**Adatfeldolgozási folyamat és tisztítási döntések leírása**

**Adatforrás:**

A projekt adatait az **OpenWeatherMap API** segítségével gyűjtöttem, amely valós idejű időjárási adatokat szolgáltat világszerte.  
A lekérdezések Python segítségével, az API current weather végpontján keresztül történtek, 12 kiválasztott európai városra (pl. Budapest, London, München, Bordeaux, Glasgow stb.).  
Minden városról három különböző időpontban történt adatlekérés, így a dataset időbeli varianciát is tartalmaz.

**Adatgyűjtés és -struktúra**

Az API-hívások a requests könyvtárral készültek.  
Az adatok JSON formátumban érkeztek, majd a projektben írt flatten\_weather\_record() függvény segítségével **lapos, táblázatos szerkezetbe** lettek alakítva.  
A függvény a nyers JSON-ból csak a releváns kulcsokat tartotta meg:

* **Fő meteorológiai jellemzők:** temp\_c, feels\_like\_c, humidity\_pct, wind\_speed\_ms
* **Leíró mezők:** weather\_main, weather\_desc, cloudiness\_pct
* **Idő és hely adatok:** data\_calc\_unix, latitude, longitude, country, city

Az összesített adatokat pandas.DataFrame-be rendeztem, majd CSV formátumban mentettem (raw\_weather.csv).

**Adattisztítás és transzformáció**

Az adattisztítás célja az elemzéshez és vizualizációhoz megfelelő, konzisztens adattábla előállítása volt.

1. **Hiányzó értékek kezelése:**

A cloudiness\_pct és visibility\_m oszlopokat átvizsgáltam.

A visibility\_m mező konstans értékeket tartalmazott (10000 m), ezért **eltávolítottam** redundancia miatt.

1. **Idődimenzió kezelése:**

Az API által szolgáltatott UNIX timestamp (data\_calc\_unix) mezőt átkonvertáltam **UTC** időformátumba (data\_calc\_utc), majd a helyi időzónára (data\_calc\_local – *Europe/Budapest*).

Ezzel a későbbi időalapú vizualizációk (felhősség, hőmérséklet) konzisztensen értelmezhetők lettek.

1. **Új származtatott változó:**

Létrehoztam az is\_weekend indikátort, amely 1/0 értékkel jelzi, hogy a mért időpont hétvégére esik-e.

Ez később a viselkedési minták vagy városi aktivitások összevetéséhez is hasznos lehet.

1. **Földrajzi adatok egységesítése:**

A latitude / longitude mezők közvetlenül az API-ból származnak, így pontos koordinátákat biztosítanak a térképes megjelenítéshez.

A country mező ISO-formátumban (HU, DE, FR, stb.) szerepel, amely alkalmas volt ország-szintű aggregálásokhoz (pl. átlaghőmérséklet-heatmap).

**Vizualizáció és elemzés**

A tisztított adatokat a **Tableau** segítségével vizualizáltam.  
A végső dashboard négy interaktív elemet tartalmaz:

1. **Heatmap:** országonkénti átlaghőmérséklet az aktuális városi mérések alapján
2. **Scatter plot:** hőmérséklet és hőérzet összevetése városonként
3. **Bar chart:** városonkénti átlagos hőmérséklet, páratartalom és szélsebesség
4. **Line chart:** a felhősség százalékos arányának időbeli alakulása

A dashboard automatikusan méretezhető és részben interaktív, lehetővé téve az adatok időbeli, földrajzi és meteorológiai szempontból történő összehasonlítását.

**Összegzés**

A megoldás egy teljes adatfeldolgozási folyamatot mutat be – az adatok lekérésétől és tisztításától kezdve az interaktív vizualizációig.  
A kód moduláris, Python-dokumentált és bővíthető más városokra vagy időintervallumokra is.  
A Tableau dashboard megfelelően **áttekinthető, adatvezérelt döntéstámogató eszközt** biztosít.