INFOLVATORS
INFORMATION INOVATORS

DATA PREPROCESSING

STAGE 1

ANGGOTA TEAM

INFOLVATORS
INFORMATION INOVATORS

- M RIZQI FADHILAH
- M ARVIN FADRIANSYAH
- MELLIZA NASTASIA IZAZI
- THUFAEL BINTANG ALFATTAH
- ZULFIKAR FAUZI
- ANNISA SULISTYANINGSIH
- NIKEN MUSTIKAWENI
- GALIH REFA

DAFTAR PEMBAHASAN

DATA CLEANSING

A. HANDLE MISSING VALUES
B. HANDLE DUPLICATED DATA
C. HANDLE OUTLIERS
D. FEATURE TRANSFORMATION
E. FEATURE ENCODING
F. HANDLE CLASS IMBALANCE

FEATURE ENGINEERING

A. FEATURE SELECTION
B. FEATURE EXTRACTION
C. FEATURE TAMBAHAN

GIT

A. BUAT REPOSITORY GIT
B. UPLOAD FILE NOTEBOOK ATAU FILE PENGERJAAN LAINNYA PADA REPOSITORY TERSEBUT

INFOLVATORS
INFORMATION INOVATORS

DATA CLEANSING

Handling Missing Value

```
Jumlah missing values per kolom:
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
```

DataSet yang tersedia yaitu Bank Target Marketing tidak adanya Missing Value atau kolom yang berisi nilai kosong

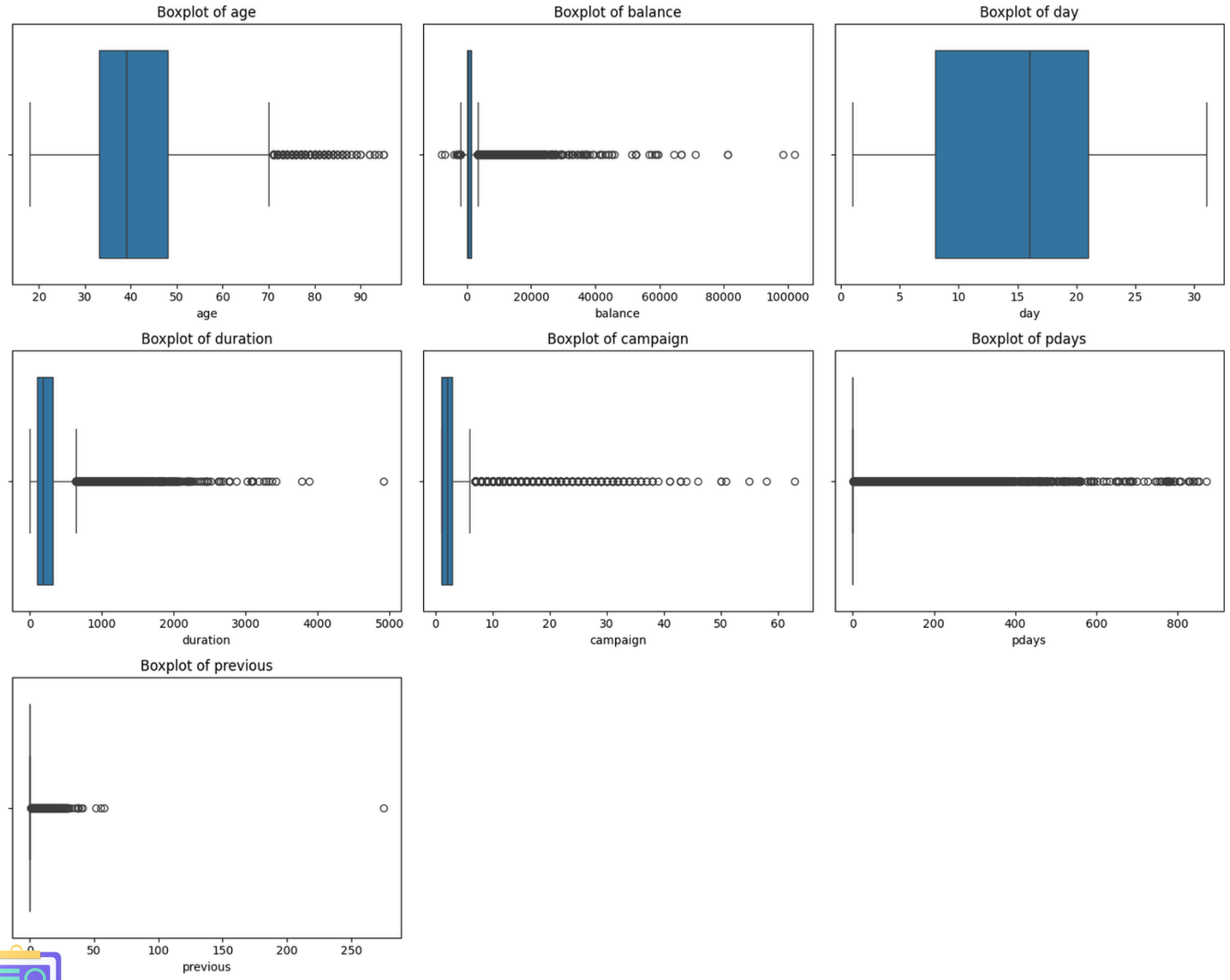
Handling Duplicate Data

```
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        45211 non-null  int64
 1   job        45211 non-null  object
 2   marital    45211 non-null  object
 3   education  45211 non-null  object
 4   default    45211 non-null  object
 5   balance    45211 non-null  int64
 6   housing    45211 non-null  object
 7   loan       45211 non-null  object
 8   contact    45211 non-null  object
 9   day        45211 non-null  int64
 10  month      45211 non-null  object
 11  duration   45211 non-null  int64
 12  campaign   45211 non-null  int64
 13  pdays      45211 non-null  int64
 14  previous   45211 non-null  int64
 15  poutcome   45211 non-null  object
 16  y          45211 non-null  object
dtypes: int64(7), object(10)
memory usage: 5.9+ MB
```

DataSet Kolom yang sudah dibuat , Dapat dilihat bahwa tidak adanya Duplikasi data pada tiap kolom

```
# Melakukan Cheking untuk mendeteksi data duplikat
df.duplicated().sum()
```

```
0
```



Handling Outliers

Handling outliers dilakukan untuk mengatasi nilai-nilai yang sangat jauh atau tidak biasa dalam suatu dataset yang dapat mempengaruhi analisis statistik dan model prediktif, sehingga menangani mereka membantu mencegah kesalahan atau distorsi dalam interpretasi hasil analisis data

INFOLVATORS
INFORMATION INOVATORS

```
# Fungsi untuk mendeteksi outliers menggunakan IQR
def detect_outliers_iqr(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return df[(df[column] < lower_bound) | (df[column] > upper_bound)]

# Deteksi outliers pada kolom balance
outliers_balance = detect_outliers_iqr(df, 'balance')
print(f'Number of outliers in balance: {outliers_balance.shape[0]}')
```

```
Number of outliers in balance: 4729
```

Melakukan Cheking Outliers dan terdapat **4729 Data Outliers** Menggunakan Perhitungan IQR

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
34	51	management	married	tertiary	no	10635	yes	no	unknown	5	may	336	1	-1	0	unknown	no
65	51	management	married	tertiary	no	6530	yes	no	unknown	5	may	91	1	-1	0	unknown	no
69	35	blue-collar	single	secondary	no	12223	yes	yes	unknown	5	may	177	1	-1	0	unknown	no
70	57	blue-collar	married	secondary	no	5935	yes	yes	unknown	5	may	258	1	-1	0	unknown	no
186	40	services	divorced	unknown	no	4384	yes	no	unknown	5	may	315	1	-1	0	unknown	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45164	35	services	married	tertiary	no	4655	no	no	cellular	9	nov	111	2	-1	0	unknown	no
45181	46	blue-collar	married	secondary	no	6879	no	no	cellular	15	nov	74	2	118	3	failure	no
45185	60	services	married	tertiary	no	4256	yes	no	cellular	16	nov	200	1	92	4	success	yes
45191	75	retired	divorced	tertiary	no	3810	yes	no	cellular	16	nov	262	1	183	1	failure	yes
45208	72	retired	married	secondary	no	5715	no	no	cellular	17	nov	1127	5	184	3	success	yes

4729 rows × 17 columns

Kolom '**balance**' sering kali memiliki distribusi yang sangat skewed, di mana sebagian besar nilai berada di rentang rendah dan beberapa nilai sangat tinggi. Skewness yang ekstrem ini dapat mempengaruhi performa model machine learning yang sensitif terhadap **distribusi data.**

```
# Fungsi untuk menghapus outliers menggunakan IQR
def remove_outliers_iqr(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    df_clean = df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]
    return df_clean

# Menghapus outliers pada kolom balance
df_no_outliers = remove_outliers_iqr(df, 'balance')
print(f'Shape of dataset after removing outliers: {df_no_outliers.shape}')
```

```
Shape of dataset after removing outliers: (40482, 17)
```

Setelah **Menghapus Data Outliers dari 45211 Menjadi 40482** hal ini dirasa masih cukup karena data yang tersedia relatif banyak , hal ini dilakukan demi mendapatkan hasil Data berkualitas Tinggi

Handling Outliers

INFOLVATORS
INFORMATION INOVATORS

Features Transformation

```
# Menghapus nilai 0 atau nilai minus pada kolom 'balance'
df_no_outliers_clean = df_no_outliers[df_no_outliers['balance'] > 0].copy()  # Buat salinan eksplisit setelah penyaringan

# Log transformasi kolom 'balance'
df_no_outliers_clean['balance_log'] = np.log1p(df_no_outliers_clean['balance'])

# Tampilkan beberapa baris dari dataset yang telah dibersihkan dan ditransformasi
display(df_no_outliers_clean[['balance', 'balance_log']])
```

	balance	balance_log
0	2143	7.670429
1	29	3.401197
2	2	1.098612
3	1506	7.317876
4	1	0.693147
...	...	...
45205	505	6.226537
45206	825	6.716595
45207	1729	7.455877
45209	668	6.505784
45210	2971	7.996990

33219 rows × 2 columns

Kemudian dalam **membangun model Machine Learning**, kami **menambah feature baru** berupa melakukan **Log Transformation pada kolom Balance** , mengapa hal ini dilakukan ? Mengingat dari Data Kolom Balance **terdapat skewnes** pada kolom tersebut dan **perlu melakukan Log untuk memperkecil skala dan mengurangi skewnes yang terjadi karena nilai diluar atau 0 dan minus**

INFOLVATORS
INFORMATION INOVATORS

Features Transformation

```
# Mendefinisikan mapping dari nama bulan ke nomor bulan
month_mapping = {
    'jan': 1,
    'feb': 2,
    'mar': 3,
    'apr': 4,
    'may': 5,
    'jun': 6,
    'jul': 7,
    'aug': 8,
    'sep': 9,
    'oct': 10,
    'nov': 11,
    'dec': 12
}

# Menambahkan kolom baru yang berisi nomor bulan berdasarkan kolom 'month'
df_no_outliers_clean['month_num'] = df_no_outliers_clean['month'].map(month_mapping)

# Menampilkan hasil DataFrame
display(df_no_outliers_clean)
```

mpaign	pdays	previous	poutcome	y	balance_log	month_num
1	-1	0	unknown	no	7.670429	5
1	-1	0	unknown	no	3.401197	5
1	-1	0	unknown	no	1.098612	5
1	-1	0	unknown	no	7.317876	5
1	-1	0	unknown	no	0.693147	5
...	...	...	...	...	...	...
2	-1	0	unknown	yes	6.226537	11
3	-1	0	unknown	yes	6.716595	11
2	-1	0	unknown	yes	7.455877	11
4	-1	0	unknown	no	6.505784	11
2	188	11	other	no	7.996990	11

Kami **menambahkan juga kolom baru berupa “month_num”** (nomor bulan) untuk membantu pengurutan data jikalau dilakukan visualisasi **berdasarkan bulan**

INFOLVATORS
INFORMATION INOVATORS

Features Encoding

```
# Melakukan one-hot encoding pada beberapa kolom kategorikal
categorical_columns = ['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact', 'poutcome']
df_encoded = pd.get_dummies(df_no_outliers_clean, columns=categorical_columns, drop_first=True)

# Menampilkan hasil one-hot encoding
display(df_encoded)
```

Melakukan One - Hot Encoding untuk mengubah nilai dari Kategorikal Kolom (Job, Marital, default , education, housing, contact, poutcome.

y	balance_log	month_num	...	education_tertiary	education_unknown	default_yes	housing_yes	loan_yes
no	7.670429	5	...	True	False	False	True	False
no	3.401197	5	...	False	False	False	True	False
no	1.098612	5	...	False	False	False	True	True
no	7.317876	5	...	False	True	False	True	False
no	0.693147	5	...	False	True	False	False	False
...	...	...	...	...	...	...	...	...
yes	6.226537	11	...	False	False	False	False	True
yes	6.716595	11	...	True	False	False	False	False
yes	7.455877	11	...	False	False	False	False	False
no	6.505784	11	...	False	False	False	False	False
no	7.996990	11	...	False	False	False	False	False

Hasil table menunjukan bahwa data masih belum bernilai binary atau 1:0 yang seharusnya ini bernilai 1:0 maka perlu diubah pada kolom **y yang yes 1 dan no 0, kemudian true 1 false 0**

INFOLVATORS
INFORMATION INOVATORS

Features Encoding

```
[ ]  # Mengubah nilai 'y' menjadi nilai biner (1 untuk 'yes' dan 0 untuk 'no')
     df_no_outliers_clean['y'] = df['y'].map({'no': 0, 'yes': 1})
```

```
[ ]  # Menampilkan hasil feature encoding
     display(df_no_outliers_clean)
```

Untuk memudahkan membangun model machine learning di tahap selanjutnya, **kami akan melakukan convert kolom"Y" yang berisi 'Yes/No' menjadi 1/0 (binary) pada kolom**

	age	job	marital	education	default	balance	housing	loan	contact	day	duration	campaign	pdays	previous	poutcome	y	balance_log	month_num
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	261	1	-1	0	unknown	0	7.670429	5
1	44	technician	single	secondary	no	29	yes	no	unknown	5	151	1	-1	0	unknown	0	3.401197	5
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	76	1	-1	0	unknown	0	1.098612	5
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	92	1	-1	0	unknown	0	7.317876	5
4	33	unknown	single	unknown	no	1	no	no	unknown	5	198	1	-1	0	unknown	0	0.693147	5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45205	25	technician	single	secondary	no	505	no	yes	cellular	17	386	2	-1	0	unknown	1	6.226537	11
45206	51	technician	married	tertiary	no	825	no	no	cellular	17	977	3	-1	0	unknown	1	6.716595	11
45207	71	retired	divorced	primary	no	1729	no	no	cellular	17	456	2	-1	0	unknown	1	7.455877	11
45209	57	blue-collar	married	secondary	no	668	no	no	telephone	17	508	4	-1	0	unknown	0	6.505784	11
45210	37	entrepreneur	married	secondary	no	2971	no	no	cellular	17	361	2	188	11	other	0	7.996990	11

33219 rows × 18 columns

INFOLVATORS
INFORMATION INOVATORS

Handle Class Imbalance

```
df_no_outliers_clean['y'].value_counts()
```

y	count
0	29198
1	4021

dtype: int64

Melakukan pengecekan value pada kolom y yang berisikan informasi **y**: Respons target, menunjukkan apakah nasabah telah berlangganan deposito berjangka (biner: "yes", "no")

```
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import SMOTE
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
```

```
# Pisahkan fitur dan target
X = df_no_outliers_clean.drop('y', axis=1)  # Fitur
y = df_no_outliers_clean['y']  # Target
```

```
# Pisahkan data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

```
# Identifikasi fitur kategorikal dan numerik
categorical_features = X_train.select_dtypes(include=['object']).columns
numerical_features = X_train.select_dtypes(exclude=['object']).columns
```

INFOLVATORS
INFORMATION INOVATORS

Handle Class Imbalance

```
# Pipeline untuk preprocessing fitur
preprocessor = ColumnTransformer(
    transformers=[
        ('num', StandardScaler(), numerical_features),
        ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_features)
    ])
```

```
# Terapkan preprocessing pada data latih dan uji
X_train_preprocessed = preprocessor.fit_transform(X_train)
X_test_preprocessed = preprocessor.transform(X_test)
```

```
# Terapkan SMOTE pada data latih
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train_preprocessed, y_train)
```

```
# Tampilkan jumlah sampel untuk memverifikasi
print("Jumlah data y sebelum SMOTE:")
print(y_train.value_counts())

print("\nJumlah data y setelah SMOTE:")
print(pd.Series(y_train_resampled).value_counts())
```

```
Jumlah data y sebelum SMOTE:
y
0    23358
1     3217
Name: count, dtype: int64

Jumlah data y setelah SMOTE:
y
0    23358
1    23358
Name: count, dtype: int64
```

Pada dataset ini, kami meningkatkan jumlah sample dengan menciptakan sample sintetis menggunakan oversampling **metode SMOTE.**

INFOLVATORS
INFORMATION INOVATORS

FEATURE ENGINEERING

Features Engineering

Berikut adalah daftar lengkap fitur baru yang bisa diekstrakdari dataset:

1. **Binning (Age Group)**
   Youth <= 25, Adult 26-45, Senior 46-5
2. **Derived Feature**
   - Balance per Campaign (Menilai sebarapa besar saldo rata-rata per kampanye)
   Membuat kolom baru per campaign = balance/(campaign +1)
   - Contact Duration per Day (Mengukur durasi rata-rata kontak per hari)
   Membuat kolom baru duration per day = duration/(day+1)

Features Selection

```python
from sklearn.feature_selection import SelectFromModel
from sklearn.ensemble import RandomForestClassifier

# Buat model untuk seleksi fitur
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train_resampled, y_train_resampled)

# Seleksi fitur menggunakan model
selector = SelectFromModel(model, threshold="mean", prefit=True)
X_train_selected = selector.transform(X_train_resampled)

# Mendapatkan fitur yang terpilih
feature_names = preprocessor.transformers_[1][1].get_feature_names_out(categorical_features).tolist() + numerical_features.tolist()
selected_features = [feature_names[i] for i in range(len(feature_names)) if selector.get_support()[i]]

print("Fitur yang terpilih:")
print(selected_features)
```

```
Fitur yang terpilih:
['job_admin.', 'job_blue-collar', 'job_entrepreneur', 'job_housemaid', 'job_management', 'job_services', 'job_student', 'poutcome_success', 'poutcome_unknown', 'day', 'campaign', 'balance_log']
```

HASIL SELEKSI FITUR MENUNJUKKAN FITUR-FITUR YANG DIANGGAP PALING PENTING OLEH MODEL RANDOMFORESTCLASSIFIER BERDASARKAN KEPENTINGAN MEREKA DALAM PREDIKSI TARGET Y (MISALNYA, JENIS PEKERJAAN EMPLOYMENT TYPE).

INFOLVATORS
INFORMATION INOVATORS

Features Selection

1. Fitur Kategorikal yang Terpilih:

- **job** dan **poutcome**: Kategori pekerjaan dan hasil kampanye yang di-encode menjadi fitur biner (OneHotEncoding). Fitur ini menunjukkan bahwa jenis pekerjaan dan hasil kampanye memiliki pengaruh signifikan terhadap target. Misalnya, jenis pekerjaan tertentu mungkin lebih relevan dalam memprediksi target dibandingkan dengan yang lain.

- **poutcome_success** dan **poutcome_unknown**: Ini menunjukkan bahwa hasil kampanye success dan unknown berperan penting dalam menentukan hasil target.

2. Fitur Numerik yang Terpilih:

- **day**: Ini menunjukkan bahwa hari dalam bulan kampanye juga mempengaruhi target. Mungkin ada pola musiman atau temporal dalam data.

- **campaign**: Jumlah kontak selama kampanye berperan penting. Ini mungkin menunjukkan seberapa sering nasabah dihubungi dan bagaimana hal ini mempengaruhi keputusan mereka.

- **balance_log**: Saldo yang telah ditransformasi dengan log juga penting. Transformasi ini membantu menstabilkan varians dan menangani skewness, membuat fitur ini lebih informatif untuk model.

Kenapa Fitur Tersebut Terpilih:

- **Pengaruh terhadap Target**: Fitur-fitur yang terpilih adalah fitur yang memiliki kekuatan prediktif yang signifikan untuk target y. Model RandomForestClassifier memilih fitur-fitur ini karena mereka memberikan kontribusi yang lebih besar dalam prediksi dibandingkan fitur lainnya.

- **Seleksi Fitur Berbasis Model:** RandomForestClassifier menggunakan ukuran pentingnya fitur untuk menentukan fitur mana yang memiliki dampak paling besar terhadap prediksi. Fitur yang memiliki nilai penting yang lebih tinggi dari rata-rata (threshold="mean") terpilih.

INFOLVATORS
INFORMATION INOVATORS

Features Extraction

```
#Creating new feature 'balance_per_duration'
df_no_outliers_clean['balance_per_duration'] = df_no_outliers_clean['balance'] / df_no_outliers_clean['duration']

#Creating new feature 'campaign_duration_ratio'
df_no_outliers_clean['campaign_duration_ratio'] = df_no_outliers_clean['campaign'] / df_no_outliers_clean['duration']
display(df_no_outliers_clean[['balance_per_duration', 'campaign_duration_ratio']].head())
```

	balance_per_duration	campaign_duration_ratio
0	8.210728	0.003831
1	0.192053	0.006623
2	0.026316	0.013158
3	16.369565	0.010870
4	0.005051	0.005051

Balance_per_duration: Rasio saldo bank terhadap durasi panggilan.

1. **Korelasi Potensia**l: Saldo lebih tinggi bisa terkait dengan durasi panggilan yang lebih panjang, mungkin mempengaruhi keputusan nasabah.
2. **Insight Ekonomi:** Menggambarkan stabilitas ekonomi dan kemungkinan nasabah untuk tertarik pada penawaran.

Campaign_Duration_Ratio: Rasio jumlah kampanye terhadap durasi panggilan.

1. **Efisiensi Kampanye:** Menilai efektivitas kampanye berdasarkan durasi panggilan.
2. **Keterlibatan Nasabah:** Menunjukkan respons nasabah terhadap kampanye; rasio rendah bisa berarti respons yang lambat.

INFOLVATORS
INFORMATION INOVATORS

Features Tambahan

Berikut tiga ide fitur tambahan yang mungkin akan membantu performansi model:

1. average_balance_per_contact: Rata-rata balance per kontak pelanggan.
2. previous_campaign_success_rate: Rasio keberhasilan kampanye sebelumnya.
3. age_group: Kategorisasi umur menjadi beberapa kelompok umur.

GIT

Menggunakan Github sebagai media untuk melakukan kolaborasi antar anggota tim , untuk mempermudah pengerjaan Google Collabs

Dengan langkah langkah

1. **Membuat Repositori untuk menyimpan File yang telah dibuat**
2. **Melakukan Uploading Pengerjaan yang telah dikerjakan**
3. **Membuat File Readme yang berisi summary atau rangkuman dari yang sudah dikerjakan.**

Berikut link Github : https://github.com/aizenciel/Preprocessing_Infolvators

INFOLVATORS
INFORMATION INOVATORS

SEKIAN TERIMAKASIH