İstanbul
GEDİK
Üniversitesi

İstanbul GEDİK Üniversitesi

Veri Madenciliği

Öğr. Gör. Zeki ÇIPLAK

Overfitting,
Balanced ve Underfitting

Overfitting (Aşırı Uyum)

- Eğitim verisinin, belli bir modele karşı aşırı uyum göstermesi durumudur.

- Overfitting durumunda, eğitim verisiyle kurulmuş model, veriyi ezberlemiş olur ve **sadece eğitim verisinde bulunan değerler için iyi sonuçlar verir**.

- Eğitim verisinin dışındaki verilerde, modelimiz iyi tahminler yapamaz. Bu istenmeyen bir durumdur.

- Overfitting durumunda, eğitim setinin MAE ve MSE gibi hata değerleri düşüktür.

- Overfitting modellerde, **düşük bias** (yanlılık) ve **yüksek varyans** durumu gözlenir.

Underfitting (Eksik Uyum)

- Eğitim verisinin, belli bir modele karşı eksik uyum göstermesi durumudur. Underfitting'te, modele sunulan eğitim verisi yetersiz gelmiş ve **eksik öğrenme** yaşanmış demektir.

- Underfitting, genellikle **bağımsız değişken sayısının yetersiz olması** durumunda ortaya çıkar. Overfitting kadar çok karşılaşılmaz ama bu da istenmeyen bir durumdur.

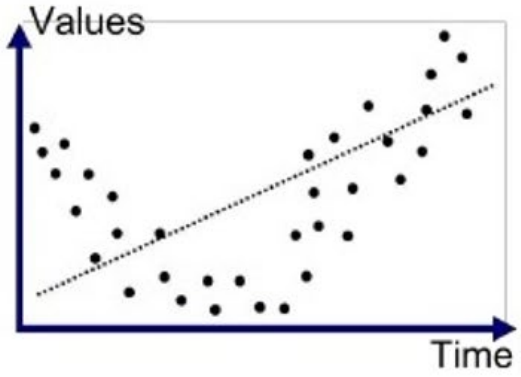
- Underfitting durumunda, eğitim ve test verisinin MAE ve MSE gibi hata değerleri yüksek çıkar.

- Overfitting modellerde, **yüksek bias** (yanlılık) ve **düşük varyans** durumu gözlenir.

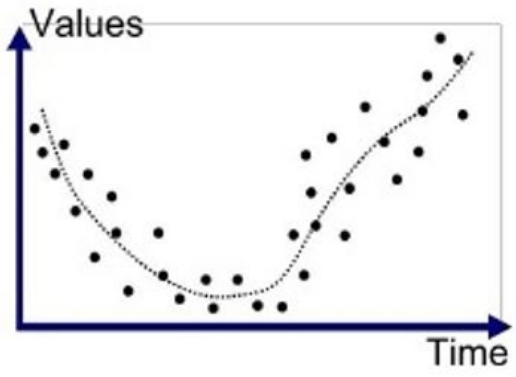
Fit / Balanced (Tam Uyum)

- Eğitim ve test verisinin, belli bir modele karşı tam uyum göstermesi durumudur. Fit olma durumu, veri madenciliği projesinde hedeflenen bir durumdur.

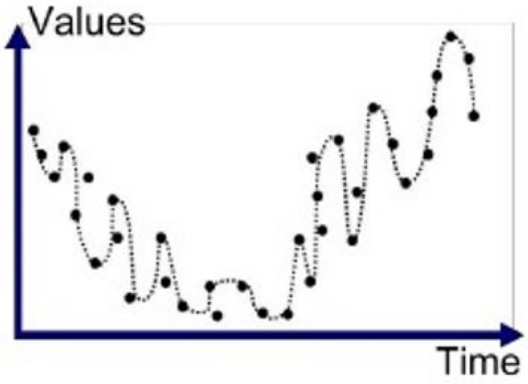
- Modelin balanced (fit) olma durumunda, hata değerleri makul ve kabul edilebilir bir seviyede çıkar.



Underfitting



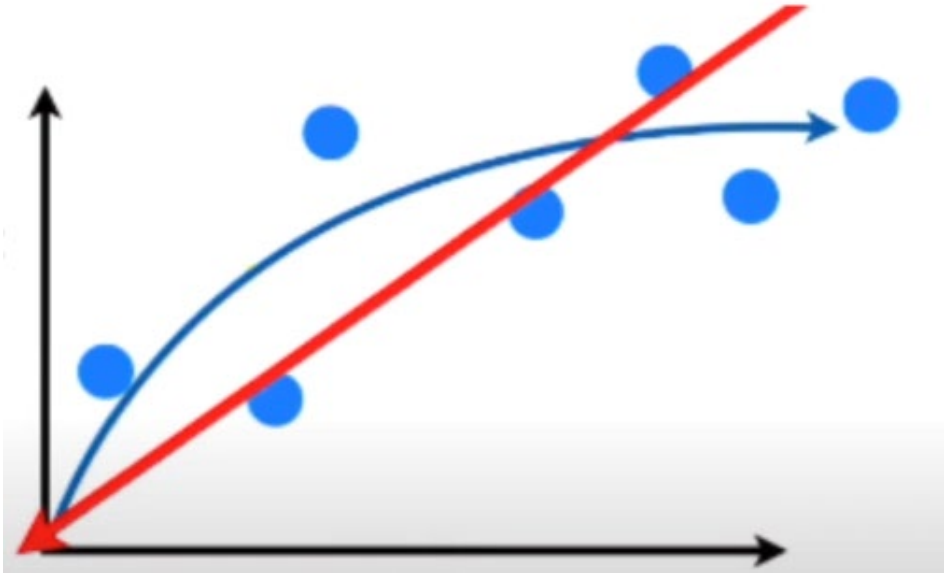
Fitting (Balanced)



Overfitting

Yanlılık (Bias) ve Varyans

Yanlılık (Bias)

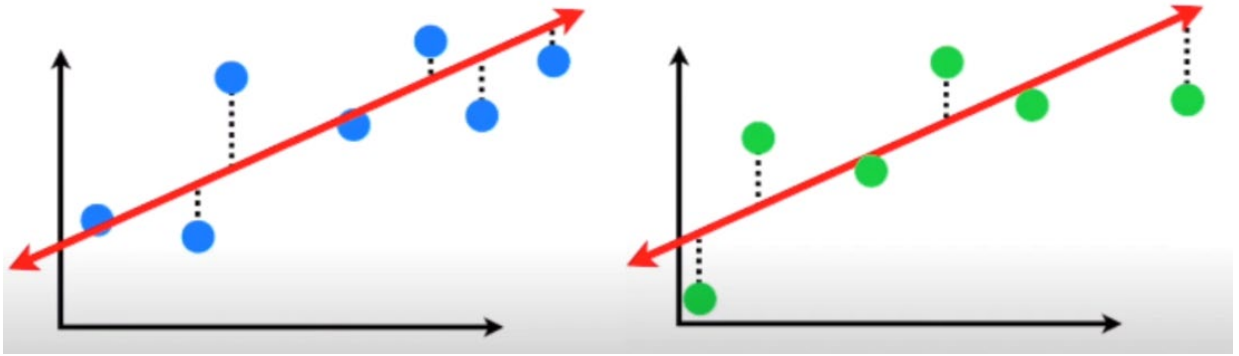


Bir modelin, bağımsız değişken ve bağımlı değişken arasındaki doğrusal veya doğrusal olmayan ilişkiyi, belli bir yönde baskın olarak çıkarmasına Bias (Yanlılık) denir.

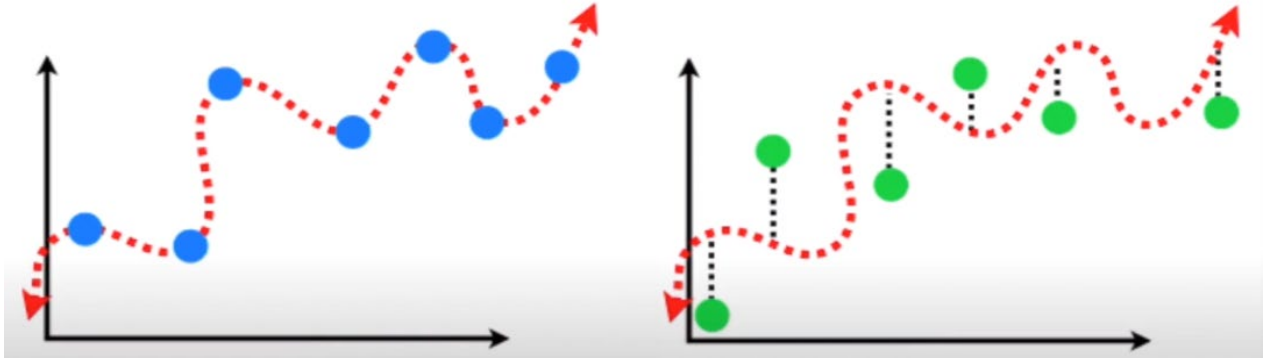
- Eğitim verisinin doğrusal olmaması durumunda, o veriye doğrusal bir model önerilirse, Underfitting durumu ve **yüksek yanlılık** durumu ortaya çıkacaktır.
- Model; eğitim verisindeki değişkenlerin ilişkisinin, kendi belirlediği yönde olmasını isteyecek, bu da yüksek yanlılığa ve **hata hesabının yüksek çıkmasına** sebep olacaktır.

Varyans

- Eğitim ve test verisinin hataları arasındaki fark, bir değişkenlik ölçüsüdür. Bu da **varyans** olarak ifade edilir.
- **Underfitting** bir modelde; yanlılık **yüksek**, varyans **düşük** olur.



- **Overfitting** modelde ise, yanlılık **düşük** ve varyans **yüksek** olur.



Varyans

- Bazı durumlarda, (veri setindeki değerlerin durumuna göre) **yanlılık ile varyans aynı anda düşük** ve **yanlılık ile varyans aynı anda yüksek** de olabilmektedir.

Sınıflandırma (Classification)

Sınıflandırma

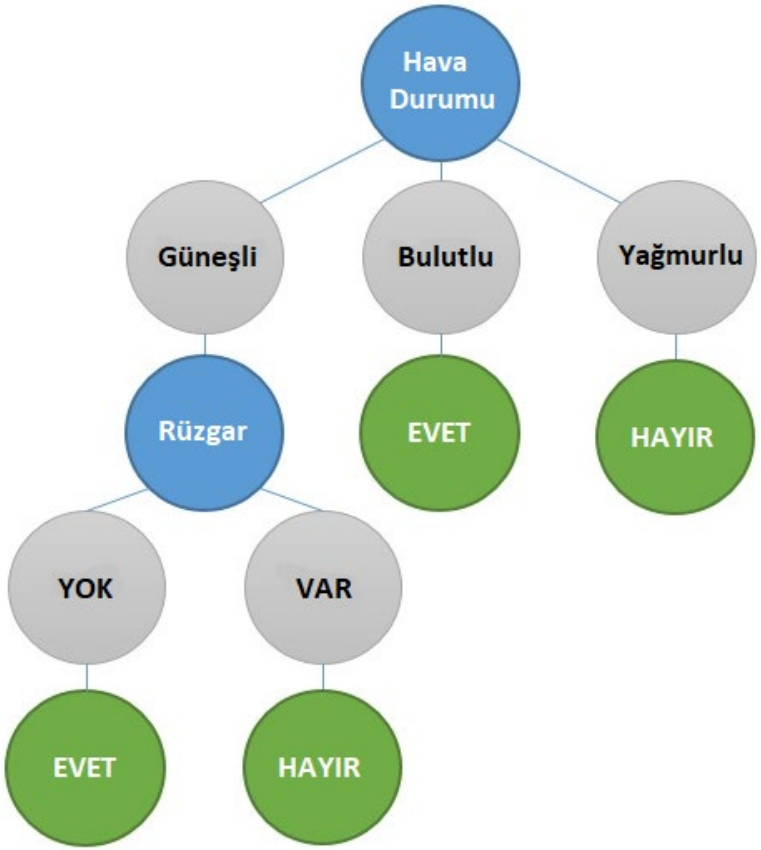
- Literatürde, binlerce sınıflandırma algoritması mevcuttur. Bazı algoritmalar, çok az değişiklikle yeni bir algoritma olarak literatüre girebilmektedir.

- En çok kullanılan sınıflandırma algoritmaları aşağıdaki gibidir:

  - ***Decision Trees (Karar Ağaçları)***
  - ***Random Forest***
  - ***Support Vector Machine (SVM)***
  - ***Naive Bayes***
  - ***K-Nearest Neighbor (En yakın komşu)***
  - ***Lojistik Regresyon***

Karar Ağaçları
(Decision Trees)

Karar Ağaçları



Veri setindeki değişkenlere dayalı bir **karar verme sürec**ini oluşturmaya yarar.

Karar Ağaçları düğümlerden oluşur. Her bir düğüm bir soru içerir. Soruya verilen cevaba göre, diğer bir düğüme gidilir.

Ağacın en tepesindeki düğüm kök (root) düğümdür. Karar varılan düğüm, yaprak (leaf) düğümdür.

Karar Ağaçları

- Sadece sınıflandırma probleminde değil, aynı zamanda veri setindeki **değişkenler arasındaki ilişkiler**in ortaya çıkarılmasında da kullanılır.

- Karar Ağacında **yaprak düğüme ulaşıldığında** bir çıktı üretilmiş olur, yani karar verilir.

- **Soruların doğru bir sırada sorulması**, karar ağacının verimli olarak çalışmasını sağlar.

- Bir veri setinden, farklı **ayırma kriterleri**ne göre, farklı karar ağaçları oluşturulabilir.

Karar Ağaçları

- Karar Ağaçlarında Entropi, Twoing ve Gini indeksi vb. gibi ayırma kriterleri uygulanır.

- Ayırma kriteri, diğer nitelikler ile karşılaştırıldığında **en iyi ayırıcı nitelik** (değişken) olmalıdır.

- Ayırma işlemi, **karar ağacının aşırı büyümesi** veya **ağaç derinliğinin belirlenen bir limite ulaşması** gibi, bir **durma kriteri**ne ulaşılana kadar devam eder.

- Oluşturulan karar ağacı çok uzunsa, **budama** denilen işlem yapılır ve ağacın yüksekliği azaltılmış olur. Burada «**minsplit**» parametresi de büyük rol oynar.

Sınıflandırma Algoritmalarında Model Değerlendirme

Model Değerlendirme

- Sınıflandırma modellerinin başarı/hata oranları, regresyon modellerinde kullandığımız (MAE, MSE vb. gibi) metriklerle hesaplanmaz.

- Sınıflandırma modellerinin değerlendirmesi, genellikle **Karmaşıklık/Hata Matrisi** (**Confusion Matrix**) ile yapılır. Bu matristen elde edilen Accuracy, Precision, Recall ve F1 Score gibi metriklerle modelin başarısı ölçülmüş olur.

- Bu metriklerin hepsi birlikte, **1.0**'a ne kadar yakınlaşmışsa, model o kadar başarılı sayılabilir.

- "***Sadece bu metrikler üzerinden bir model başarısı ölçülebilir.***" diyemeyiz ama bu metrikler, modelin başarısını anlamamıza büyük katkı sağlarlar.

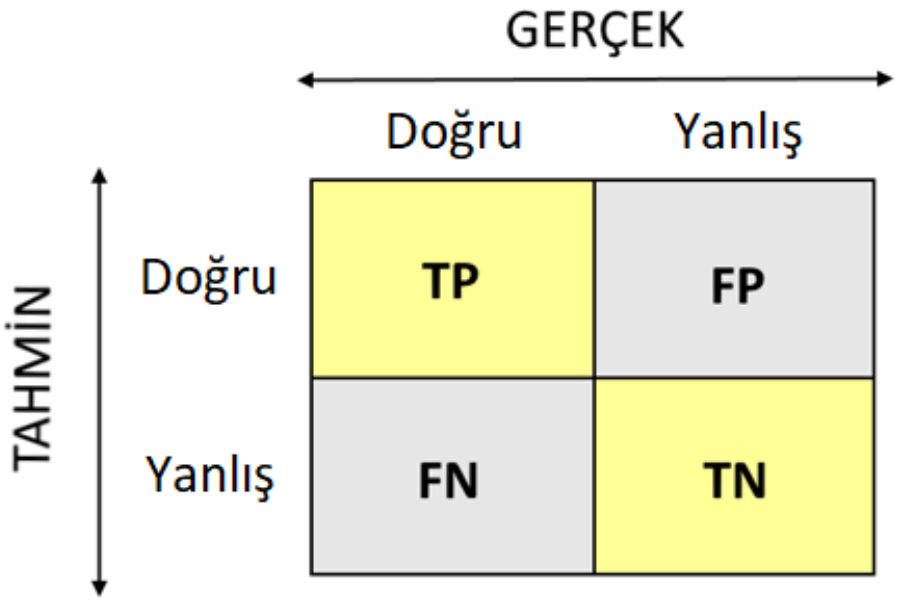
Karmaşıklık/Hata Matrisi

- Tüm hata matrislerinde dört adet değerlendirme terimi bulunur. Az önce bahsettiğimiz dört metrik de, bu dört terim üzerinden hesaplanır.

- Hata matrisindeki dört terim şöyledir:
  - Doğruya doğru demek (True Positive – **TP**) DOĞRU
  - Yanlışa yanlış demek (True Negative – **TN**) DOĞRU
  - Doğruya yanlış demek (False Positive – **FP**) YANLIŞ
  - Yanlışa doğru demek (False Negative – **FN**) YANLIŞ

- Sadece doğrular ve yanlışlar üzerinden yapılacak olan Accuracy (Doğruluk) hesabı, modelin gerçek başarısı hakkında bilgi vermek yönünden yetersiz kalır. (Spam Mail Sınıflandırma Örneği)

Karmaşıklık/Hata Matrisi



Accuracy (Doğruluk):

Doğru sınıflandırmanın toplama bölünmesi ile bulunur.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Karmaşıklık/Hata Matrisi

Precision:

Doğru olarak tahmin edilenlerin,
ne kadarının gerçekten doğru olduğunu gösterir.

Precision = TP / (TP + FP)

Recall:

Gerçekte doğru olanların,
ne kadarının doğru olarak tahmin edildiğini gösterir.

Recall = TP / (TP + FN)

F1 Score:

Precision ve Recall değerlerinin harmonik ortalamasını verir.

F1 Score = 2 * [(Precision * Recall) / (Precision + Recall)]