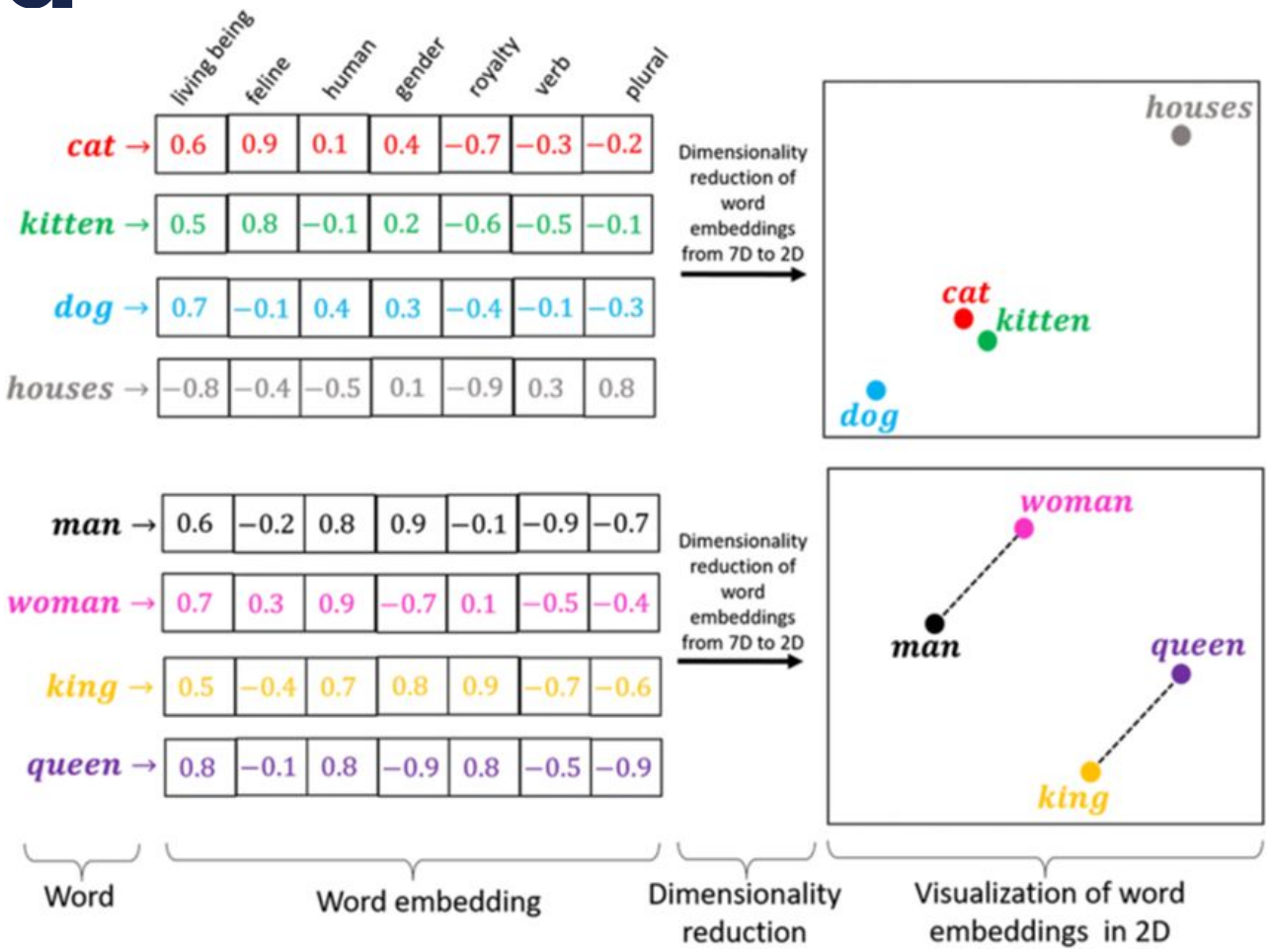
Term	Review 1	Review 2	Review 3	IDF	TF-IDF (Review 1)	TF-IDF (Review 2)	TF-IDF (Review 3)
This	1	1	1	0.00	0.000	0.000	0.000
movie	1	1	1	0.00	0.000	0.000	0.000
is	1	2	1	0.00	0.000	0.000	0.000
very	1	0	0	0.48	0.068	0.000	0.000
scary	1	1	0	0.18	0.025	0.022	0.000
and	1	1	1	0.00	0.000	0.000	0.000
long	1	0	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 Embeddina

Word embedding adalah sebuah teknik dalam pengolahan bahasa alami (NLP) yang digunakan untuk merepresentasikan kata-kata dalam bentuk vektor dalam ruang multidimensional yang berisi informasi tentang arti, makna, dan hubungan antar kata tersebut. Dalam word embedding, setiap kata direpresentasikan sebagai vektor numerik, dan kedekatan antara vektor-vektor ini merefleksikan kemiripan makna antara kata-kata tersebut.

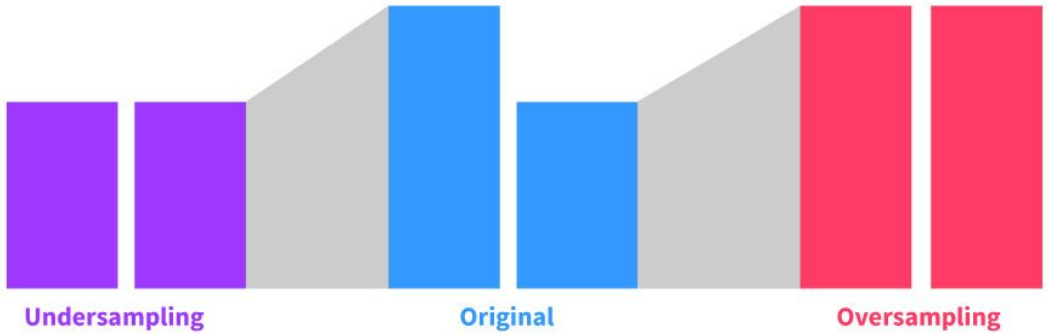
Word embedding banyak digunakan dalam aplikasi NLP seperti analisis teks, klasifikasi dokumen, pemodelan bahasa, dan mesin pencari. Teknik ini membantu model NLP untuk memahami makna dan konteks dari kata-kata dalam dokumen dan memperoleh representasi numerik dari teks yang dapat diolah oleh model machine learning.

Contoh teknik word embedding yang paling populer adalah Word2Vec dan GloVe. Word2Vec membangun representasi vektor untuk kata-kata dari distribusi kata-kata di lingkungan sekitarnya dalam korpus teks, sedangkan GloVe menggunakan matriks co-occurrence untuk menemukan representasi vektor untuk kata-kata.



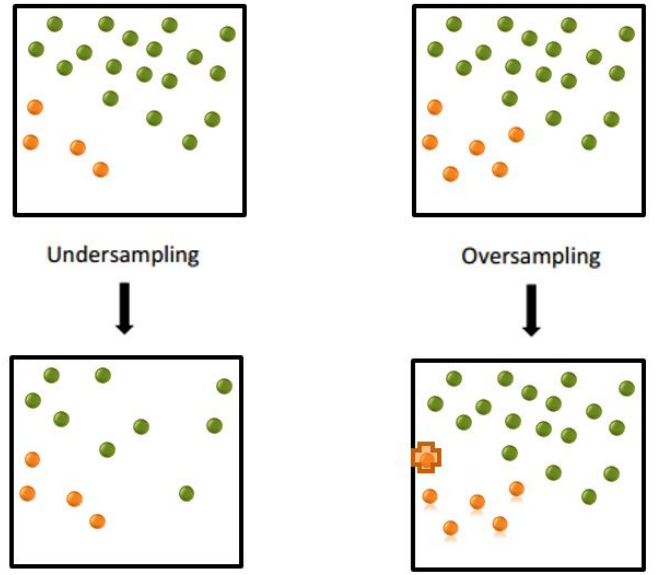
Imbalance Dataset

Oversampling and Undersampling



In undersampling, we pull all the rare events while pulling a sample of the abundant events in order to equalize the datasets.

These methods can be used separately or together;one is not better than the other. Which method a data scientist uses depends on the dataset and analysis.



SMOTE

SMOTE

- ✓ **Synthetic Minority Oversampling Technique (SMOTE)** adalah metode untuk melakukan oversampling.
- ✓ Pada dasarnya, SMOTE akan membuat data tambahan yang bersifat "sintetis" yang memiliki karakteristik yang mirip dengan data aslinya.
- ✓ SMOTE bekerja dengan bantuan algoritma KNN (K-Nearest Neighbor) –yang akan dijelaskan di materi Machine Learning Classification

Synthetic Minority Oversampling Technique