

**SHARING VISION  
DATA SCIENCE BOOTCAMP**

# **FINAL PROJECT**

**TRIYOZA APRIANDA**

Menggunakan Metodologi CRISP-DM  
untuk Model Klasifikasi Memprediksi  
*Buyer Rating* di Suatu Marketplace

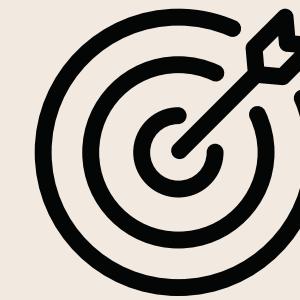
- Menggunakan metode CRISP DM untuk masalah klasifikasi memprediksi kolom 'label' (rating yang diberikan oleh customer) pada dataset suatu marketplace

- METODE CRISP DM  
*(The CRoss Industry Standard Process for Data Mining)*, terdiri dari beberapa tahapan:

- ▶ BUSSINESS UNDERSTANDING
- ▶ DATA UNDERSTANDING
- ▶ DATA PREPARATION
- ▶ FEATURE ENGINEERING
- ▶ MODELING
- ▶ EVALUASI

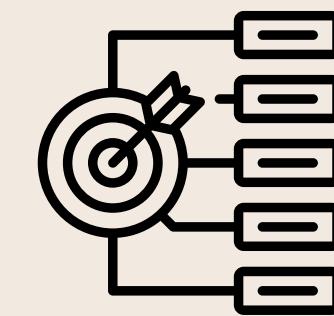


# Bussines Understanding



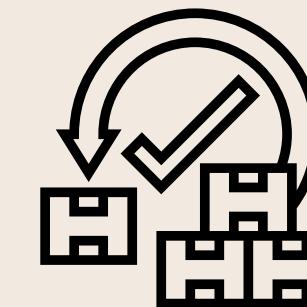
## BUSSINESS OBJECTIVES

Suatu perusahaan marketplace ingin membuat guideline berisi tips untuk seller bagaimana agar mendapatkan rating 5 dari buyer.



## MODEL OBJECTIVES

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

**Recall > 0.6; Precision > 0.6; FPR < 0.45**

Model yang dibuat mencapai atau bahkan melebihi model success criteria. Apabila tidak berhasil, pilih model dengan performace terbaik.)

# Data Understanding

## ◎ Data Description

Data yang digunakan:

- 'model\_development\_set.csv', digunakan untuk pembuatan model.
- 'back\_testing\_set.csv' sebagai dataset yang digunakan untuk menguji model yang dibuat dan memprediksi kolom 'label' (*buyer rating*), karena pada back\_testing\_set ini kolom 'label' disembunyikan oleh instruktor.

Data 'model\_development\_set.csv' terdiri dari:

- 13.645 baris
- 40 kolom/fitur

Dari 40 fitur, terbagi menjadi 3 tipe data:

- Numerik
- Kategorik
- Date Time



# ◎ Data Description

FITUR

NUMERIK

Fitur dengan tipe data numerik

Data columns (total 19 columns):			
#	Column	Non-Null Count	Dtype
0	customer_zip_code_prefix	13645	non-null float64
1	geolocation_zip_code_prefix	13645	non-null float64
2	geolocation_lat	13645	non-null float64
3	geolocation_lng	13645	non-null float64
4	order_item_id	13645	non-null float64
5	price	13645	non-null float64
6	freight_value	13645	non-null float64
7	payment_sequential	13644	non-null float64
8	payment_installments	13644	non-null float64
9	payment_value	13644	non-null float64
10	product_name_length	13453	non-null float64
11	product_description_length	13453	non-null float64
12	product_photos_qty	13453	non-null float64
13	product_weight_g	13640	non-null float64
14	product_length_cm	13640	non-null float64
15	product_height_cm	13640	non-null float64
16	product_width_cm	13640	non-null float64
17	seller_zip_code_prefix	13645	non-null float64
18	label	13645	non-null int64

dtypes: float64(18), int64(1)

→ Kode pos customer

→ Lokasi (longitude dan latitude)

→ Kode unik item yang dipesan

→ Harga item

→ Pembayaran

→ Berkaitan dengan detail produk (panjang karakter nama produk, deksripsi produk, jumlah foto produk, berat produk, dan dimensi produk: panjang, lebar, tinggi)

→ Kode pos penjual

→ Label untuk buyer rating

# ◎ Data Description

FITUR

KATEGORIK

Data columns (total 15 columns):			
#	Column	Non-Null Count	Dtype
0	customer_id	13645 non-null	object
1	customer_unique_id	13645 non-null	object
2	customer_city	13645 non-null	object
3	customer_state	13645 non-null	object
4	geolocation_city	13645 non-null	object
5	geolocation_state	13645 non-null	object
6	order_id	13645 non-null	object
7	order_status	13645 non-null	object
8	product_id	13645 non-null	object
9	seller_id	13645 non-null	object
10	payment_type	13644 non-null	object
11	product_category_name	13453 non-null	object
12	seller_city	13645 non-null	object
13	seller_state	13645 non-null	object
14	product_category_name_english	13451 non-null	object

→ ID dan kode unik customer

→ Alamat customer terdaftar dan alamat saat memesan yang terdeteksi

→ Kode unik pesanan,  
Status pesanan

→ Kode unik/id produk dan penjual

→ Jenis pembayaran

→ Nama produk

→ Kota dan state penjual

→ Nama produk (English)

# ◎ Data Description

FITUR

DATE TIME

#	Column	Non-Null Count	Dtype	
0	order_purchase_timestamp	13645	non-null	datetime64[ns]
1	order_approved_at	13642	non-null	datetime64[ns]
2	order_delivered_carrier_date	13645	non-null	datetime64[ns]
3	order_delivered_customer_date	13645	non-null	datetime64[ns]
4	order_estimated_delivery_date	13645	non-null	datetime64[ns]
5	shipping_limit_date	13645	non-null	datetime64[ns]
dtypes: datetime64[ns](6)				

0. Waktu pembelian
1. Waktu pesanan diaprove
2. Tanggal mulai dibawa kurir
3. Tanggal pesanan diterima customer
4. Tanggal estimasi sampai
5. Tanggal batas pengiriman

# ○ Exploratory Data Analysis (EDA)

- Fitur Numerik
- Fitur Kategorik
- Fitur Datetime



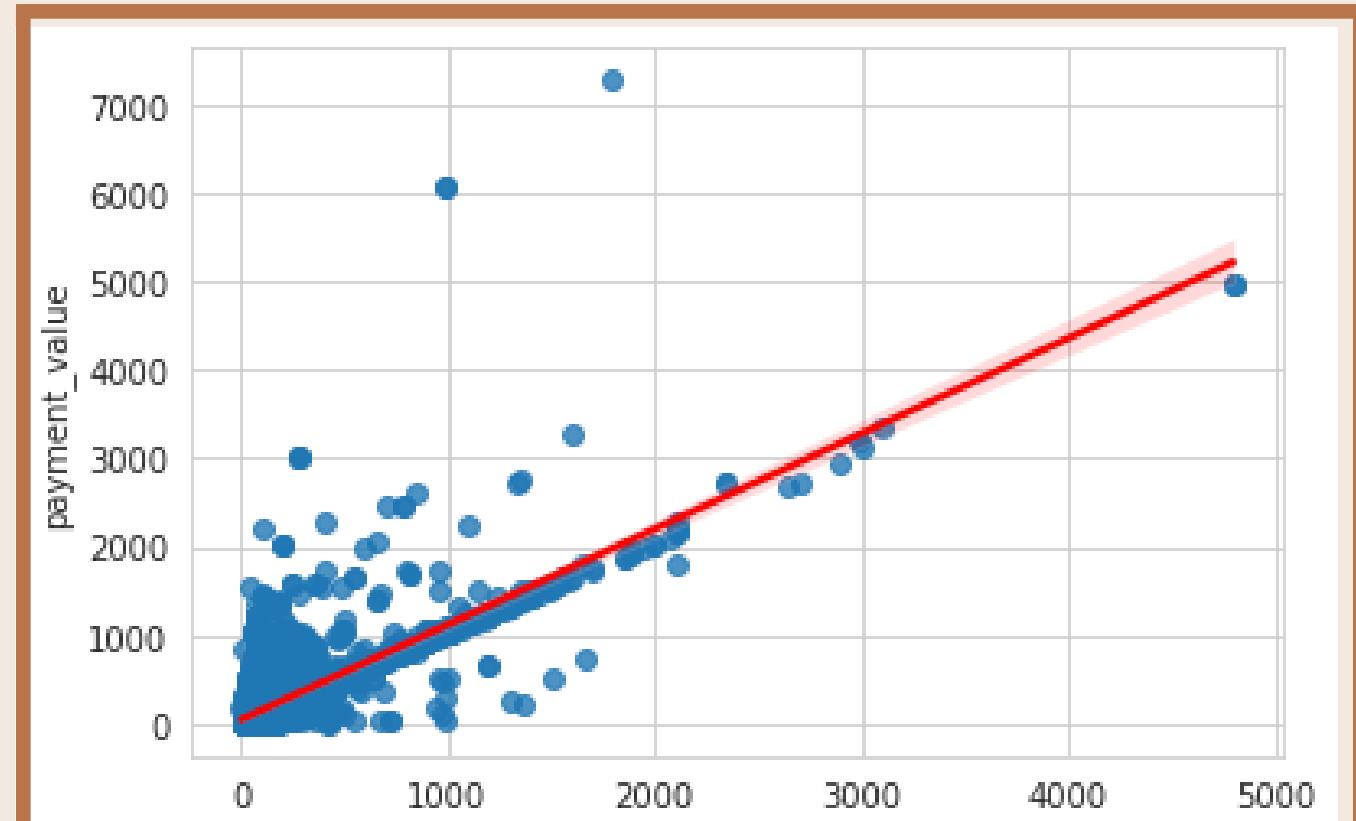
# Fitur Numerik

## ■ Univariat Numerik

- Membuat histogram dan boxplot setiap fitur numerik
- Dari plot yang didapatkan, setiap fitur memiliki sebaran yang beragam

## ■ Multivariat Numerik-Numerik

- Membuat regplot untuk melihat hubungan antar fitur.
- Salah satu regplot dengan korealsi paling tinggi

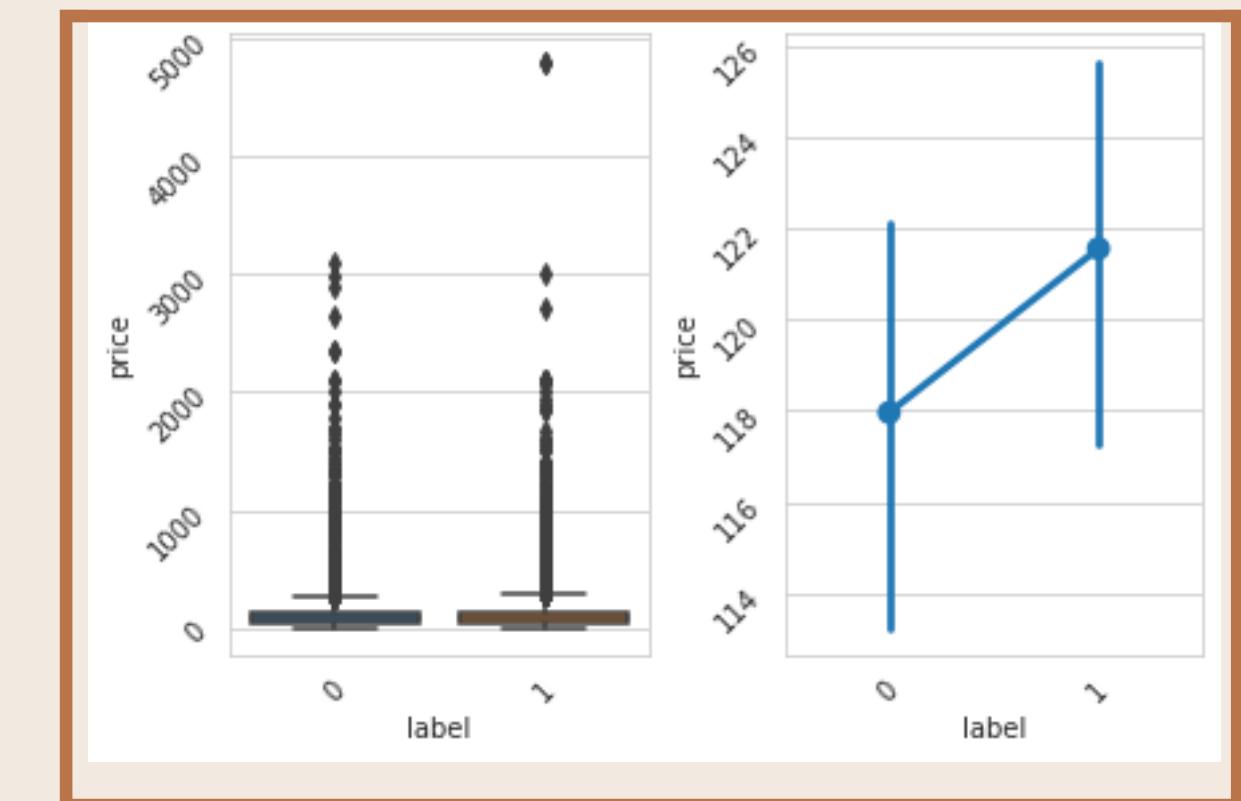
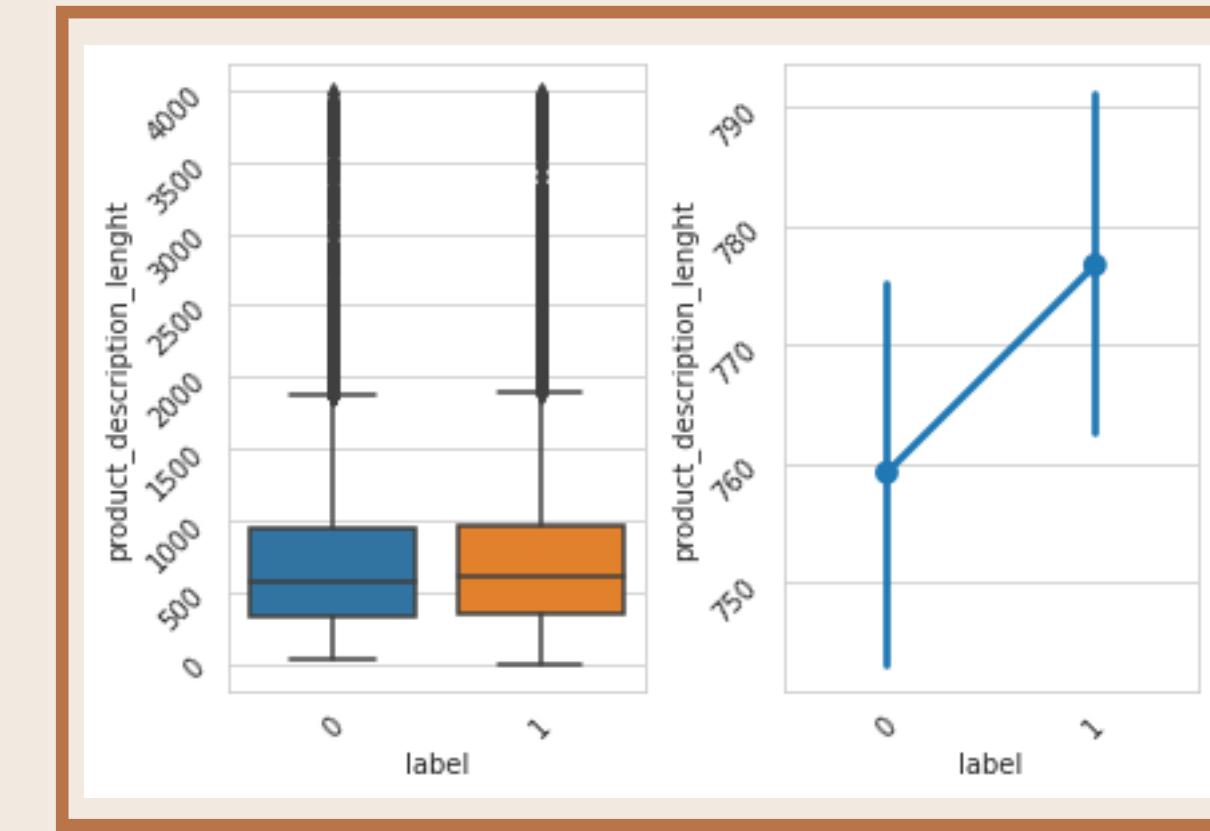


SpearmanResult (correlation=0.783)  
(x: price, y = payment\_value)

# Fitur Numerik

## Numerik - Label

- Membuat boxplot dan pointplot fitur numerik dengan sumbu x berupa 'label'
- Dari plot yang dihasilkan, pada umumnya di setiap fitur numerik memiliki rata- rata nilai yang lebih tinggi pada label 0 (rating di bawah 5)
- Namun untuk fitur 'price' dan 'description length' berlaku sebaliknya, memiliki rata- rata nilai yang lebih tinggi pada label 1 (buyer memberi rating 5)



Boxplot dan pointplot untuk fitur 'price' dan 'desc. length' dengan hue = 'label'

# Fitur Kategorik

## ■ Univariat Kategorik

- Membuat countplot dan stacked barplot untuk setiap fitur kategorik

## ■ Kategorik - Label

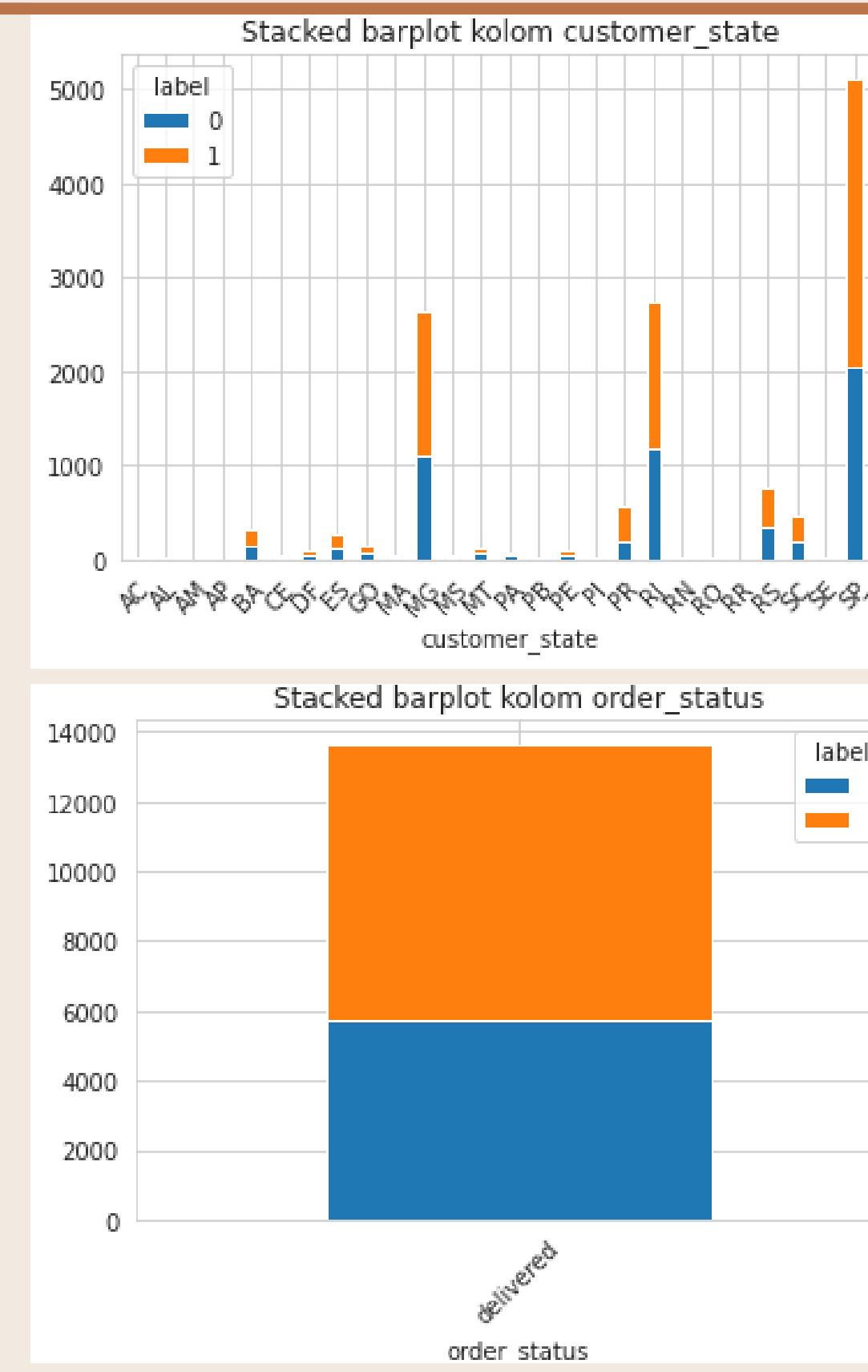
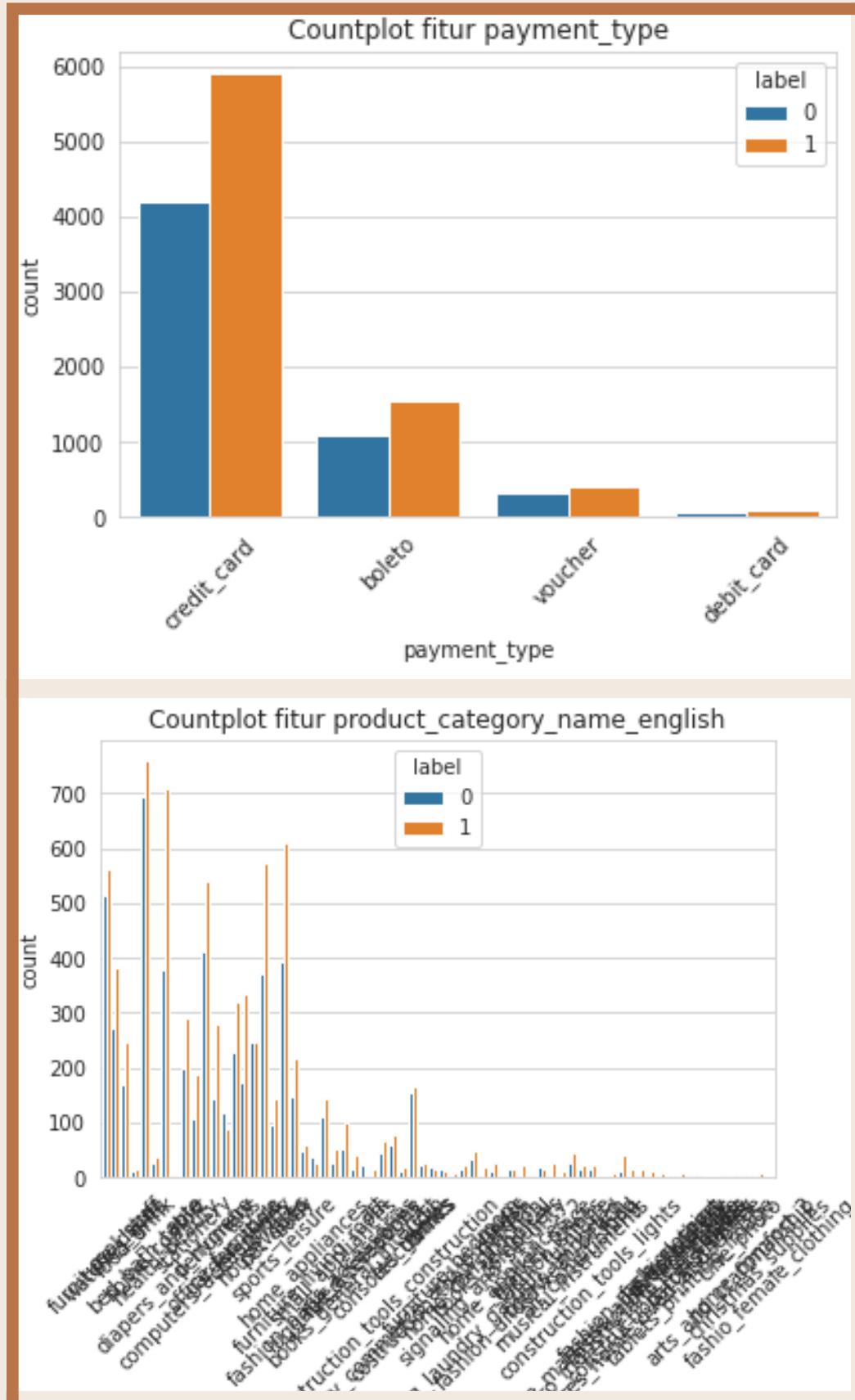
- Membuat countplot dan stacked barplot untuk setiap fitur kategorik dengan hue = 'label'

Pada umumnya untuk setiap fitur kategorik, jika dilihat dari visualnya, label 1 lebih banyak dari pada label 0



# Fitur Kategorik

# Count plot dan Stacked bar plot



Dari beberapa barplot dan stacked barplot yang ditampilkan dapat dilihat bahwa untuk setiap fitur kategorik, label 1 atau pemberian rating 5 lebih unggul untuk setiap kategori yang ada dalam fitur

# Date Time

- Dari fitur datetime dapat diidentifikasi lama waktu pemrosesan pesanan
- Dari data yang tersedia, dicari selisih antar fitur, sehingga dapat dibuat kolom baru khusus untuk selisih waktu

#	Column	Dtype
0	difference_approved_purchase	timedelta64[ns]
1	difference_deliveredcarrier_purchase	timedelta64[ns]
2	difference_deliveredcustomer_purchase	timedelta64[ns]
3	difference_deliveredcarrier_approved	timedelta64[ns]
4	difference_deliveredcustomer_approved	timedelta64[ns]
5	difference_deliveredcarrier_delivered_customer	timedelta64[ns]

Keterangan:

- 0: Lama waktu yang dibutuhkan untuk diapprove setelah customer membuat pesanan
- 1: Lama waktu mulai dibawa kurir setelah customer membuat pesanan
- 2: Lama waktu pesanan sampai ke customer sejak pesanan dibuat
- 3: Lama waktu mulai dibawa kurir setelah diapprove penjual
- 4: Lama waktu pesanan sampai ke customer setelah diapprove penjual
- 5: Lama waktu yang dibutuhkan sejak mulai dibawa kurir hingga sampai ke customer

# Date Time

approved_purchase	difference_deliveredcarrier_purchase	difference_deliveredcustomer_purchase
0 days 00:00:00	2 days 14:27:46	9 days 16:02:05
0 days 00:28:34	6 days 04:05:57	12 days 07:40:46
0 days 00:15:49	1 days 05:44:00	14 days 10:22:01
0 days 00:09:46	4 days 04:24:33	11 days 07:03:31
0 days 18:01:04	6 days 09:10:36	20 days 11:46:45

(1) Dalam hari, jam, menit, detik

approved_purchase	difference_deliveredcarrier_purchase	difference_deliveredcustomer_purchase
0.000000	2.602616	9.668113
0.476111	6.170799	12.319977
0.263611	1.238889	14.431956
0.162778	4.183715	11.294109
18.017778	6.382361	20.490799

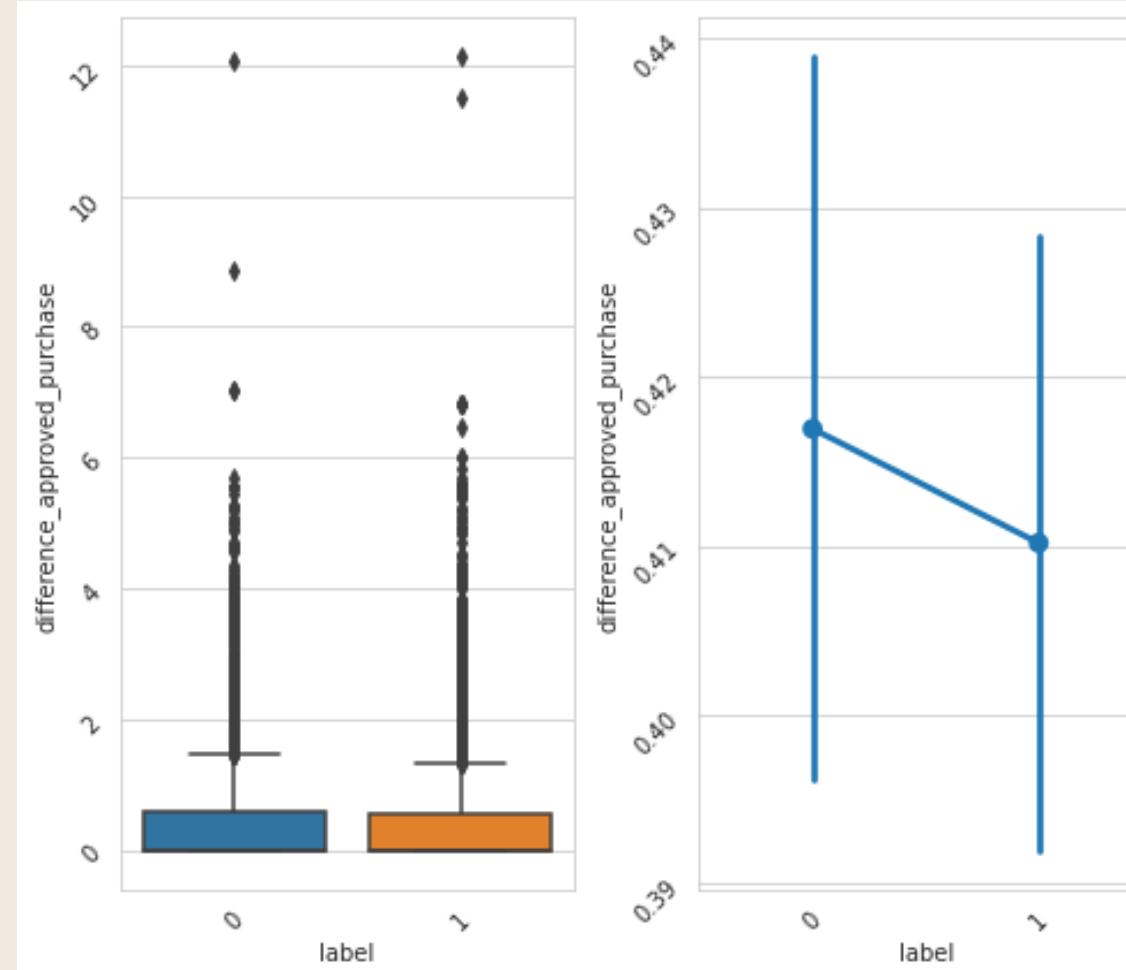
(2) Dalam total hari

(Cuplikan kolom hasil selisih waktu yang telah dilakukan)

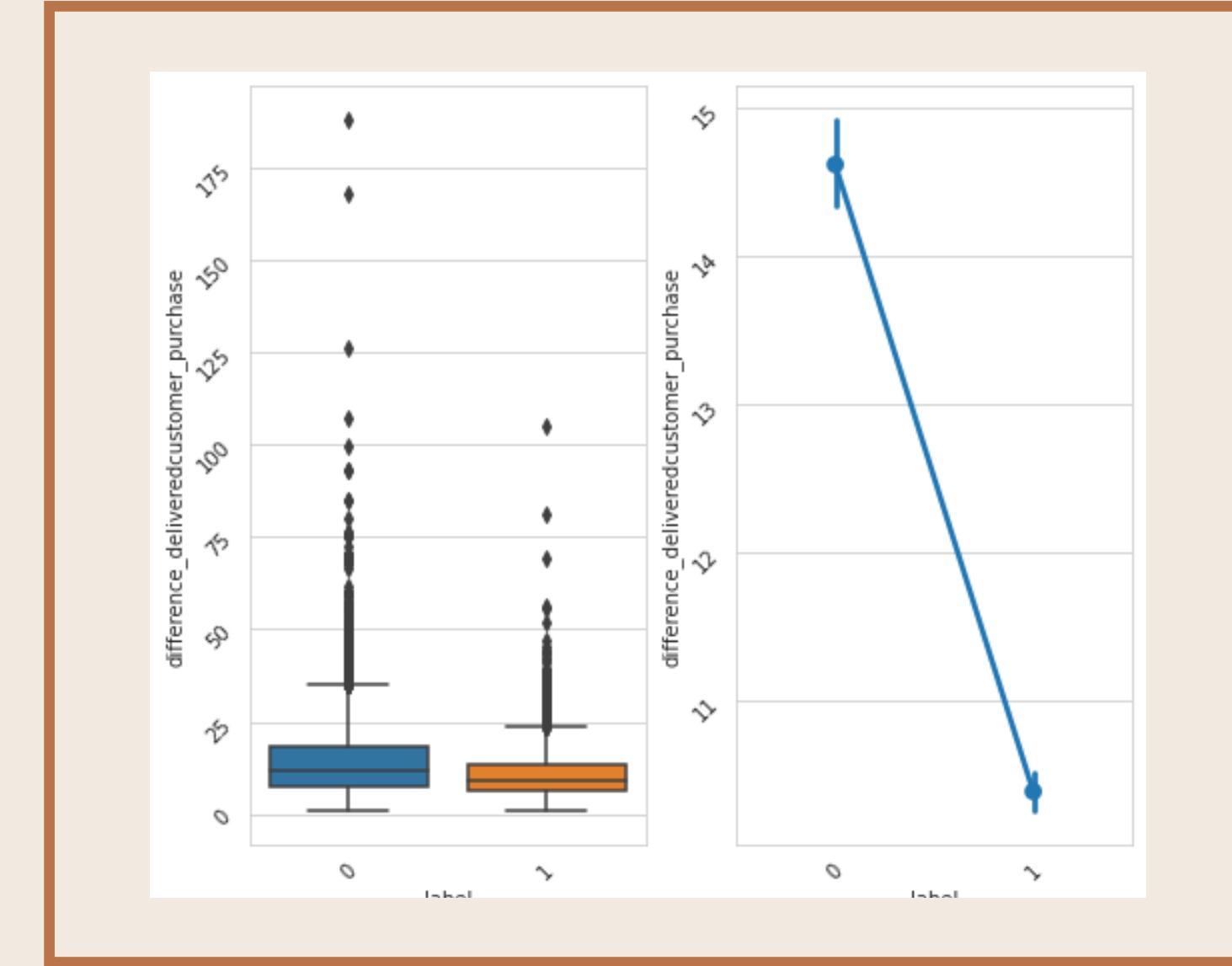
- Awalnya selisih waktu yang didapatkan dalam format hari, jam, menit, dan detik (1)
- Hal ini akan menyulitkan dan kurang sesuai jika dilakukan pemrosesan lebih lanjut seperti membuat plot
- Dengan engan menggunakan 'time delta' pada python format semua selisih waktu diseragamkan menjadi total hari (2).

# Date Time

## Boxplot dan poinplot

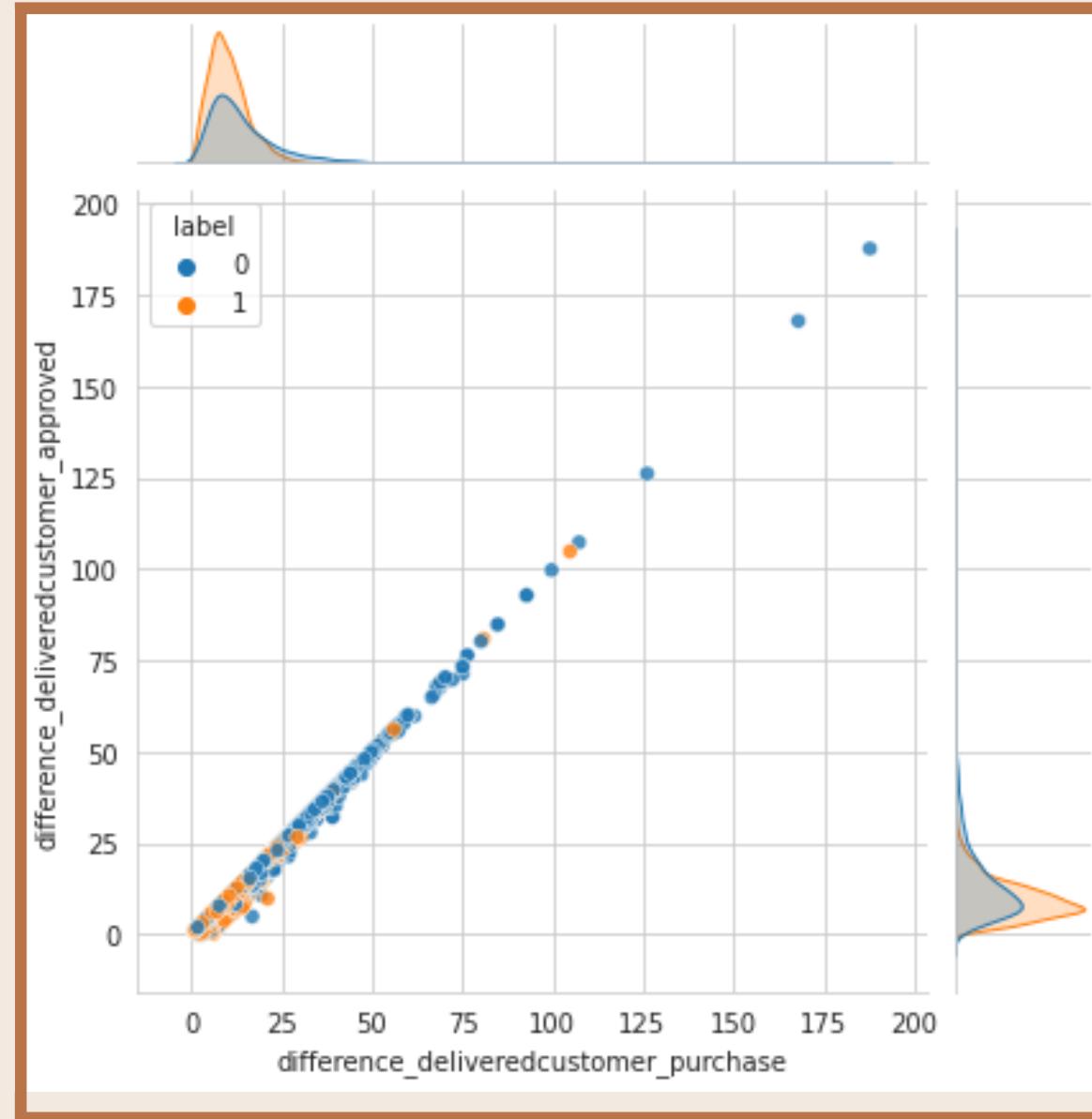


Boxplot dan point plot dengan hue = 'label' untuk waktu yang dibutuhkan (dalam hari), dari pembelian hingga pesanan diapproved seller

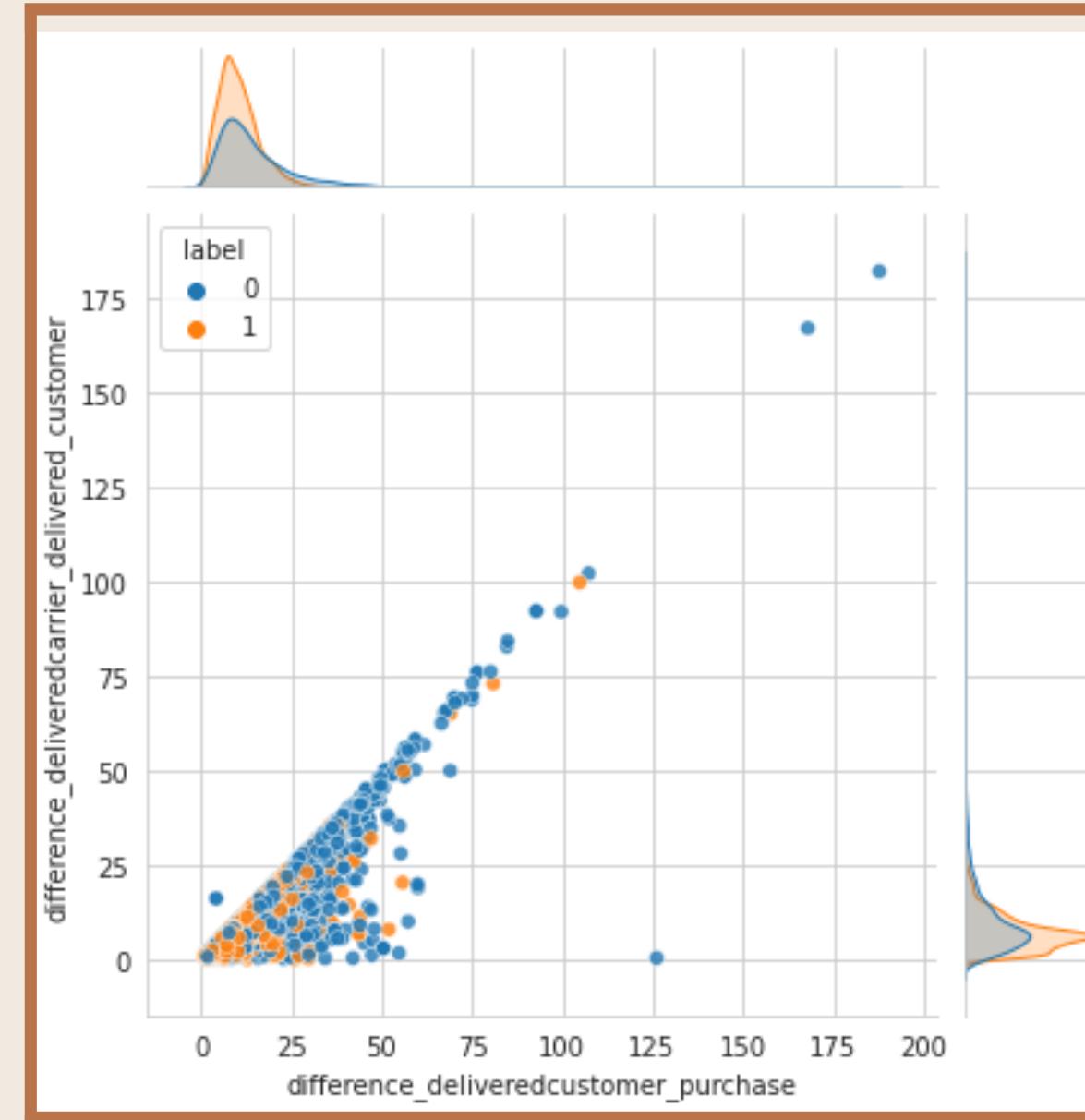


Boxplot dan point plot dengan hue = 'label' untuk waktu yang dibutuhkan (dalam hari ) sejak pembelian hingga sampai ke customer

# Date Time



(a) Jointplot selisih waktu (x : waktu yang dibutuhkan dari pembelian hingga diterima customer, y: waktu yang dibutuh setelah pesanan di-approved seller hingga diterima customer)



(b) Jointplot selisih waktu (x : waktu yang dibutuhkan dari pembelian hingga diterima customer, y: waktu yang dibutuhkan dari mulai dibawa kurir hingga sampai ke customer)

# EDA

- Pada fitur numerik untuk 'harga' dan 'panjang deskripsi' cukup unik karena rata-rata untuk nilai fitur tersebut lebih tinggi pada pemberian rating 5 daripada dibawah 5, berbeda dengan fitur numerik lain yang berlaku sebaliknya
- Pada fitur kategorik, di setiap kategori dalam fitur lebih banyak pemberian dengan rating 5 daripada di bawah 5, namun kurang dapat dijelaskan hubungannya.
- Dari ketiga tipe data, yang paling berpengaruh terhadap pemberian rating terdapat pada fitur datetime yaitu lama waktu yang dibutuhkan dalam proses pesanan. Dimana semakin lama waktu tahapan pemrosesan, maka rating yang diberikan oleh buyer lebih dominan di bawah 5

# Data Preparation & Feature Engineering

## TRAIN TEST SPLIT

- Drop fitur yang kurang diperlukan

```
✓ [52] devset_fs = devset.drop(['customer_id', 'customer_unique_id', 'customer_zip_code_prefix',
0s                                'geolocation_zip_code_prefix', 'geolocation_lat', 'geolocation_lng',
                                'order_id', 'order_purchase_timestamp', 'order_approved_at', 'order_delivered_carrier_date',
                                'order_delivered_customer_date', 'order_estimated_delivery_date', 'product_id',
                                'seller_id', 'shipping_limit_date', 'order_item_id', 'seller_zip_code_prefix'], axis = 1)
```

- Melakukan train test split

```
✓ [53] X = devset_fs.drop(['label'], axis =1)
0s
y = devset_fs['label']
```

```
✓ [54] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, stratify = y, random_state =42)
0s
```

```
✓ [55] X_train.shape
```

```
(10916, 22)
```

```
[ ] X_test.shape
```

```
(2729, 22)
```

# Data Preparation & Feature Engineering

## MISSING VALUE HANDLING

```
customer_city          0.000000
customer_state         0.000000
geolocation_city       0.000000
geolocation_state      0.000000
order_status            0.000000
price                   0.000000
freight_value           0.000000
payment_sequential      0.009161
payment_type             0.009161
payment_installments    0.009161
payment_value            0.009161
product_category_name   1.310004
product_name_length     1.310004
product_description_length 1.310004
product_photos_qty       1.310004
product_weight_g          0.036643
product_length_cm        0.036643
product_height_cm        0.036643
product_width_cm         0.036643
seller_city               0.000000
seller_state              0.000000
product_category_name_english 1.328325
dtype: float64
```

## TRAINING SET (X\_TRAIN)

- simple imputer (strategy = median) untuk fitur numerik
- simple imputer (strategy = most\_frequent ) untuk fitur kategorik

# Data Preparation & Feature Engineering

## TRANSFORMASI

Fitur Numerik

- Scaling

Fitur Kategorik

- one hot encoder

```
[ ] x_train_transformed.shape
```

```
(10916, 2934)
```

## FEATURE SELECTION

- Multiicolinearity Reduction

```
x_train_sel.shape
```

```
(10916, 1773)
```

- Mutual Information

```
x_train_sel2.shape
```

```
(10916, 872)
```

# Data Preparation & Feature Engineering

## TESTING SET

Melakukan semua tahapan:

- Missing value handling
- Transformasi
- Feature Selection

yang dilakukan terhadap training set sebelumnya ke **testing set** dengan cara transform tanpa fit ulang

```
[66] X_test_sel.shape
```

```
(2729, 872)
```

# Modeling

- Menentukan model
- Tuning Hyperparameter menggunakan GridSearchCV
- Mendapatkan parameter terbaik
- Fitting ke training set
- Periksa performance (train dan test)
- Sesuai kriteria : Ya | Tidak

↓  
Selesai  
(Final Model)



# Modeling

Model klasifikasi yang dilakukan:

- Logistic Regression
- Decision Tree
- Random Forest
- AdaBoost
- XGBoost

Hyper parameter

```
[ ] xgbfinal = XGBClassifier(random_state=42,  
n_estimators=180,  
learning_rate=0.1,  
max_depth=20,  
gamma=0.01,  
reg_lambda = 1,  
min_child_weight=1)
```

```
[ ] classif_pr(X_train_sel2, y_train, xgbfinal, 0.57)
```

	precision	recall	f1-score	support
1	0.99	0.98	0.99	6359
0	0.98	0.99	0.98	4557
accuracy			0.99	10916
macro avg	0.98	0.99	0.99	10916
weighted avg	0.99	0.99	0.99	10916

```
[ ] classif_pr(X_test_sel, y_test, xgbfinal, 0.57)
```

	precision	recall	f1-score	support
1	0.69	0.70	0.70	1590
0	0.58	0.56	0.57	1139
accuracy	TNR = 1 - 0.56 = 0.44		0.64	2729
macro avg	0.63	0.63	0.63	2729
weighted avg	0.64	0.64	0.64	2729

# Evaluasi

- Menggunakan data 'back\_testing\_set.csv' untuk memprediksi kolom label menggunakan final model (XGBoost)
- Sebelumnya data back\_testing\_set ditransformasi dan diseleksi mengikuti apa yang dilakukan kepada training set
- Didapatkan kolom prediksi label yang berisi 0 dan 1 ( 0: buyer memberi rating di bawah 5, 1: buyer memberi rating 5), kemudian dijadikan kolom sendiri dan diextract ke file csv

## Evaluasi

```
[ ] y_pred = xgbfinal.predict_proba(eval_sel)[:,1] > 0.57
```

```
[ ] y_pred
```

```
array([ True,  True,  True, ...,  True, False, False])
```

```
[ ] y_pred_int = y_pred.astype(int)
```

```
[ ] label_btest = pd.DataFrame(y_pred_int, columns= ['label'])  
label_btest.head()
```

	label
0	1
1	1
2	1
3	0
4	1

```
[ ] label_btest.to_csv('Labelint.csv', index = False)  
files.download('Labelint.csv')
```

# TERIMA KASIH