DATA ENGINEER BATCH 4
Mentee: Yovina Silvia
Mentor: Bilal Benefit

alterra academy

TASK 1 - Ingestion Data

1. We have already learned how to create DataFrame from files here. Now, we are going to create a DataFrame from a larger csv file on our datasets.

   - `df = pd.read_csv('C:/Users/USER/Alta/belajar-bc/ingestion-demo/dataset/yellow_tripdata_2020-07.csv', low_memory=False)`: Membaca file CSV menggunakan `pandas` dan menyimpannya dalam variabel `df`. `low_memory=False` digunakan untuk menghindari `DtypeWarning` dengan membaca file dalam mode chunk.
   - `print(df.head())`: Menampilkan 5 baris pertama dari DataFrame untuk melihat data yang baru dibaca.

```python
import pandas as pd

# Step 1: Read the CSV file with low_memory=False to avoid DtypeWarning
print("1. Membaca file CSV")
df = pd.read_csv('C:/Users/USER/Alta/belajar-bc/ingestion-demo/dataset/yellow_tripdata_2020-07.csv', low_memory=False)
print(df.head())
print("--------------------------------------------------------------------------\n")
```

```
(.venv)
USER@Luna MINGW64 ~/Alta/belajar-bc/ingestion-demo/ingestion_data (main)
$ python task-1-ingestion-data.py
1. Membaca file CSV
   VendorID tpep_pickup_datetime  ... total_amount  congestion_surcharge
0       1.0  2020-07-01 00:25:32  ...         9.30                   0.0
1       1.0  2020-07-01 00:03:19  ...        27.80                   0.0
2       2.0  2020-07-01 00:15:11  ...        22.30                   2.5
3       2.0  2020-07-01 00:30:49  ...        14.16                   2.5
4       2.0  2020-07-01 00:31:26  ...         7.80                   0.0

[5 rows x 18 columns]
--------------------------------------------------------------------------
```

2. Rename all the columns with snake_case format.

- `df.columns = [col.lower().replace(' ', '_') for col in df.columns]`: Mengubah semua nama kolom menjadi huruf kecil dan mengganti spasi dengan garis bawah.
- `df = df.rename(columns={'vendorid': 'vendor_id'})`: Mengganti nama kolom `vendorid` menjadi `vendor_id` untuk konsistensi.
- `print(df.head())`: Menampilkan 5 baris pertama dari DataFrame setelah perubahan nama kolom.

```
# Step 2: Rename all the columns with snake_case format
print("2. Mengganti nama kolom dengan format snake_case")
df.columns = [col.lower().replace(' ', '_') for col in df.columns]
df = df.rename(columns={'vendorid': 'vendor_id'})
print(df.head())
print("--------------------------------------------------------------------------------\n")
```

```
2. Mengganti nama kolom dengan format snake_case
   vendor_id tpep_pickup_datetime  ... total_amount  congestion_surcharge
0        1.0  2020-07-01 00:25:32  ...         9.30                   0.0
1        1.0  2020-07-01 00:03:19  ...        27.80                   0.0
2        2.0  2020-07-01 00:15:11  ...        22.30                   2.5
3        2.0  2020-07-01 00:30:49  ...        14.16                   2.5
4        2.0  2020-07-01 00:31:26  ...         7.80                   0.0

[5 rows x 18 columns]
--------------------------------------------------------------------------------
```

3. Select only 10 top of highest number of passenger_count, show only columns vendor_id, passenger_count, trip_distance, payment_type, fare_amount, extra, mta_tax, tip_amount, tolls_amount, improvement_surcharge, total_amount, congestion_surcharge from the DataFrame.

- `selected_columns`: Daftar kolom yang dipilih untuk analisis lebih lanjut.
- `df_selected = df[selected_columns]`: Membuat DataFrame baru `df_selected` yang hanya berisi kolom yang dipilih.
- `print(df_selected.head())`: Menampilkan 5 baris pertama dari DataFrame `df_selected` untuk memverifikasi kolom yang dipilih.
- `top_passenger_data = df_selected.nlargest(10, 'passenger_count')`: Memilih 10 baris teratas dari `df_selected` berdasarkan kolom `passenger_count`.
- `print(top_passenger_data)`: Menampilkan DataFrame `top_passenger_data` yang berisi 10 baris teratas dengan jumlah penumpang terbanyak.

```
# Step 3: Select the relevant columns and Select the top 10 rows with the highest passenger_count
print("3. Memilih kolom yang relevan dan pilih 10 baris teratas dengan jumlah penumpang terbanyak")
print("Memilih kolom yang relevan")
selected_columns = [
    'vendor_id', 'passenger_count', 'trip_distance', 'payment_type',
    'fare_amount', 'extra', 'mta_tax', 'tip_amount', 'tolls_amount',
    'improvement_surcharge', 'total_amount', 'congestion_surcharge'
]
df_selected = df[selected_columns]
print(df_selected.head())
# Select the top 10 rows with the highest passenger_count
print("Memilih 10 baris teratas dengan jumlah penumpang terbanyak")
top_passenger_data = df_selected.nlargest(10, 'passenger_count')
print(top_passenger_data)
print("-------------------------------------------------------------------------\n")
```

```
3. Memilih kolom yang relevan dan pilih 10 baris teratas dengan jumlah penumpang terbanyak
Memilih kolom yang relevan
   vendor_id  passenger_count  ...  total_amount  congestion_surcharge
0        1.0              1.0  ...          9.30                   0.0
1        1.0              1.0  ...         27.80                   0.0
2        2.0              1.0  ...         22.30                   2.5
3        2.0              1.0  ...         14.16                   2.5
4        2.0              1.0  ...          7.80                   0.0

[5 rows x 12 columns]
Memilih 10 baris teratas dengan jumlah penumpang terbanyak
        vendor_id  passenger_count  ...  total_amount  congestion_surcharge
214141        2.0              9.0  ...         11.76                   0.0
79823         2.0              8.0  ...         12.30                   2.5
737023        2.0              8.0  ...          8.30                   0.0
164792        2.0              7.0  ...          8.80                   0.0
385688        2.0              7.0  ...          8.10                   0.0
385689        2.0              7.0  ...         13.78                   2.5
385689        2.0              7.0  ...         13.78                   2.5
690829        2.0              7.0  ...          8.80                   0.0
732901        2.0              7.0  ...         72.80                   2.5
65            2.0              6.0  ...          5.30                   0.0
144           2.0              6.0  ...         11.44                   2.5

[10 rows x 12 columns]
------------------------------------------------------------------------
```

4. [Extra] Cast the data type to the appropriate value.

- `df_selected.loc[:, 'column_name']`: Menggunakan `.loc` untuk mengubah nilai di DataFrame secara aman tanpa menimbulkan `SettingWithCopyWarning`.
- `fillna(0).astype(type)`: Mengisi nilai `NaN` dengan 0 dan mengubah tipe data kolom sesuai dengan yang diinginkan.
- `print(df_selected.head())`: Menampilkan 5 baris pertama dari DataFrame `df_selected` setelah perubahan tipe data.

```
# Step 4: Cast the data types to the appropriate values using .loc to avoid SettingWithCopyWarning
print("4. Mengubah tipe data ke nilai yang sesuai")
df_selected.loc[:, 'vendor_id'] = df_selected['vendor_id'].fillna(0).astype(int)
df_selected.loc[:, 'passenger_count'] = df_selected['passenger_count'].fillna(0).astype(int)
df_selected.loc[:, 'trip_distance'] = df_selected['trip_distance'].fillna(0).astype(float)
df_selected.loc[:, 'payment_type'] = df_selected['payment_type'].fillna(0).astype(int)
df_selected.loc[:, 'fare_amount'] = df_selected['fare_amount'].fillna(0).astype(float)
df_selected.loc[:, 'extra'] = df_selected['extra'].fillna(0).astype(float)
df_selected.loc[:, 'mta_tax'] = df_selected['mta_tax'].fillna(0).astype(float)
df_selected.loc[:, 'tip_amount'] = df_selected['tip_amount'].fillna(0).astype(float)
df_selected.loc[:, 'tolls_amount'] = df_selected['tolls_amount'].fillna(0).astype(float)
df_selected.loc[:, 'improvement_surcharge'] = df_selected['improvement_surcharge'].fillna(0).astype(float)
df_selected.loc[:, 'total_amount'] = df_selected['total_amount'].fillna(0).astype(float)
df_selected.loc[:, 'congestion_surcharge'] = df_selected['congestion_surcharge'].fillna(0).astype(float)
print(df_selected.head())
print("------------------------------------------------------------------------\n")
```

```
4. Mengubah tipe data ke nilai yang sesuai
   vendor_id  passenger_count  ...  total_amount  congestion_surcharge
0        1.0              1.0  ...          9.30                   0.0
1        1.0              1.0  ...         27.80                   0.0
2        2.0              1.0  ...         22.30                   2.5
3        2.0              1.0  ...         14.16                   2.5
4        2.0              1.0  ...          7.80                   0.0

[5 rows x 12 columns]
------------------------------------------------------------------------
```