Rakamin Academy

peerData

Data Pre-processing

STAGE 2

Data Cleansing

Data Cleansing (Handle missing values)

```
[4]  1 # Missing values
     2 missing_values = df.isna().sum()
     3 missing_values
```

```
age          0
job          0
marital      0
education    0
default      0
balance      0
housing      0
loan         0
contact      0
day          0
month        0
duration     0
campaign     0
pdays        0
previous     0
poutcome     0
y            0
dtype: int64
```

Tidak ada missing values yang terdapat pada dataset, sehingga tidak perlu untuk melakukan tindak lebih lanjut.

Data Cleansing (Handle duplicated data)

```
[5]  1 # Menghitung jumlah data yang duplikat
     2 duplicate_count = df.duplicated().sum()
     3 print("Jumlah data yang duplikat:", duplicate_count)
```

```
Jumlah data yang duplikat: 0
```

Tidak ada data yang duplikat pada dataset, sehingga tidak perlu untuk melakukan tindak lebih lanjut.

-IQR METHOD-

Data Cleansing (Handle outliers)

```
print(f'Number of rows before removing outlier: {len(df)}')

filtered = np.array([True] * len(df))
for f in ['duration', 'campaign', 'pdays', 'previous']:
    Q1 = df[f].quantile(0.25)
    Q3 = df[f].quantile(0.75)
    iqr = Q3 - Q1
    b_thresh = Q1 - (1.5 * iqr)
    u_thresh = Q3 + (1.5 * iqr)

    filtered = ((df[f] >= b_thresh) & (df[f] <= u_thresh))
df = df[filtered]

print(f'Number of rows after removing outlier: {len(df)}')
```

```
Number of rows before removing outlier: 45211
Number of rows after removing outlier: 36954
```

Berdasarkan karakteristik dataset, untuk menghilangkan outlier kami memilih **metode IQR** yang **lebih relevan**. Alasannya karena pada dataset terdapat beberapa fitur memiliki distribusi yang sangat miring dengan ekor yang panjang.

Metode Z-score mengasumsikan distribusi normal, yang mungkin tidak sesuai untuk beberapa fitur. Metode IQR lebih tahan terhadap outlier dalam distribusi yang miring.

-Z-SCORE METHOD-

Data Cleansing (Handle outliers)

```
from scipy import stats
print(f'Number of rows before removing outlier: {len(df)}')

filtered = np.array([True] * len(df))
for f in ['day']:
    zscore = abs(stats.zscore(df[f]))
    filtered = (zscore < 3) & filtered

df = df[filtered]
print(f'Number of rows after removing outlier: {len(df)}')
```

```
Number of rows before removing outlier: 36954
Number of rows after removing outlier: 36954
```

Berdasarkan karakteristik dataset, untuk menghilangkan outlier kami memilih **metode Z-SCORE** yang **lebih relevan**. Z-score lebih relevan pada distribusi normal karena distribusi normal memberikan dasar statistik yang lebih jelas dan interpretabil untuk mengukur posisi relatif dan probabilitas suatu titik data dalam distribusi.

Data Cleansing (Feature transformation)

age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
53	services	divorced	primary	no	-291	yes	yes	unknown	7	may	591	1	-1	0	unknown	yes
49	services	married	secondary	no	-8	yes	no	unknown	8	may	1119	1	-1	0	unknown	yes
43	blue-collar	married	primary	no	-192	yes	no	unknown	8	may	1120	2	-1	0	unknown	yes
32	blue-collar	married	secondary	yes	-1	yes	no	unknown	9	may	653	1	-1	0	unknown	yes
28	blue-collar	single	secondary	no	-197	yes	no	unknown	9	may	2016	2	-1	0	unknown	yes

Pemutusan untuk mengubah nilai **negatif** pada saldo menjadi nilai **absolut** dikarenakan terdapat beberapa pelanggan dengan nilai negatif yang memiliki hasil **target (y)** yang bernilai **'yes'**. Dalam hal ini mungkin terjadi **kesalahan input** dikarenakan pelanggan dengan saldo negatif tidak mungkin atau tidak memenuhi syarat untuk membuka term deposit.

Before

	age	balance	day	duration	campaign	pdays	previous
count	36954.000000	36954.000000	36954.000000	36954.000000	36954.000000	36954.0	36954.0
mean	40.932430	1318.788846	16.145424	257.726119	2.921957	-1.0	0.0
std	10.430218	3039.557077	8.372554	262.256406	3.325791	0.0	0.0
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.0	0.0
25%	33.000000	55.000000	9.000000	101.000000	1.000000	-1.0	0.0
50%	39.000000	414.000000	17.000000	177.000000	2.000000	-1.0	0.0
75%	49.000000	1358.000000	22.000000	318.000000	3.000000	-1.0	0.0
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	-1.0	0.0

After

	age	balance	day	duration	campaign	pdays	previous
count	36954.000000	36954.000000	36954.000000	36954.000000	36954.000000	36954.0	36954.0
mean	40.932430	1375.862451	16.145424	257.726119	2.921957	-1.0	0.0
std	10.430218	3014.151556	8.372554	262.256406	3.325791	0.0	0.0
min	18.000000	0.000000	1.000000	0.000000	1.000000	-1.0	0.0
25%	33.000000	123.000000	9.000000	101.000000	1.000000	-1.0	0.0
50%	39.000000	457.000000	17.000000	177.000000	2.000000	-1.0	0.0
75%	49.000000	1373.000000	22.000000	318.000000	3.000000	-1.0	0.0
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	-1.0	0.0

Data Cleansing (Feature encoding)

Label Encode Variable Target (y)

```
[ ] #dataframe before label encoding 'y'
    df.tail()
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
45203	23	student	single	tertiary	no	113	no	no	cellular	17	nov	266	1	-1	0	unknown	yes
45205	25	technician	single	secondary	no	505	no	yes	cellular	17	nov	386	2	-1	0	unknown	yes
45206	51	technician	married	tertiary	no	825	no	no	cellular	17	nov	977	3	-1	0	unknown	yes
45207	71	retired	divorced	primary	no	1729	no	no	cellular	17	nov	456	2	-1	0	unknown	yes
45209	57	blue-collar	married	secondary	no	668	no	no	telephone	17	nov	508	4	-1	0	unknown	no

```
from sklearn.preprocessing import LabelEncoder

# Initialize LabelEncoder
encoder = LabelEncoder()

# Fit and transform the 'y' column
df['y'] = encoder.fit_transform(df['y'])

# Print the encoded DataFrame
df.tail()
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
45203	23	student	single	tertiary	no	113	no	no	cellular	17	nov	266	1	-1	0	unknown	1
45205	25	technician	single	secondary	no	505	no	yes	cellular	17	nov	386	2	-1	0	unknown	1
45206	51	technician	married	tertiary	no	825	no	no	cellular	17	nov	977	3	-1	0	unknown	1
45207	71	retired	divorced	primary	no	1729	no	no	cellular	17	nov	456	2	-1	0	unknown	1
45209	57	blue-collar	married	secondary	no	668	no	no	telephone	17	nov	508	4	-1	0	unknown	0

Untuk feature ‘y’, kami melakukan label encoding karena feature tersebut akan kami jadikan target, dimana untuk “yes” di encode 1 dan “no” di encode 0.

Data Cleansing (Feature encoding)

▾ One-Hot-Encoding for 3 debt predictor (default, housing, loan)

```
[ ] #OHE default, housing, loan
    encodee1 = ['default', 'housing', 'loan']

    for e in encodee1:
        ohe= pd.get_dummies(df[e], prefix=e)
        df = df.join(ohe)

    df = df.drop(encodee1, axis=1)
```

```
[ ] df.tail()
```

	age	job	marital	education	balance	contact	day	month	duration	campaign	pdays	previous	poutcome	y	default_no	default_yes	housing_no	housing_yes	loan_no	lo
45203	23	student	single	tertiary	113	cellular	17	nov	266	1	-1	0	unknown	1	1	0	1	0	1	
45205	25	technician	single	secondary	505	cellular	17	nov	386	2	-1	0	unknown	1	1	0	1	0	0	
45206	51	technician	married	tertiary	825	cellular	17	nov	977	3	-1	0	unknown	1	1	0	1	0	1	
45207	71	retired	divorced	primary	1729	cellular	17	nov	456	2	-1	0	unknown	1	1	0	1	0	1	
45209	57	blue-collar	married	secondary	668	telephone	17	nov	508	4	-1	0	unknown	0	1	0	1	0	1	

Data Cleansing (Feature encoding)

OHE for 2 demographic predictors (education, marital)

```
[ ] #OHE education, marital
    encodee1 = ['education', 'marital']

    for e in encodee1:
        ohe= pd.get_dummies(df[e], prefix=e)
        df = df.join(ohe)

    df = df.drop(encodee1, axis=1)
```

```
[ ] df.head()
```

	age	job	balance	contact	day	month	duration	campaign	pdays	previous	...	housing_yes	loan_no	loan_yes	education_primary	education_secondary	education_tertiary
0	58	management	2143	unknown	5	may	261	1	-1	0	...	1	1	0	0	0	1
1	44	technician	29	unknown	5	may	151	1	-1	0	...	1	1	0	0	1	0
2	33	entrepreneur	2	unknown	5	may	76	1	-1	0	...	1	0	1	0	1	0
3	47	blue-collar	1506	unknown	5	may	92	1	-1	0	...	1	1	0	0	0	0
4	33	unknown	1	unknown	5	may	198	1	-1	0	...	0	1	0	0	0	0

5 rows × 25 columns

Data Cleansing (Feature encoding)

OHE for 2 campaign method related (contact, poutcome)

```
[ ] #OHE contact, poutcome
    encodee2 = ['contact', 'poutcome']

    for e in encodee2:
        ohe= pd.get_dummies(df[e], prefix=e)
        df = df.join(ohe)

    df = df.drop(encodee2, axis=1)
```

```
[ ] df.describe()
```

	age	balance	day	duration	campaign	pdays	previous	y	default_no	default_yes	...	education_secondary	education_tertiary	ed
count	36954.000000	36954.000000	36954.000000	36954.000000	36954.000000	36954.0	36954.0	36954.000000	36954.000000	36954.000000	...	36954.000000	36954.000000	
mean	40.932430	1375.862451	16.145424	257.726119	2.921957	-1.0	0.0	0.091573	0.979488	0.020512	...	0.512746	0.287953	
std	10.430218	3014.151556	8.372554	262.256406	3.325791	0.0	0.0	0.288427	0.141746	0.141746	...	0.499844	0.452815	
min	18.000000	0.000000	1.000000	0.000000	1.000000	-1.0	0.0	0.000000	0.000000	0.000000	...	0.000000	0.000000	
25%	33.000000	123.000000	9.000000	101.000000	1.000000	-1.0	0.0	0.000000	1.000000	0.000000	...	0.000000	0.000000	
50%	39.000000	457.000000	17.000000	177.000000	2.000000	-1.0	0.0	0.000000	1.000000	0.000000	...	1.000000	0.000000	
75%	49.000000	1373.000000	22.000000	318.000000	3.000000	-1.0	0.0	0.000000	1.000000	0.000000	...	1.000000	1.000000	
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	-1.0	0.0	1.000000	1.000000	1.000000	...	1.000000	1.000000	

8 rows × 25 columns

```
[ ] #OHE contact, poutcome
    encodee3 = ['job']

    for e in encodee3:
        ohe= pd.get_dummies(df[e], prefix=e)
        df = df.join(ohe)

    df = df.drop(encodee3, axis=1)
```

Data Cleansing (Feature encoding)

OHE for 2 campaign method related (contact, poutcome)

```
[ ] #OHE contact, poutcome
    encodee2 = ['contact', 'poutcome']

    for e in encodee2:
        ohe= pd.get_dummies(df[e], prefix=e)
        df = df.join(ohe)

    df = df.drop(encodee2, axis=1)
```

```
[ ] df.describe()
```

	age	balance	day	duration	campaign	pdays	previous	y	default_no	default_yes	...	education_secondary	education_tertiary	ed
count	36954.000000	36954.000000	36954.000000	36954.000000	36954.000000	36954.0	36954.0	36954.000000	36954.000000	36954.000000	...	36954.000000	36954.000000	
mean	40.932430	1375.862451	16.145424	257.726119	2.921957	-1.0	0.0	0.091573	0.979488	0.020512	...	0.512746	0.287953	
std	10.430218	3014.151556	8.372554	262.256406	3.325791	0.0	0.0	0.288427	0.141746	0.141746	...	0.499844	0.452815	
min	18.000000	0.000000	1.000000	0.000000	1.000000	-1.0	0.0	0.000000	0.000000	0.000000	...	0.000000	0.000000	
25%	33.000000	123.000000	9.000000	101.000000	1.000000	-1.0	0.0	0.000000	1.000000	0.000000	...	0.000000	0.000000	
50%	39.000000	457.000000	17.000000	177.000000	2.000000	-1.0	0.0	0.000000	1.000000	0.000000	...	1.000000	0.000000	
75%	49.000000	1373.000000	22.000000	318.000000	3.000000	-1.0	0.0	0.000000	1.000000	0.000000	...	1.000000	1.000000	
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	-1.0	0.0	1.000000	1.000000	1.000000	...	1.000000	1.000000	

8 rows × 25 columns

```
[ ] #OHE contact, poutcome
    encodee3 = ['job']

    for e in encodee3:
        ohe= pd.get_dummies(df[e], prefix=e)
        df = df.join(ohe)

    df = df.drop(encodee3, axis=1)
```

Data Cleansing (Feature encoding)

Untuk fitur housing, loan, default, marital, job, education, contact, outcome kami lakukan One-Hot-Encoding. Ini juga berlaku terhadap fitur Duration, balance dan month yang merupakan hasil dari feature yang telah di ekstraksi. OHE digunakan agar tidak ada mispersepsi pada variabel kategorikal khususnya untuk variabel yang nilainya tidak ter inherit dengan nilai ordinal.

Data Cleansing (Handle class imbalance)

```
[ ]  # x nya adalah atribut yang mempengaruhi target
     # y nya adalah target itu sendiri
     X = df_new.drop(['y'], axis = 1)
     y = df_new['y']

     print(x.shape)
     print(y.shape)

     (36951, 33)
     (36951,)
```

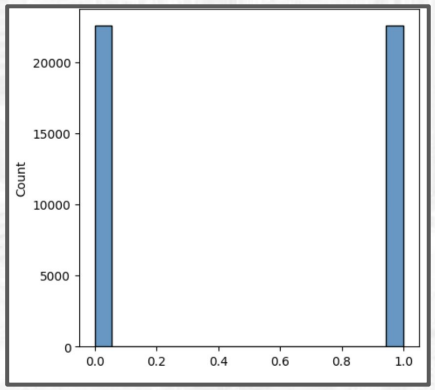
Menentukan variabel dataset

```
[ ]  from sklearn.model_selection import train_test_split
     from imblearn.over_sampling import SMOTE
     from collections import Counter

     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

     # Apply SMOTE to the training data
     smote = SMOTE(random_state=42)
     X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)

     # Print class distribution before and after applying SMOTE
     print("Class distribution before SMOTE:", Counter(y_train))
     print("Class distribution after SMOTE:", Counter(y_train_resampled))

     Class distribution before SMOTE: Counter({0: 26846, 1: 2714})
     Class distribution after SMOTE: Counter({0: 26846, 1: 26846})
```

Pertama, kita mengimpor perpustakaan yang diperlukan untuk menggunakan SMOTE dan memeriksa distribusi kelas. Kemudian, kita memuat dataset dan membaginya menjadi subset train dan test. Penerapan SMOTE pada data train menggunakan metode fit_resample.

```
[ ]  print(pd.Series(y_train_resampled).value_counts())

     0    26846
     1    26846
     Name: y, dtype: int64
```

Hasil akhir menunjukkan jumlah yang sama antara kedua kelas, yaitu ‘yes’ dan ‘no’ dengan jumlah yang sama yaitu 26846. Jumlah total sampel dapat meningkat secara signifikan, karena SMOTE menghasilkan sampel sintetik dengan melakukan interpolasi antar sampel yang ada di kelas minoritas, dan dengan melakukan hal tersebut, berpotensi membuat sampel baru dalam jumlah besar.



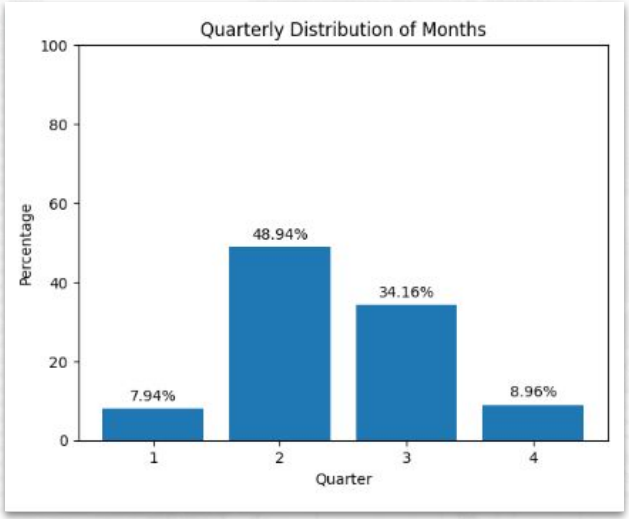
Hasil akhir menunjukkan jumlah yang seimbang

Feature Engineering

Feature Engineering (Feature extraction)

month_quartal

```
#Months to Quarter
def map_to_quartal(month):
    quarters = {'jan': 1, 'feb': 1, 'mar': 1,
                'apr': 2, 'may': 2, 'jun': 2,
                'jul': 3, 'aug': 3, 'sep': 3,
                'oct': 4, 'nov': 4, 'dec': 4}
    return quarters[month]

# Apply the function to create the new column
df['month_quartal'] = df['month'].apply(map_to_quartal)

print(df)
```

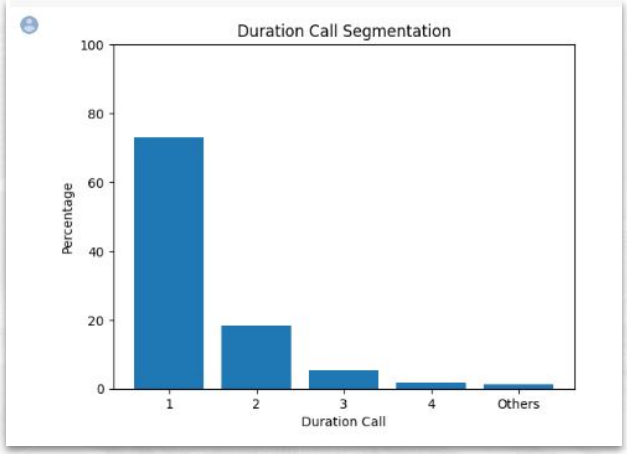


Fitur month_quartal diekstraksi dari fitur month, dengan pertimbangan untuk melihat banyaknya nasabah yang dihubungi pada quarter tertentu. Hasilnya, quarter terbanyak jatuh pada quarter 2, yang berarti nasabah paling banyak dihubungi pada bulan April, Mei, Juni.

Feature Engineering (Feature extraction)

duration_bin

```
[ ]  # Make segment for duration feature
     # Define the bin edges per 5 mins
     bin_edges = [1, 300, 600, 900, 1200, 10000]

     # Define the bin labels
     bin_labels = ['1', '2', '3', '4', 'Others']

     # Add a new column with bin categories
     df['duration_bin'] = pd.cut(df['duration'], bins=bin_edges, labels=bin_labels, right=False)

     print(df)
```

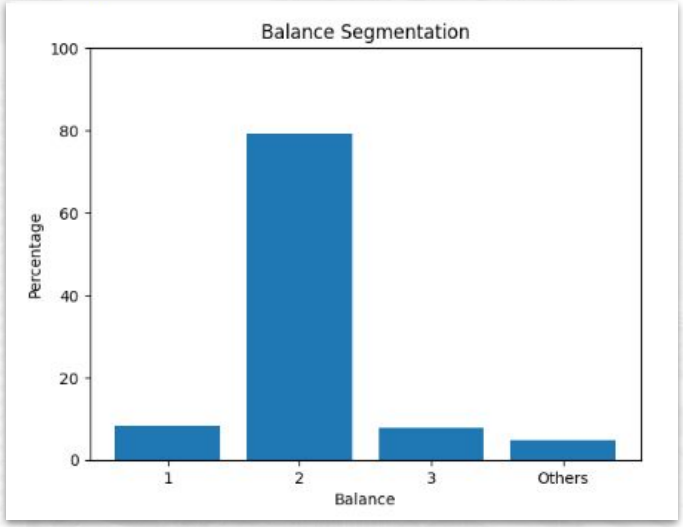


Fitur duration_bin diekstraksi dari fitur duration, dengan pertimbangan untuk melihat durasi telepon/telemarketing. Hasilnya, angka durasi terbanyak berada pada kisaran 1 - 300 detik / 5 menit pertama.

Feature Engineering (Feature Extraction)

balance_bin

```
[14] # Make segment for balance feature
     # Define the bin edges
     bin_edges = [-5000, 0, 3000, 6000, 100000]

     # Define the bin labels
     bin_labels = ['1', '2', '3', 'Others']

     # Add a new column with bin categories
     df['balance_bin'] = pd.cut(df['balance'], bins=bin_edges, labels=bin_labels, right=False)

     print(df)
```



Fitur balance diekstraksi dari fitur balance, dengan pertimbangan untuk mengelompokkan nasabah berdasarkan jumlah balance. Hasilnya, Hasilnya, kebanyakan nasabah memiliki saldo dalam rentang 0 - 3000.

Feature Engineering (Feature selection)

- Feature **'pdays'** dipertimbangkan untuk **tidak dimasukkan** dalam modelling karena memiliki korelasi tinggi dengan fitur ‘previous’ (menghindari multikolinearitas). Hal ini didasari 'pdays' memiliki banyak **outliers** yang lebih ekstrim, memiliki nilai **mayoritas -1** dan **standar deviasi** yang sangat besar ketimbang 'previous'.
- Feature **'duration'** tidak dimasukkan dalam modelling dikarenakan telah dibuat ekstraksi duration_bin.
- Feature **‘default’** tidak dimasukkan ke dalam modelling dikarenakan terdapat ***class imbalance***: 98% value no dan 2% yes.
- Feature **‘poutcome’** tidak dimasukkan ke dalam modelling dikarenakan setelah handling outliers, value yang muncul hanya **unknown**, sehingga tidak dapat memberikan informasi yang cukup.
- Untuk feature selection lanjutan, akan menggunakan metode **RFE** setelah diketahui model machine learning yang akan digunakan.

Feature Engineering (Scaling Dataset)

Robust Scaler

```
from sklearn.preprocessing import RobustScaler

scaler = RobustScaler()

columns_to_scale = ['campaign','previous']
data_to_scale = df[columns_to_scale]

scaler.fit(data_to_scale)

scaled_data = scaler.transform(data_to_scale)

df[columns_to_scale] = scaled_data
df[columns_to_scale].describe()
```

	campaign	previous
count	36954.000000	36954.0
mean	0.460979	0.0
std	1.662896	0.0
min	-0.500000	0.0
25%	-0.500000	0.0
50%	0.000000	0.0
75%	0.500000	0.0
max	30.500000	0.0

Robust scaler perlu digunakan pada kolom 'campaign' dan 'previous' untuk mengatasi distribusi skewed. Dengan menggunakan median dan IQR, robust scaler dapat mengurangi pengaruh outlier pada proses penskalaan dan membantu menghasilkan data yang memiliki distribusi yang lebih terpusat dan terdistribusi secara merata.

Feature Engineering (Scaling Dataset)

Standard Scaler

```
from sklearn.preprocessing import StandardScaler

standard_scaler = StandardScaler()
columns_to_standard_scale = [i for i in df.columns.to_list() if not (i in columns_to_scale or i=='month_quartal')]


data_to_standard_scale = df[columns_to_standard_scale]
standard_scaler.fit(data_to_standard_scale)

standard_scaled_data = standard_scaler.transform(data_to_standard_scale)

df[columns_to_standard_scale] = standard_scaled_data


df.describe()
```

	age	balance	day	campaign	previous	y	housing_no	housing_yes	loan_no
count	3.695400e+04	3.695400e+04	3.695400e+04	36954.000000	36954.0	3.695400e+04	3.695400e+04	3.695400e+04	3.695400e+04
mean	1.045990e-16	3.999375e-17	1.045990e-16	0.460979	0.0	-8.614038e-17	-1.015226e-16	1.692043e-16	-4.614663e-17
std	1.000014e+00	1.000014e+00	1.000014e+00	1.662896	0.0	1.000014e+00	1.000014e+00	1.000014e+00	1.000014e+00
min	-2.198683e+00	-4.564738e-01	-1.808962e+00	-0.500000	0.0	-3.174969e-01	-9.213085e-01	-1.085413e+00	-2.244382e+00
25%	-7.605342e-01	-4.156657e-01	-8.534458e-01	-0.500000	0.0	-3.174969e-01	-9.213085e-01	-1.085413e+00	4.455570e-01
50%	-1.852747e-01	-3.048536e-01	1.020701e-01	0.000000	0.0	-3.174969e-01	-9.213085e-01	9.213085e-01	4.455570e-01
75%	7.734910e-01	-9.496833e-04	6.992676e-01	0.500000	0.0	-3.174969e-01	1.085413e+00	9.213085e-01	4.455570e-01
max	5.183813e+00	3.342649e+01	1.774223e+00	30.500000	0.0	3.149637e+00	1.085413e+00	9.213085e-01	4.455570e-01

8 rows × 41 columns

Standard Scaler perlu digunakan pada kolom sisanya untuk mengubah skala data menjadi standar dan sesuai dengan asumsi distribusi normal. data akan diubah sehingga memiliki rata-rata nol dan simpangan baku satu, sesuai dengan distribusi normal standar.

Feature Engineering (4 Additional Features)

Fitur yang perlu ada dalam dataset ini adalah sebagai berikut:

1. Pendapatan per bulan (bisa gaji atau upah).

   Karena pendapatan tiap bulan mencerminkan seberapa besar kemampuan seorang nasabah untuk bisa menyisihkan pendapatannya untuk investasi di deposito berjangka.

2. Rata-rata pengeluaran per tahun

   Karena pengeluaran pertahun bisa mencerminkan berapa total belanja yang dilakukan seorang nasabah, dibuat per tahun lebih baik karena tiap periode waktu pengeluaran nasabah biasanya berbeda-beda.

3. Total instrumen investasi dimana uang nasabah di invest

   Karena ini menunjukkan betapa teredukasinya seorang nasabah terhadap investasi dan juga melihat apakah melakukan deposit berjangka menjadi prioritas bagi nasabah yang juga sudah berinvestasi di instrumen investasi lain

4. Gender

   Karena gender memperlihatkan perbedaan yang cukup signifikan terhadap pola investasi seorang nasabah. Wanita cenderung belanja lebih banyak dan Pria lebih terkesan cenderung lebih mempersiapkan hari tua dan berinvestasi. Hasil dari adanya fitur ini akan memberikan informasi yang menarik pada model.