**TÜRKİYE TEKNOLOJİ TAKIMI VAKFI YAPAY ZEKA EĞİTİMİ LİSE**

**DERS PLANI 5**

**DERS ETİKETLERİ**

Sınıf: **Lise**

Zorluk Düzeyi: **Zor**

Süre: **200 Dakika**

**Konular**

**1.** Etkinlik

**2.** Makine Öğrenmesi Metodu

**3.** Uygulama

**4.** Makine Öğrenmesi Metodu

**5.** Uygulama

**6.** Model Selection ve Örneği

**KAZANIM**

● Öğrenciler makine öğrenmesi metodları hakkında bilgi sahibi olurlar.

● Öğrenciler, “Makine Öğrenmesi Algoritmalarını” kavrarlar.

● Öğrenciler algoritmalar kurarlar.

●Python üzerinde yapay zeka örnekleri yaparlar.

**BECERİLER**

● Eleştirel Düşünme

● Metabiliş Beceriler

● İşbirliği ile çalışma

**DERS GENEL HATLARI**

**Harekete geç:** Kahoot uygulaması ile derse geçiş yapılır.

**Keşfet:** Etkinliklerle makine öğrenmesi metotları ve algoritmalarını anlamlandırır.

**Üret/İlerlet:** Python üzerinde uygulamalar yapar.

**Değerlendir:**Sohbet edilerek günün dönüşümlü değerlendirmesi yapılır.

**1.Adım Harekete Geç:**

**4. Hafta Kahoot**

Kahoot uygulaması ile eğlenerek derse başlayınız.Hemde burada elde edeceğimiz veriler ile öğrencilerimizin 4.dersteki eğitimimizin verimi hakkında bilgi sahibi olacağız.

●Öğrenciler bilgisayarlarında veya eğitmen etkileşimli tahta, projeksiyonda

https://kahoot.it/ ortamını açarlar.

● Eğitmenin daha önceden oluşturmuş olduğu ölçme değerlendirme aracı olan

quiz öğrenciler tarafından yarışma yapılarak çözülür.

● Kahoot it üzerinde her öğrenci öğrendiklerini pekiştirir ve kendini değerlendirir.

|  |  |
| --- | --- |
|  | **Eğitmene Not:** Kahoot uygulaması ile:  ● Öğrencilerin Turnuva tekniği ile bilgilerinin ölçme ve değerlendirilmesi amaçlanır.  ● Öğretmen sınıfın konu hakkındaki sahip oldukları bilgi düzeyini ölçer.  <https://create.kahoot.it/share/5-hafta-egitim-baslangc-testi-lise/5a763a5a-1e18-4ce0-93db-4145fe7ee671>  Kahoot uygulamasına yukarıda bulunan linkten ulaşabilirsiniz. |

**Videolar ile Önceki Hafta Tekrarı**

• <https://www.youtube.com/watch?v=nKW8Ndu7Mjw>

• <https://www.youtube.com/watch?v=ggIk08PNcBo>

**2.Adım Keşfet:**

**Regresyon Etkinliği**

Aşağıdaki linkten ulaşacağınız klasörde etkinlik ile ilgili tüm detayları bulabilirsiniz.

<https://drive.google.com/drive/u/0/folders/1wfO74PASnYyB-tJK9p0JHw3rC2inFgMz>

**Regresyon Anlatım**

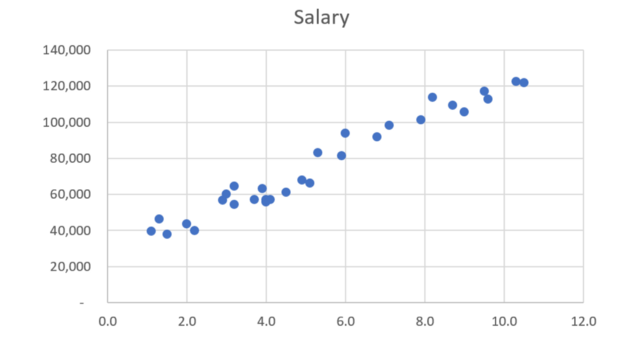
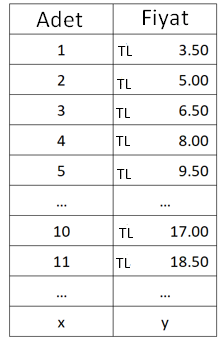
Eğitmenimiz dersin bu bölümünde sırasıyla Simple Lineer, Multiple Lineer ve Polynomial Regression algoritmaları anlatılır. Algoritmalar anlatılırken birbirlerinden farkı ve onlarla olan ilişkisinden bahsedilmesi öğrenciler için faydalı olabilir.

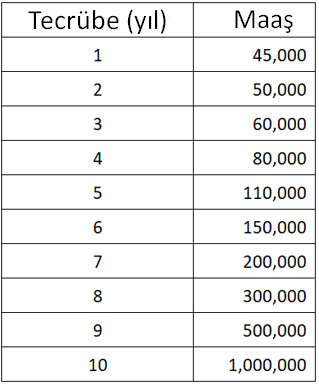
Aşağıdaki linkte bulunan sunumdan dersin anlatımını sağlayabilirsiniz.

<https://drive.google.com/file/d/1RT9ZC5wgLkyy3kcfJxVm5-fPjQh7BVub/view>

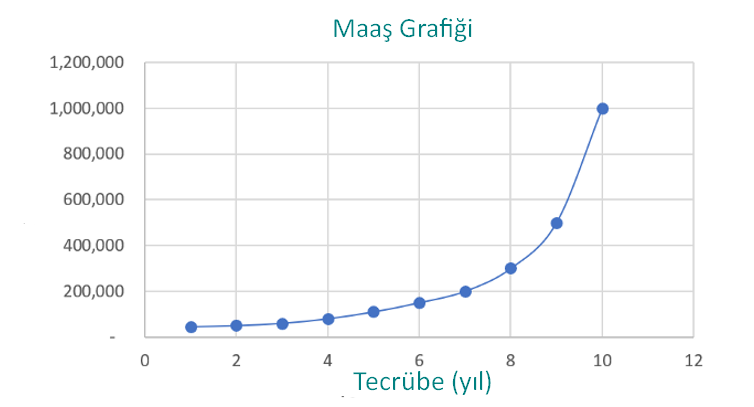
**Regresyon**

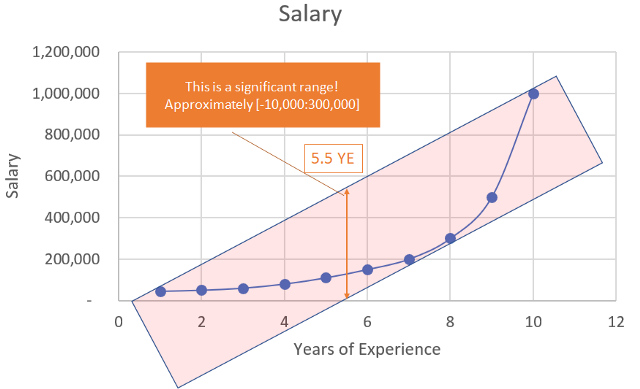
Diyelim ki, BİM’de alışveriş yapıyorsunuz. Bir şey alsanız da almasanız da, otopark için 2,00 TL ödemeniz gerekiyor. Elma alacaksınız, her elmanın fiyatı 1.5 TL ve en az bir elma satın almak zorundasınız. Bu bilgiler ışığında aşağıdaki tabloyu doldurabiliriz (Fig.1).

Bu örnek için y = 2 + 1.5x denklemini kullanarak temel alarak toplam fiyatı tahmin etmek (veya hesaplamak) kolay görünüyor. Lineer denklemi çıkardıktan sonra sayı fark etmeksizin toplam maliyeti hesaplayabiliyoruz. Fakat, maalesef aslında işler o kadar basit ve kolay değil. Bu örnekte tüm verilerimiz aynı y = 2 + 1.5x doğrusunun üzerinde oldu. Normalde verilerin aynı doğru üzerinde olması beklenmez fakat belli sapmalara rağmen doğrusallık gözlenebiliyorsa lineer regresyon uygulayıp, bazı isabetli tahminler yapabiliriz.

Az önceki BİM örneğinde maliyet denklemimiz y = 2+ 1.5x bulunmuştu. Peki ya bu denklemimiz kareli, küplü ifadelerle bir polinom olsaydı o zaman ne olurdu?

Şimdi bir de maaş örneğimize bakalım. 1 yıl tecrübeli öğretmeninin maaşı 1 yıl tecrübeli mühendisten farklıdır; 1 yıl tecrübeli inşaat mühendisi makine mühendisinkinden farklıdır. Ve eğer 2 farklı firmadan 2 makine mühendisini karşılaştırırsanız, maaşları da yine çoğunlukla farklıdır. Peki, şirketinize iş başvurusu yapan bir adayın maaşını nasıl tahmin edebilirsiniz? Fig.3’te görülen tabloda aynı mesleğe sahip, yıllık tecrübelerine göre sıralanmış 10 farklı maaş verisi bulunmaktadır. Öncelikle bu veriyi biraz görselleştirmek amacıyla grafiğini çizelim.

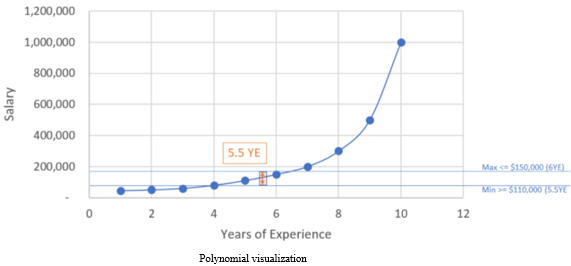


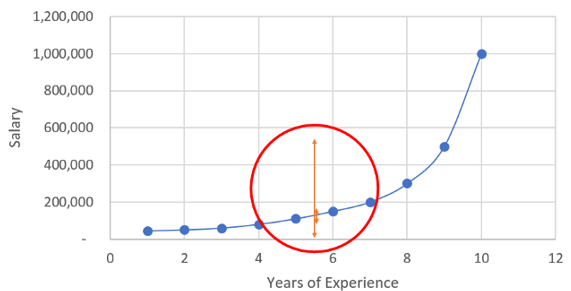


Şimdi adayımızın 5 yıllık bir tecrübesi olduğunu varsayalım. Lineer modele kullanırsak eğer nasıl bir sonuçla karşı karşıya kalacağız?

Fig.5’te görüldüğü üzere, 5 yıl tecrübesi bulunan adayımızın tahmini maaş aralığı -10,000 ile 300,000 arasında. Neden böyle bir sonuça karşılaştık? Çünkü bizim verilerimiz lineer yani doğrusal değiller, kavisli polinomal yapıda bir grafiğimiz var. İşte bu yüzden lineer regresyon bu durumda bize doğru sonuçları vermiyor. Bunun gibi durumlarda daha doğru sonuçlar elde edebilmek için polinom regresyonu kullanacağız.

Polinom regresyon formülü : ß0 + ß1 X1 + ß2 X22 + …. +  ßi Xni

Örneğimize geri dönersek, biz zaten 5 yıl ve 6 yıl tecrübesi olanların maaşlarını biliyoruz. Yani, 5.5 yıllık bir tecrübesi olan çalışan muhtemelen bunların maaşları arasında bir maaşla çalışacaktır. 



Polinom formülümüz gerçek grafiğe ne kadar yakın olursa o kadar doğru ve tutarlı sonuçlar elde ederiz. Sadece şimdiki bilgilerimizle lineer ile polinom regresyonunu grafik üzerinden karşılaştırarak farkı daha iyi anlayalım.



Şuanki ilkel haliyle bile polim regresyonu lineer regresyondan 7.75 kat daha iyi bir sonuç veriyor. Peki 5.5 yıllık adayımızın tahmini maaşını tam olarak nasıl hesaplayacağız? Ortalama değeri kullanarak hızlıca hesaplayabiliriz. 5.5 5 ile 6’nın ortalama değeri olduğu için, maaşı da (110+150)/2’den 130 olarak hesaplayabiliriz. Bu, elle yaptığımız çok basit hesaplama fakat bilgisayarlarımızı kullanarak polinom denklemimizi verilerimize daha iyi uydurduğumuzda çok daha doğru ve tutarlı sonuçlar elde edebileceğiz.

**Regresyon Video**

• <https://www.youtube.com/watch?v=CtKeHnfK5uA>

**Python ile Regresyon Uygulaması**

Aşağıdaki linkten ulaşacağınız klasörde uygulama ile ilgili tüm detayları bulabilirsiniz.

<https://drive.google.com/drive/u/0/folders/1YYS8JgM8Mu_NzfMJ0RRWerPv_Nc3oqIE>

**Apriori Anlatım**

Aşağıdaki linkte bulunan sunumdan dersin anlatımını sağlayabilirsiniz.

<https://drive.google.com/file/d/1RT9ZC5wgLkyy3kcfJxVm5-fPjQh7BVub/view>

|  |  |  |  |
| --- | --- | --- | --- |
| **Association Rule Learning:** Olayların birlikte gerçekleşme durumlarını değerlendirip veriler arasında ilişkiler kuran bir makine öğrenmesi yöntemidir. Birçok makine öğrenme algoritması sayısal verilerle çalışır ve bu algoritmalar daha çok matematiksel olma eğilimindedirler. (Regression algorithms, Support Vector Machines ). Ancak, Association Rule Mining algoritmaları kategorik veriler ile çok başarılı bir şekilde çalışırlar.  Association Rule Mining’e örnek olarak genellikle market sepeti uygulaması verilir. Bu işlem, müşterilerin yaptıkları alışverişlerdeki ürünler arasındaki ilişkileri bularak müşterilerin satın alma alışkanlıklarını ortaya koyar. Örneğin; kola alan müşteri ekmek de alır ya da un alan müşteri şeker de alır gibi… Bu tip ilişkilerin keşfedilmesi, müşterilerin hangi ürünleri bir arada aldıkları bilgisini ortaya çıkarır ve market yöneticileri de bu bilgi ışığında raf düzenlerini belirleyerek satış oranlarını artırabilir ve etkili satış stratejileri geliştirebilirler.  **Destek (Support):** Bir varlığı içeren eylem sayısının toplam eylem sayısına oranıdır. ( A / Tüm eylem sayısı )  **Güven (Confidence):** İki varlığı içeren eylem sayısının birine oranıdır.((A+B ) / A ) )  🡪Bir birliktelik kuralı oluşturmak için destek ve güven değerlerine bakılır. Kural için minumum destek ve minumum güven şartı aranır.  Birliktelik kuralı için kullanılabilecek diğer ölçüler **Lift** (Kaldırmak), **Conviction** (Kanaat), **All-Confidence** (Tam Güven), **Collective** **Strength** (Kolektif güç) ve **Leverage**’tır. (Kaldıraç)  Bir market sepeti örneğiyle Apriori algoritmasının nasıl çalıştığını çözelim.  Örneğin yukarıdaki şekilin bir marketteki müşterilerin alışveriş sepetleri olduğunu düşünelim. 1. tabloya baktığımızda alınan ürünleri görüyoruz. ( 1 3 4 – 2 3 5 – 1 2 3 5 – 2 5) Algoritma ilk olarak bu ürünlerin frekansını bulur yani toplam alınma sayısını (1. ürün 2 defa alınmış, 3. ürün 3 defa gibi). Durumlara göre belirlenen minimum support değerinin %50 olarak belirleyelim. Bu frekans değerlerini bulduktan sonra en yüksek frekanslının minumum support değeri alınır ( %50–3\*50/100 = 1.5 ) ve frekansı bu değerden küçük olanlar listeden elenir. Kalan değerlerin ikili kombinasyonu alınarak aynı işlem tekrarlanır ve tablo daha da küçültülür.    Ürünlerin ikili kombinasyonları alındıktan sonra bu kombinasyonların frekansları belirlenir, en büyük freakans kullanılarak minimum support değeri bulunur ve minimum support değeri altında kalan kombinasyonlar elenir (1 2 ve 1 5 ikili kombinasyonları elenmiştir.). Daha sonra ürünlerin üçlü kombinasyonları ele alınarak aynı işlem tekrarlanır. Bu aşamada ise 1 2 3 – 1 3 5 – 1 2 5 üçlü kombinasyonları elenir ve sadece 2 3 5 üçlü kombinasyonu kalır.  Sonuç olarak 2 3 ve 5 numaralı ürünler arasında bir ilişki bulmuş olduk. Yani, bu üç ürün sıklıkla beraber alınıyor ve aynı alışveriş sepetlerinde bulunuyor. Bu durum göz önüne alınarak; ürünlerin raf yerleri düzenlenerek, bu ürünlere özel promosyonlar yaparak market satışları maksimize edilebilir. Şimdi bu algoritmayı bir de Python üzerinde bir örnekle beraber inceleyelim.  **Not:** Apriori algoritması “sklearn” içinde yer almadığından dolayı, işlemi github’tan “apyori” adında bir kütüphane ile gerçekleştireceğiz. Kütüphaneyi indirdikten sonra apyori.py dosyasını projemizle aynı dizine atmanız gerekmektedir. | | | |

**Apriori Video**

• <https://www.youtube.com/watch?v=WGlMlS_Yydk>

**Apriori Etkinliği**

Aşağıdaki linkten ulaşacağınız klasörde etkinlik ile ilgili tüm detayları bulabilirsiniz.

<https://drive.google.com/drive/u/0/folders/1yJ5hmaDGzeGAVxtI-Ip8B-oN89XffH7P>

**Python ile Apriori Uygulaması**

Aşağıdaki linkten ulaşacağınız klasörde uygulama ile ilgili tüm detayları bulabilirsiniz.

<https://drive.google.com/drive/u/0/folders/13vazyDkHWnjPyZ8nGOGmA7ZlrjMWzpR4>

**3. 4. Adım Üret/İlerlet :**

**Model Selection Anlatım**

|  |  |  |  |
| --- | --- | --- | --- |
| Bir makine öğrenme algoritmasını verilerimize uyguladıktan sonraki adım, modelin veriye ne kadar uygun olduğunu bulmaktır. Farklı makine öğrenmesi algoritmalarını değerlendirmek için farklı performans metrikleri kullanılmaktadır. Yani, elimizdeki veri kümesine bağlı olarak farklı algoritmaların verimliliğini ölçmek için farklı metrikler gereklidir.  Bir makine öğrenmesi modelinin performansını değerlendirirken bir sonraki önemli nokta, model performansını değerlendirmek için hangi veri kümesinin kullanılması gerektiğidir. Makine öğrenmesi modeli, eğitim seti kullanılarak basitçe test edilemez, çünkü çıktılar önyargılı olabilir.Train edilen data üzerinde test etmek gibi. Sınavı yapmadan önce öğrencilere cevap anahtarı dağıtmak gibi bir şey aslında bu. Bu nedenle genelleme hatasını tahmin etmek için modelin henüz görmediği bir veri setini test etmesi gerekmektedir yani test veri setini. Bu nedenle, modelin test edilmesi amacıyla etiketli bir veri kümesi gerekir.  **K-Fold Cross Validation:**  Verilerle uğraşırken karşılaştığımız temel zorluklardan biri, overfitting ve underfitting’dir. Bu durumda variance ve bias’tan bahsetmek gerekecektir. Bias**,**tahminimizin gerçek değerden ne kadar uzak olduğudur. Varianceise Veri setindeki değerlerin ortalamaya göre dağılımını gösterir.    Veri setini eğitim ve test set olarak ayırmamızın amacı, olası overfitting’den kaçınmak ve modelin daha önceden görmediği veri seti üzerinde nasıl performans gösterdiğini anlamak içindir. Fakat modelimizin eğitim ve test aşamasında dağılımdan kaynaklı bazı hatalar olabilir. Bu hataları minimum seviyeye indirmek için K – Fold Cross Validation tekniği kullanılır. Eğitim veri setini rasgele k parçaya böler. k-1 parça eğitim için kullanılırken 1 parçada test seti için kullanılır ve k defa bu işlem tekrar eder. Her raund’da elde edilen değerler toplanır ; ortalaması alınır ve modelin performansı değerlendirilir.  Yandaki görselde veri seti 10 parçaya bölünmüştür. Her turda, mavi boyalı alan test setine ayrılmışken diğer parçalar eğitim için ayrılmıştır.Her turun sonunda sınıflandırıcımızdan gelen performans puanlarıda E değişkenine kaydedilir. Bütün turlar bittiginde E’nin aritmetik ortalaması modelimizin performansını gösterir.kfold  **Grid Search Algoritması**  Öncelikle hiperparametre kavramından biraz bahsedelim. Veriden öğrenen makine öğrenmesi modelleri tasarlanırken, modelde kullanılan algoritma ya da teknikler ne olması gerektiğine tasarımcının karar vermesi gereken bazı parametreleri de beraberinde getirmektedir. Örneğin, KNN sınıflandırma algoritmasında k değerinin ne olacağına modeli tasarlayan kişi karar vermektedir. Benzer şekilde SVM algoritmasında hangi kernel fonksiyonunun kullanılacağına da tasarımcı karar verir. Aynı şekilde derin ağ modellerinde seyreltme (dropout) değerine, katman sayısına, nöron sayısına modeli tasarlayan kişi karar vermektedir. Genelde bu parametreler için hangi tercihte bulunulacağı başlangıçta açık ve kesin değildir; probleme, verisetine vb. gibi etmenlere göre değişiklik göstermektedir. Bu nedenle parametrelerin seçimi modeli tasarlayan kişiye bırakılmıştır. Ne olması gerektiği, modeli tasarlayan kişiye bırakılmış, probleme, veri setine göre değişiklik gösteren parametreler hiper-parametre (hyperparameters) olarak adlandırılmaktadır. Model tasarımı yapılırken hiper parametrelere yönelik yaptığımız ilk seçimler genelde bizi doğru sonuçlara götürmez. İteratif bir şekilde ardı ardına hiper parametreler değiştirilerek, modelin başaırımı gözlenir ve model için en uygun hiper parametre grubu seçilmeye çalışılır. Bunu yanı sıra bu seçim işini otomatize eden yöntemler de mevcuttur. Yazının devamında bu yöntemlerden birisi olan grid search algoritması üzerinde durulacaktır.  Hiper parametrelerden bazıları sonsuz sayıda değer alabilecek konumdadırlar. Bununla birlikte biz problem hakkında sahip olduğumuz ön bilgileri kullanarak hiper parametrelerin alabilecekleri değerler için aralıklar belirleyebiliriz. Belirlediğimiz bu aralıklardan belirli ana noktalar seçilerek hiper parametreler için değer listeleri oluşturulur. Grid search ile hiper parametre seçim işleminde; belirlenen aralıkta bulunan tüm değerlerin kombinasyonları için ağ eğitilip sonuçlar gözlenir duruma göre en iyi kombinasyon hiper parametre grubu olarak seçilir.  **Grid Search bazı özellikleri;**  - Sadece iyi başarım veren sonuca odaklanılmaması gerekmektedir. Konfigürasyon değerlerine karar vermek için grid search’teki trendler gözlenmelidir. Bu nedenle grid search sonucundaki tüm sonuçlar gözden geçirilmelidir. Yani parametreler arasındaki ilişkiler, eğilimler gözlenerek sabit parametreler yada aralıklar gözlenmelidir.  - Grid search’de farklı parametre kombinasyonlarının denenebilmesi için paralel işlemler koşturulabilir. Böylece zamandan tasarruf sağlanmış olur.  - Derin ağları eğitmek uzun zaman alan işlem olduğu için, hiper parametre seçimi için veri kümesinin bir alt kümesi üzerinde çalışılabilir böylece zamandan tasarruf edilmiş olur. Bu işlemde amaç hiper parametreler için en uygun değerleri bulmak değil, genel bir kanı oluşturmaktır. Böylece hiper parametrelerin hangi aralıklarına odaklanacağı belirlenir.  - Hiper parametreler için başlangıçta değerleri büyük tutmak, daha sonra uygun bulunan değer aralıklarına yoğunlaşmak zaman kazancı sağlayacaktır. Örneğin ağdaki nöron sayısı için başlangıçta [10, 100, 500, 1000] gibi bir aralık seçip daha sonra sonuçlara göre kapsamı genişleterek [100, 150, 200, 250, 300, 350 , 400] gibi derinlemesine ilerlemek zaman tasarrufu sağlayacaktır. | | | |

**Model Selection Video**

• <https://www.youtube.com/watch?v=hihuMBCuSlU>

**Python ile Model Selection Uygulaması**

Aşağıdaki linkten ulaşacağınız klasörde uygulama ile ilgili tüm detayları bulabilirsiniz.

<https://drive.google.com/drive/u/0/folders/1M2cllRBJi0whMidPMZpRU-5i8I-TAPtN>

**Not:** Bu uygulama klasöründe kodlanması gereken iki konu vardır, Model Selection anlatımında yer aldığı gibi önce K-Fold Cross Validation dosyası sonra Grid Search dosyası kodlanmalıdır.

**5. Adım Değerlendir:**

Sohbet edilerek günün dönüşümlü değerlendirmesi yapılır.