BME AUT
Automatizálási és
Alkalmazott
Informatikai Tanszék

Félévi NagyHF dokumentáció

Üzleti Intelligencia Labor

2025 tavasz

Időjárás adatok és DAM ár összefüggésének vizsgálata

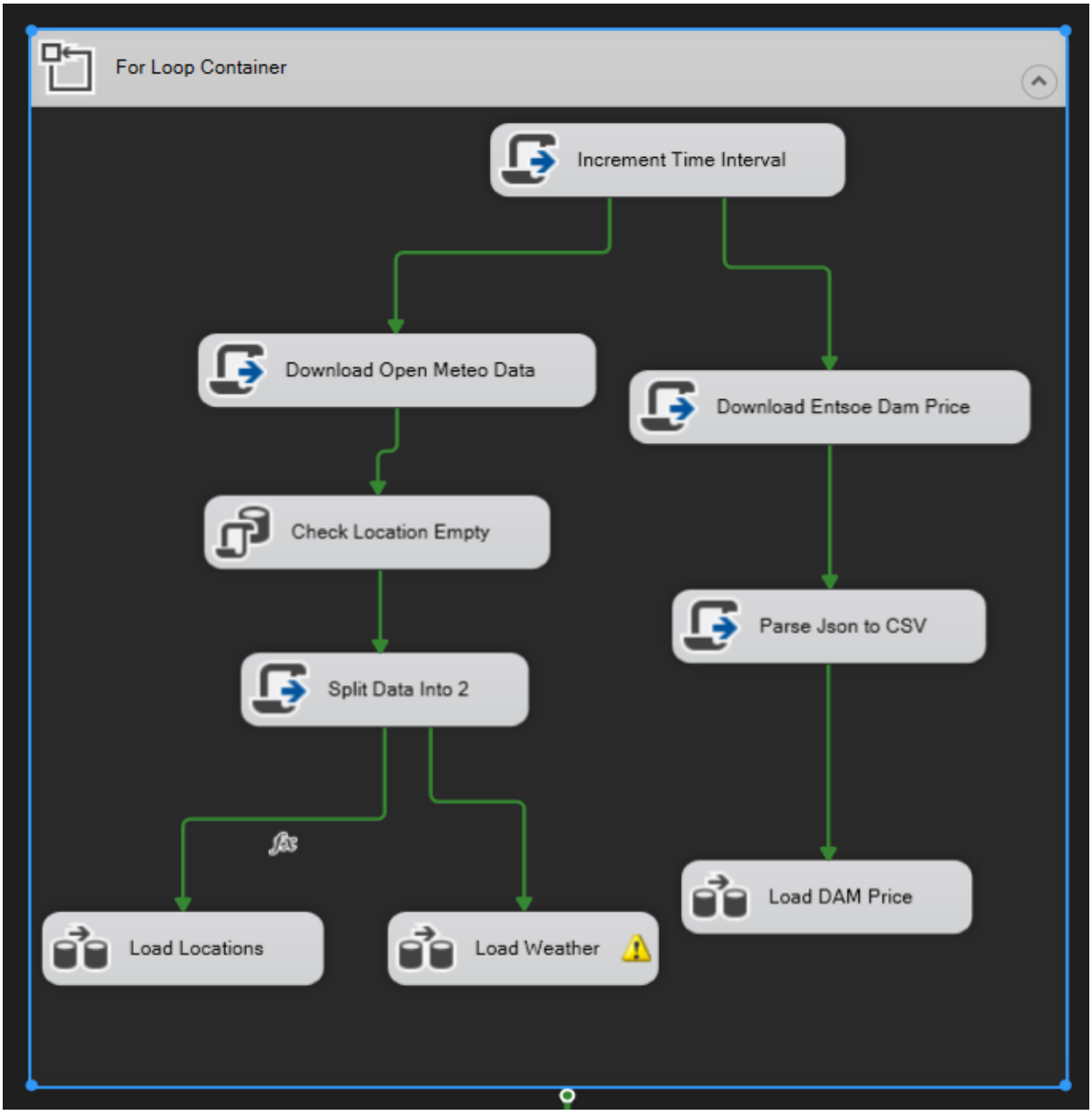
Biró Bence - (SM5P0J)

birben628@gmail.com

MŰEGYETEM 1782

ETL:

Az feladat adat migrációs és feldolgozásért felelős részét SQL Server Integration Service(SSIS)-ben készítettem.



Az adatok lekérését és feldolgozását egy loop-ba raktam be mivel az API-t, amit használtam csak egy hetes intervallumokba tudtam lekérni. Valamint a csv fájlok és JSON válaszok könnyű feldolgozásához Script Task-at használtam.

Valamint az API-nak volt limitációja arra nézve is, hogy mikortól tudok adatot lekérni. Ez február 23.

Valamint az egyes idő intervallumok megadásához variable-et használtam.

Name	Scope	Data type	Value	Expression
currentEnd	Package	DateTime	12/30/1899	
currentStart	Package	DateTime	2/23/2025	
endDate	Package	String		
loc_row_count	Package	Int32	0	
loopEnd	Package	DateTime	5/3/2025	
loopStart	Package	DateTime	2/23/2025	
startDate	Package	String		

Ezek szerepét a későbbiekben fogom kifejteni.

Increment Time Interval

```
public void Main()
{
    DateTime start = (DateTime)Dts.Variables["User::currentStart"].Value;
    DateTime finalEnd = (DateTime)Dts.Variables["User::loopEnd"].Value;

    DateTime end = start.AddDays(6);
    if (end > finalEnd) end = finalEnd;

    string startDate = start.ToString("yyyy-MM-dd");
    string endDate = end.ToString("yyyy-MM-dd");

    Boolean fireAgain = true;

    Dts.Variables["User::startDate"].Value = startDate;
    Dts.Variables["User::endDate"].Value = endDate;
    Dts.Variables["User::currentEnd"].Value = end;


    Dts.Events.FireInformation(0, "Script Task", $"Start: {startDate}", "", 0, ref fireAgain);
    Dts.Events.FireInformation(0, "Script Task", $"End: {endDate}", "", 0, ref fireAgain);


    DateTime newStart = start.AddDays(7);
    Dts.Variables["User::currentStart"].Value = newStart;

    Dts.TaskResult = (int)ScriptResults.Success;
}
```

Ennek a task-nak a feladata a loop elején a változtassa, hogy melyik hétre kéri le az adatot.

A currentStart a loop jelenlegi loop kezdetét jelzi. Ez folyamatosan nő 7 nappal.

A loopEnd azt jelzi, hogy mi az utolsó nap ameddig tartson az iterálás.

A startDate és endDate-et használja a többi Script Task az adatok lekéréséhez. Ez jelzi az egy hetes intervallumot.

A currentEnd a jelenlegi vég időpontot menti el.

Download Open Meteo Data

Ez a script felel azért, hogy az openMeteo oldaláról letöltse az adatokat csv formában, majd azokat elmenti a C:\SSIS mappába open-meteo.csv néven.

Ebben a Script task-ban beégetve Vannak, hogy milyen lokációkra kérjen le.

```
var hungaryCoordinates = new List<(double Latitude, double Longitude)>
{
    (48.100, 19.800),   // Salgótarján
    (47.687, 17.650),   // Győr
    (46.840, 16.850),   // Zalaegerszeg
    (46.460, 16.980),   // Nagykanizsa
    (46.080, 18.230),   // Pécs
    (46.250, 20.150),   // Szeged
    (47.530, 21.630),   // Debrecen
    (48.103, 20.778),   // Miskolc
    (47.783, 19.133),   // Vác
    (47.497, 19.040),   // Budapest
    (47.183, 20.200),   // Szolnok
    (46.907, 19.691),   // Kecskemét
    (47.000, 17.500),   // Pápa
    (47.350, 18.900),   // Tatabánya
    (47.800, 18.100),   // Esztergom
    (46.366, 17.783),   // Kaposvár
    (47.200, 16.616),   // Sárvár
    (46.983, 20.383),   // Békéscsaba
    (47.250, 20.150),   // Karcag
    (47.600, 20.500),   // Mezőtúr
    (46.683, 21.083),   // Orosháza
    (48.000, 21.700),   // Sátoraljaújhely
    (46.767, 18.383),   // Dombóvár
    (47.733, 21.083),   // Hajdúböszörmény
};
```

Check Location Empty

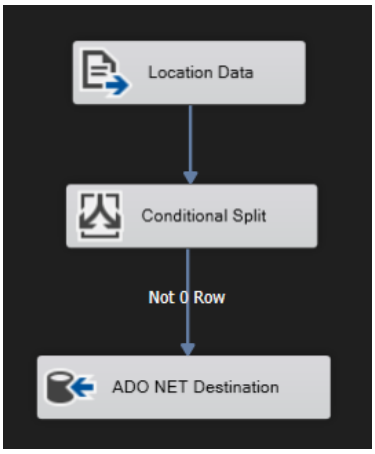
Ez a SQL task executer megszámolja hány érték van az adatbázis OMLocations táblájában és elmenti az értéket a loc_row_count-ba, ez később van felhasználva.

Split Data Into 2

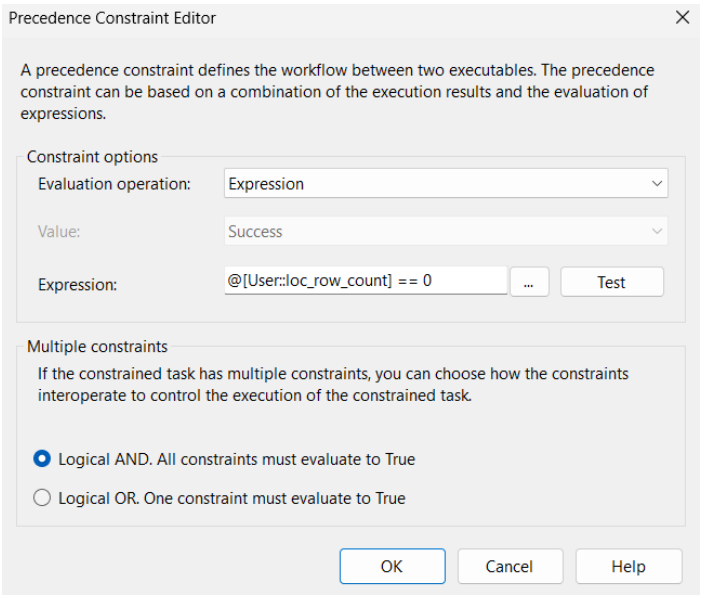
Ez a script task felel azért, hogy a letöltött Open Meteo adatok külön csv fájlba osztódjon szét, mivel a letöltött csv-ben együtt szerepelnek a helyrajzi adatok és az időjárás adatok. Ezeket a C:\SSIS mappába om-locations és om-weather néven menti el.

Load Locations

Ez a dataflow fele azért, hogy kiszűrje a null értéket és elmentse a csv-ben lévő adatot adatbázisba.

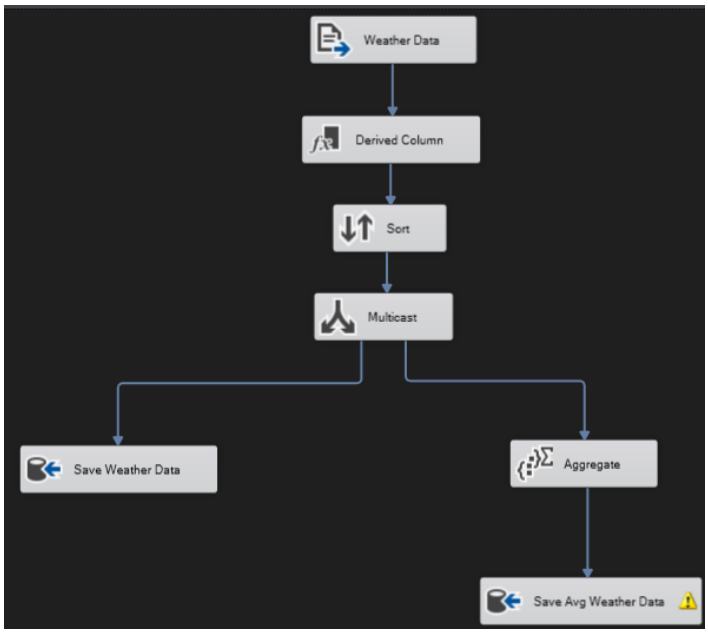


Ez a dataflow csak akkor fut le, ha az adatbázisban nincs az OMLocations táblában elem, amit a loc_row_count jelez.



Load Weather

Ez a Data Flow felel az időjárás adatok, feldolgozásáért. Egy részről, hogy ne legyen benne duplikáló más részről, hogy aggregálja aza adatokat. Az aggregálást úgy végzi, hogy országos szinten egy átlagot vesz a letöltött lokációkra. Ezeket az adatokat külön táblába menti.



Download Entsoe Dam Price

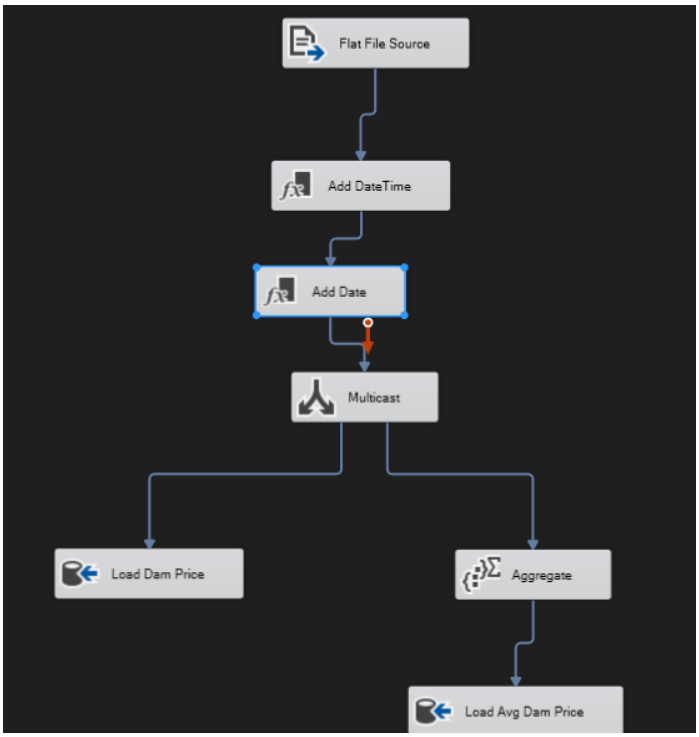
Ez a Script Task felel azért, hogy az Entsoe oldaláról letöltse a DAM árakat JSON formába majd azokat elmentse a C:\SSIS mappába entsoe_price_result.json néven.

Parse Json to CSV

Ez a Script Task felel az előzőekben elmentet JSON csv-é alakításáért. Amely ugyan úgy elmenti a C:\SSIS mappába entsoe_price_result.csv néven.

Load DAM Price

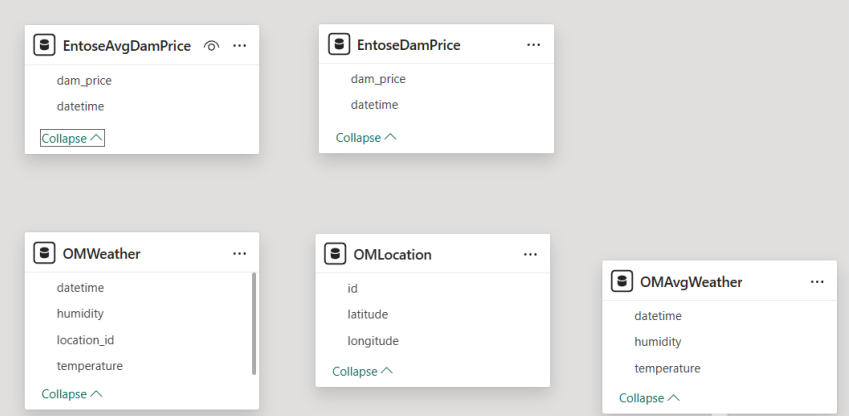
Az előzőekben átalakított CSV fájl fogja a Data Flow feldolgozni, hozzáadva egy DATETIME oszlopot, valamint egy DATE oszlopot, amely alapján lesz ez aggregálva, az órás adatok napos átlagra.



Adatbázis

Az adatbázis kezeléséhez Microsoft SQL Server Management Studio-t használtam.

Táblák:



EntsoeDamPrice: óránkénti DAM ár található

EntsoeAvgDamPrice: napi átlag DAM ár

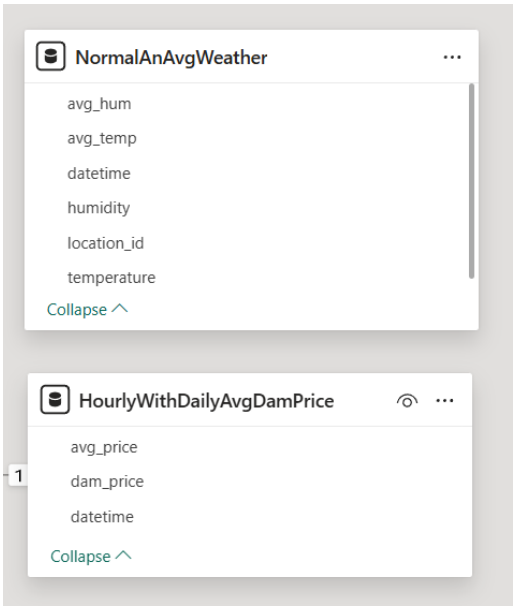
OMLocation: adott location ID-hoz tartozó koordináta

OMWeather: órás szinten a lokációkhoz tartozó hömérséklet és páratartalom

OMAvgWeather: órás szinten átlagolt hőmérséklet és páratartelom

Nézetek

Az adatok egyszerűbb megjelenítése érdekében, nézetekett, hoztam létre.



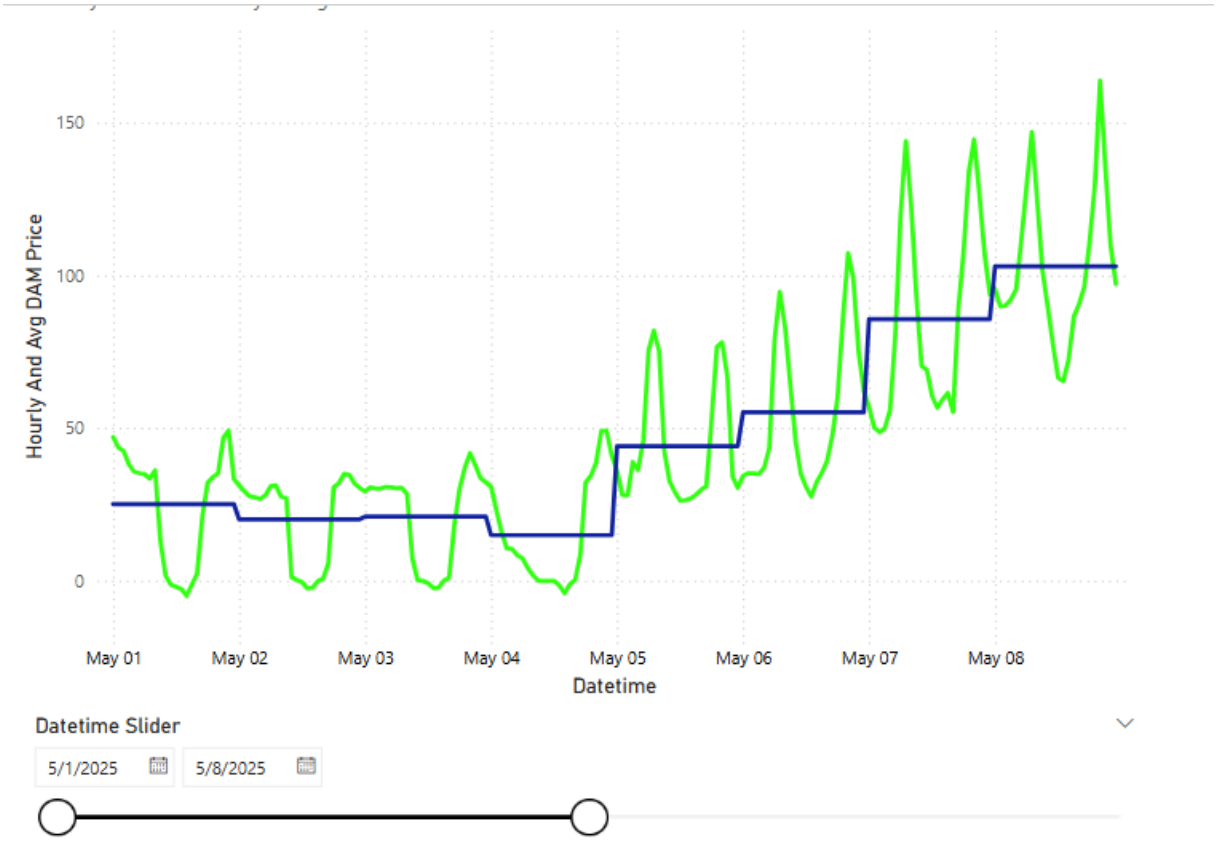
NormalAndAvgWeather: a lokációnkkénti és átlagolt időjárás adatok egy nézetben

HourlyWithDailyAvgDamPrice: órás DAM árak mellé az arra a napra vonatkozó DAM ár

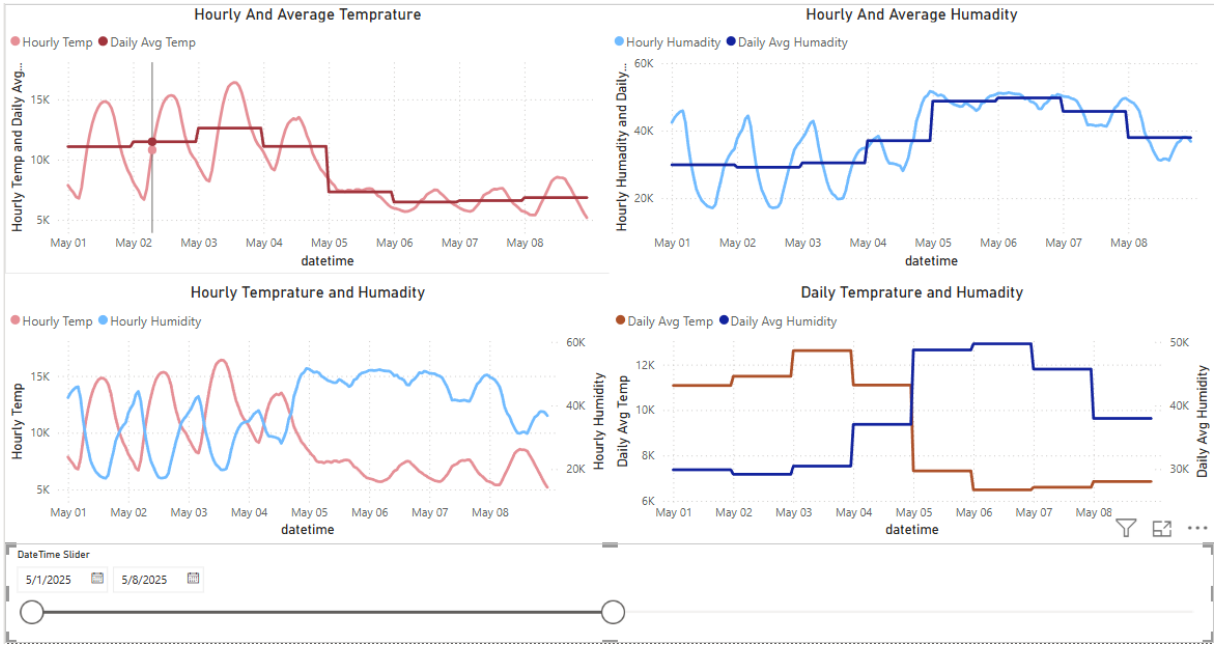
Vizualizáció:

Az adatok megjelenítéséhez Power BI-t használtam.

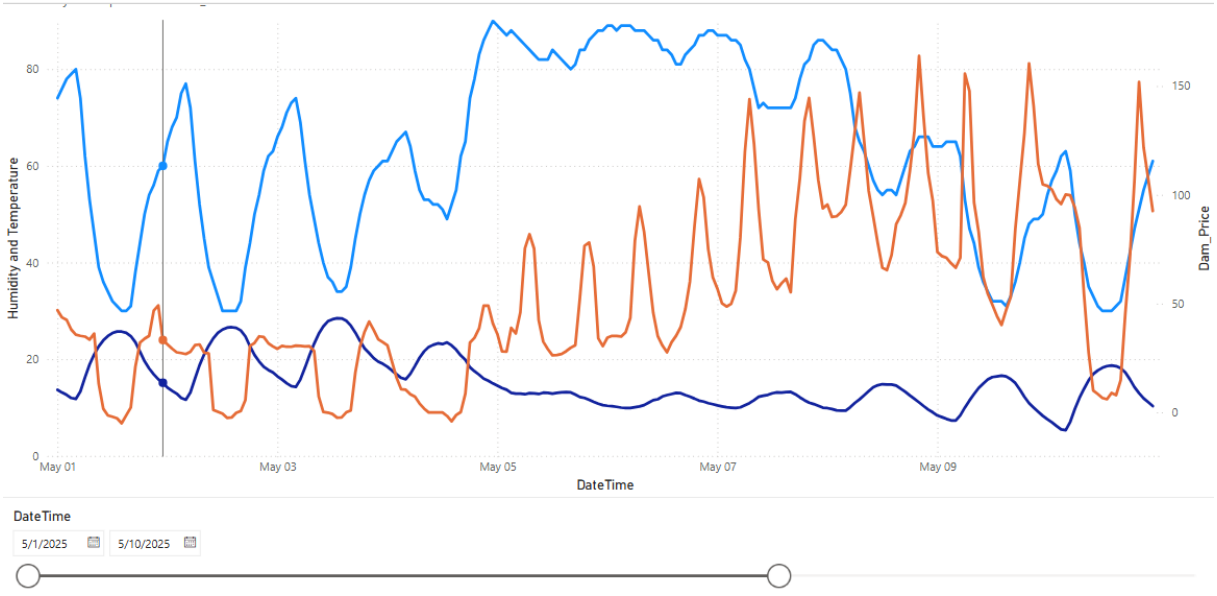
A diagramjaim egy sémára épülnek, ez pedig egy Line Chart-ot és egy Slicer-t jelent. A Line diagrammon jelenik meg az ár vagy az időjárás adat. A Slicer-rel pedig lehet állítani az idő intervallumot, amin megjelenjen.



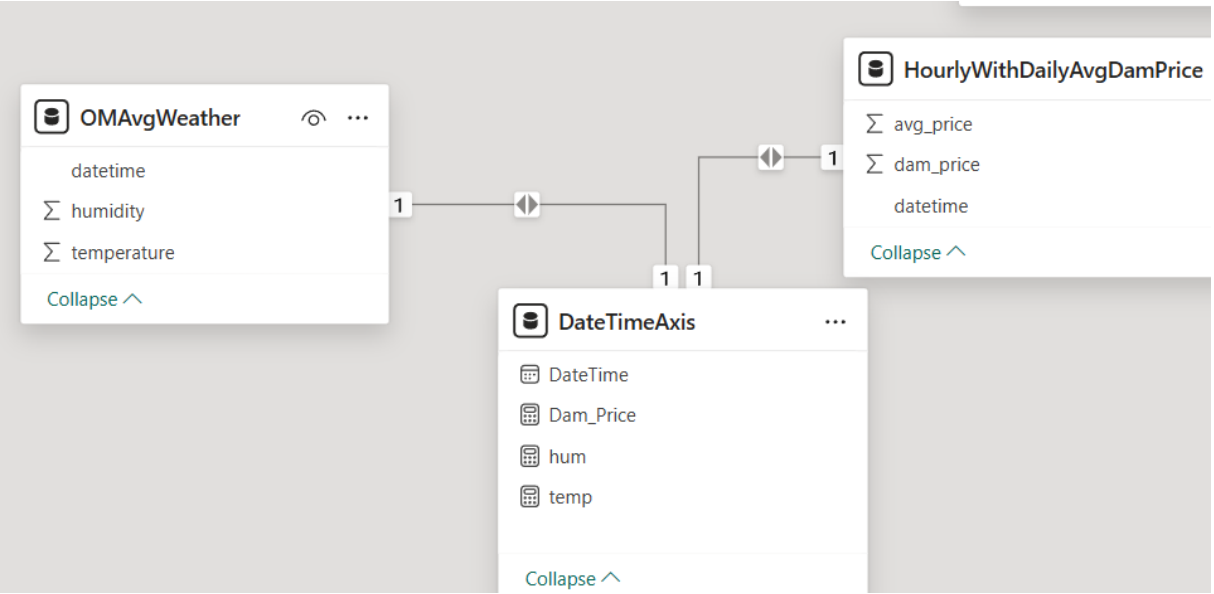
Ezen a diagrammon az órás DAM ár és a napi átlag jelenik meg.



Ezeken a diagramokon a hőmérséklet és a páratartalom jelenik meg egymáshoz viszonyítva vagy külön-külön.



Az utolsó diagrammon az időjárás adatok vannak összevetve a DAM árral.



Ehhez szükségem volt a modellingben egy új táblát létrehoznom. Ez a DateTimeAxis lett ebben lettek a datetime alapján össze join-va az értékek. Ez igazából a Nézet megoldás Power BI oldalon.

Adatelemzés

Az adatelemzést Python-ban írtam PyCharm-ot használva. Ezt a részt a félév közbeni labor mintájára készítettem.

Az eredmények nem lettek használhatóak, egyrészről csak a hőmérséklet és a páratartalomból nem lehet egy olyan komplex árat meghatározni, mint a DAM ár, más részről az időjárás API, amit használtam sem adott elegendő adatot, csak február 23-tól kezdődően.

Az adatokat nem csv-ből hanem az adatbázisból töltöttem-be. Ezzel egy nehézség adódott, mikor szerettem volna az adatokat Pandas DataFrame-re alakítani, akkor az egyes sorok <class 'pyodbc.Row'> voltak, amelyet nem tudott feldolgozni ezt átalakítottam <class 'tuple'>.

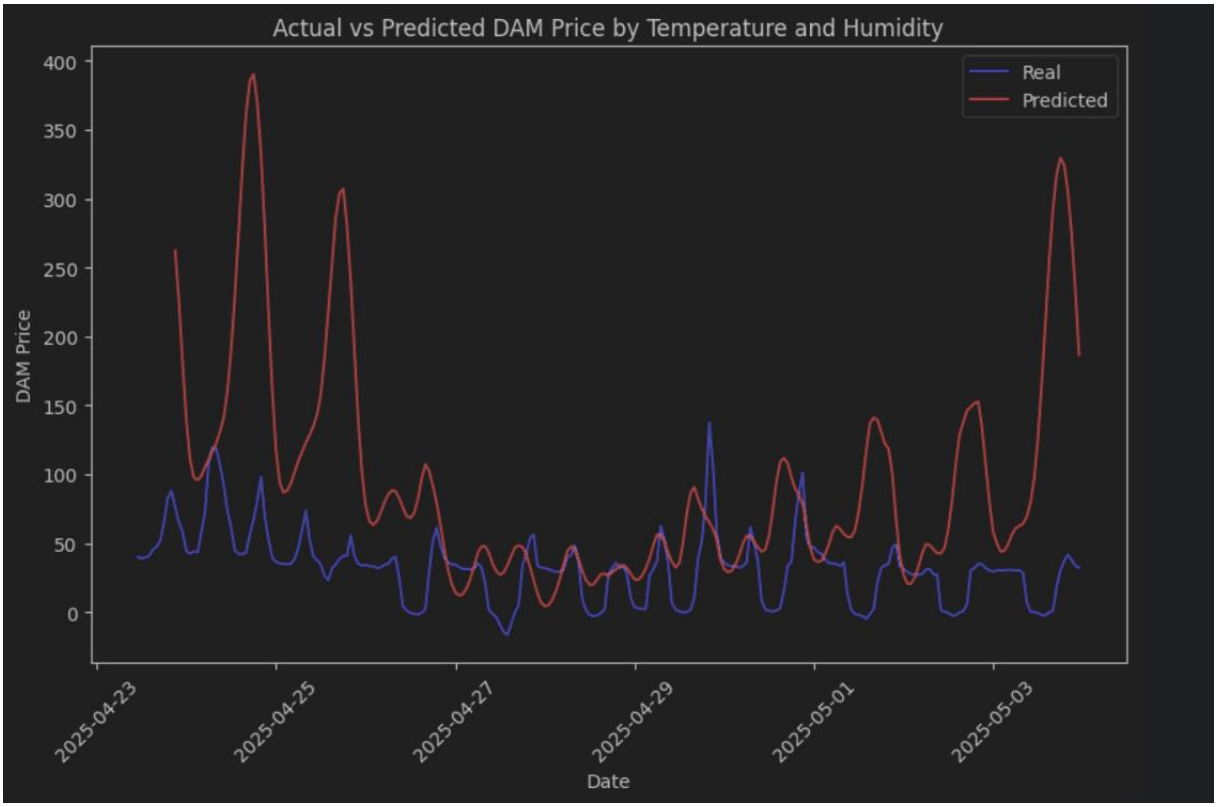
Két táblából kértem le adatot EntoseDamPrice és a OMAvgWeather, Ezeket DateTime alapján egy DateFrame-be alakítottam.

```
import numpy as np

df["Hour.Of.Day.X"] = np.sin(2 * np.pi * df["datetime"].dt.hour / 24)
df["Hour.Of.Day.Y"] = np.cos(2 * np.pi * df["datetime"].dt.hour / 24)
[6]
```

Ahhoz, hogy napszakot tudjon követni, az órát sin és cos függvényként leképeztem.

```python
from sklearn.preprocessing import StandardScaler

scaler_input = StandardScaler()
scaler_output = StandardScaler()

X_train = df_train.drop(columns=["dam_price"])
y_train = df_train[["dam_price"]]

X_valid = df_val.drop(columns=["dam_price"])
y_valid = df_val[["dam_price"]]

X_test = df_test.drop(columns=["dam_price"])
y_test = df_test[["dam_price"]]

scaled_train = scaler_input.fit_transform(X_train)
target_train = scaler_output.fit_transform(y_train)
scaled_valid = scaler_input.transform(X_valid)
target_valid = scaler_output.transform(y_valid)
scaled_test = scaler_input.transform(X_test)
target_test = scaler_output.transform(y_test)
```

Az adatokat StandardScaler-el normalizáltam 0-1 közötti értékre. A tanuló adathalmazból kivettem az értéket amire ráakarom tanítani(DAM ár).



Láthatóan a predikált értékek nagyon pontatlanok. De az észrevehető, hogy az órás trendeket, mikor nagyobb mikor alacsonyabb, arra rátudott tanulni.