Analyze the behavior of loan property customers

Supported by:
Rakamin Academy
Career Acceleration School
www.rakamin.com

Rakamin Academy

NI MADE SRI KUMALA DEWI OKA

Freshgraduate Mathematics
Udayana University

Connect with me!

dewiikumala_

Sri Kumala Dewi

Overview

Tulisan ini memuat **analisa perilaku pelanggan** yang mengajukan pinjaman kepada bank yang menyediakan layanan pinjaman uang untuk pembelian properti. Analisa dilakukan **untuk mengetahui minat pelanggan** terhadap tipe properti dan pengaruh status perkawinan terhadap jangka waktu peminjaman.

Dataset

Dataset yang Digunakan

Loan Customer

- **Deskripsi**

Dataset ini menggambarkan perilaku dan profil pelanggan di sebuah bank yang menyediakan program pinjaman untuk pembelian property

- **Data**

Setiap baris mewakili pelanggan, setiap kolom berisi atribut pelanggan

Data Preprocessing

Informasi Dataframe

```
df_loan.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 621 entries, 0 to 620
Data columns (total 17 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   loan_id             621 non-null    object 
 1   birth_date          572 non-null    object 
 2   phone_number        569 non-null    float64
 3   gender              608 non-null    object 
 4   married             618 non-null    object 
 5   dependents          605 non-null    object 
 6   education           620 non-null    object 
 7   self_employed       588 non-null    object 
 8   applicant_income    612 non-null    float64
 9   coapplicant_income  621 non-null    float64
 10  loan_amount         621 non-null    int64  
 11  loan_term_month     621 non-null    int64  
 12  loan_term_year      621 non-null    int64  
 13  credit_history      570 non-null    float64
 14  has_credit_card     88 non-null     object 
 15  property_type       620 non-null    object 
 16  loan_status         621 non-null    object 
dtypes: float64(4), int64(3), object(10)
memory usage: 82.6+ KB
```

Hasil Analisis

1. Terdapat 621 baris (data) dengan 17 kolom (features)
2. Pada beberapa features terlihat memiliki null/missing values yang ditandai dengan null count < baris
3. Ada kesalahan type data pada kolom phone_number dan birth_date sehingga perlu dilakukan konversi tipe data.

Untuk selengkapnya, dapat melihat google collaboration disini

Informasi Dataframe

Handling dtypes error

```
df_loan['birth_date'] = pd.to_datetime(df_loan['birth_date'])
df_loan['phone_number'] = df_loan['phone_number'].astype(str)
df_loan.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 621 entries, 0 to 620
Data columns (total 17 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   loan_id             621 non-null    object
 1   birth_date          572 non-null    datetime64[ns]
 2   phone_number        621 non-null    object
 3   gender              608 non-null    object
 4   married             618 non-null    object
 5   dependents          605 non-null    object
 6   education           620 non-null    object
 7   self_employed       588 non-null    object
 8   applicant_income    612 non-null    float64
 9   coapplicant_income  621 non-null    float64
 10  loan_amount         621 non-null    int64
 11  loan_term_month     621 non-null    int64
 12  loan_term_year      621 non-null    int64
 13  credit_history      570 non-null    float64
 14  has_credit_card     88 non-null     object
 15  property_type       620 non-null    object
 16  loan_status         621 non-null    object
dtypes: datetime64[ns](1), float64(3), int64(3), object(10)
memory usage: 82.6+ KB
```

Hasil Analisis

Semua tipe data sudah sesuai

Missing Values

```
[ ]  # Informasi banyaknya missing values di tiap feature
     df_loan.isnull().sum()
```

```
loan_id               0
birth_date           49
phone_number         52
gender               13
married               3
dependents           16
education             1
self_employed        33
applicant_income      9
coapplicant_income    0
loan_amount           0
loan_term_month       0
loan_term_year        0
credit_history       51
has_credit_card     533
property_type         1
loan_status           0
dtype: int64
```

Hasil Analisis

Feature has_credit_card memiliki missing values terbanyak, sehingga feature ini akan dibuang (drop) bersamaan dengan baris data yang memiliki missing values lainnya.

Missing Values

Handling Missing Values

```
# Membuang kolom yang memiliki missing values terbanyak
df_loan.drop('has_credit_card', axis=1, inplace=True)

# Menghapus baris NaN
df_loan = df_loan.dropna()

# Melihat jumlah data
print(df_loan.info())
print('\nUkuran data : {}'.format(df_loan.shape))
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 472 entries, 0 to 619
Data columns (total 16 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   loan_id             472 non-null    object
 1   birth_date          472 non-null    datetime64[ns]
 2   phone_number        472 non-null    object
 3   gender              472 non-null    object
 4   married             472 non-null    object
 5   dependents          472 non-null    object
 6   education           472 non-null    object
 7   self_employed       472 non-null    object
 8   applicant_income    472 non-null    float64
 9   coapplicant_income  472 non-null    float64
 10  loan_amount         472 non-null    int64
 11  loan_term_month     472 non-null    int64
 12  loan_term_year      472 non-null    int64
 13  credit_history      472 non-null    float64
 14  property_type       472 non-null    object
 15  loan_status         472 non-null    object
dtypes: datetime64[ns](1), float64(3), int64(3), object(9)
memory usage: 62.7+ KB
None

Ukuran data : (472, 16)
```

Hasil Analisis

1. Jumlah features menjadi 16 karena ada features yang dibuang
2. Jumlah data yang lengkap sebanyak 472 data

Duplicated Values

```
# Melihat contoh dan banyaknya data duplikat
duplicate_rows = df_loan[df_loan.duplicated()]
print(f'Banyaknya data duplikat : {df_loan.duplicated().sum()}\nDengan rincian sebagai berikut:')
duplicate_rows
```

```
Banyaknya data duplikat : 5
Dengan rincian sebagai berikut:
```

	loan_id	birth_date	phone_number	gender	married	dependents	education	self_employed	applica
90	LP001310	1987-10-18	628131021360.0	Male	Yes	0	Graduate	No	
279	LP001904	1986-01-30	62811520563.0	Male	Yes	0	Graduate	No	
308	LP001990	1980-08-03	628527921833.0	Male	No	0	Not Graduate	No	
363	LP002160	1990-11-27	628115255742.0	Male	Yes	3+	Graduate	No	
401	LP002277	1972-06-12	628138392877.0	Female	No	0	Graduate	No	

Next steps: View recommended plots

Hasil Analisis

Terdapat 5 data duplikat pada dataset sehingga harus dihapus agar tidak merusak kesimpulan analisis akhir.

Duplicated Values

Handling duplicated Data

```
# Menghapus data duplikat
df_loan.drop_duplicates(inplace=True)

# Mengecek kembali apakah ada data duplikat
print(f'Banyaknya data duplikat : {df_loan.duplicated().sum()}')
print(f'Jumlah data setelah data duplikat dihapus : {len(df_loan)}')
```

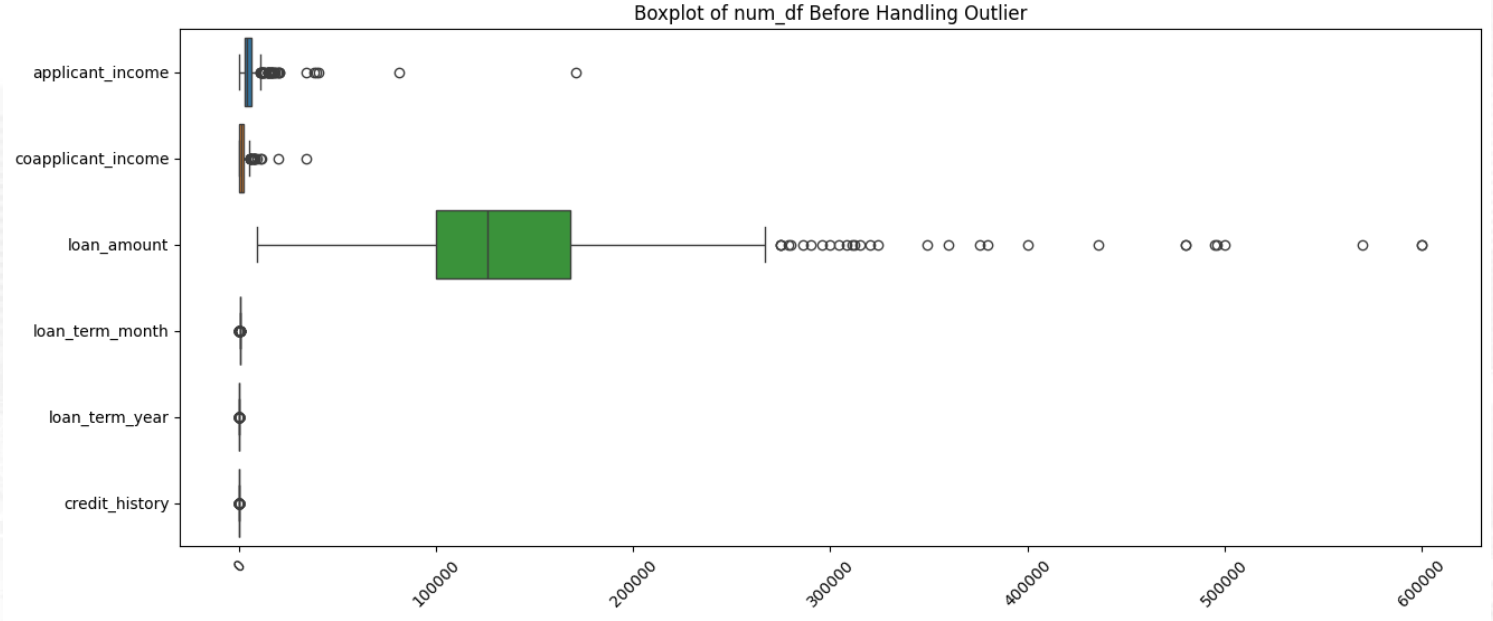
```
Banyaknya data duplikat : 0
Jumlah data setelah data duplikat dihapus : 467
```

Hasil Analisis

Data duplikat sudah dihapus dan menyisakan 467 data.

Handling Outlier

Hasil Analisis



Outlier terlihat jelas pada data applicant_income, coapplicant_income, dan loan_amount. Data outlier dapat merusak sebaran data sehingga akan dilakukan metode zscores untuk menghapus data tersebut.

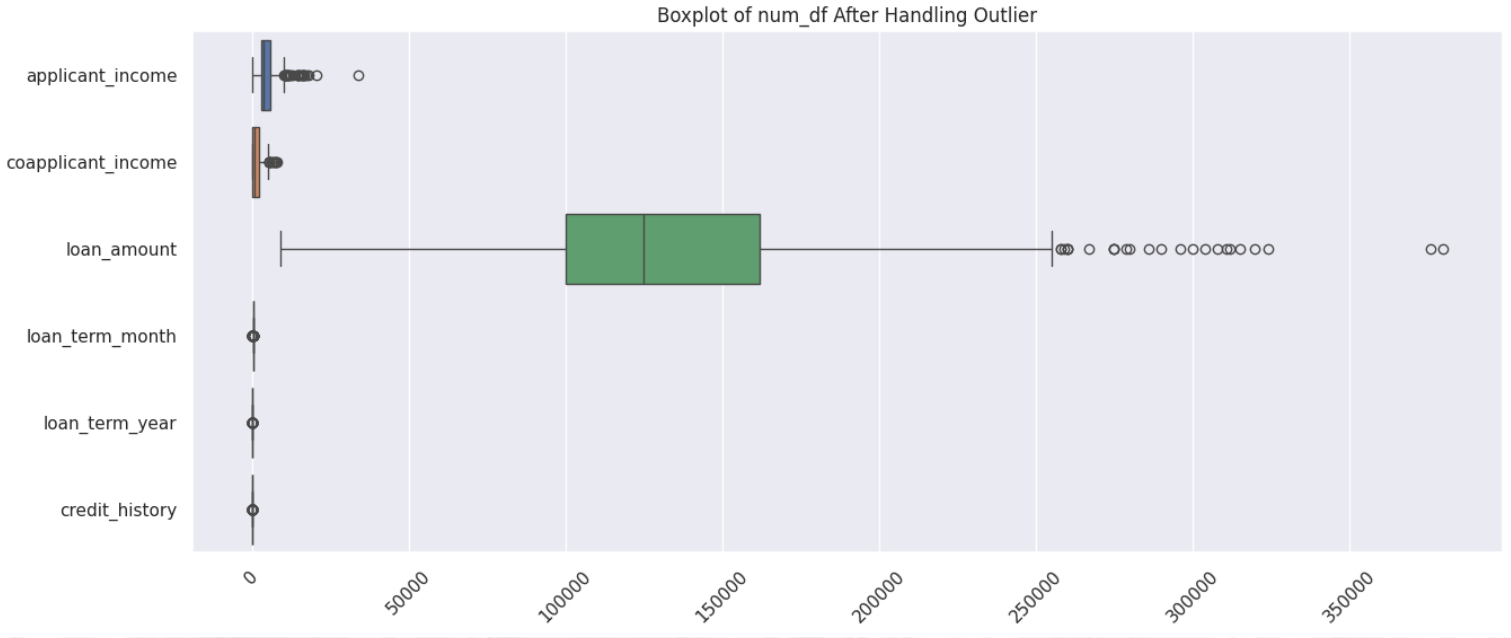
Handling Outlier

Handling Outlier

```
[296] # # Handling Outlier

      from scipy.stats import zscore

      print(f'Jumlah baris sebelum handling outier : {len(df_loan)}')

      z = np.abs(zscore(df_loan[num_df]))
      threshold = 3
      df_loan_clean = df_loan[(z < threshold).all(axis=1)]

      print(f'Jumlah baris sesudah handling outier : {len(df_loan_clean)}')

      Jumlah baris sebelum handling outier : 467
      Jumlah baris sesudah handling outier : 439
```

Handling Outlier

Hasil Analisis



Outlier tidak hilang sepenuhnya namun sudah lebih baik dari sebelumnya

Feature Encoding

Rakamin Academy

Mapping Encoding

```
[56] # Mapping cat_df

mapping_gender = {
    'Female' : 0,
    'Male' : 1
}

mapping_married = {
    'No' : 0,
    'Yes' : 1
}

mapping_education = {
    'Not Graduate' : 0,
    'Graduate' : 1
}

mapping_dependents = {
    '0' : 0,
    '1' : 1,
    '2' : 2,
    '3+' : 3
}

mapping_self_employed = {
    'No' : 0,
    'Yes' : 1
}

mapping_property_type = {
    'apartment' : 0,
    'house' : 1,
    'studio' : 2
}


df_encode['gender'] = df_encode['gender'].map(mapping_gender)
df_encode['married'] = df_encode['married'].map(mapping_married)
df_encode['education'] = df_encode['education'].map(mapping_education)
df_encode['self_employed'] = df_encode['self_employed'].map(mapping_self_employed)
df_encode['property_type'] = df_encode['property_type'].map(mapping_property_type)
df_encode['dependents'] = df_encode['dependents'].map(mapping_dependents)
```

Feature Encoding

Hasil Analisis

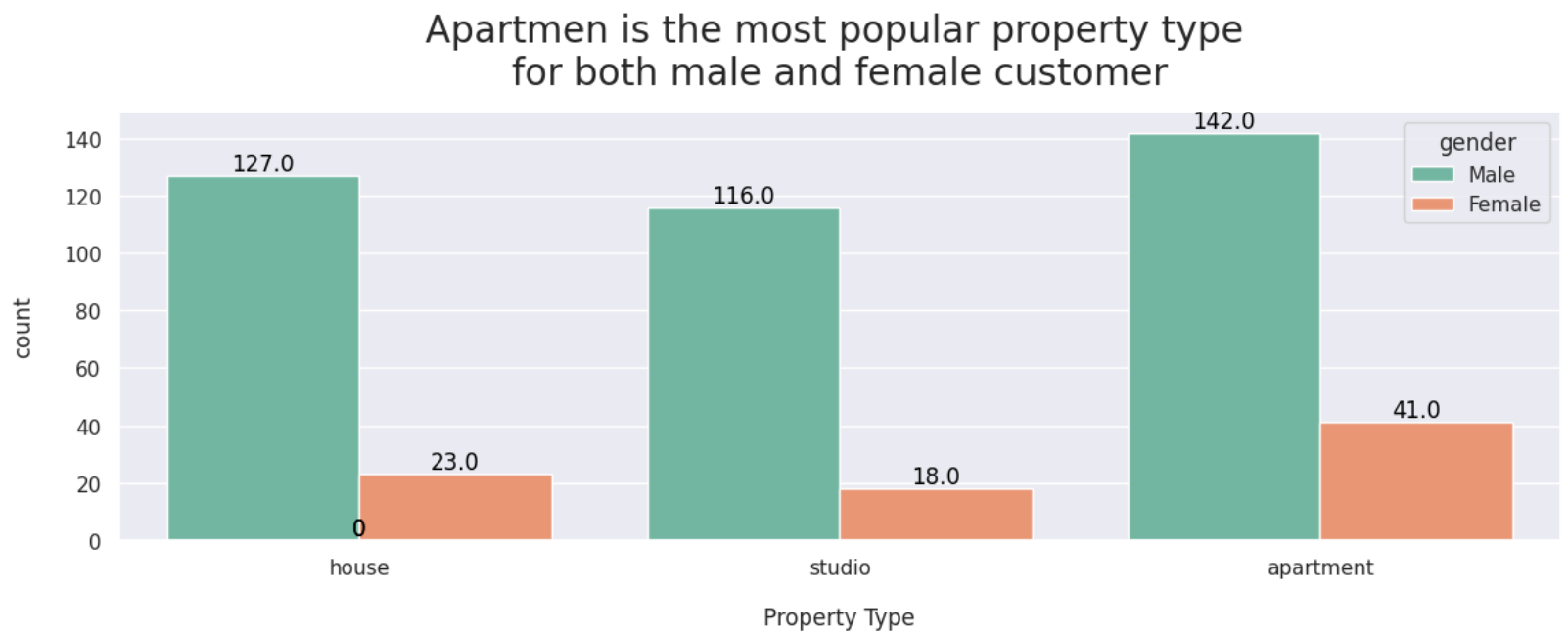
```
[57] df_encode.head()
```

	loan_id	birth_date	phone_number	gender	married	dependents	education	self_employed	applicant_income	coapplicant_income	loan_a
0	LP001002	1977-05-15	628114203187.0	1	0	0	1	0	5849.0	0.0	1
1	LP001003	1979-08-24	628776449212.0	1	1	1	1	0	4583.0	1508.0	1
2	LP001005	1991-03-28	628119240537.0	1	1	0	1	1	3000.0	0.0	
3	LP001006	1980-06-02	62856343702.0	1	1	0	0	0	2583.0	2358.0	1
4	LP001008	1989-05-19	628113008687.0	1	0	0	1	0	6000.0	0.0	1

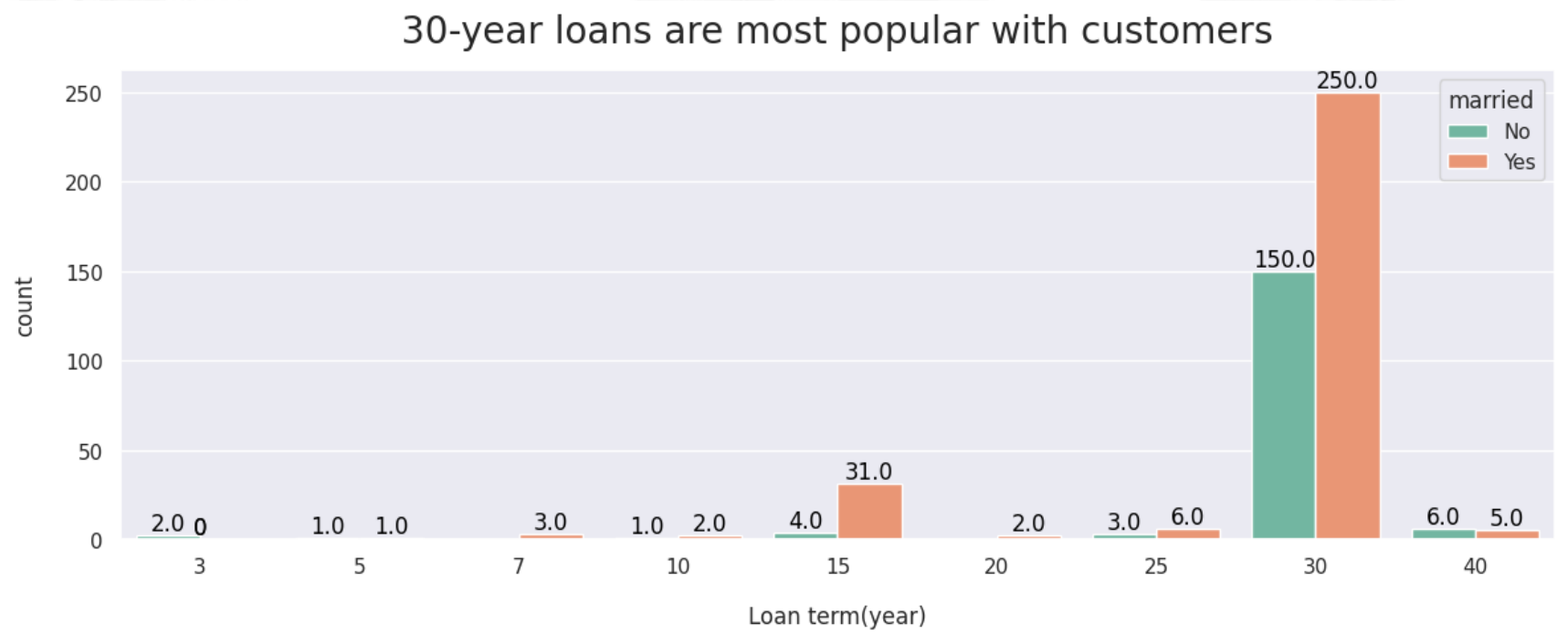
Nilai pada feature categorical sudah berubah sesuai dengan proses mapping encode sebelumnya.

BUSINESS INSIGHT

Analisis Minat Pelanggan Berdasarkan Pada Tipe Properti



Analisis Pengaruh Status Perkawinan Terhadap Jangka Waktu Meminjam



Kesimpulan

Kesimpulan

Berdasarkan analisa yang telah dilakukan, nasabah yang mengajukan peminjaman pada bank memiliki karakteristik sebagai berikut :

1. Lebih meminati properti apartemen dibanding studio dan rumah
2. Cenderung melakukan peminjaman dengan jangka waktu 30 tahun

Terima Kasih!