Data Gathering from Kaggle using API

July 27, 2024

0.1 Importing Necessary Libraries

[1]:

```
import pandas as pd
import zipfile
import os
import kaggle
```

0.2 Downloading dataset using Kaggle API

[2]:

```
!kaggle datasets download -d hmavrodiev/london-bike-sharing-dataset
```

```
Dataset URL: https://www.kaggle.com/datasets/hmavrodiev/london-bike-sharing-
dataset
License(s): other
Downloading london-bike-sharing-dataset.zip to C:\Users\HH\Downloads\Tableau +
Python Project



  0%|          | 0.00/165k [00:00<?, ?B/s]
100%|##########| 165k/165k [00:00<00:00, 173kB/s]
100%|##########| 165k/165k [00:00<00:00, 173kB/s]
```

0.3 Extracting the file from the downloaded Zip File

[3]:

```
zipfile_name= 'london-bike-sharing-dataset.zip'
with zipfile.ZipFile(zipfile_name, 'r') as file:
    file.extractall()
```

[5]:

```
bikes= pd.read_csv('london_merged.csv')
bikes.head()
```

[5]:

```
             timestamp  cnt   t1   t2    hum  wind_speed  weather_code  \
0  2015-01-04 00:00:00  182  3.0  2.0   93.0         6.0           3.0
1  2015-01-04 01:00:00  138  3.0  2.5   93.0         5.0           1.0
2  2015-01-04 02:00:00  134  2.5  2.5   96.5         0.0           1.0
3  2015-01-04 03:00:00   72  2.0  2.0  100.0         0.0           1.0
4  2015-01-04 04:00:00   47  2.0  0.0   93.0         6.5           1.0
```

```
   is_holiday  is_weekend  season
0         0.0         1.0     3.0
1         0.0         1.0     3.0
2         0.0         1.0     3.0
3         0.0         1.0     3.0
4         0.0         1.0     3.0
```

[6]:
```
bikes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17414 entries, 0 to 17413
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   timestamp     17414 non-null  object
 1   cnt           17414 non-null  int64
 2   t1            17414 non-null  float64
 3   t2            17414 non-null  float64
 4   hum           17414 non-null  float64
 5   wind_speed    17414 non-null  float64
 6   weather_code  17414 non-null  float64
 7   is_holiday    17414 non-null  float64
 8   is_weekend    17414 non-null  float64
 9   season        17414 non-null  float64
dtypes: float64(8), int64(1), object(1)
memory usage: 1.3+ MB
```

[7]:
```
bikes['weather_code'].value_counts()
```

[7]:
```
weather_code
1.0     6150
2.0     4034
3.0     3551
7.0     2141
4.0     1464
26.0      60
10.0      14
Name: count, dtype: int64
```

[9]:
```
bikes.shape
```

[9]:
```
(17414, 10)
```

0.4 Specifying the new column names

[11]:

```
new_cols_dict= {
    'timestamp': 'time',
    'cnt': 'count',
    't1': 'temp_real_C',
    't2': 'temp_feels_like_C',
    'hum': 'humidity_percent',
    'wind_speed': 'wind_speed_kph',
    'weather_code': 'weather',
    'is_holiday': 'is_holiday',
    'is_weekend': 'is_weekend',
    'season': 'season'
}

bikes.rename(new_cols_dict, axis= 1, inplace= True)
```

[ ]:

```
# Changing the humidity values to percentage
bikes.humidity_percent= bikes.humidity_percent / 100
```

[14]:

```
# Creating a seasons dictionary so that we can map the integers 0-3 to the␣
 ↪actual written values
season_dict= {
    '0.0': 'spring',
    '1.0': 'summer',
    '2.0': 'autumn',
    '3.0': 'winter'
}

# Creating a weather dictionary so that we can map the integers to the actual␣
 ↪written values
weather_dict= {
    '1.0': 'Clear',
    '2.0': 'Scattered clouds',
    '3.0': 'Broken clouds',
    '4.0': 'Cloudy',
    '7.0': 'Rain',
    '10.0': 'Rain with thunderstorm',
    '26.0': 'Snowfall'
}

# Changing the seasons column data type to text
bikes.season= bikes.season.astype('str')
bikes.season= bikes.season.map(season_dict)

# Changing the weather column data type to text
bikes.weather= bikes.weather.astype('str')
bikes.weather= bikes.weather.map(weather_dict)
```

0.5 Checking the updated DataFrame

[15]:
```
bikes.head()
```

[15]:
```
                  time  count  temp_real_C  temp_feels_like_C  \
0  2015-01-04 00:00:00    182          3.0                2.0
1  2015-01-04 01:00:00    138          3.0                2.5
2  2015-01-04 02:00:00    134          2.5                2.5
3  2015-01-04 03:00:00     72          2.0                2.0
4  2015-01-04 04:00:00     47          2.0                0.0

   humidity_percent  wind_speed_kph         weather  is_holiday  is_weekend  \
0              93.0             6.0  Broken clouds         0.0         1.0
1              93.0             5.0           Clear         0.0         1.0
2              96.5             0.0           Clear         0.0         1.0
3             100.0             0.0           Clear         0.0         1.0
4              93.0             6.5           Clear         0.0         1.0

   season
0  winter
1  winter
2  winter
3  winter
4  winter
```

0.6 Writing the DataFrame to an Excel file which will used in Tableau

[16]:
```
bikes.to_excel('london_bikes_final.xlsx', sheet_name= 'Data')
```