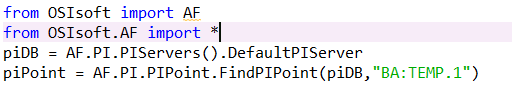
**MAKİNE ÖĞRENİMİ DÖKÜMANTASYONU**

**Adım 1:** PI System’den AFSDK ile Python’da veri çekilmesi

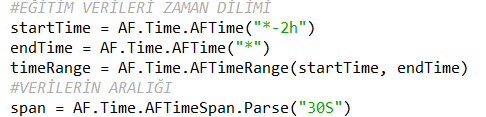
1. Bir kurulumu içe aktarmak ve kullanmak için AddReference() metodu kullanılır.



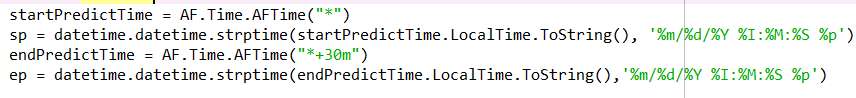
1. Database bağlantısını yapıp ilgili tag’i çekebiliriz.(Örnekte “BA:TEMP.1” tagi çekilmiştir.)



1. Çekilecek verilerin zaman diliminin ayarlanması.(Örnekte “\*” şu anki zaman dilimini , “\*-2h” şu anki zamandan 2 saat öncesine kadar ki zaman dilimini göstermektedir.

“span” değişkeni ise verileri çekerken ne kadar süre aralıklarla çektiğimizi gösterecek.)

1. Tahmin edilecek zaman diliminin belirtilmesi.



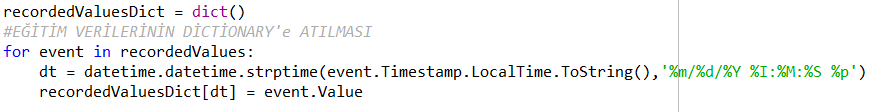


1. Verilerin PI Data Archive’den çekilmesi.(Parametre olarak yukarda belirlediğimiz değişkenleri koyuyoruz.)

C:\Users\huseyinguveli\AppData\Local\Microsoft\Windows\INetCache\Content.Word\verilerin çekilmesi1.png

**Adım 2:** Verilerin önişlemesi.(Dataframe türüne çevirme, tarihleri sıralama)

1. Bir dictionary oluşturup, çekilen verileri bu dictionary’nin içine atıyoruz



1. Bir “dataframe” oluşturup, oluşturduğumuz dictionary içindeki verileri bu dataframe’e atıyoruz. İkinci satırdaki kod ile “TimeStamp”i “datetime”a çeviriyoruz.



1. Daha sonra yine bir dataframe oluşturup bu dataframe’in index’ini “TimeStamp”deki verilerle doldurmuş oluyoruz. Ardından “Index”i sıralıyoruz.



**Adım 3:** Regresyon ve tahmin

1. Kullanacağımız regresyonun kütüphanesini çağırıyoruz.



1. Autoregressive Modeli oluşturuyoruz. C:\Users\huseyinguveli\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ar_model.png
2. Oluşturduğumuz modelimizin eğitimini fit() fonksiyonu ile yapıyoruz. C:\Users\huseyinguveli\AppData\Local\Microsoft\Windows\INetCache\Content.Word\modelin eğitilmesi.png
3. Tahminimizi predict() fonksiyonu ile yapıyoruz. “start” ve “end” parametrelerine daha önce oluşturduğumuz(Adım1: (d)) “sp” ve “ep” değişkenlerini string türüne çevirip yazıyoruz.



**Adım 4:** Verilerin grafiksel olarak gösterilmesi.

* Öncelikle interaktif modu aktif etmek için plt.ion() komutunu kullanıyoruz.
* Ardından sonsuz döngü olduğundan dolayı grafikleri yazdırmadan önce plt.show() yapıyoruz. Grafikler üst üste binmesin diye.
* plt.plot() komutu ile eğitim ve tahmin verilerimizi çizdiriyoruz.
* Değiştirilmiş ancak otomatik olarak grafiğe çizilmemiş değerleri grafiğe çizdirmek için plt.draw() komutunu kullanıyoruz.
* plt.gcf().clear() eski grafiği silip figure’ü boşaltıyoruz. Böylece hızlı şekilde veriler geldiğinde rahat bir şekilde üst üste binmeden grafiğimizi canlı hale getiriyoruz.

