FINAL PROJECT: BINARY CLASSIFICATION

FIVE STAR PRODUCT RATING CLASSIFICATION

Wilhelmus Medhavi

Data Science Bootcamp Sharing Vision

Batch May 2022

OUTLINE

1. **Business Understanding**
   Memahami masalah dan goal yang ingin dicapai dalam project

2. **Data Understanding**
   Memahami dan memeriksa dataset untuk mendapatkan informasi yang dapat digunakan dalam model

3. **Data Preparation**
   Menyiapkan dan membersihkan data apa saja yang akan digunakan dalam model

4. **Feature Engineering**
   Membuat, mentransformasi, dan memilih fitur apa saja yang akan digunakan

5. **Modelling**
   Memilih machine learning model berdasarkan kriteria yang diinginkan

6. **Evaluation**
   Mengevaluasi model terpilih dengan memprediksi dataset baru

01

Business Understanding

Final Project:
Binary Classification
FIVE STAR PRODUCT RATING CLASSIFICATION

BUSINESS UNDERSTANDING

Business Objective

Suatu perusahaan marketplace Ecommerce ingin membuat guideline berisi tips untuk penjual bagaimana agar mendapatkan rating 5 dari pembeli.

Model Objective

Membuat mesin klasifikasi untuk menentukan apakah buyer memberikan rating 5 terhadap barang yang dibeli (label 1) atau rating di bawah 5 (label 0).

Model Success Criteria

Model memiliki nilai *recall*, *precission*, dan *True Negative Rate* (TNR) > 0.7 serta nilai *False Positive Rate* (FPR) < 0.3.

02

Data Understanding

Final Project:
Binary Classification
FIVE STAR PRODUCT RATING CLASSIFICATION

DATA UNDERSTANDING

Data Description

- Dataset terdiri dari dua file csv yakni model_development_set untuk melatih model dan model_backtesting_set untuk pengecekan model dalam memprediksi data baru.
- Dataset model_development_set memiliki ukuran dimensi 6814 baris dengan 40 kolom, yang terdiri dari 3 jenis data (object, number, dan datetime).
  - Jumlah data tipe number : 19 kolom
  - Jumlah data tipe object : 15 kolom
  - Jumlah data tipe datetime : 6 kolom

Data tipe Datetime

Nama kolom	Keterangan
order_purchase_timestamp	Waktu tanggal pemesanan
order_delivered_carrier_date	Waktu tanggal pengiriman barang dari penjual ke kurir
order_delivered_customer_date	Waktu tanggal barang sampai ke pembeli
order_estimated_delivery_date	Waktu estimasi tanggal barang dari kurir sampai ke pembeli
shipping_limit_date	Waktu tanggal batas pengiriman barang dari penjual ke kurir
order_approved_at	Waktu tanggal verifikasi transaksi oleh sistem

Nama kolom	Keterangan
customer_zip_code_prefix	Awalan kode pos pembeli
geolocation_zip_code_prefix	Awalan kode pos lokasi pembeli
geolocation_lat	Koordinat X lokasi pembeli
geolocation_lng	Koordinat Y lokasi pembeli
order_item_id	Jumlah barang yang dibeli
price	Harga barang
freight_value	Biaya pengiriman
payment_sequential	Urutan pembayaran
payment_installments	Tahap pembayaran
payment_value	Total harga yang dibayar
product_name_lenght	Panjang nama produk
product_description_length	Panjang deskripsi produk
product_photos_qty	Jumlah foto produk

Data tipe Kontinu/Numerik

Nama Kolom	Keterangan
product_weight_g	Ukuran berat produk
product_length_cm	Ukuran panjang produk
product_height_cm	Ukuran tinggi produk
product_width_cm	Ukuran lebar produk
seller_zip_code_prefix	Awalan kode pos penjual
label	Rating bintang 5 atau tidak

Nama Kolom	Keterangan
customer_id	Kode unik pembeli dalam pesanan
customer_unique_id	Kode unik pembeli
customer_city	Kota asal pembeli
customer_state	Negara asal pembeli
geolocation_city	Lokasi kota asal pembeli
geolocation_state	Lokasi negara asal pembeli
order_id	Kode unik pesanan
order_status	Status pesanan
product_id	Kode unik produk
seller_id	Kode unik penjual
payment_type	Tipe pembayaran
product_category_name	Nama kategori produk dalam Bahasa Spanyol
seller_city	Kota asal penjual

Data tipe Kategorik/Objek

Nama Kolom	Keterangan
seller_state	Negara asal penjual
product_category_name_english	Nama kategori produk dalam Bahasa Inggris

Kolom terdapat Missing Values

Nama Kolom	Jumlah missing	Persentase (%)
order_approved_at	2	0.029351
product_category_name	95	1.394188
product_name_length	95	1.394188
product_description_length	95	1.394188
product_photos_qty	95	1.394188
product_category_name_english	96	1.408864

DATA UNDERSTANDING

```
data1_num.describe().T
```

	count	mean	std	min	25%	50%	75%	max
customer_zip_code_prefix	6814.0	33074.708248	27186.295524	1151.000000	13087.750000	24030.000000	38500.000000	99950.000000
geolocation_zip_code_prefix	6814.0	33074.708248	27186.295524	1151.000000	13087.750000	24030.000000	38500.000000	99950.000000
geolocation_lat	6814.0	-21.936902	4.408352	-33.519139	-23.600512	-22.913169	-20.421852	2.836945
geolocation_lng	6814.0	-45.901907	3.541532	-72.685562	-47.567456	-46.454536	-43.356398	-34.820877
order_item_id	6814.0	1.197388	0.657077	1.000000	1.000000	1.000000	1.000000	17.000000
price	6814.0	119.431415	181.548486	2.200000	39.990000	74.900000	134.900000	3980.000000
freight_value	6814.0	19.559297	13.698708	0.000000	13.612500	16.330000	20.907500	192.840000
payment_sequential	6814.0	1.091576	0.638881	1.000000	1.000000	1.000000	1.000000	18.000000
payment_installments	6814.0	2.900205	2.731707	1.000000	1.000000	1.000000	4.000000	24.000000
payment_value	6814.0	172.836651	258.207677	0.010000	61.595000	109.345000	192.277500	7274.880000
product_name_lenght	6719.0	48.896711	9.876962	12.000000	43.000000	52.000000	57.000000	64.000000
product_description_lenght	6719.0	770.090341	639.562855	8.000000	339.000000	589.000000	974.000000	3963.000000
product_photos_qty	6719.0	2.167584	1.715007	1.000000	1.000000	1.000000	3.000000	18.000000
product_weight_g	6814.0	2133.938215	3735.769992	0.000000	300.000000	741.500000	1825.000000	40425.000000
product_length_cm	6814.0	30.510420	16.452926	7.000000	18.000000	25.000000	38.000000	105.000000
product_height_cm	6814.0	16.634723	13.606891	2.000000	8.000000	13.000000	20.000000	105.000000
product_width_cm	6814.0	23.314500	11.937758	8.000000	15.000000	20.000000	30.000000	105.000000
seller_zip_code_prefix	6814.0	25225.881127	28184.961478	1021.000000	6871.000000	14020.000000	30882.000000	99730.000000
label	6814.0	0.586440	0.492508	0.000000	0.000000	1.000000	1.000000	1.000000

```
data1_cat.describe().T
```

	count	unique	top	freq
customer_id	6814	6285	5f79f446984a2ba4f1cb86bf26f3ef64	5
customer_unique_id	6814	6248	79b23d9300163e8db6ae01c4d2eb3394	5
customer_city	6814	866	rio de janeiro	859
customer_state	6814	27	SP	2532
geolocation_city	6814	980	rio de janeiro	859
geolocation_state	6814	27	SP	2532
order_id	6814	6285	37d928cc42067f69b394747c5c71a260	5
order_status	6814	1	delivered	6814
order_purchase_timestamp	6814	6278	2017-05-25 22:27:50	5
order_approved_at	6812	6239	2017-08-08 16:15:18	5
order_delivered_carrier_date	6814	6177	2017-08-10 11:58:14	5
order_delivered_customer_date	6814	6283	2017-08-11 21:55:50	5
order_estimated_delivery_date	6814	410	2018-03-15 00:00:00	41
product_id	6814	4566	99a4788cb24856965c36a24e339b6058	36
seller_id	6814	1402	4a3ca9315b744ce9f8e9374361493884	137
shipping_limit_date	6814	6268	2017-08-14 20:43:31	5
payment_type	6814	4	credit_card	4991
product_category_name	6719	68	cama_mesa_banho	761
seller_city	6814	367	sao paulo	1669
seller_state	6814	17	SP	4787
product_category_name_english	6718	67	bed_bath_table	761

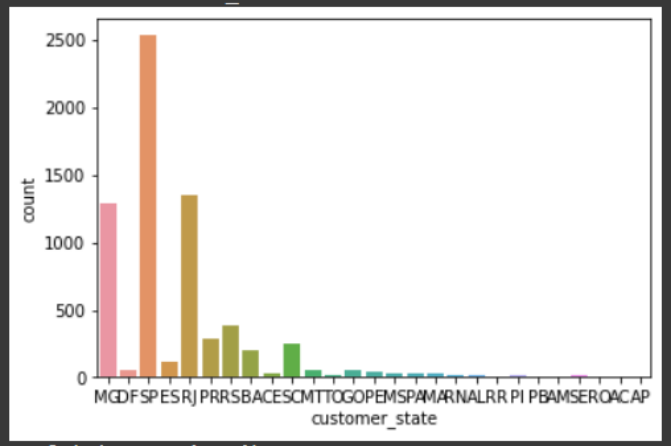
EXPLORATORY DATA ANALYSIS

- Sebelum data dilakukan EDA, beberapa kolom yang kurang diperlukan dibuang terlebih dahulu (secara total ada 20 kolom yang dibuang bersama dengan kolom tipe datetime).
- Dimensi ukuran data yang dilakukan EDA tersisa menjadi 20 kolom yang terbagi menjadi 16 kolom data kontinu dan 4 kolom data kategorik.

Alasan di drop	Nama Kolom
Hanya berisi nilai kode unik	'customer_id', 'customer_unique_id', 'order_id', 'product_id', 'seller_id',
Nilai dalam kolom hanya satu jenis kategorik	'order_status',
Sudah diwakili oleh fitur lain, seperti kota dan negara (kota dan negara dipilih salah satu, dalam kasus ini negara dipilih karena jumlah nilai uniknya lebih sedikit sehingga lebih sederhana)	'geolocation_city', 'geolocation_state', 'geolocation_zip_code_prefix', 'geolocation_lat', 'geolocation_lng',
Sudah dipilih fitur negara (diwakilkan)	'customer_city', 'seller_city',
Sudah dipilih fitur kategori dalam Bahasa inggris agar memudahkan intrepetasi	'product_category_name'
Fitur non numerik ataupun kategorik	Seluruh data tipe datetime

EXPLORATORY DATA ANALYSIS

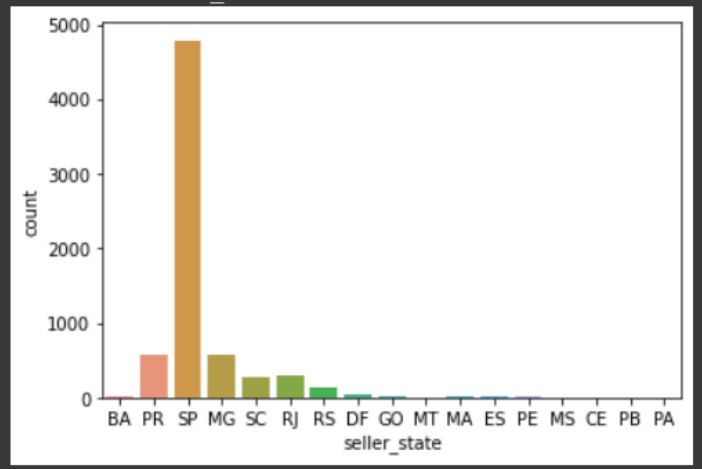
Kolom : customer_state



Jumlah kategori unik : 27

Negara asal pembeli	Jumlah data
SP	2532
RJ	1347
MG	1284
RS	387
PR	282

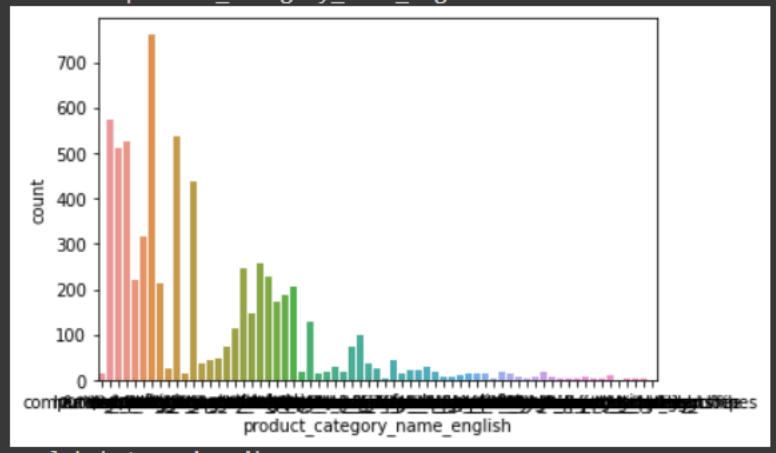
Kolom : seller_state



Jumlah kategori unik : 17

Negara asal penjual	Jumlah data
SP	4787
PR	570
MG	568
RJ	296
SC	268

Kolom : product_category_name_english

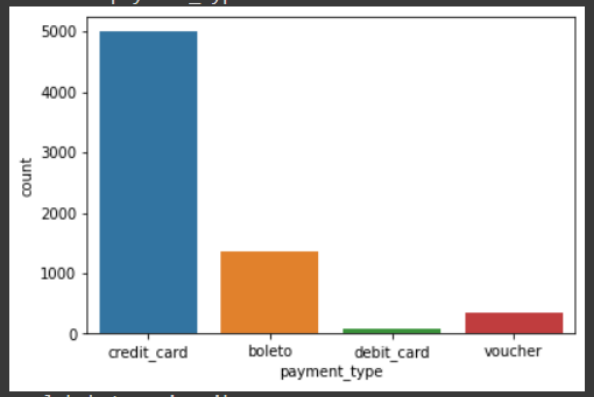


Jumlah kategori unik : 68

Jumlah total nilai kategori unik :

bed_bath_table	761
health_beauty	572
sports_leisure	536
furniture_decor	527
computers_accessories	510
	...

Kolom : payment_type



Jumlah kategori unik : 4
Jumlah nilai kategori unik : ['credit_card', 'boleto', 'debit_card', 'voucher']
Jumlah total nilai kategori unik :

credit_card	4991
boleto	1369
voucher	360
debit_card	94

EXPLORATORY DATA ANALYSIS

Data secara keseluruhan

Price	Nilai
Mean	$ 119.43
Median	**$ 74.90**
Min	$ 2.20
Max	$ 3980

Freight value	Nilai
Mean	$ 19.559
Median	**$ 16.33**
Min	$ 0
Max	$ 192.84

Payment value	Nilai
Mean	$ 172.836
Median	**$ 109.345**
Min	$ 0
Max	$ 7274.88

Secara keseluruhan, median dari harga total transaksi adalah sekitar $109, dengan rincian sekitar $74 harga barang dan $16 biaya pengiriman.

Name Lenght	Nilai
Mean	48.89
Median	**52**
Min	12
Max	64

Desc Lenght	Nilai
Mean	770.09
Median	**589**
Min	8
Max	3963

Photos Qty	Nilai
Mean	2.16
Median	**1**
Min	1
Max	18

Secara keseluruhan, median dari jumlah karakter nama produk adalah 52 huruf, deskripsi produk sebesar 589 huruf, dan satu jumlah foto produk.

EXPLORATORY DATA ANALYSIS

Data pada kondisi seller dengan rate 5*

Price	Nilai
Mean	$ 118.617
Median	**$ 69.995**
Min	$ 2.20
Max	$ 3980

Freight value	Nilai
Mean	$ 19.24
Median	**$ 16.11**
Min	$ 0
Max	$ 177.98

Payment value	Nilai
Mean	$ 159.088
Median	**$ 102.425**
Min	$ 0
Max	$ 4042.740

Median dari harga total transaksi, biaya pengiriman, dan harga barang cenderung lebih rendah dari data secara keseluruhan.

Name Lenght	Nilai
Mean	48.74
Median	52
Min	12
Max	64

Desc Lenght	Nilai
Mean	767.85
Median	589
Min	8
Max	3963

Photos Qty	Nilai
Mean	2.15
Median	1
Min	1
Max	14

Median dari jumlah karakter nama produk, deskripsi produk, dan jumlah foto produk tidak berbeda dengan data keseluruhan.

EXPLORATORY DATA ANALYSIS

Data pada kondisi seller tanpa rate 5*

Price	Nilai
Mean	$ 120.58
Median	**$ 78**
Min	**$ 4.20**
Max	$ 3109.99

Freight value	Nilai
Mean	$ 20
Median	**$ 16.79**
Min	$ 0
Max	$ 192.84

Payment value	Nilai
Mean	$ 192.331
Median	**$ 119.065**
Min	$ 0
Max	$ 7274.88

Median dari harga total transaksi, biaya pengiriman, dan harga barang **cenderung lebih tinggi** dari data secara keseluruhan.

Name Lenght	Nilai
Mean	49.21
Median	52
Min	12
Max	64

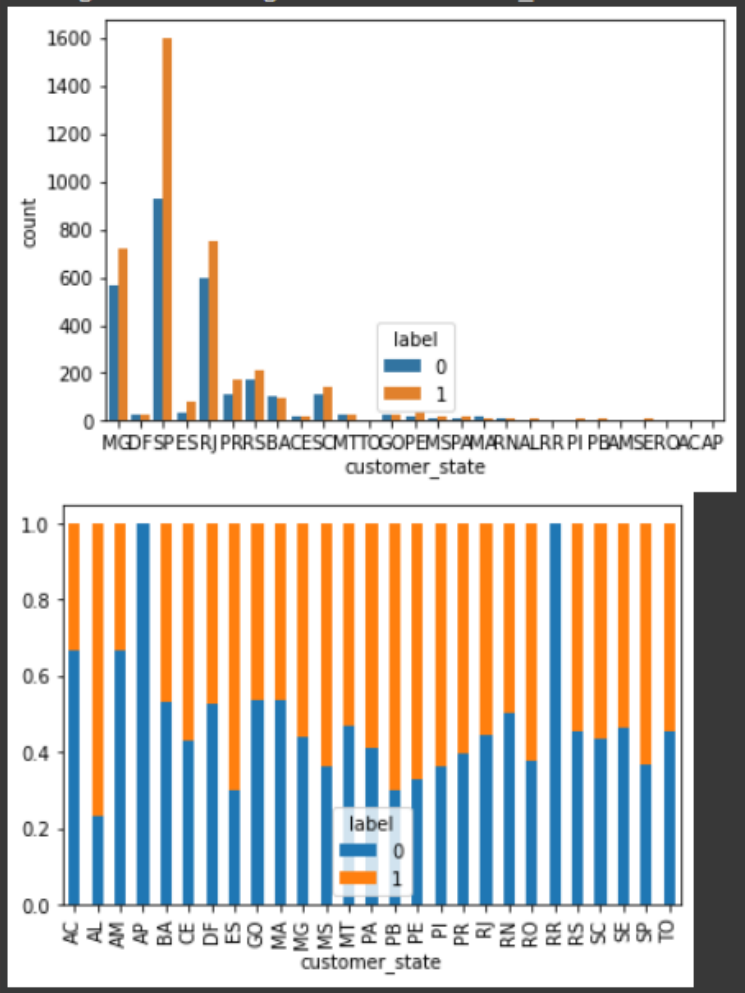
Desc Lenght	Nilai
Mean	767.15
Median	582.5
Min	40
Max	3921

Photos Qty	Nilai
Mean	2.14
Median	1
Min	1
Max	18

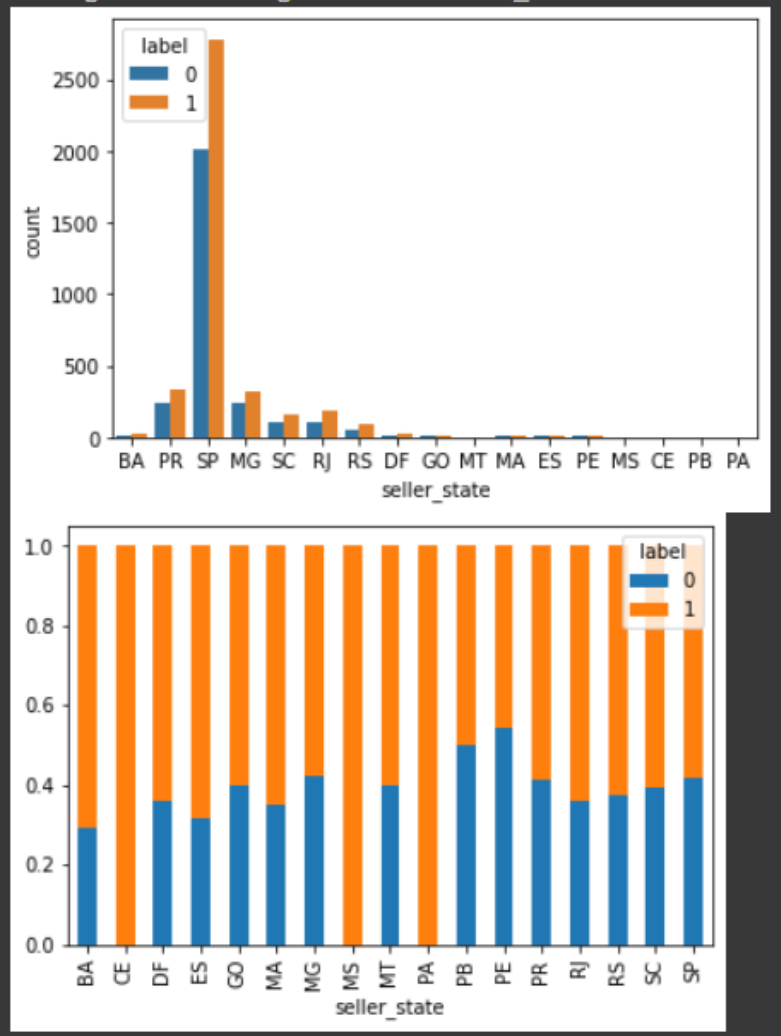
Jumlah karakter pada deskripsi produk untuk seller yang tidak mendapat bintang 5 cenderung lebih banyak ketimbang dengan seller dengan bintang 5.

EXPLORATORY DATA ANALYSIS

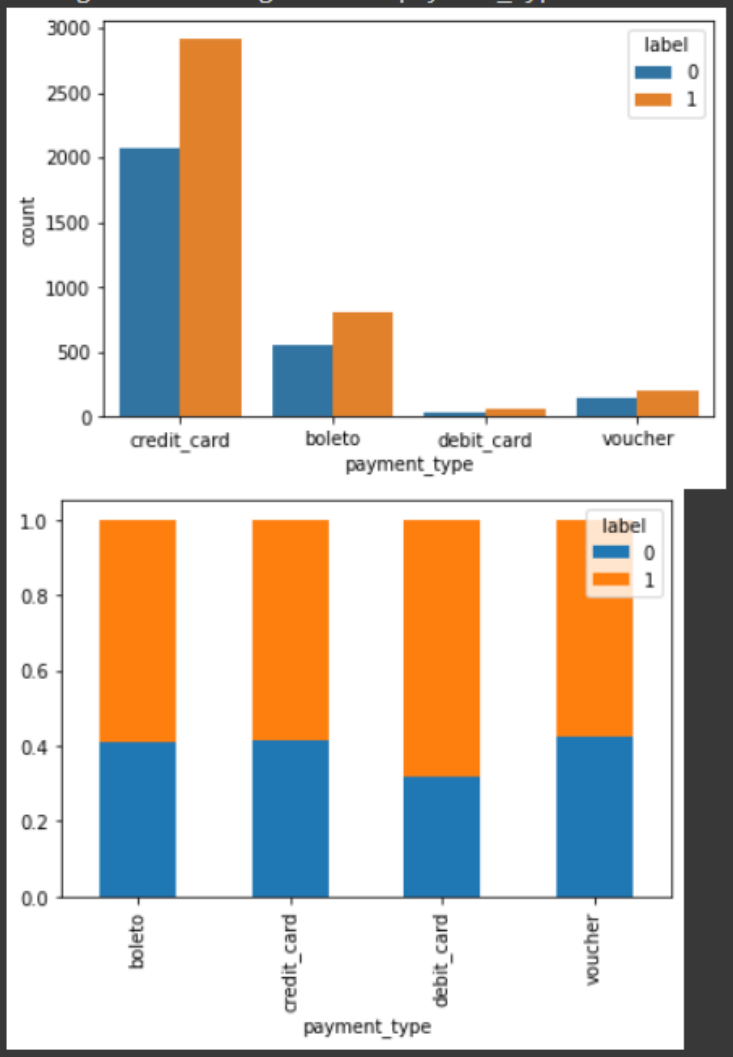
Hubungan label dengan kolom customer_state



Hubungan label dengan kolom seller_state



Hubungan label dengan kolom payment_type



INSIGHT

1. Kisaran harga barang yang dijual berada dalam interval $2.2 - $3980 dengan rata-rata sebesar $119.43 (belum termasuk biaya pengiriman kapal)
2. Rata-rata transaksi customer menghabiskan total sekitar $172.83 dengan rata-rata biaya pengiriman sebesar $19.55
3. Pembayaran transaksi paling banyak digunakan adalah kartu kredit dengan jenis barang paling banyak dibeli adalah bed_bath_table
4. Customer yang berdomisili di negara AP dan RR belum pernah memberikan review bintang 5 ketika membeli barang
5. Seller yang berdomisili di negara CE dan PA belum pernah mendapatkan review dibawah bintang 5

03

Data Preparation

Final Project:
Binary Classification
FIVE STAR PRODUCT RATING CLASSIFICATION

DATA PREPARATION

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6814 entries, 0 to 6813
Data columns (total 26 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   customer_zip_code_prefix       6814 non-null   int64
 1   customer_state                 6814 non-null   object
 2   order_purchase_timestamp       6814 non-null   datetime64[ns]
 3   order_approved_at              6812 non-null   datetime64[ns]
 4   order_delivered_carrier_date   6814 non-null   datetime64[ns]
 5   order_delivered_customer_date  6814 non-null   datetime64[ns]
 6   order_estimated_delivery_date  6814 non-null   datetime64[ns]
 7   order_item_id                  6814 non-null   int64
 8   shipping_limit_date            6814 non-null   datetime64[ns]
 9   price                          6814 non-null   float64
 10  freight_value                  6814 non-null   float64
 11  payment_sequential             6814 non-null   int64
 12  payment_type                   6814 non-null   object
 13  payment_installments           6814 non-null   int64
 14  payment_value                  6814 non-null   float64
 15  product_name_lenght            6719 non-null   float64
 16  product_description_lenght     6719 non-null   float64
 17  product_photos_qty             6719 non-null   float64
 18  product_weight_g               6814 non-null   int64
 19  product_length_cm              6814 non-null   int64
 20  product_height_cm              6814 non-null   int64
 21  product_width_cm               6814 non-null   int64
 22  seller_zip_code_prefix         6814 non-null   int64
 23  seller_state                   6814 non-null   object
 24  product_category_name_english  6718 non-null   object
 25  label                          6814 non-null   int64
dtypes: datetime64[ns](6), float64(6), int64(10), object(4)
memory usage: 1.4+ MB
```

```
df.isna().sum()
```

```
customer_zip_code_prefix          0
customer_state                    0
order_purchase_timestamp          0
order_approved_at                 2
order_delivered_carrier_date      0
order_delivered_customer_date     0
order_estimated_delivery_date     0
order_item_id                     0
shipping_limit_date               0
price                             0
freight_value                     0
payment_sequential                0
payment_type                      0
payment_installments              0
payment_value                     0
product_name_lenght              95
product_description_lenght       95
product_photos_qty               95
product_weight_g                  0
product_length_cm                 0
product_height_cm                 0
product_width_cm                  0
seller_zip_code_prefix            0
seller_state                      0
product_category_name_english    96
label                             0
system_approve_time               0
seller_response_time              0
delivery_time                     0
delivery_on_time                  0
shipping_date_on_time             0
dtype: int64
```

```
df.isna().sum()/len(df)*100
```

```
customer_zip_code_prefix         0.000000
customer_state                   0.000000
order_purchase_timestamp         0.000000
order_approved_at                0.029351
order_delivered_carrier_date     0.000000
order_delivered_customer_date    0.000000
order_estimated_delivery_date    0.000000
order_item_id                    0.000000
shipping_limit_date              0.000000
price                            0.000000
freight_value                    0.000000
payment_sequential               0.000000
payment_type                     0.000000
payment_installments             0.000000
payment_value                    0.000000
product_name_lenght              1.394188
product_description_lenght       1.394188
product_photos_qty               1.394188
product_weight_g                 0.000000
product_length_cm                0.000000
product_height_cm                0.000000
product_width_cm                 0.000000
seller_zip_code_prefix           0.000000
seller_state                     0.000000
product_category_name_english    1.408864
label                            0.000000
system_approve_time              0.000000
seller_response_time             0.000000
delivery_time                    0.000000
delivery_on_time                 0.000000
shipping_date_on_time            0.000000
dtype: float64
```

DATA PREPARATION

Metode penanganan missing value :

1. Missing value ≤10% : gunakan simple imputer yaitu median untuk kolom numerik dan modus untuk kolom kategorik.
2. Missing value >10% dan ≤30% : gunakan KNNImputer.
3. Missing value >30% : drop kolom.

- Karena keseluruhan nilai missing value tidak melebihi 10%, maka tindakan yang dilakukan hanyalah menggunakan Simple Imputer.
- Tahapannya adalah dengan memisahkan data kontinu dan kategorikal dahulu, kemudian dilakukan onehotencoder dan diimpute.

04

Feature Engineering

Final Project:
Binary Classification
FIVE STAR PRODUCT RATING CLASSIFICATION

FEATURE ENGINEERING

Nama fitur	Tipe data	Keterangan
system_approve_time	float	waktu sistem konfirmasi pesanan dalam satuan hari
seller_response_time	float	waktu sseller mengirimkan barang ke kapal dalam satuan hari
delivery_time	float	waktu pengiriman barang hingga ke konsumen dalam satuan hari
delivery_on_time	Int	ketepatan waktu pengiriman atau tidak (0,1) [0=telat, 1=tepat]
shipping_date_on_time	int	keterlambatan pengiriman barang ke kapal atau tidak (0,1) [0=telat, 1=tepat]

```
# Berdasarkan fitur waktu/datetime, maka dapat diperoleh 5 fitur baru sebagai berikut :

df['system_approve_time'] = (df.order_approved_at - df.order_purchase_timestamp) / pd.Timedelta(days=1)
df['seller_response_time'] = (df.order_delivered_carrier_date - df.order_approved_at) / pd.Timedelta(days=1)
df['delivery_time'] = (df.order_delivered_customer_date - df.order_delivered_carrier_date) / pd.Timedelta(days=1)
df['delivery_on_time'] = df.apply(lambda x : 1 if x['order_estimated_delivery_date'] > x['order_delivered_customer_date'] else 0,axis=1)
df['shipping_date_on_time'] = df.apply(lambda x : 1 if x['shipping_limit_date'] > x['order_delivered_carrier_date'] else 0,axis=1)
```

Pesanan rata-rata (median) diverifikasi sistem cukup cepat, yakni 0.014 hari atau sekitar 20 menit; namun terdapat proses verifikasi terlama yakni 12 hari.

Respon penjual dalam mengirimkan barang ke kurir pengiriman kapal rata-rata membutuhkan waktu hampir 3 hari, dengan lama waktu pengiriman dari kapal sekitar 9 hari.

FEATURE ENGINEERING

- Untuk mengurangi jumlah kolom pada saat OHE, maka negara asal penjual dan pembeli diubah menjadi integer berdasarkan peringkat rasio label 1.
- Keseluruhan data negara asal penjual dan pembeli tersebut ditransformasi dengan menggunakan fungsi map() dan dictionary yang telah dibuat berdasarkan tabel urutan peringkat disamping.

```
# Menyimpan data ranking kedalam dict

df_st_cons =  state_consumer.loc[:,['customer_state','Rank']]
df_st_sell =  state_seller.loc[:,['seller_state','Rank']]
dict_st_cons = dict(df_st_cons.values)
dict_st_sell = dict(df_st_sell.values)
```

```
# Mentransformasi negara asal penjual dan pembeli

df_concat_final['customer_state'] = df_concat_final['customer_state'].map(dict_st_cons)
df_concat_final['seller_state'] = df_concat_final['seller_state'].map(dict_st_sell)
```

	seller_state	label_1	label_0	Ratio	Rank
0	PE	11	13.0	0.458333	1
1	PB	2	2.0	0.500000	2
2	MG	327	241.0	0.575704	3
3	SP	2778	2009.0	0.580322	4
4	PR	334	236.0	0.585965	5
5	GO	18	12.0	0.600000	6
6	MT	3	2.0	0.600000	7
7	SC	162	106.0	0.604478	8
8	RS	86	51.0	0.627737	9
9	DF	30	17.0	0.638298	10
10	RJ	190	106.0	0.641892	11
11	MA	15	8.0	0.652174	12
12	ES	13	6.0	0.684211	13
13	BA	22	9.0	0.709677	14
14	CE	2	NaN	NaN	15
15	MS	2	NaN	NaN	16
16	PA	1	NaN	NaN	17

	customer_state	label_1	label_0	Ratio	Rank
0	AC	1.0	2	0.333333	1
1	AM	1.0	2	0.333333	2
2	GO	25.0	29	0.462963	3
3	MA	13.0	15	0.464286	4
4	BA	94.0	107	0.467662	5
5	DF	26.0	29	0.472727	6
6	RN	9.0	9	0.500000	7
7	MT	26.0	23	0.530612	8
8	SE	7.0	6	0.538462	9
9	TO	6.0	5	0.545455	10
10	RS	212.0	175	0.547804	11
11	RJ	751.0	596	0.557535	12
12	MG	720.0	564	0.560748	13
13	SC	141.0	109	0.564000	14
14	CE	20.0	15	0.571429	15
15	PA	20.0	14	0.588235	16
16	PR	170.0	112	0.602837	17
17	RO	5.0	3	0.625000	18
18	SP	1599.0	933	0.631517	19
19	PI	7.0	4	0.636364	20
20	MS	16.0	9	0.640000	21
21	PE	31.0	15	0.673913	22
22	ES	79.0	34	0.699115	23
23	PB	7.0	3	0.700000	24
24	AL	10.0	3	0.769231	25
25	RR	NaN	1	NaN	26
26	AP	NaN	1	NaN	27

FEATURE ENGINEERING

```
df_concat_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6814 entries, 0 to 6813
Data columns (total 25 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   customer_state                 6814 non-null   int64  
 1   payment_type                   6814 non-null   object 
 2   seller_state                   6814 non-null   int64  
 3   product_category_name_english  6814 non-null   object 
 4   customer_zip_code_prefix       6814 non-null   int64  
 5   order_item_id                  6814 non-null   int64  
 6   price                          6814 non-null   float64
 7   freight_value                  6814 non-null   float64
 8   payment_sequential             6814 non-null   int64  
 9   payment_installments           6814 non-null   int64  
 10  payment_value                  6814 non-null   float64
 11  product_name_lenght            6814 non-null   float64
 12  product_description_lenght     6814 non-null   float64
 13  product_photos_qty             6814 non-null   float64
 14  product_weight_g               6814 non-null   int64  
 15  product_length_cm              6814 non-null   int64  
 16  product_height_cm              6814 non-null   int64  
 17  product_width_cm               6814 non-null   int64  
 18  seller_zip_code_prefix         6814 non-null   int64  
 19  label                          6814 non-null   int64  
 20  system_approve_time            6814 non-null   float64
 21  seller_response_time           6814 non-null   float64
 22  delivery_time                  6814 non-null   float64
 23  delivery_on_time               6814 non-null   int64  
 24  shipping_date_on_time          6814 non-null   int64  
dtypes: float64(9), int64(14), object(2)
memory usage: 1.3+ MB
```

```
# Melakukan pemisahan variabel X dan Y

X = df_concat_final.drop(['label'],axis=1)
Y = df_concat_final['label']
```

```
# Melihat rasio label 0 dan 1

Y.value_counts(normalize=True)
```

```
1    0.58644
0    0.41356
Name: label, dtype: float64
```

```
# Melakukan pemisahan train test split

X_train,X_test,y_train,y_test = train_test_split(X,Y,test_size=0.2,stratify=Y,random_state=42)
```

```
# Melihat ukuran dataset dari train test

print('X train :',X_train.shape)
print('y train :',y_train.shape)
print('X test :',X_test.shape)
print('y test :',y_test.shape)
```

```
X train : (5451, 24)
y train : (5451,)
X test : (1363, 24)
y test : (1363,)
```

FEATURE ENGINEERING

- Setelah dipisahkan train dan test, selanjutnya dilakukan label encoding dan scaling numerik dengan OneHotEncoder dan RobustScaler.
- Fitur pada data kemudian dieliminasi dengan pengecekan multikolinearitas (r >= 0.7) sehingga terdapat fitur yang dibuang.
- Selanjutnya fitur dipilih lagi menggunakan KBest dengan random seed 42 dan nilai K=15 fitur sehingga menyederhanakan input fitur dalam model.
- Terakhir, data test ditransformasi mengikuti data train dan dipilih fitur apa saja yang akan dipakai berdasarkan seleksi fitur Kbest.

```
# Melihat list feature yang didrop

list_drop
```

```
['price', 'payment_sequential', 'payment_type_boleto']
```

```
# Melihat feature yang digunakan dengan Kbest dan mutual info classifier

final_fitur = X_train_concat.columns[kbest.get_support()]
final_fitur
```

```
Index(['customer_zip_code_prefix', 'freight_value', 'payment_value',
       'product_description_lenght', 'product_weight_g',
       'seller_zip_code_prefix', 'system_approve_time', 'seller_response_time',
       'delivery_time', 'payment_type_debit_card',
       'product_category_name_english_diapers_and_hygiene',
       'product_category_name_english_kitchen_dining_laundry_garden_furniture',
       'product_category_name_english_musical_instruments',
       'product_category_name_english_sports_leisure',
       'product_category_name_english_watches_gifts'],
      dtype='object')
```

05
Modelling

Final Project:
Binary Classification
FIVE STAR PRODUCT RATING CLASSIFICATION

MODELLING

- Pada tahap ini, secara keseluruhan semua model klasifikasi diikut sertakan (total terdapat 6 model) dengan parameter default.
- Selanjutnya, keseluruhan model akan dilakukan tuning parameter berdasarkan urutan hasil peforma model.

```
# Mendeklarasikan model-model klasifikasi

log = LogisticRegression(random_state=42)
dt = DecisionTreeClassifier(random_state=42)
rf = RandomForestClassifier(random_state=42)
ada = AdaBoostClassifier(random_state=42)
xg = XGBClassifier(random_state=42)
svc = SVC(random_state=42)
```

```
# Mendeklarasikan fungsi yang dibutuhkan

def classif(estimator, x, y):
    y_pred = estimator.predict(x)
    print(classification_report(y, y_pred, labels=[1,0]))

def list_metrcis_model(model,y_test, y_pred):
    from sklearn import metrics
    report = metrics.classification_report(y_test, y_pred, output_dict=True)
    df_classification_report = pd.DataFrame(report)
    df_classification = df_classification_report.iloc[0:3,0:3]
    preci = df_classification.iloc[0,1]
    reca = df_classification.iloc[1,1]
    tnr = df_classification.iloc[1,0]
    fpr = 1-tnr
    npv = df_classification.iloc[0,0]
    f1 = df_classification.iloc[2,1]
    acc = df_classification.iloc[0,2]
    output_list = [model,preci,reca,tnr,fpr,npv,f1,acc]
    return output_list
```

MODELLING

Model		Precision	Recall	TNR	FPR	NPV	F1 Score	Accuracy
Log	Train	63.63	88.36	28.35	71.65	63.20	73.98	63.55
	Test	62.68	88.49	25.35	74.65	60.85	73.38	62.36
DT	Train	100.00	99.94	100.00	0.00	99.91	99.97	99.96
	Test	65.94	64.21	53.01	46.99	51.11	65.06	59.57
RF	Train	99.97	99.97	99.96	0.04	99.96	99.97	99.96
	Test	67.00	83.85	41.49	58.51	64.46	74.49	66.32
Ada	Train	64.70	89.30	30.88	69.12	67.05	75.03	65.14
	Test	62.58	88.74	24.82	75.18	60.87	73.40	62.29
XG	Train	67.20	92.27	36.11	63.89	76.72	77.76	69.05
	Test	63.93	90.49	27.66	72.34	67.24	74.92	64.49
SVC	Train	64.02	95.50	23.87	76.13	78.89	76.65	65.88
	Test	62.51	94.12	20.04	79.96	70.63	75.12	63.46

Berdasarkan hasil perbandingan kesluruhan model klasifikasi default, akan dipilih satu model untuk dilakukan tunning optimasi.

- Kriteria model : Recall, Precision, TNR > 0.7
- Model tidak underfit ataupun overfit

Berdasarkan hasil tersebut secara keseluruhan belum ada memenuhi, sehingga akan dilakukan tuning seluruhnya dengan urutan sebagai berikut:

1. XGBoost
2. AdaBoost
3. Decission Tree
4. Random Forest
5. Log
6. SVC

MODELLING

Model		Precision	Recall	TNR	FPR	NPV	F1 Score	Accuracy
Log	Train	67.05	61.68	57.01	42.99	51.20	64.26	59.75
	Test	65.11	60.95	53.72	46.28	49.27	62.96	57.96
DT	Train	64.72	85.08	34.21	65.79	61.78	73.51	64.04
	Test	63.04	84.73	29.61	70.39	57.79	72.29	61.92
RF	Train	64.80	86.36	33.45	66.55	63.36	74.04	64.48
	Test	63.51	86.48	29.61	70.39	60.73	73.24	62.95
Ada	Train	67.47	88.11	39.75	60.25	70.22	76.42	68.12
	Test	63.16	84.98	29.79	70.21	58.33	72.47	62.14
XG	Train	96.60	98.69	95.08	4.92	98.08	97.63	97.19
	Test	67.53	75.22	48.76	51.24	58.14	71.17	64.27
SVC	Train	76.78	97.00	58.39	41.61	93.20	85.71	81.03
	Test	64.06	86.98	30.85	69.15	62.59	73.78	63.76

Hasil metrics model optimum masih belum ada yang memenuhi seluruh kriteria model (recall,precission,TNR masih dibawah 0.7)
Jika diurutkan berdasarkan model optimum yang paling mendekati adalah :

1. XGBoost optimal (cenderung overfit)
2. Logistic optimal (tuning dengan ROC AUC Curve)
3. SVC optimal dengan kernel rbf
4. AdaBoost optimal
5. Random Forest optimal
6. Decission Tree optimal

MODELLING

Model XGBoost optimal menggunakan parameter sebagai berikut:

```
[ ]  # Mendefiniskan dan fitting model xgboost optimum

     xg_opt = XGBClassifier(n_estimators=200,
                            learning_rate=0.7,
                            max_depth=4,
                            gamma=1,
                            reg_lambda=0.7,
                            random_state=42)

     xg_opt.fit(X_train_final, y_train)

     y_train_pred_xg_opt = xg_opt.predict(X_train_final)
     y_test_pred_xg_opt = xg_opt.predict(X_test_final)
```

```
[ ]  # Melihat hasil fitting model xg default pada data train

     print(classification_report(y_train, y_train_pred_xg_opt, labels=[1,0]))
```

	precision	recall	f1-score	support
1	0.97	0.99	0.98	3197
0	0.98	0.95	0.97	2254
accuracy			0.97	5451
macro avg	0.97	0.97	0.97	5451
weighted avg	0.97	0.97	0.97	5451

```
[ ]  # Melihat hasil fitting model xg default pada data test

     print(classification_report(y_test, y_test_pred_xg_opt, labels=[1,0]))
```

	precision	recall	f1-score	support
1	0.68	0.75	0.71	799
0	0.58	0.49	0.53	564
accuracy			0.64	1363
macro avg	0.63	0.62	0.62	1363
weighted avg	0.64	0.64	0.64	1363

06

Evaluation

Final Project:
Binary Classification
FIVE STAR PRODUCT RATING CLASSIFICATION

EVALUATION

- Model XGBoost optimal memiliki nilai metrics paling mendekati kriteria model yang diiginkan sehingga dipilih sebagai model terbaik untuk memprediksi dataset baru.
- Model XGBoost optimal (tuned) tersebut kemudian digunakan untuk memprediksi data model_backtest_set yang telah disesuaikan isi kolom fiturnya dengan model.
- Hasil prediksi tersebut ditemukan sebanyak 1879 data diberi label 1 (diberikan rate *5) dan sebanyak 1042 data dilabeli 0.

```
[ ]  # Prediksi hasil backtest
     predicted_data_xg = xg_opt.predict(backtest_final)
     predicted_data_xg
```

```
array([1, 1, 1, ..., 1, 1, 1])
```

```
[ ]  # konversi array menjadi dataframe

     df_predicted_xg = pd.DataFrame(data=predicted_data_xg,columns=['predicted'])
```

```
[ ]  df_predicted_xg.value_counts()
```

```
predicted
1            1879
0            1042
dtype: int64
```

CONCLUSION

Model terbaik dalam kasus ini adalah model **Logistic Regression** yang telah dioptimasi dengan peforma precision, recall, TNR, FPR secara berturut-turut adalah 67%, 75%, 48%, dan 51%.

RECOMMENDATION

Untuk meningkatkan peforma model, perlu dilakukan feature engineering (missal resampling) atau tuning hyperparameter kembali dengan menggunakan GridSearchCV.

THANK YOU!