



smartPulse

Leveraging Real Time API Services to measure the impact of weather conditions on Production Facilities

Amaç

Size göndermiş olduğum ilçe merkez koordinatlarını aşağıda belirttiğimiz api servislerine göndererek öncelikle 2020 sonrası tarihsel verileri, sonrasında da gerçek zamanlı verileri ve tahmin bilgilerini parse edilerek ön analizinin yapılması.

Hedefler

1. 2020 Mart ayından başlayarak belirtilen 973 ilçe merkez koordinatı için, saat bazında teknik bilgiler kısmında belirtilen verilerin elde edilmesi
2. Gerçek Zamanlı Verilerin aynı veri yapısında elde edilmesi ve python ($\geq 3.6.0$) ile her saat düzenli olarak çalışacak bir scheduler yazılması.
3. Elde edilen verilerin ön analizinin yapılarak rüzgar, güneş akısı ve benzeri verilerin gün içindeki sapmalarının, hangi il ve ilçeler için, hangi ölçekte gerçekleştiğinin belirlenmesi.

Teknik Bilgiler

Bu çalışma kullanacağınız **WeatherStack API** bilgisi aşağıdaki gibidir :

- 51d7bfab86b02431a6a6cc5d489ea60b

Gerçek zamanlı elde ettiğiniz verileri cross-check etmek için <https://openweathermap.org/>

sitesinden free bir API aktive ederek ekstra bir çalışma yapabilirsiniz. Bu siteden aynı zamanda 3 saat sonrasına kadar hava tahminlerini alabildiğiniz için, servisin bu tahminlerinin tutarlılığını da ölçmeniz mümkün.

Veri Yapısı

Her ilçe merkezi için saatlik olarak parse etmenizi istediğim değerler şöyle :

<https://weatherstack.com/documentation>

astro > sunrise	Returns the local sunrise time in the format hh:mm am/pm.
astro > sunset	Returns the local sunset time in the format hh:mm am/pm.
astro > moonrise am/pm.	Returns the local moonrise time in the format hh:mm
astro > moonset am/pm.	Returns the local moonset time in the format hh:mm
astro > moon_illumination	Returns the moon illumination level as percentage.

hourly > time	Returns the time as a number in 24h format
hourly > temperature Celsius)	Returns the temperature in the selected unit. (Default:
hourly > wind_speed kilometers/hour)	Returns the wind speed in the selected unit. (Default:
hourly > wind_degree	Returns the wind degree.
hourly > wind_dir	Returns the wind direction.
hourly > weather_code	Returns the universal weather condition code
hourly > weather_descriptions	Returns one or more weather description texts
hourly > precip (Default: MM - millimeters)	Returns the precipitation level in the selected unit.
hourly > humidity	Returns the air humidity level in percentage.
hourly > visibility kilometers)	Returns the visibility level in the selected unit. (Default:

hourly > pressure millibar)	Returns the air pressure in the selected unit. (Default: MB -
hourly > cloudcover	Returns the cloud cover level in percentage.
hourly > heatindex (Default: Celsius)	Returns the heat index temperature in the selected unit.
hourly > dewpoint (Default: Celsius)	Returns the dew point temperature in the selected unit.
hourly > windchill (Default: Celsius)	Returns the wind chill temperature in the selected unit.
hourly > windgust kilometers/hour)	Returns the wind gust speed in the selected unit. (Default:
hourly > feelslike (Default: Celsius)	Returns the "Feels Like" temperature in the selected unit.
hourly > chanceofrain	Retuns the chance of rain (precipitation) in percentage.
hourly > chanceofremdry	Retuns the chance of remaining dry in percentage.
hourly > chanceofwindy	Retuns the chance of being windy in percentage.
hourly > chanceofovercast	Retuns the chance of being overcast in percentage.
hourly > chanceofsunshine	Retuns the chance of sunshine in percentage.
hourly > chanceoffrost	Retuns the chance of frost in percentage.
hourly > chanceofhightemp	Retuns the chance of high temperatures in percentage.
hourly > chanceoffog	Retuns the chance of fog in percentage.
hourly > chanceofsnow	Retuns the chance of snow in percentage.
hourly > chanceofthunder	Retuns the chance of thunder in percentage.

Çıktı olarak csv / excel formatında aşağıdaki gibi bir veri modeli bekliyorum

ILCEKOD	TIME	TEMPRATURE	WIND_SPEED	etc		
int64	datetime64	float64	float64	...		

Python içerisindeki pandas modülünü kullanırsanız rahat edersiniz.

Parametrelerin birim formatı olarak metrik sistemini kullanabilirsiniz. Servisin default değerleri zaten bu birime dayalıdır.

`units = m`

temperature: Celsius

`units = m`

Wind Speed/Visibility: Kilometers/Hour

`units = m`

Pressure: MB - Millibar

`units = m`

Precip: MM - Millimeters

`units = m`

Total Snow: CM - Centimeters

Notlar :

- Saatlik scheduling islemi icin python icindeki [schedule](#) veya [cron](#) kutuphanelerini yapabilirsiniz.
- Wrapper olarak apixu kullanabilirsiniz ya da <https://weatherstack.com/documentation> sitesinde yer alan orneklerden baslayabilirsiniz.
- Openweathermap ile forecast degerlerini de elde etmek isterseniz kendi github sayfali ve dokumantasyonlarinin yani sira asagidaki modulden de yararlanabilirsiniz <https://github.com/crazycapivara/openweathermapy>
- API'nin request kotasi biterse lütfen benimle iletisime gecin (usurat@smartpulse.io)
- Başarılar !