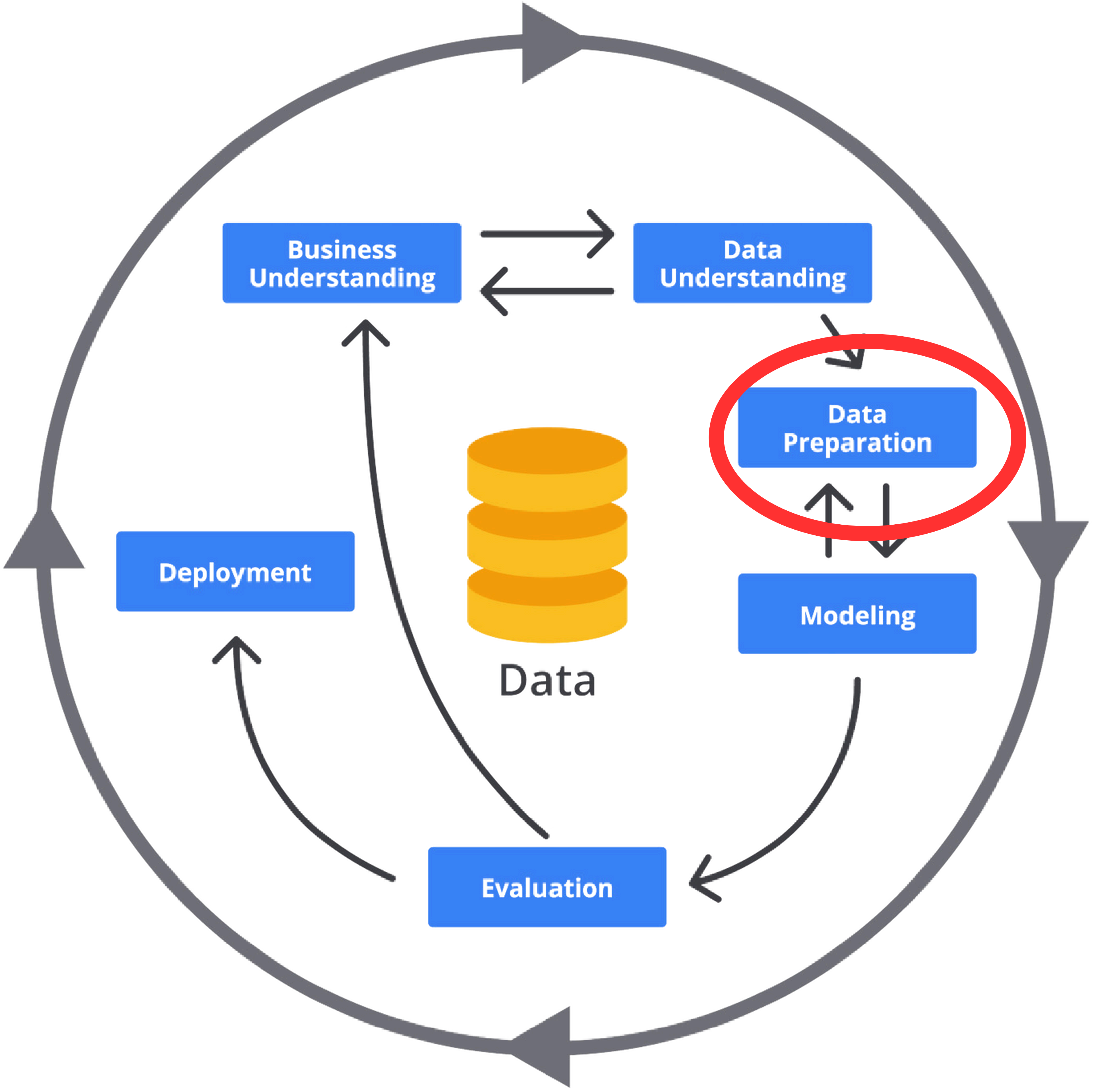
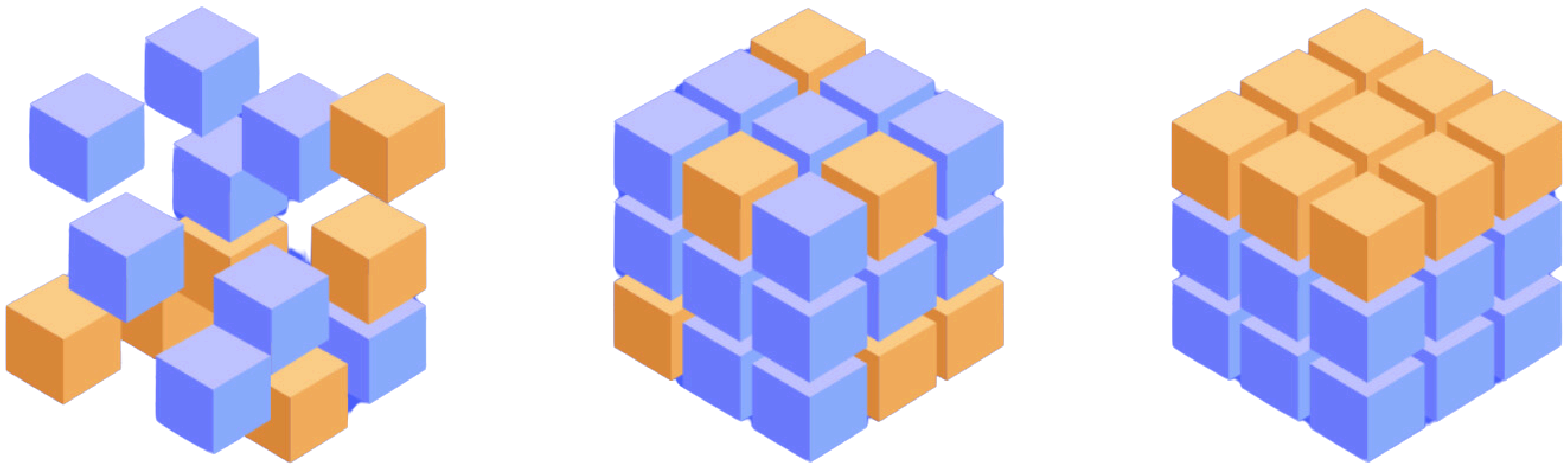
GWE #4 :
Data Preprocessing



Step ke 3 dari Crisp DM

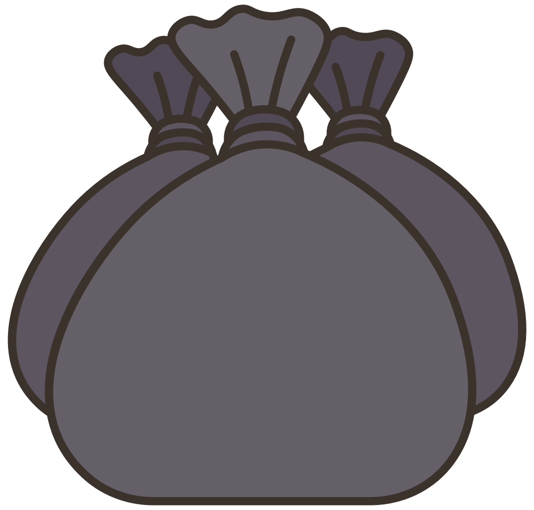
Apa itu data preprocessing ?



Data Preprocessing adalah proses **menyiapkan dan membersihkan data** sebelum dianalisis atau digunakan dalam pemodelan machine learning. Proses ini melibatkan **pembersihan, transformasi, dan pengorganisasian data** agar model dapat memahami pola dengan lebih baik.

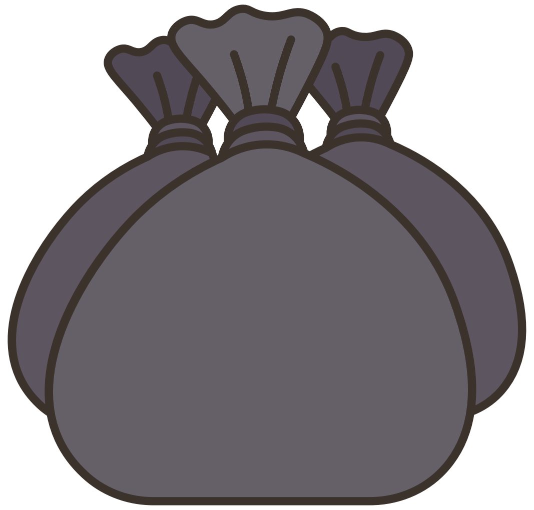
Kenapa Data Preprocessing penting ?

Input



Model

Output



- Model machine learning hanya sebaik kualitas datanya.
- Data mentah sering kali mengandung noise, duplikasi, dan nilai yang hilang.
- Preprocessing meningkatkan akurasi, efisiensi, dan keandalan model.

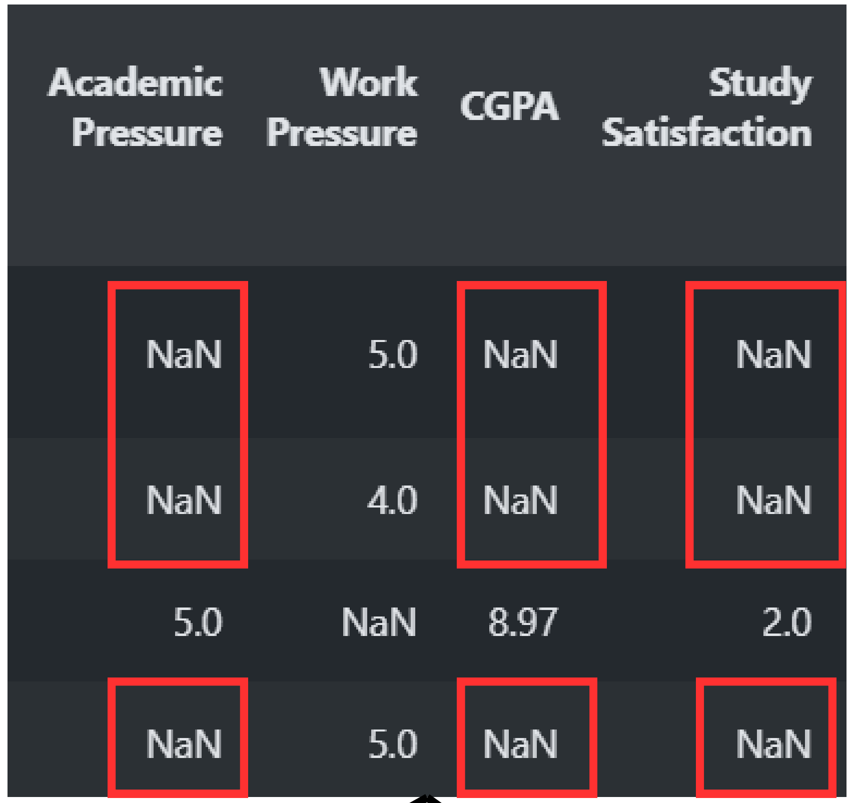
80% waktu seorang Data Scientist dihabiskan untuk membersihkan data, bukan membangun model.

Tahapan Data Preprocessing

- Data Cleaning
- Data Integration
- Data Transformation
- Data Reduction

Data Cleaning

Handling Missing Values



Academic Pressure	Work Pressure	CGPA	Study Satisfaction
NaN	5.0	NaN	NaN
NaN	4.0	NaN	NaN
5.0	NaN	8.97	2.0
NaN	5.0	NaN	NaN

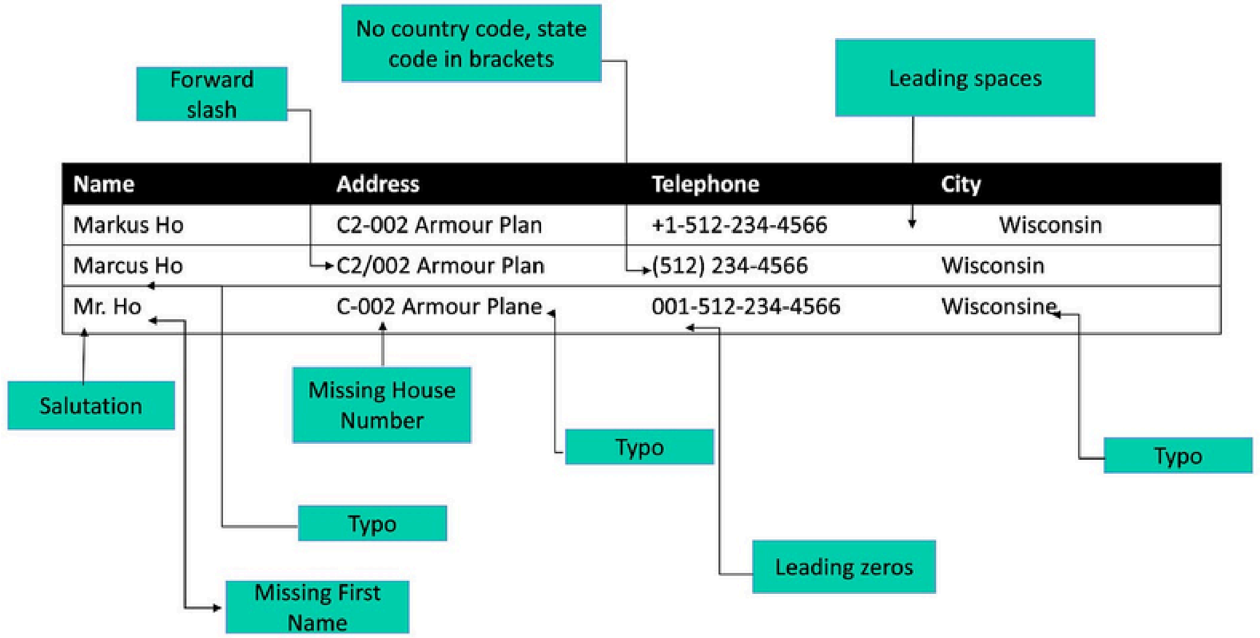
Drop Column

Imputasi

Statistical Method

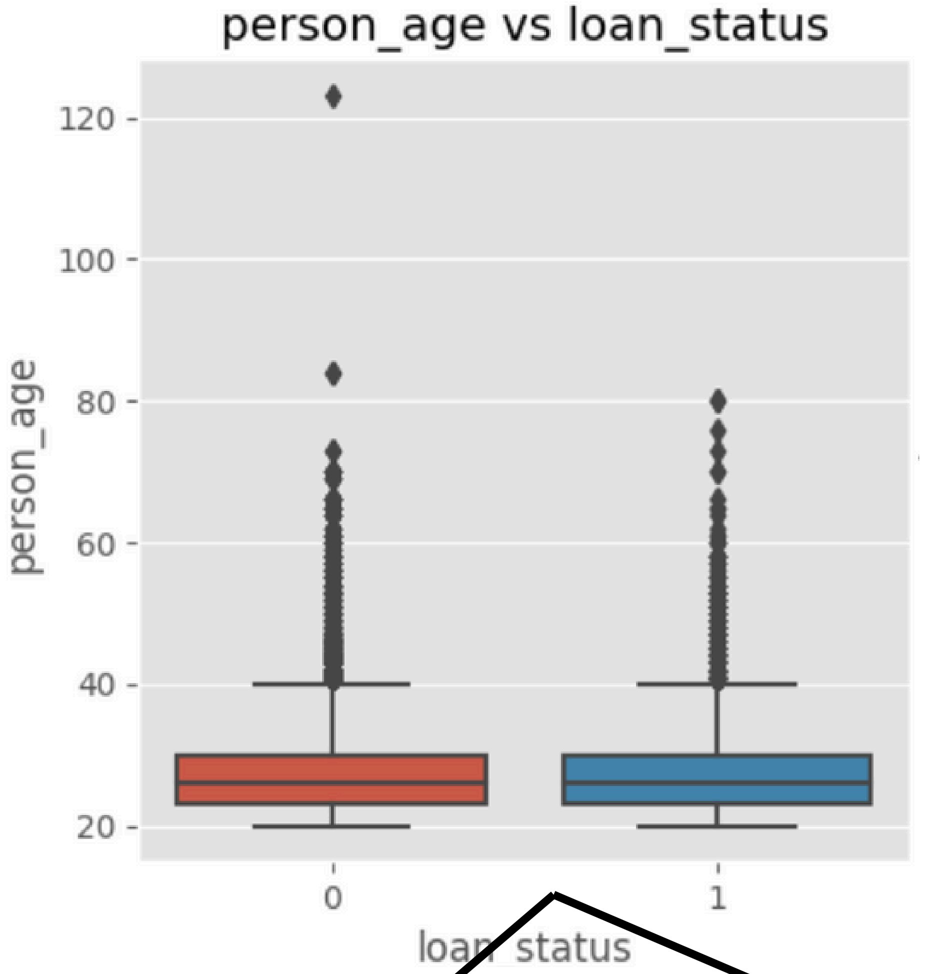
Using predictive model ex. KNNImputer

Handling Duplicate Values



Proses **mengidentifikasi**, **memperbaiki**, atau menghapus **kesalahan** dan **inkonsistensi** dalam dataset.

Handling Outliers



IQR Method

Z-Score Method

Data Cleaning

Code examples

Handling Missing Values

```
#Drop Missing Values
df = df.dropna()

#Fill Missing Values With Mean
df = df.fillna(df.mean())

#Fill Missing Values With Median
df = df.fillna(df.median())
```

Handling Duplicate Values

```
# Drop Duplicate Values

df = df.drop_duplicates()
```

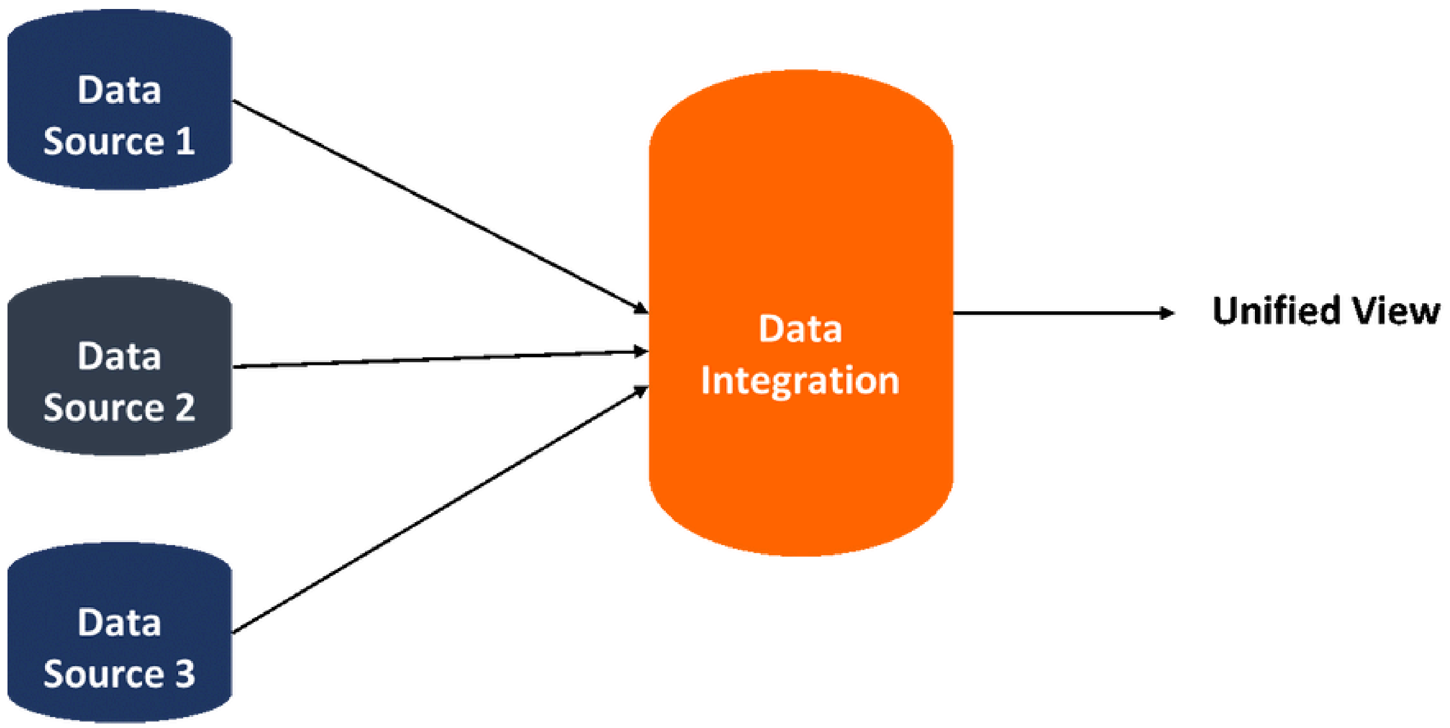
Handling Outliers

```
# Drop Outlier on column flipper_length_mm Using IQR

Q1 = df['flipper_length_mm'].quantile(0.25)
Q3 = df['flipper_length_mm'].quantile(0.75)
IQR = Q3 - Q1
df = df[~((df['flipper_length_mm'] < (Q1 - 1.5 * IQR)) | (df['flipper_length_mm'] > (Q3 + 1.5 * IQR)))]
```

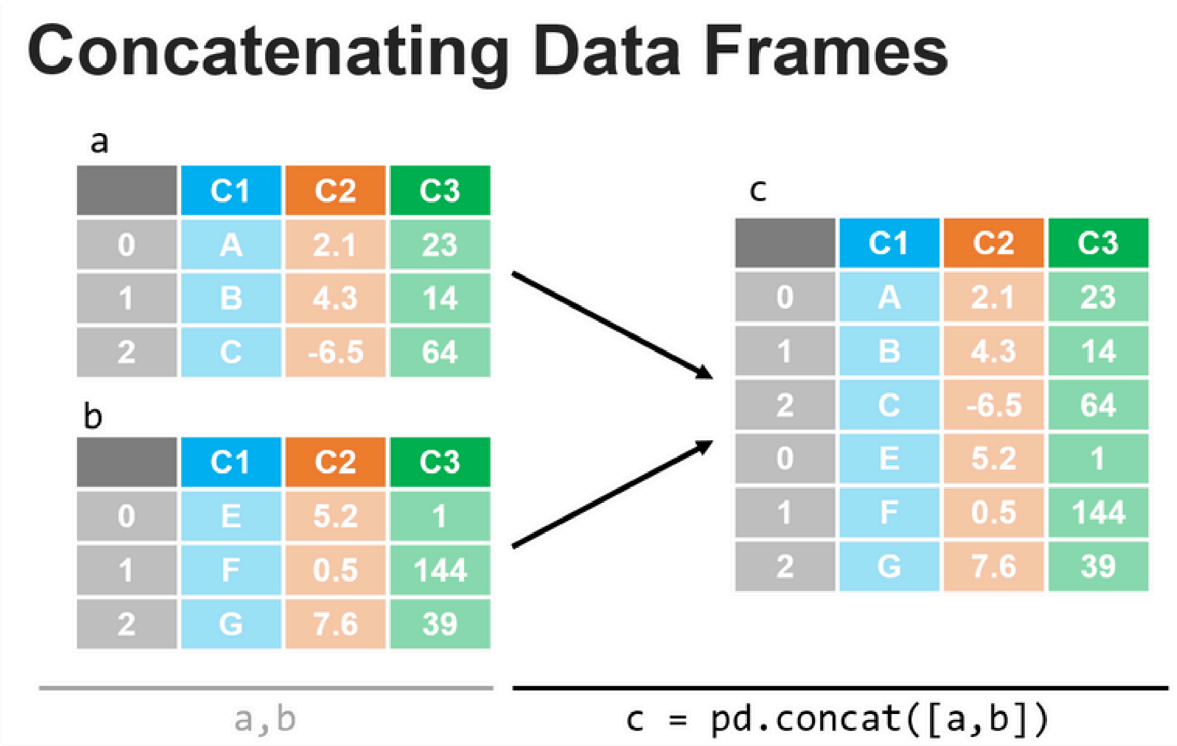
```
from scipy import stats

# Menghitung Z-Score
df['z_score'] = np.abs(stats.zscore(df['nilai']))

# Menentukan threshold (misal, Z > 3 dianggap outlier)
threshold = 3
df_cleaned = df[df['z_score'] <= threshold]
```

Data Integration

Merging Dataset



Concatenating Dataset



Data Integration adalah **proses menggabungkan data** dari berbagai sumber menjadi **satu kesatuan** yang konsisten dan dapat dianalisis secara efektif

Data Integration

Code Examples

Merging Dataset

```
# Dataset 1
df1 = pd.DataFrame({
    'id': [1, 2, 3],
    'name': ['Alice', 'Bob', 'Charlie'],
    'age': [25, 30, 35]
})

# Dataset 2
df2 = pd.DataFrame({
    'id': [1, 2, 4],
    'salary': [50000, 60000, 70000],
    'department': ['HR', 'IT', 'Finance']
})

# Merge the two datasets on 'id'
merged_df = pd.merge(df1, df2, on='id', how='outer')
# how itu ada outer, inner, left, right
```

Concatenating Dataset

```
# Dataset 1
df1 = pd.DataFrame({
    'id': [1, 2, 3],
    'name': ['Alice', 'Bob', 'Charlie'],
    'age': [25, 30, 35]
})

# Dataset 2
df2 = pd.DataFrame({
    'id': [4, 5, 6],
    'name': ['David', 'Eve', 'Frank'],
    'age': [28, 32, 40]
})

# Concatenate datasets
df_concat = pd.concat([df1, df2], ignore_index=True)
```

Data Transformation

Encoding Categorical Variable

Ordinal Encoding

Grades
A
B
C
D
Fail

Grades	Encoded
A	4
B	3
C	2
D	1
Fail	0

Label encoder digunakan ketika data memiliki hirarki/urutan

Index	Animal
0	Dog
1	Cat
2	Sheep
3	Horse
4	Lion

One-Hot code

Index	Dog	Cat	Sheep	Lion	Horse
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	0	0	1
4	0	0	0	1	0

Yang ini sebaliknya..

Feature Engineering

jadi_anggota	tahun	bulan	hari
2014-05-05	2014.0	5.0	5.0
2013-03-17	2013.0	3.0	17.0

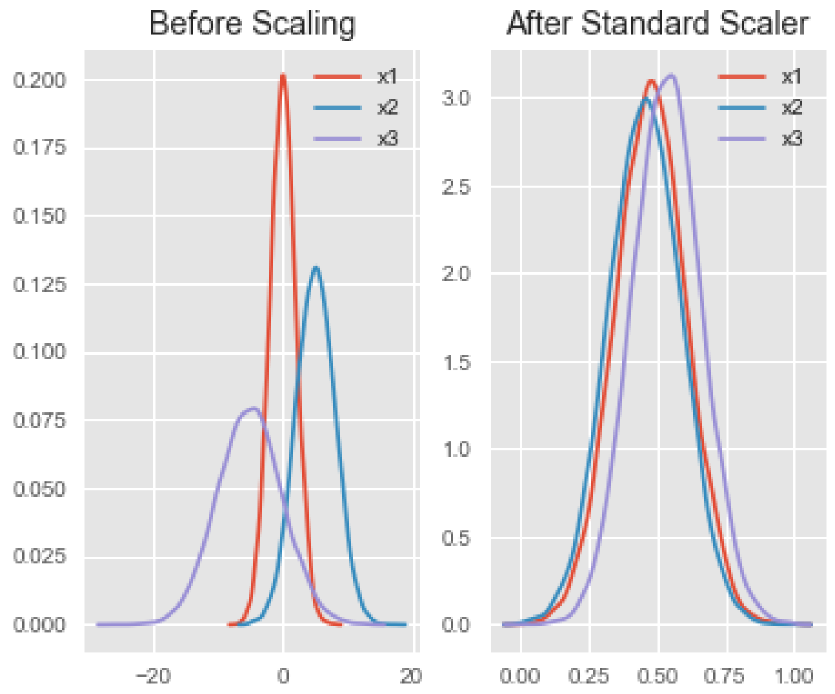
tahun_kelahiran	umur
1979	45
1950	74
1966	58
1961	63

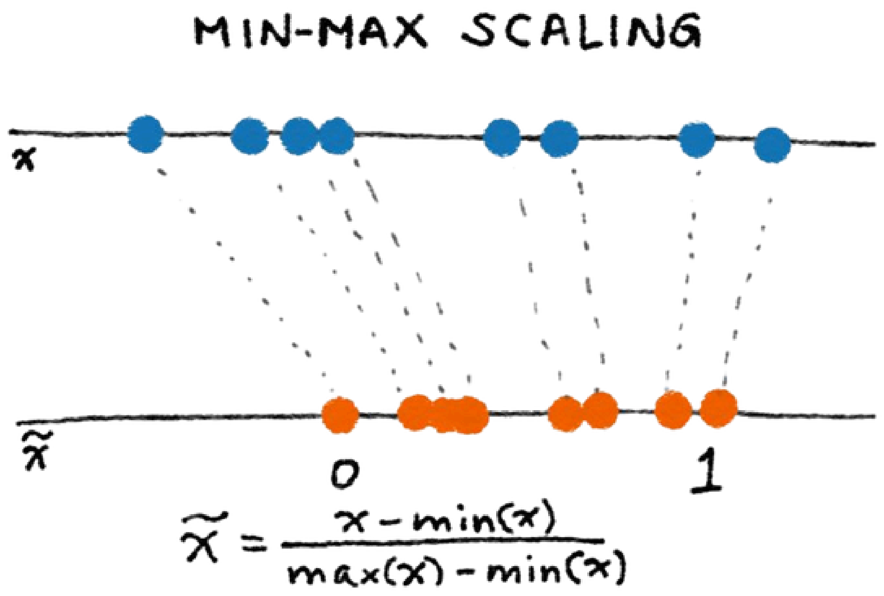
Yang ini dilakukan pake pemahaman dari business understanding

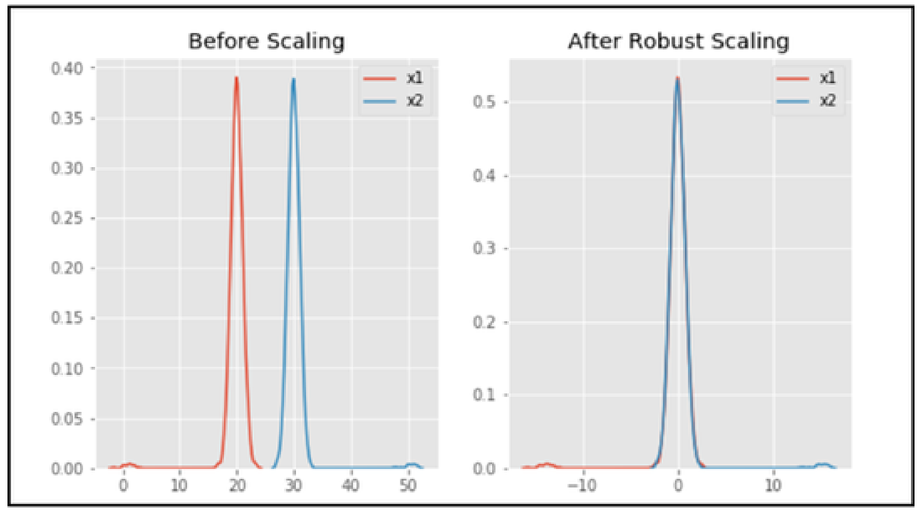
Proses mengubah data mentah menjadi **format yang lebih sesuai** untuk analisis atau model machine learning.

Data Transformation

Scaling







Standard Scaler:

- Deskripsi: Menskalakan data sehingga memiliki mean 0 dan standar deviasi 1.
- Penggunaan: Cocok untuk data dengan distribusi normal.
- Catatan: Sensitif terhadap outlier.

Min-Max Scaler:

- Deskripsi: Menskalakan data ke dalam rentang 0 hingga 1.
- Penggunaan: Digunakan saat ingin mempertahankan distribusi asli data.
- Catatan: Rentan terhadap outlier.

Robust Scaler:

- Deskripsi: Menskalakan data berdasarkan median dan interquartile range (IQR).
- Penggunaan: Ideal untuk data dengan outlier atau distribusi non-normal.
- Catatan: Lebih tahan terhadap outlier.

Data Transformation

Code Examples

Encoding Categorical Variable

```
#one hot encoding for sex and embarked columns
col = ['sex','embarked']
df_sample = pd.get_dummies(df_sample, columns=col)
```

```
#Label encoder for sex and embarked columns
from sklearn.preprocessing import LabelEncoder
col = ['sex','embarked']
le = LabelEncoder()
df_sample[col] = df_sample[col].apply(le.fit_transform)
```

Scaling

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
col = ['fare','age']
df_sample[col] = scaler.fit_transform(df_sample[col])
```

Feature Engineering

```
#tahun_kelahiran menjadi umur
train_features['umur'] = 2024 - train_features['tahun_kelahiran']
train_features[['tahun_kelahiran', 'umur']].head()
```

```
#split tanggal_menjadi_anggota menjadi tahun, bulan, dan hari
train_features['tanggal_menjadi_anggota'] = pd.to_datetime(train_features['tanggal_menjadi_anggota'])
train_features['tahun'] = train_features['tanggal_menjadi_anggota'].dt.year
train_features['bulan'] = train_features['tanggal_menjadi_anggota'].dt.month
train_features['hari'] = train_features['tanggal_menjadi_anggota'].dt.day
```

Data Reduction

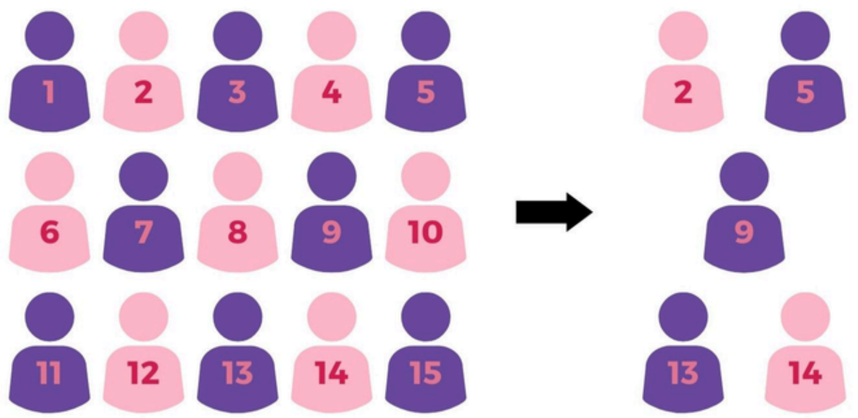
Feature selection

ID	Tahun Kelahiran	Kelas Pekerjaan	fnlwgt
~~478~~	1992	Swasta	~~37210~~
~~479~~	1981	Swasta	~~101950~~

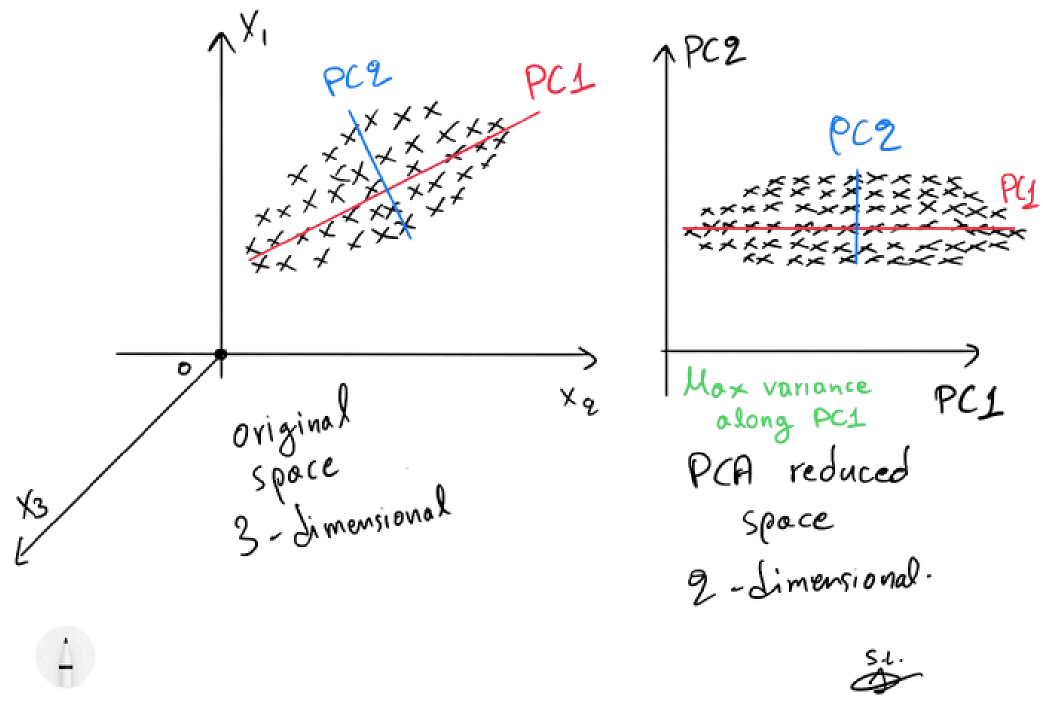
Features	Importance
capital	29.121907
Umur	13.670925
Hubungan_is not husband or wife	10.901339
Status_Not Married	10.228424
hours per week	9.188713

Feature Selection hanya ambil fitur yang dianggap memberikan value/penting aja. yang ga penting di buangggg

Sampling



Dimensionality Reduction



PCA (Principal Component Analysis)

Proses **mengurangi jumlah data** untuk **mempercepat pemrosesan** dan meningkatkan efisiensi tanpa mengorbankan kualitas informasi.

Data Reduction

Code Examples

Feature selection

```
df.drop(['tahun_kelahiran','pendidikan'],
axis=1, inplace=True)
```

Sampling

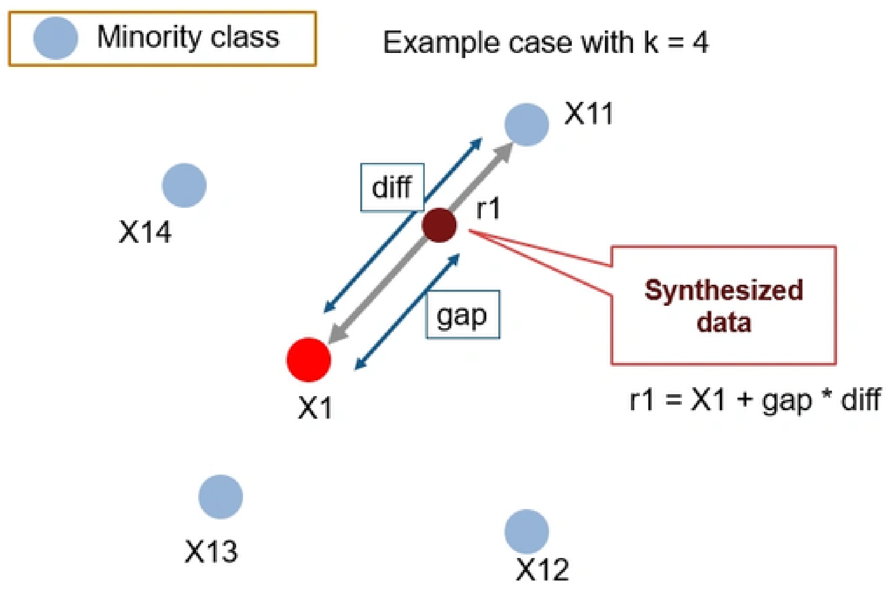
```
#sample 50% of the data
df = df.sample(frac=0.5)
```

Dimensionality Reduction

```
#pca 2 component
pca = PCA(n_components=2)
penguins_pca = pca.fit_transform(penguins_preprocessed)
penguins_pca = pd.DataFrame(data=penguins_pca, columns=["PC1", "PC2"])
```

Handling Imbalance Data

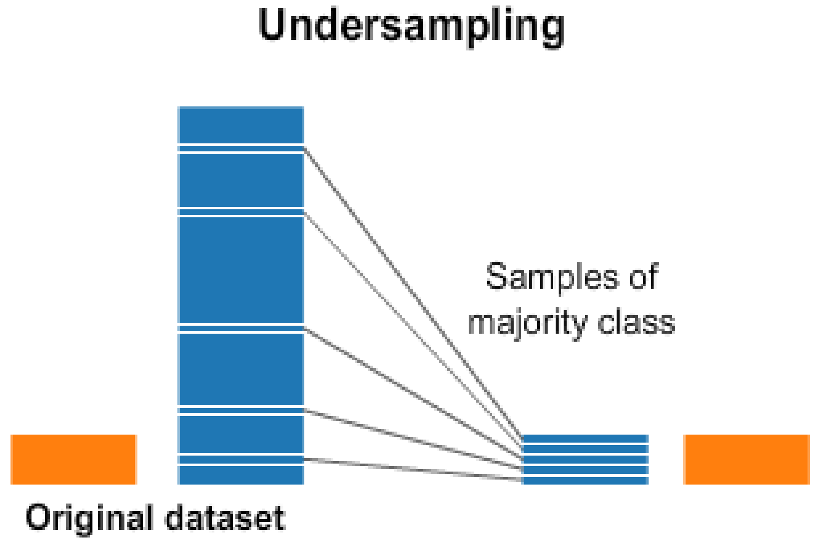
SMOTE (Synthetic Minority Over-sampling Technique).

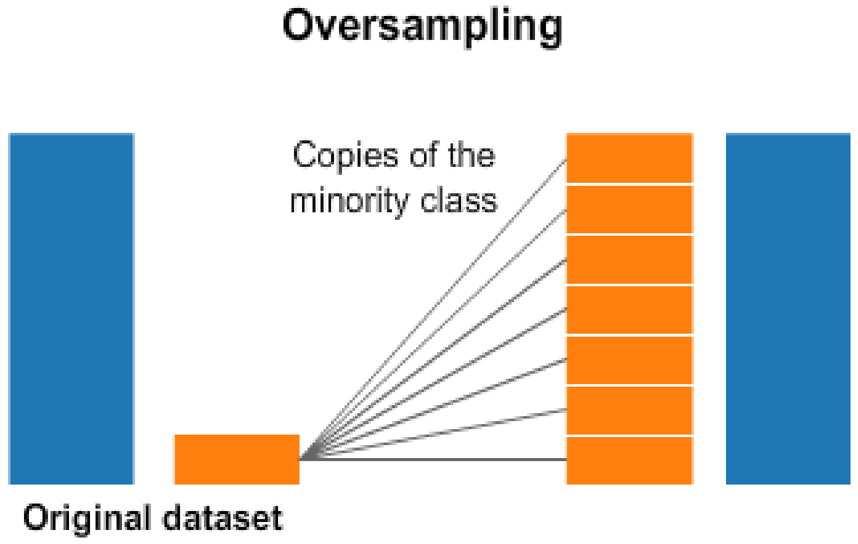


Penyesuaian Bobot (Cost-Sensitive Learning)

Decision Tree
Random Forest
Support Vector Machine (SVM)
Logistic Regression
Gradient Boosting Machines (GBM)
Neural Networks
etc

Random Oversampling / Undersampling





Handling Imbalance Target Variable

Code Examples

SMOTE (Synthetic Minority Over-sampling Technique).

```
from imblearn.over_sampling import SMOTE

smote = SMOTE(random_state=42)
X_smote, y_smote = smote.fit_resample(X_train, y_train)
```

Penyesuaian Bobot (Cost-Sensitive Learning)

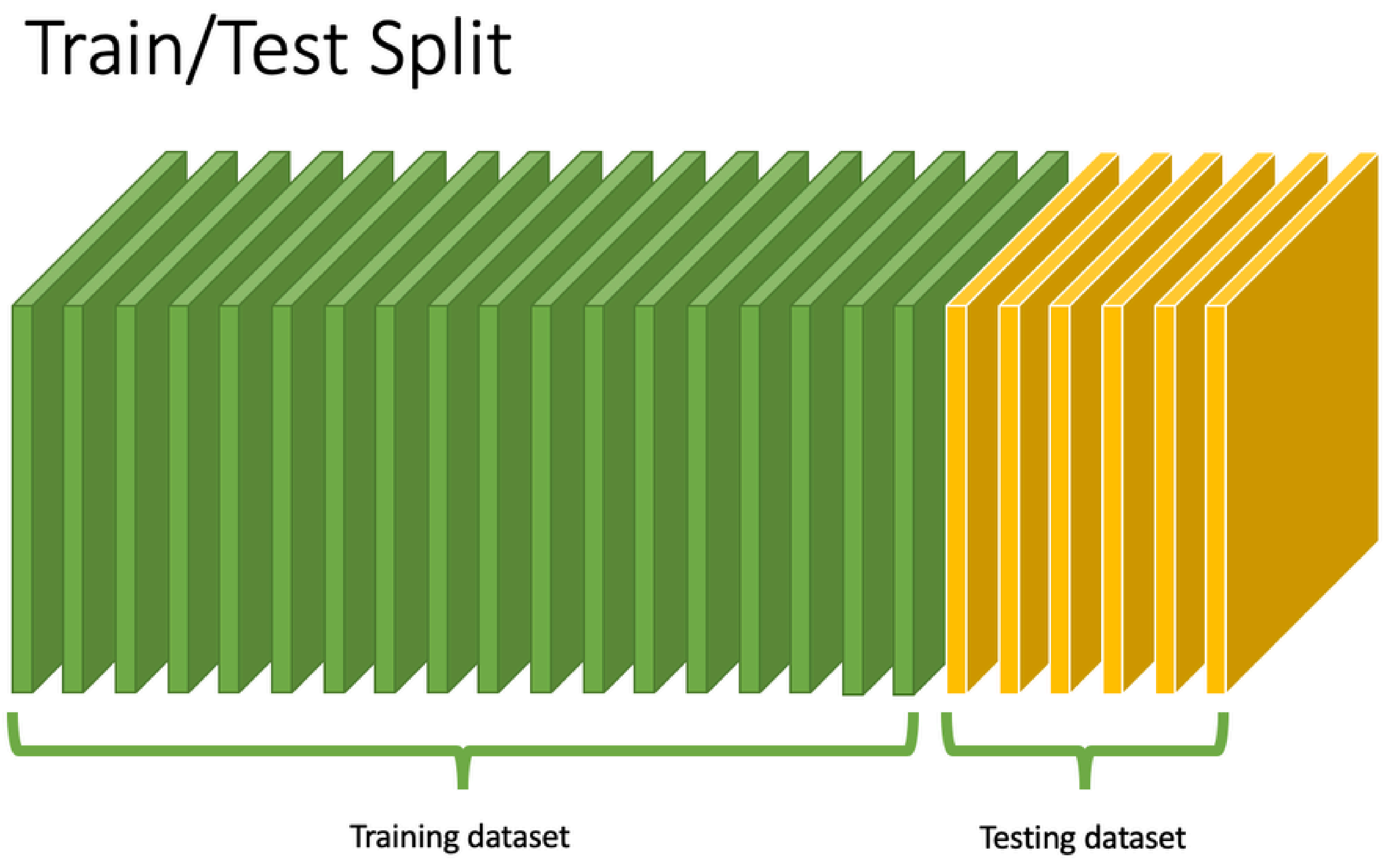
```
models = {
    'CatBoost': CatBoostClassifier(verbose=0, scale_pos_weight=1.2),
    'Random Forest': RandomForestClassifier(class_weight='balanced'),
    'XGBoost': XGBClassifier(scale_pos_weight=1.2),
    'LightGBM': LGBMClassifier(scale_pos_weight=1.2)
}
```

Random Oversampling / Undersampling

```
from imblearn.over_sampling import RandomOverSampler

# Inisialisasi RandomOverSampler
ros = RandomOverSampler(random_state=42)

# Melakukan oversampling pada training set
X_resampled, y_resampled = ros.fit_resample(X_train, y_train)
```

```
from imblearn.under_sampling import RandomUnderSampler

# Inisialisasi RandomUnderSampler
rus = RandomUnderSampler(random_state=42)

# Melakukan undersampling pada training set
X_resampled, y_resampled = rus.fit_resample(X_train, y_train)
```

Splitting Dataset



Tergantung data sizenya

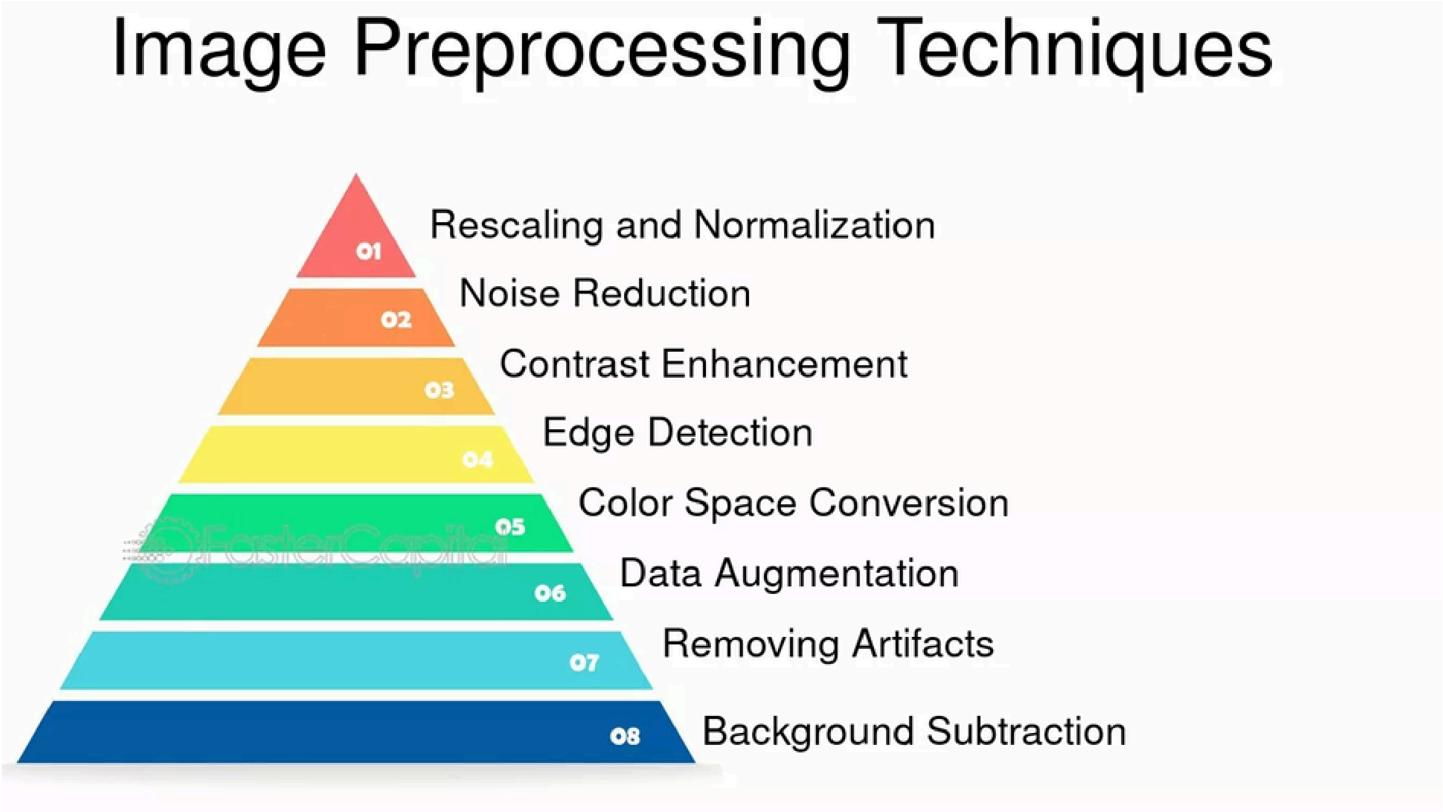
Ratio umum train-test split :

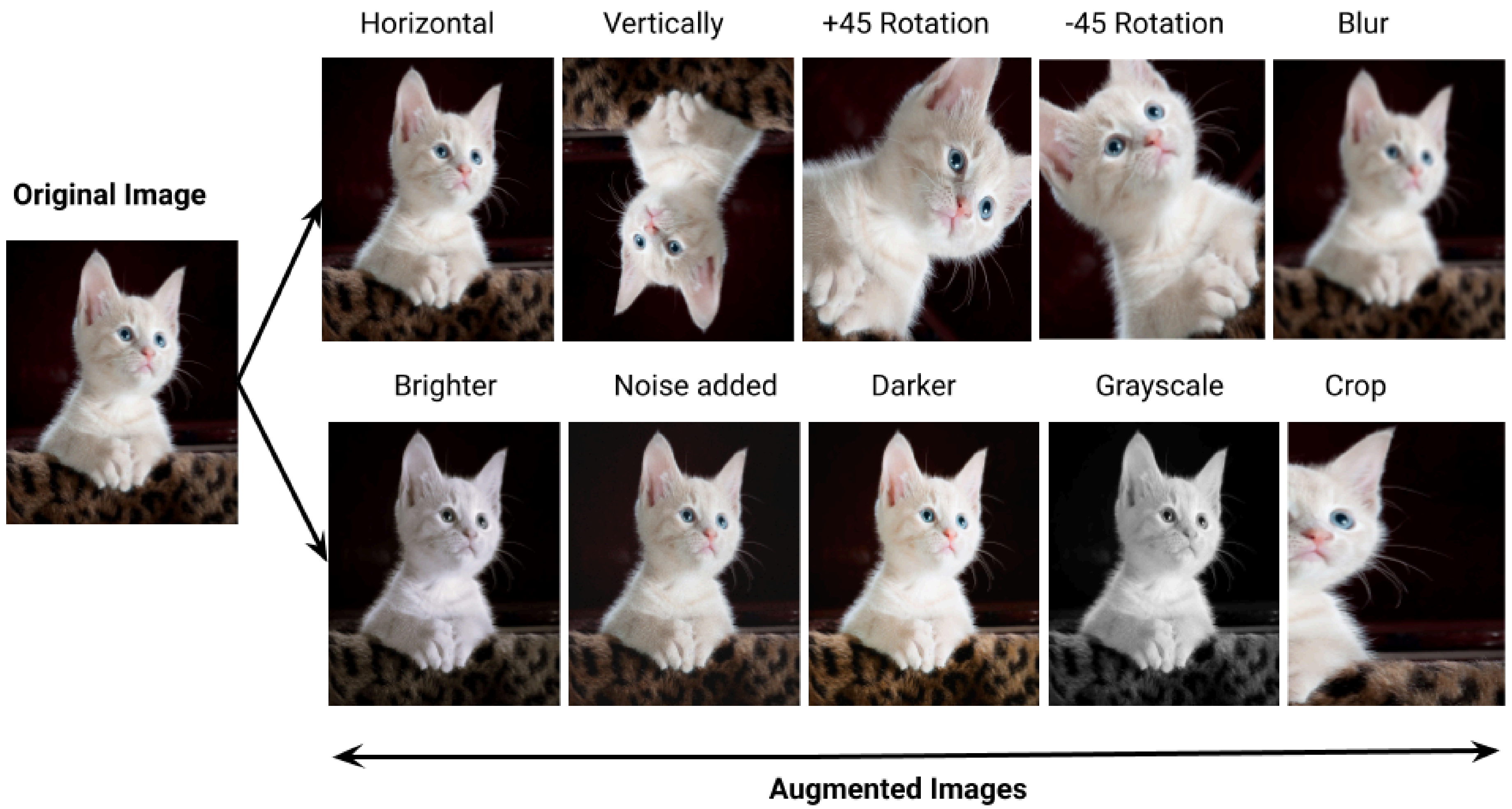
80 : 20

70 : 30

```python
from sklearn.model_selection import train_test_split
X = df.drop(columns='income')
y = df['income']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

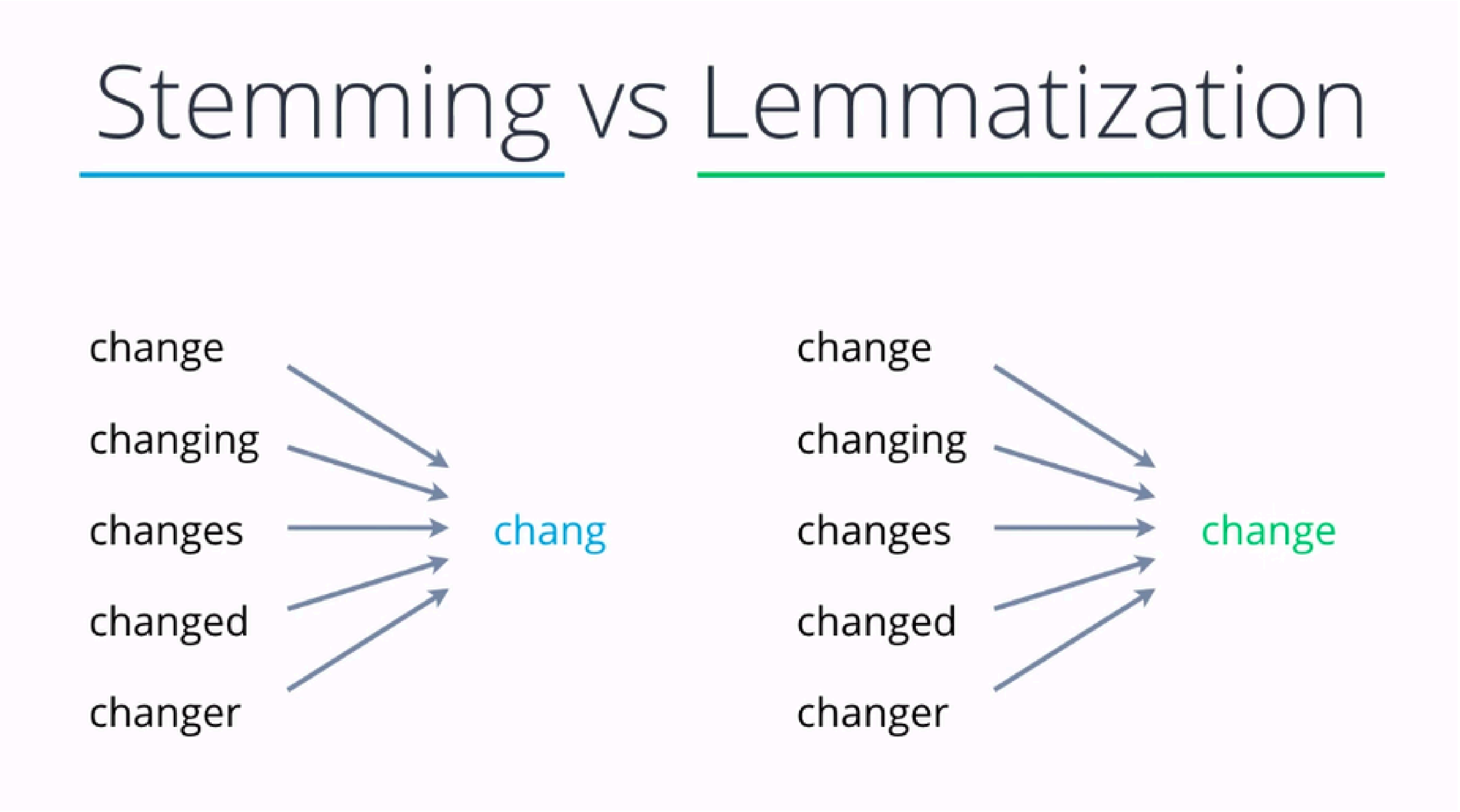
Overview preprocessing di data image



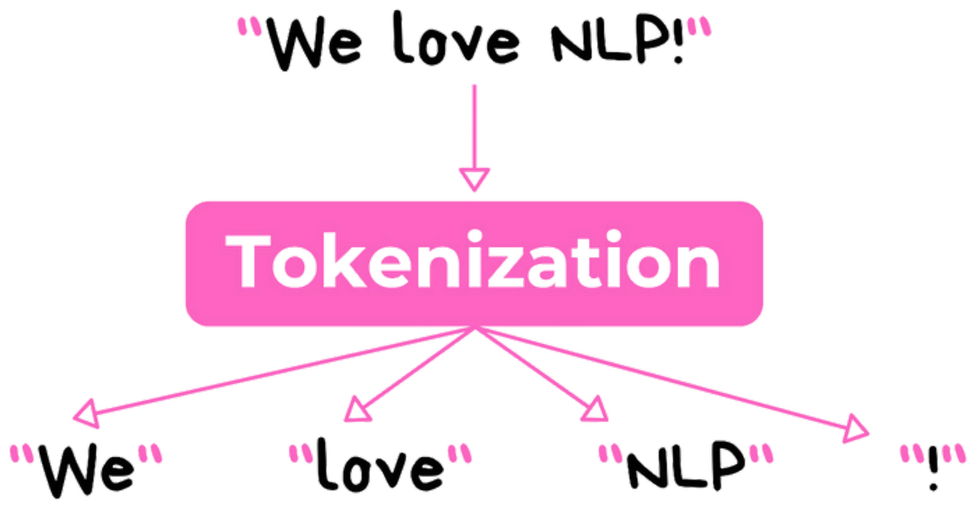


Dibahas detail di modul ke-7

Overview preprocessing di data text



Sample text with Stop Words	Without Stop Words
GeeksforGeeks – A Computer Science Portal for Geeks	GeeksforGeeks , Computer Science, Portal ,Geeks
Can listening be exhausting?	Listening, Exhausting
I like reading, so I read	Like, Reading, read



Dibahas detail di modul ke-8

ayo coba

Q & A

Quiz

Penugasan Modul 4

Absensi

https://tel-u.ac.id/presensigwe

Terima kasih !
See u on GWE #5