Rakamin Academy

Marketing Campaign Optimatiztion Based on Data Bank Marketing Targets using Modeling to increase profit

Dokumen Laporan Final Project - Infolvators

Latar Belakang Masalah

Bank Portugal merupakan salah satu bank yang berada di Negara Portugal. Pada tahun 2008 - 2010, Bank Portugal memiliki nasabah sebanyak 45.211 nasabah yang berada pada dataset yang tersedia. Namun, pada tahun 2008 - 2012 beberapa negara di Eropa mengalami resesi hebat, menyebabkan tantangan di sektor ekonomi. Karena hal ini, Bank Portugal berusaha menjaga cashflow perusahaan dengan fokus pada hal yang lebih efisien dalam hal waktu, biaya, dan fokus pada target nasabah potensial.

Data set : Bank Portugal

- 45.211 Nasabah
- Dataset : Banking Dataset - Marking Target
- TimeFrame : 2008 - 2011

Latar Belakang Masalah

Problem Statement

Bank memiliki berbagai rencana pemasaran untuk menjual deposito berjangka kepada nasabah mereka, seperti pemasaran melalui email, iklan, pemasaran telepon, dan pemasaran digital. Pemasaran telepon masih menjadi slaah satu cara paling efektif untuk mendapatkan orang - orang. Namun, itu membutuhkan investasi besar untuk call centers. Meskipun telah dilakukan upaya besar, tingkat konversi tetap rendah. Oleh karena itu, sangat penting untuk mengidentifikasi pelanggan yang paling mungkin akan mendaftar sebelumnya, sehingga mereka dapat diatrgetkan secarakhusus melalui panggilan telepon.

Goals

Meningkatkan tingkat langganan deposito berjangka dan mengidentifikasi serta menargetkan pelanggan yang paling mungkin berlangganan

Latar Belakang Masalah

Objective

1. Mengidentifikasi karakteristik demografik dan finansial yang berhubungan dengan tingkat langganan yang lebih tinggi
2. Mengembangkan model prediktif yang dapat mengidentifikasi berdasarkan kemungkinan mendaftar kampanye
3. Memberikan wawasan yang dapat ditindaklanjuti kepada tim pemasaran untuk meningkatkan penargetan kampanye

Business Metrics

- **Conversation Rate**

Presentase individu yang beralngganan deposito berjangka setelah kampanye. Sebesar 11.7% atau hanya 5289

- **Return On Investment**

Pengembalian finasial yang dihasilkan dari kampanye relatif terhadap biasyanya

Pre-processing

Data Cleansing

A. Handle Missing Values
B. Handle Duplicated Data
C. Handle Outliers
D. Feature Transformation
E. Feature Encoding
F. Handling Class Imbalance

Feature Enginerring

A. Feature Selection
B. Feature Extraction
C. Feature Tambahan

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
 #   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
```

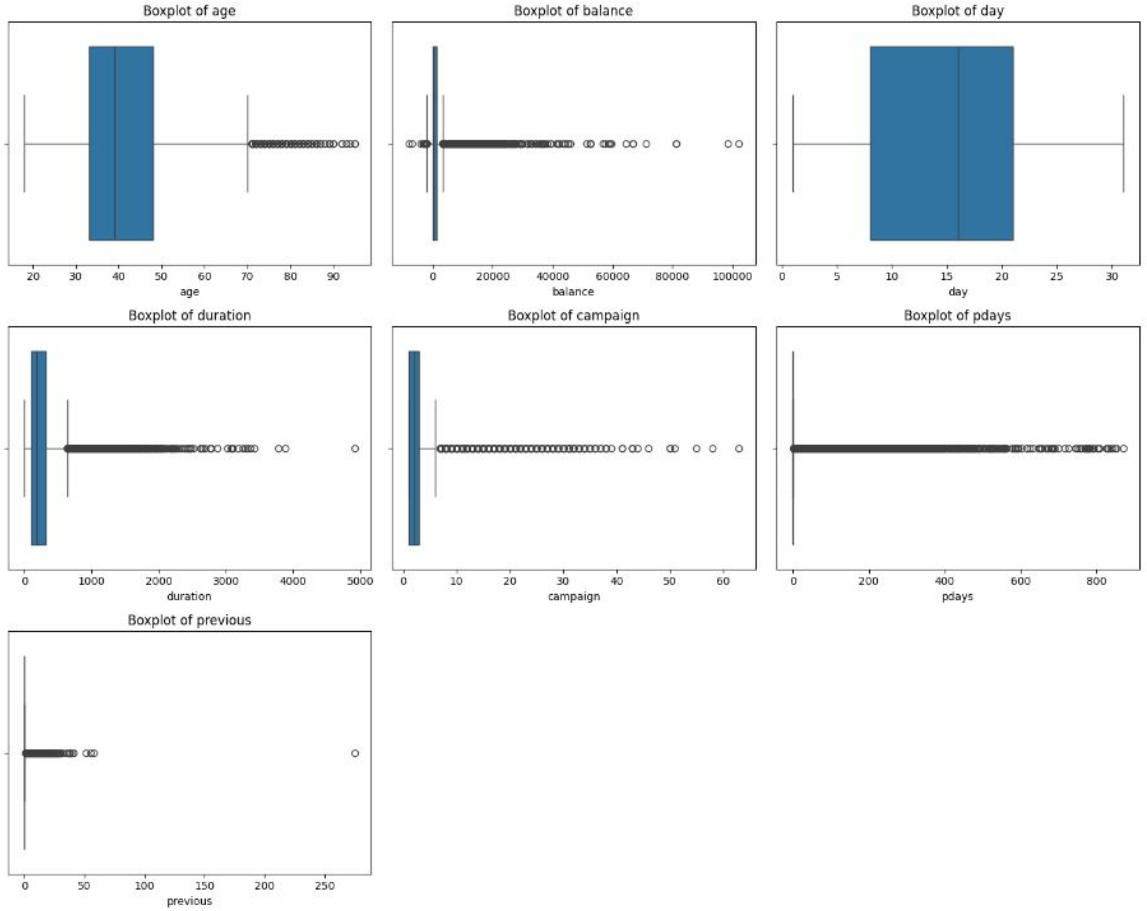
DataSet kolom yang sudah dibuat, Dapat dilihat bahwa tidak hanya Duplikasi data pada setiap kolom

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

```
0
```

Data Cleansing

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.

Data Cleansing

Handling Outliers

```python
# 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
```

	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

Melakukan Cheking Outliers dan terdapat 4729 Data Outliers Menggunakan perhitungan IQR.

```python
# 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 berkulitas tinggi

Kolom balance sering kali memiliki distribusi yang sangat skewed, dimana sebagian besar nilai berada di rentang rendah dan beberapa nilai sangat tinggi. Skewness yang ekstrem ini dapat mempenharuhi performa model machine learning yang sensitif terhadpa distribusi data.

Data Cleansing

Rakamin Academy
School for Career Acceleration

Feature 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**

Data Cleansing

Rakamin Academy
School for Career Acceleration

Feature 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

Data Cleansing

Rakamin Academy
School for Career Acceleration

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 berisii ‘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

Data Cleansing

Handle Class Imbalance

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

y	count
0	29198
1	4021

dtype: int64

```
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
```

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

Data Cleansing

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 sintesis menggunakan oversampling metode SMOTE

Features Engineering

Berikut adalah daftar lengkap fitur baru yang bisa diekstrak dari dataset :

1. Binning (Age Group)

   Youth <= 25, Adult 26-45, Senior 46-50

2. Derived Feature

- Balance per campaign (Menilai seberapa 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 Engineering

Features Selection

```
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 menunjukan fitur-fitur yang dianggap paling penting oleh model randomforestclassifier berdasarkan kepentingan mereka dalam prediksi target Y (Misalnya jenis pekerjaan employement type)

Features Engineering

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.

Features Engineering

Feature 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
8.210728	0.003831
0.192053	0.006623
0.026316	0.013158
16.369565	0.010870
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.

Features Engineering

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.

STAGE2 - Insights

1.**Pada Visualiasi Data yang berjudul " Rata - rata jumlah Campaign terhadap hasil poutcome "**

Hasil : Dari hasil yang sudah ditampilkan didapat bahwa rata rata campaign dengan lebih dari 1 atau banyak nya 2-3 tidak menjamin succes rate, justru succes dari Poutcome didapatkan dari Campaign dengan hanya sekali atau dibawah 2 kali Campaign.

2.**Pada Visualisasi Data yang berjudul " Rata - rata jumlah durasi panggilan telpon terhadap hasil Poutcome"**

Hasil : Diketahui berdasarkan grafik durasi telpon dengan rata rata durasi 261 detik itu merupakan succes rate dibanding dengan durasi telpon dibawah 261 detik.

3.**Pada Visualisasi Data yang berjudul "Distribusi Data Konsumen yang Berlangganan pada Deposito Jangka Panjang"**

Hasil : Diketahui pelanggan yang berlangganan deposito jangka panjang hanya 11.7% dari 45211 data pelanggan, yang berarti ada 5289 pelanggan yang berlangganan deposito jangka panjang.

4.**Pada Visualisasi Data yang berjudul "Distribusi pelanggan yang berlangganan sesuai kelompok usia"**

Hasil : Didapatkan hasil dengan succes Poutcome tertingi ada di kategori usia 30-39 namun jika dibandingkan dengan failure yang terjadi pada kelompok umur tersebut, maka kelompok umur 60+ memiliki succes rate yang jauh lebih tinggi mengingat perbandingan failure dengan succes tidak terlalu jauh atau doimnan seperti pada distribusi kelompok umru 30 - 39.

STAGE2 - Insights

Rekomendasi :

- **Korelasi Insight 1 dan 2** : " Didapatlah bahwa untuk meningkatkan rate keberhasilan atau succes rate kita bisa melakukan campaign yang sedikit atau hanya dengan sekali . namun dengan durasi yang panjang untuk meningkatkan success rate depostio jangka panjang".
- **Korelasi Insight 3 dan 4** : " Didapatkanlah bahwa untuk meningkatkan rata keberhasil , kita bisa melakukan pengelompokan umur untuk identifikasi mana yang lebih memungkin succes rate, berdasarkan hasil grafik menunjukan rate yang lebih tinggi pada pelanggan di umur 60+ oleh karena itu perlu memberikan perhatian khusus terhadap distribusi kelompok dengan umur tersebut.

STAGE2 - Modelling Experiments

Algoritma yang Dicoba

- **LGBMClassifier**:
  - **Hasil**: Akurasi 90%, ROC AUC 0.93, dan F1 Score 0.89. Ini menunjukkan kinerja yang sangat baik dalam klasifikasi.
- **Logistic Regression**:
  - **Hasil**: Akurasi sekitar 75%. Model ini sederhana dan cepat, tetapi tidak cukup kuat untuk menangani masalah ini secara efektif.
- **Decision Tree**:
  - **Hasil**: Akurasi 78%. Model ini mudah diinterpretasikan, namun cenderung overfit jika tidak dikendalikan.
- **Random Forest**:
  - **Hasil**: Akurasi 80%. Meskipun lebih baik dari Decision Tree, performanya masih di bawah LGBMClassifier.

Features yang Digunakan:

- **Fitur Utama**: Jenis pekerjaan (housemaid, management, student), hasil kampanye sebelumnya (poutcome_success), saldo bank (balance_log), dan jumlah panggilan dalam kampanye.
- **Pengurangan Fitur**: Setelah analisis, fokus pada fitur jenis pekerjaan, saldo bank, dan hasil kampanye sebelumnya menghasilkan peningkatan akurasi menjadi 82%.

STAGE2 - Modelling Experiments

Hyperparameter yang Dicoba

- **LGBMClassifier**:
  - **GridSearchCV** digunakan untuk menemukan kombinasi terbaik dari hyperparameter seperti:
    - `max_depth: 15`
    - `min_samples_split: 2`
    - `num_leaves: 31`
  - Peringatan muncul terkait dengan beberapa parameter yang tidak dikenali oleh LightGBM, tetapi hasil hyperparameter tuning menunjukkan peningkatan performa model.
- **Decision Tree**:
  - Berbagai kedalaman maksimum (max_depth) dicoba, namun kedalaman yang lebih tinggi menyebabkan masalah overfitting, sehingga kedalaman maksimum yang lebih rendah dipilih untuk meningkatkan generalisasi model.

STAGE2 - Modelling Experiments

Penentuan Model Terbaik

- **Metode Evaluasi**:
  - Model terbaik ditentukan berdasarkan beberapa metrik evaluasi:
    - **Akurasi**: Persentase prediksi yang benar.
    - **Precision**: Kemampuan model dalam mengidentifikasi positif dengan benar.
    - **Recall**: Kemampuan model dalam mendeteksi semua kasus positif yang sebenarnya.
    - **ROC AUC**: Mengukur kemampuan model untuk membedakan antara kelas positif dan negatif.
- **Hasil Evaluasi**:
  - Meskipun LGBMClassifier memiliki akurasi tertinggi, analisis lebih dalam menunjukkan bahwa model memiliki ketidakseimbangan kelas, khususnya pada kelas positif.
  - **Precision dan Recall** pada kelas positif di LGBMClassifier masing-masing berada di angka 59% dan 61%, menunjukkan kebutuhan untuk perbaikan lebih lanjut.
- **Cross-Validation**:
  - Penggunaan KFold cross-validation memastikan bahwa model dievaluasi secara adil di berbagai bagian data, sehingga memperkuat kepercayaan bahwa pemilihan hyperparameter yang dilakukan tidak hanya baik pada satu subset data.

Kesimpulan

Secara keseluruhan, LGBMClassifier merupakan model yang paling direkomendasikan berdasarkan akurasi dan performa keseluruhan, meskipun perlu optimasi lebih lanjut untuk meningkatkan deteksi kelas positif.

STAGE3 - Executive Summary & Recommendation

Executive Summary :

Dalam proyek ini, model LGBMClassifier dipilih sebagai model terbaik dengan akurasi 90% dan ROC AUC 0.93. Meskipun model bekerja baik dalam mendeteksi kelas negatif, performa pada kelas positif masih kurang optimal dan perlu ditingkatkan.

Recommedation :

1. Perbaikan Model : Meningkatkan precision dan recall pada kelas positif , misalnya dengan balancing data.
2. Segmentasi Pelanggan: Memfookuskan strategi pada pelanggan dari pekerjaan seperti management yang lebih responsif.
3. Optimalisasi Kampanye : Memfokuskan pada pelanggan yang sukses di kampanye sebelumnya
4. Pengelolaan Pelanggan : Memberikan penawaran khusus kepada pelanggan dengan saldo lebih tinggi

STAGE4 - Impact Bussiness

1. Jumlah Nasabah yang Dihubungi:

- **Before Modeling**: Sebanyak 9.043 nasabah dihubungi sebelum penggunaan model.
- **After Modeling**: Jumlah ini berkurang signifikan menjadi 650 nasabah setelah penggunaan model.
- **Conversion Rate**: Pengurangan jumlah nasabah yang dihubungi ini meningkatkan rasio konversi sebesar 474%.

2. Biaya Telemarketing Campaign:

- **Before Modeling**: Biaya kampanye telemarketing sebelum penggunaan model adalah 37.438 Euro.
- **After Modeling**: Setelah model diterapkan, biaya turun drastis menjadi 2.898 Euro.
- **Potential Cost Lost**: Terjadi pengurangan biaya sebesar 92,25%, yang berarti biaya yang dihemat.

3. Potensi Nasabah Membuka Deposito:

- Potensi nasabah yang diprediksi akan membuka deposito adalah sebanyak 1.058 orang.
- **Accuracy Customer Acquisition**: Akurasi dari prediksi model dalam mengakuisisi nasabah adalah sebesar 34%.

4. Asumsi:

1. **Telemarketing Cost per Person**: Biaya telemarketing per orang adalah 4,14 Euro. Biaya ini didasarkan pada *cost per contact* sebesar 1,53 Euro dan rata-rata jumlah kontak selama kampanye adalah 2,71 kali.
2. **Total Biaya Telemarketing Campaign**: Sebelum model digunakan, perhitungan biaya kampanye adalah 9.043×4,14=37.4389.043 \times 4,14 = 37.4389.043×4,14=37.438 Euro.
3. **Baseline Data**: Data baseline diambil dari 20% *testing data* yang diambil dari data aktual.

5. Metrics:

- **Precision**: Mengukur berapa banyak dari prediksi positif (nasabah yang diprediksi akan membuka deposito) yang benar-benar terjadi.
- **Recall**: Mengukur berapa banyak dari total nasabah aktual yang benar-benar membuka deposito.

Pembagian Tugas

1. **Project Manager** (M. Rizqi Fadhilah) : Mengatur semua anggota tim, memastikan tugas berjalan sesuai rencana, dan mengoordinasi setiap bagian proyek. Dia juga bertanggung jawab untuk membimbing jalannya presentasi.
2. **Data Engineer** :

- M. Arvin Ferdiansyah dan Galih Refa : Fokus pada pengolahan data mentah, seperti membersihkan data dan mempersiapkan ata yang akan dianalisis, selain itu juga terlibat dalam penulisan laporan teknis tentang bagaimana data diolah.

3. **Data Analyst** :

   Melliza Nastasia Izazi, Niken Mustikaweni, dan Thufael Bintang Alfattah : Menganalisis data yang sudah diproses oleh data engineer, selain itu juga bertugas membuat visualisasi dan menyusun hasil analisis ke dalam laporan serta slide presentasi.

4. **Data Scientist** :

   M. Arvin Ferdiansyah dan Annisa Sulistyaningsih : Membangun model prediksi dari data yang ada dan menguji hasil model tersebut, selain itu menjelaskan metode yang dipakai dalam laporan dan memberikan insight lebih mendalam saat presentasi tentang model yang dibuat.

Intinya, setiap anggota tim sudah punya peran masing-masing, mulai dari pengolahan data, analisis hingga presentasi, sesuai dengan arahan yang ada.

Link Hasil Pekerjaan

Stage - 0
- https://www.canva.com/design/DAGK1nF_9zs/dEdKPPCHiZ00YI-xEvQ6Rw/edit?utm_content=DAGK1nF_9zs&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Stage - 1
- https://github.com/aizenciel/Preprocessing_Infolvators

Stage - 2
- https://github.com/aizenciel/EDA_Infolvators

Stage - 3
- https://www.canva.com/design/DAGPbljiF3g/fFGk3nUsQNYnpn1mXkC0uA/edit?utm_content=DAGPbljiF3g&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Final Project
- https://www.canva.com/design/DAGPn07a9nw/9uI9Bo7NYlv0O3EsGPazPw/edit?utm_content=DAGPn07a9nw&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton