Report Delivery Task

Data Science Test - MileApp

Overview

Data yang digunakan yaitu data pengiriman tugas dalam 10 hari

Data Cleaning

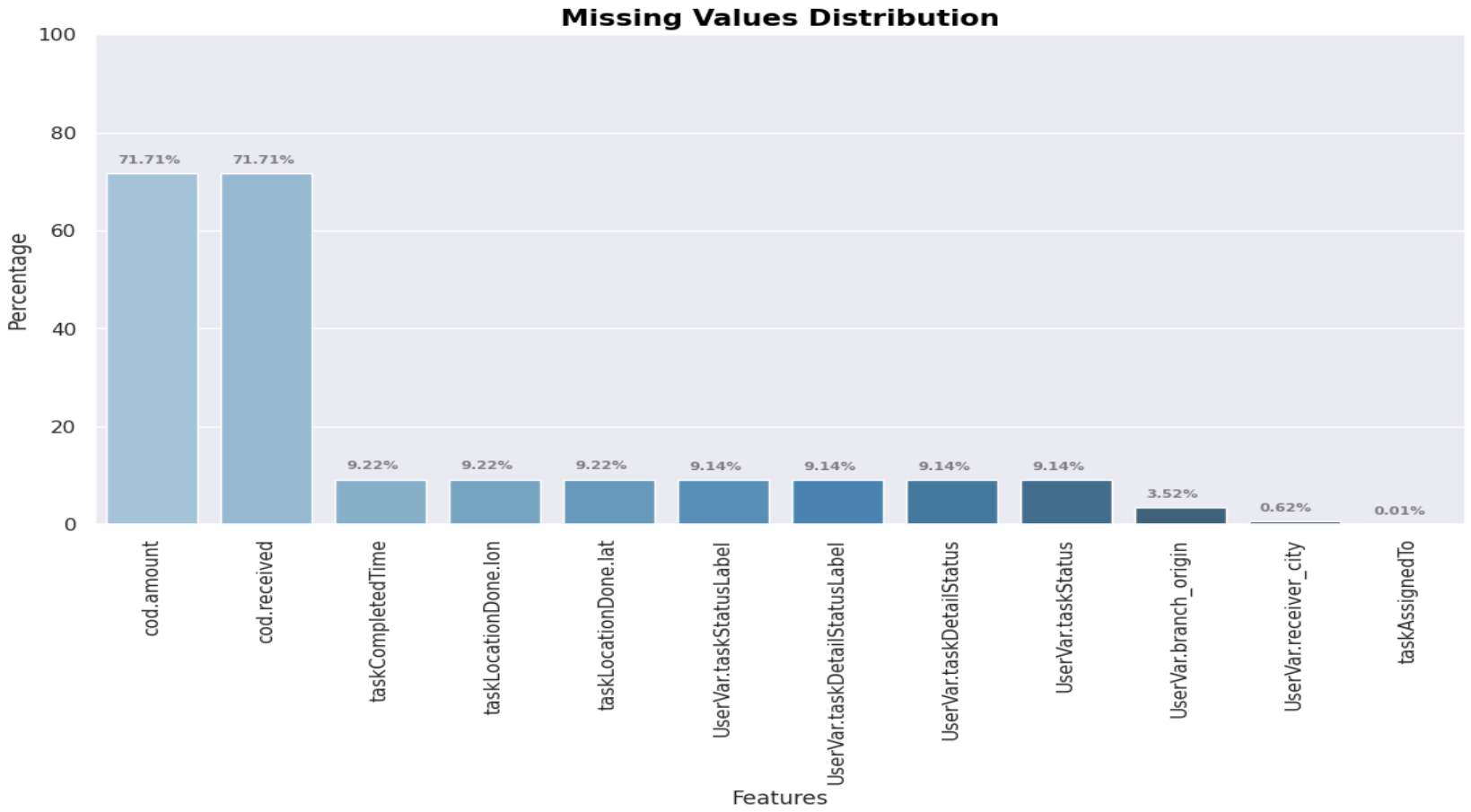
Pada tahap Data Preprocessing ini, terdapat beberapa tahapan yang dilakukan, yaitu :

1. Check Missing Values
2. Check Duplicated Data
3. Data Type Transformation
4. Check Number of Unique Values
5. Check Cardinality
6. Create New Column

Data Distionary

Field	Description
taskId	Pengidentifikasi unik untuk tugas yang dihasilkan oleh sistem.
taskCreatedTime	Waktu saat tugas dibuat.
taskCompletedTime	Waktu saat tugas selesai.
taskAssignedTo	Pekerja yang melakukan tugas
taskLocationDone	Koordinat tempat tugas diselesaikan.
flow	Alur atau jenis tugas
cod	Berisi data untuk sistem COD.
cod.amount	Jumlah uang dari COD.
cod.received	COD sudah diterima atau belum
UserVar	Berisi lebih banyak data yang ditentukan, dalam hal ini 'UserVar' adalah tentang data tugas pengiriman.
UserVar.taskStatus	Kode status pengiriman
UserVar.taskStatusLabel	Label status pengiriman
UserVar.taskDetailStatus	Kode status pengiriman terperinc
UserVar.taskDetailStatusLabel	Label status pengiriman terperinci
UserVar.branch_origin	Kode cabang asal.
UserVar.branch_dest	Kode cabang tujuan
UserVar.weight	Berat paket.

Missing Values



Pada gambar disamping, terlihat bahwa semua kolom dalam dataset yang kita miliki mempunyai missing value dengan persentase yg berbeda-beda. Kolom COD (cod.amount dan cod.received) memiliki persentasi null values mencapai lebih dari 70. Dapat kita putuskan bahwa kolom tersebut tidak akan kita gunakan dalam pemodelan machine learning kedepannya karena tidak memliki cukup data

Duplicated Data

```
[9]  # Check the amount of duplicated data
     num_duplicated = df.duplicated().sum()
     print(f"Total number of duplicate values : {num_duplicated}")

     Total number of duplicate values : 0
```

Pada kode diatas, kita melakukan pengecekan apakah data yang kita miliki mempunyai data duplikat. Setelah dilakukan pengecekan, didapatkan hasil bahwa tidak terdapat data duplikat.

Data Type Transformation

```
df['taskCreatedTime'] = pd.to_datetime(df['taskCreatedTime'], utc=True)
df['taskCompletedTime'] = pd.to_datetime(df['taskCompletedTime'], utc=True)

df['taskCreatedTime'] = df['taskCreatedTime'].dt.tz_convert('Asia/Jakarta')
df['taskCompletedTime'] = df['taskCompletedTime'].dt.tz_convert('Asia/Jakarta')

df['UserVar.weight'] = pd.to_numeric(df['UserVar.weight'], errors='coerce')
```

Terdapat beberapa kolom yang perlu diubah tipe data nya sesuai dengan semestinya. Pada gambar disamping, dapat dilihat beberapa kolom yang harus diubah tipe datanya.

Number of Unique Values

	Feature	unique_values
0	taskId	8334
1	taskAssignedTo	2787
2	UserVar.receiver_city	1830
3	UserVar.branch_dest	62
4	UserVar.branch_origin	59
5	UserVar.taskDetailStatusLabel	31
6	UserVar.taskDetailStatus	31
7	taskStatus	2
8	cod.received	2
9	UserVar.taskStatusLabel	2
11	UserVar.taskStatus	2
10	flow	1

Pada tahap ini kita akan melakukan analisis pada feature kategorikal yang hanya mempunyai satu kategori. Jika kondisi tersebut terpenuhi kita akan menghapus feature tersebut karena tidak akan berpengaruh pada model kita nantinya.

Check Cardinality

```
taskAssignedTo :  ['pacifiedLion0' 'peacefulTacos6' 'giddyCockatoo1' ... 'culturedPorpoise0'
 'ferventBoa6' 'murkyThrushe3']

taskStatus :  ['done' 'ongoing']

flow :  ['Delivery']

taskId :  ['4fe3b237c832ca4841a2' '08a4da25256affae8446' '2ff0dc469826158b7684' ...
 '1b136b5a3c60749eb571' 'e92e813c8539080c922e' 'cdb90c597655282306fd']

cod.received :  [True nan False]

UserVar.branch_dest :  ['SRG' 'MGL' 'PWT' 'CLG' 'PDG' 'BTJ' 'DTB' 'SMI' 'PKU' 'BDO' 'BTG' 'JBR'
 'BKS' 'CXP' 'SOC' 'MES' 'TKG' 'JOG' 'MXG' 'PNK' 'CBN' 'TGL' 'DJB' 'BPN'
 'BKI' 'GTO' 'MDN' 'MDC' 'KOE' 'PLM' 'SUB' 'CKR' 'UPG' 'DJJ' 'PLW' 'DPS'
 'AMQ' 'BDJ' 'BOO' 'CGK' 'TTE' 'MJK' 'AMI' 'TGR' 'KDI' 'PGK' 'BTH' 'TSM'
 'TIM' 'KDR' 'SOQ' 'PSR' 'MKQ' 'KRW' 'SMD' 'TRK' 'PBL' 'SDA' 'PKY' 'DPK'
 'TJQ' 'TNJ']

UserVar.taskStatusLabel :  ['Success' nan 'Failed']

UserVar.receiver_city :  ['BATANG ,KAB BATANG' 'PURWODADI,PURWOREJO' 'BAGELEN,PURWOREJO' ...
 'MEDAN KOTA,MEDAN' 'DENDANG,MUARASABAK' 'KOTA BANTUL']

UserVar.taskDetailStatusLabel :  ['YANG BERSANGKUTAN' 'KELUARGA/SAUDARA' 'RECEPTIONIST'
 'ATASAN/STAFF/KARYAWAN/BAWAHAN' 'SUAMI/ISTRI/ANAK' 'SECURITY'
 'PENJAGA KOS' nan 'ALAMAT TIDAK LENGKAP service/ TIDAK DIKENAL'
 'MISROUTE' 'DITOLAK OLEH PENERIMA' 'PENERIMA PINDAH ALAMAT'
 'MAILING ROOM' 'DIAMBIL SENDIRI'
 'RUMAH service/ KANTOR KOSONG (MASIH DIHUNI)' 'PENERIMA TIDAK DIKENAL'
 'PEMBANTU' 'NEW ADDRESS' 'PENERIMA MENOLAK BAYAR (KIRIMAN COD)'
 'MENUNGGU PEMBAYARAN COD'
 'PENERIMA MENOLAK MENERIMA KIRIMAN COD (TDK PESAN)' 'OFFICE BOY'
 'CRISS-CROSS' 'HOLD FOR FURTHER INSTRUCTI0N' 'SUPIR' 'DAMAGE CASE'
```

Pada tahap ini kita pastikan kembali bahwa tidak ada feature yang memiliki inconsistent data (khusus feature kategorikal).

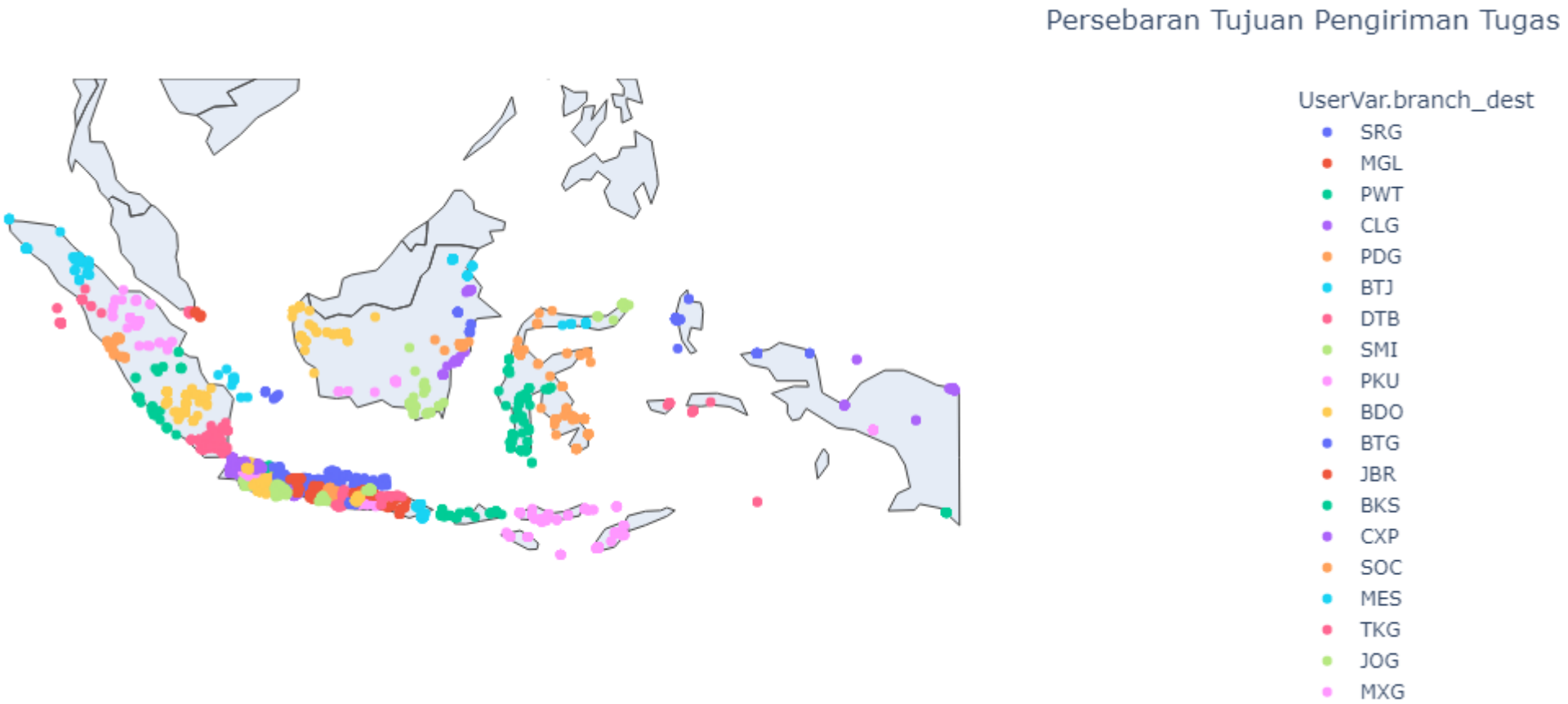
Check Cardinality

```
[14] df['taskCreatedDate'] = df['taskCreatedTime'].dt.date
     df['taskCompletedDate'] = df['taskCompletedTime'].dt.date

     df['taskCreatedDate'] = pd.to_datetime(df['taskCreatedDate'])
     df['taskCompletedDate'] = pd.to_datetime(df['taskCompletedDate'])

     df['taskCreatedHour'] = df['taskCreatedTime'].dt.hour
     df['taskCompletedHour'] = df['taskCompletedTime'].dt.hour

     df['diff_hours'] = (df['taskCompletedTime'] - df['taskCreatedTime']) / np.timedelta64(1, 'h')

     df = pd.concat([df, df['UserVar.receiver_city'].str.split(',', expand=True)], axis=1).rename(columns = {0:'kecamatan', 1:'kota/kab', 2:'provinsi'})
```

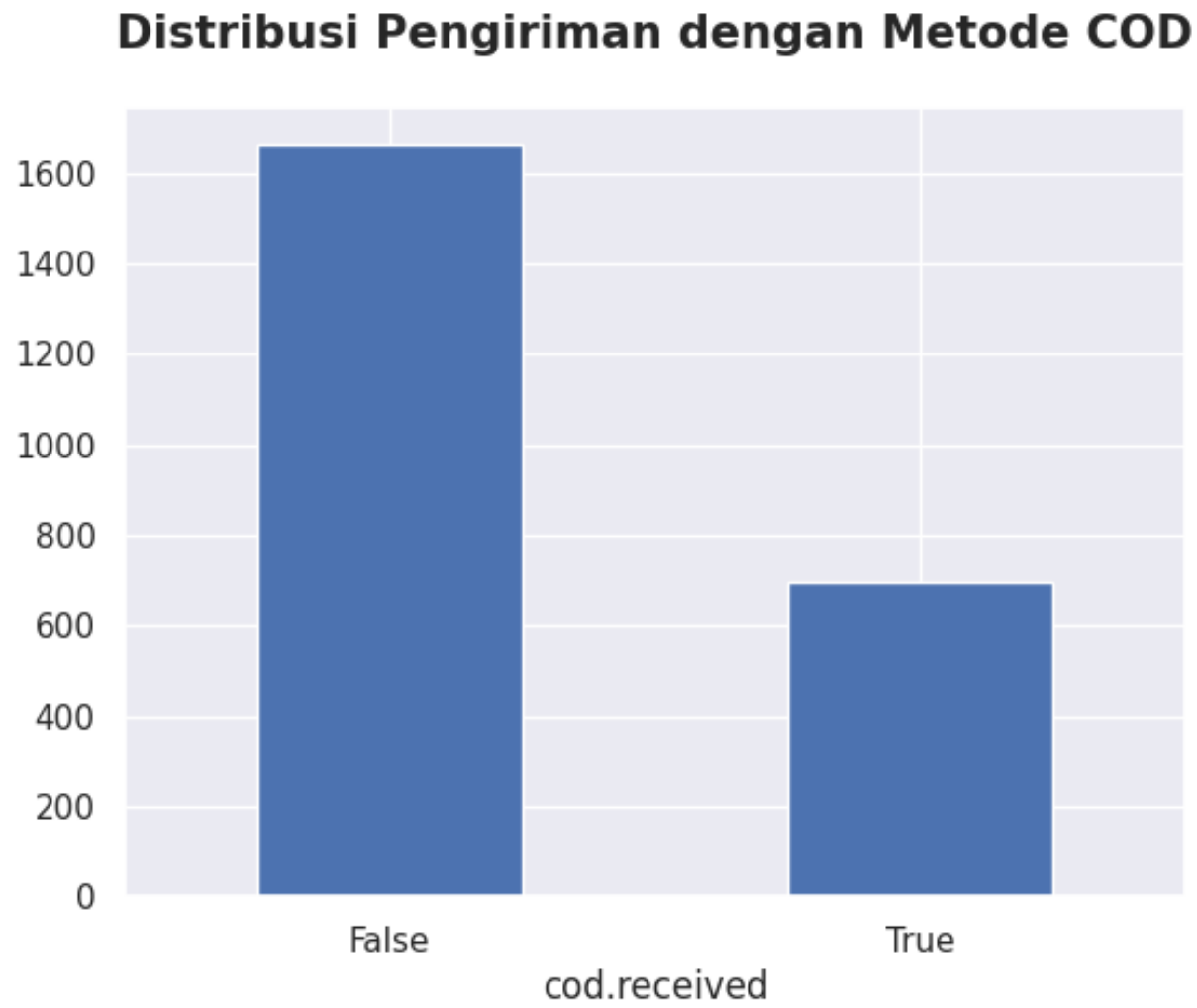
Buat beberapa kolom menggunakan data-data yang sudah tersedia.

Exploratory Data Analysis

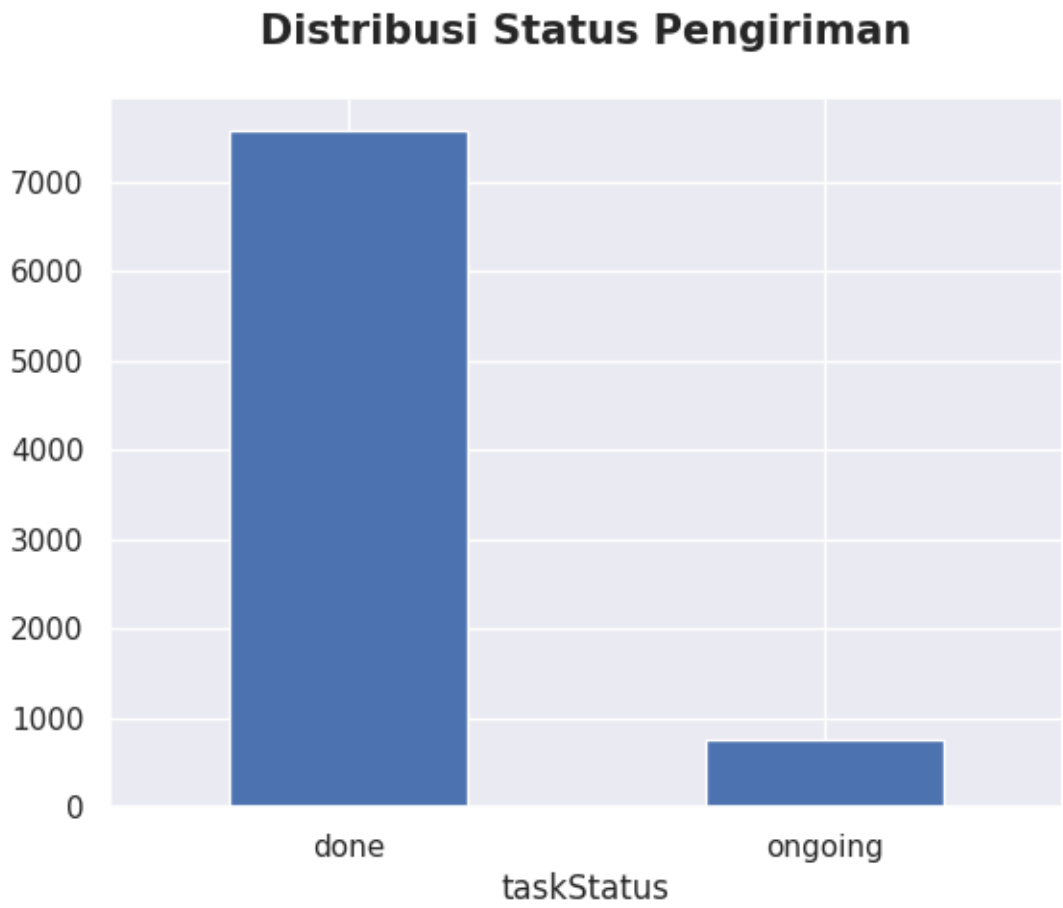
Persebaran Pengiriman Tugas (Paket)



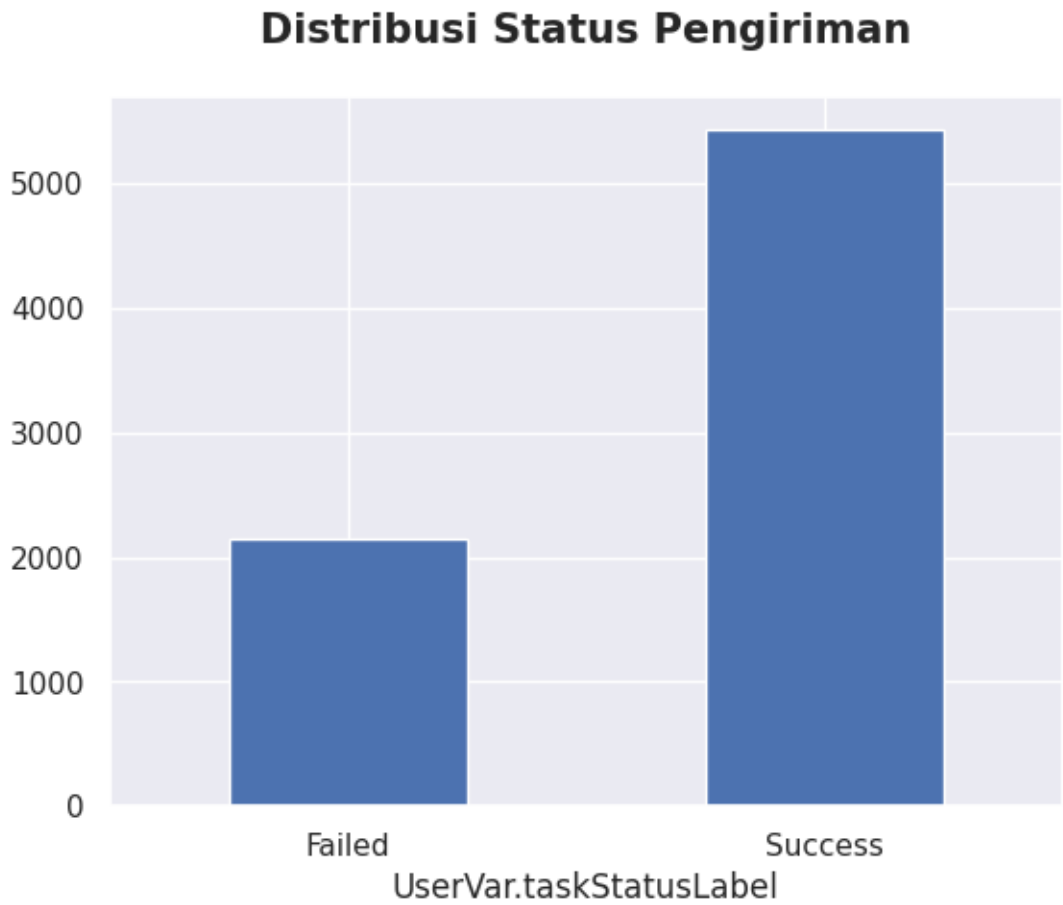
Pada persebaran pengiriman tugas diatas, dapat dilihat bahwa mayoritas terdapat di Pulau Jawa. Pulau yang paling sedikit penerima pengiriman tugas yaitu Papua.



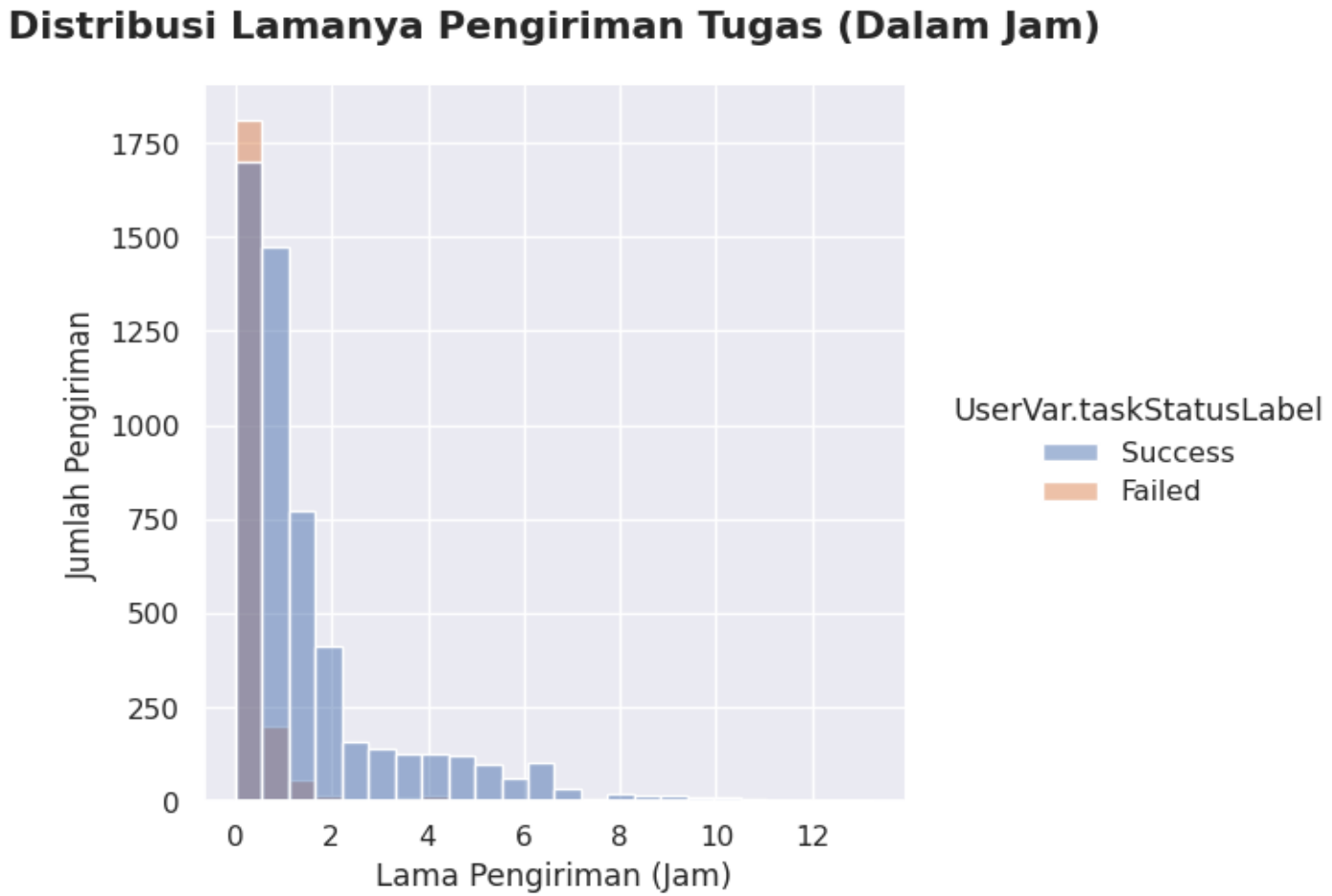
Pada grafik diatas, dapat dilihat bahwa sebagian besar pengiriman dengan metode Cash On Delivery (COD) gagal (tidak diterima), dengan lebih dari 1600 pengiriman.



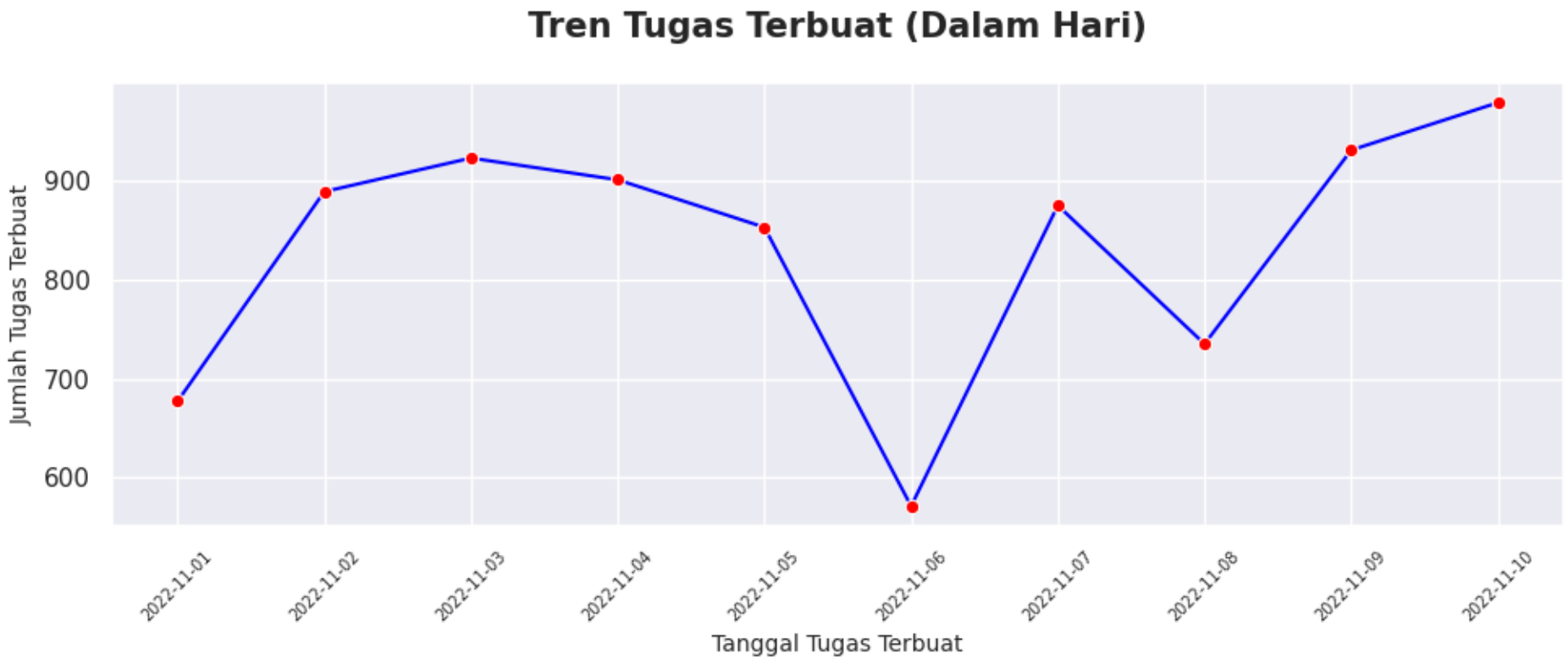
Pada grafik diatas, dapat dilihat bahwa pengiriman tugas (paket) dari cabang pengirim berhasil terkirim.



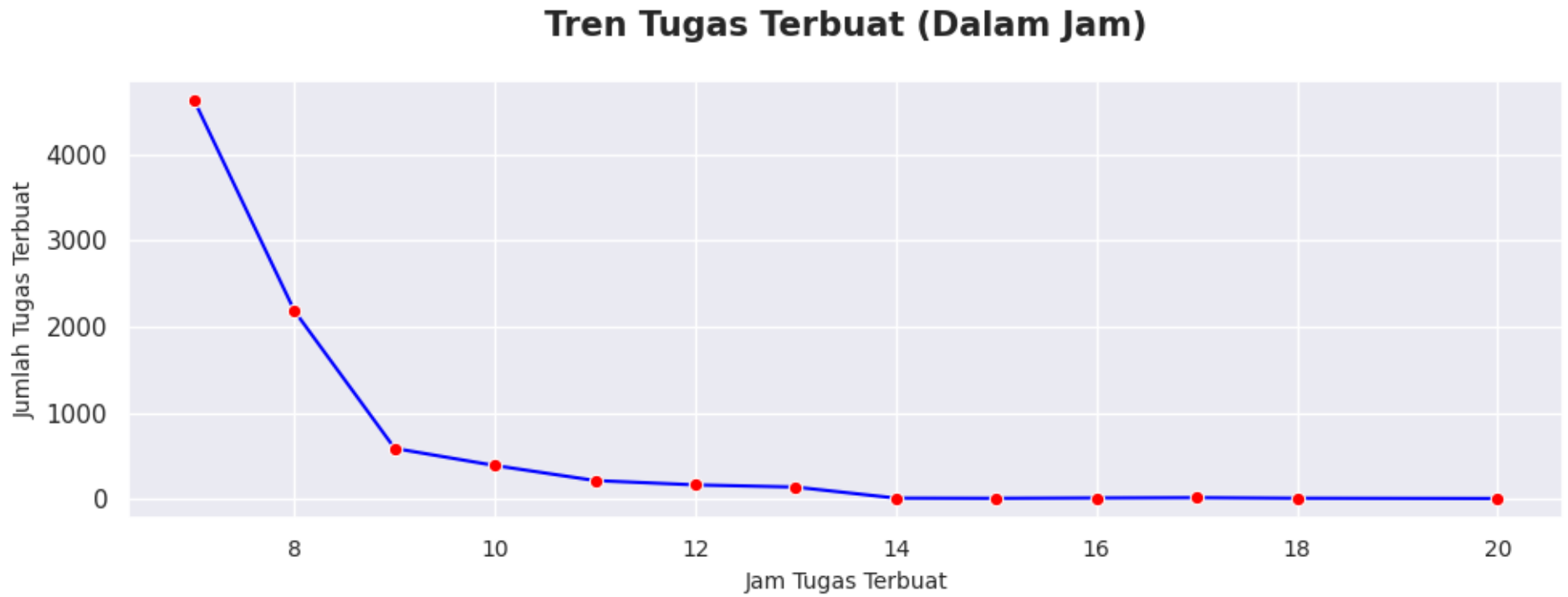
Pada grafik diatas, dapat dilihat bahwa pengiriman tugas (paket) yang diterima dari cabang pengirim berhasil(success).



Pada grafik diatas, dapat dilihat bahwa lamanya pengiriman tugas (paket) lebih banyak kurang dari 2 jam. Pola antara tugas yang sukses dan gagal pun hampir sama untuk setiap distribusinya.



Pada grafik diatas, dapat dilihat tren pembuatan tugas (dalam harian) polanya naik-turun. Pembuatan tugas paling sedikit terjadi pada tanggal 6 November 2022. Tertinggi pada tanggal 10 November 2022.



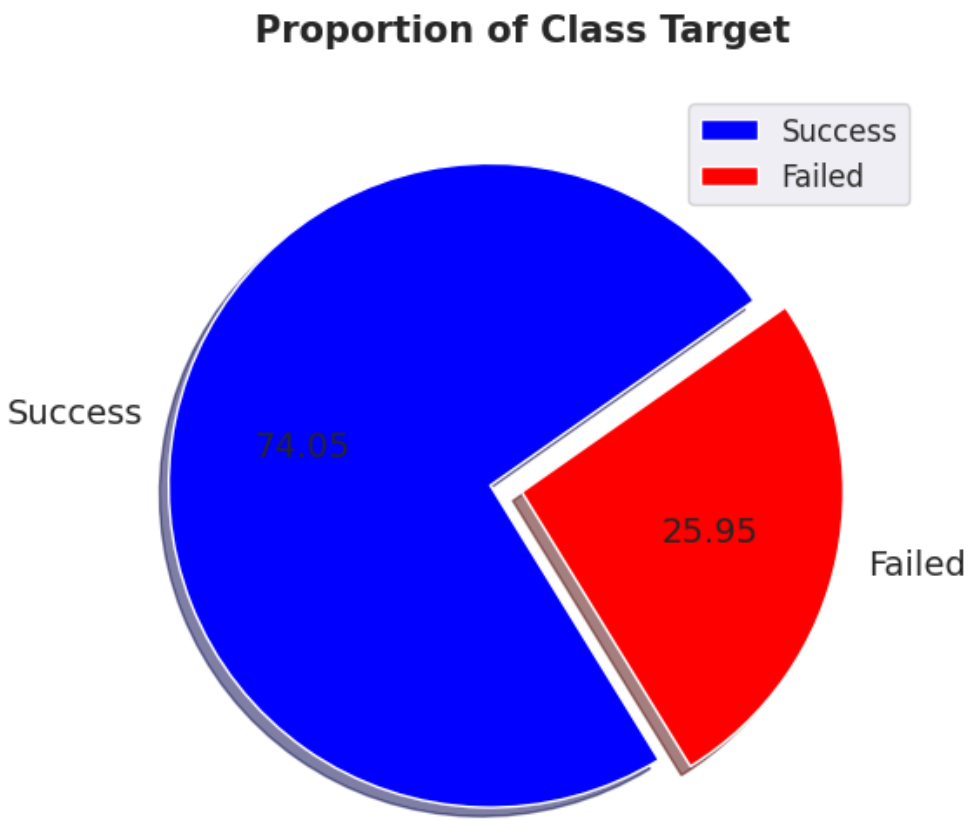
Pada grafik diatas, dapat dilihat tren pembuatan tugas (dalam jam) selama 10 hari polanya cenderung mengalami penurunan. Pembuatan tugas terbanyak terjadi antara pukul 7-9 pagi.

Berikut adalah pekerja yang mempunyai tugas pengiriman paling banyak dari semua pekerja.

Feature Engineering

Pada tahap Feature Engineering ini, terdapat beberapa tahapan yang dilakukan, yaitu :

1. Remove Unnecessary Features
2. Feature Encoding
3. Handling Missing Values
4. Sampling Dataset (Oversampling)



Karena distribusi class untuk modeling tidak merata. Maka digunakan Teknik oversampling untuk mengatasi hal tersebut.

Mohon maaf penjelasan untuk selanjutnya tidak bisa saya lanjutnya karena keterbatasan waktu. Tetapi hasil sudah ada dalam file Jupyter Notebook. Untuk lebih jelasnya saya bisa jelaskan jika memang saya maju pada tahap selanjutnya.

Terima kasih