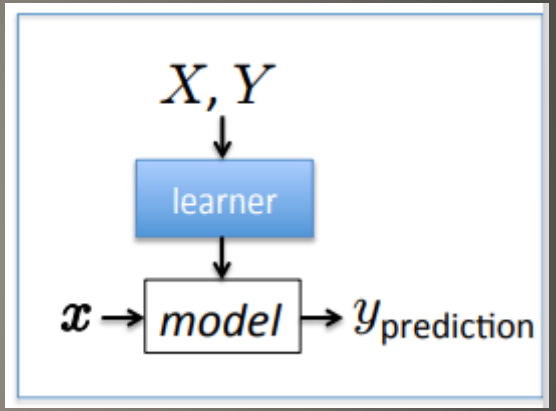
Makine Öğrenmesi

Model Değerlendirme

Makine Öğrenmesinin Aşamaları

- Verilen: Etiketli eğitim verisi X,Y=$\{\langle x_i, y_i \rangle\}$
  - Her xi D(X) in bir örneği $y_i = f_{hedef}(x_i)$
- Modeli eğit
  - Model←classifier.train(X,Y)
- Yeni veriye modeli uygula
  - Verilen: Yeni etiketsiz örnek:x D(x)'in bir örneği
  - $y_{tahmin} = model.predict(x)$



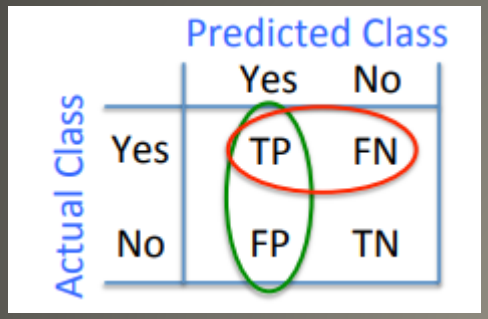
Sınıflandırma Metrikleri

- $doğruluk = \frac{doğru\ tahminlerin\ sayısı}{test\ örneklerinin\ sayısı}$

- $hata = 1 - doğruluk = \frac{Yanlış\ tahminlerin\ sayısı}{test\ örneklerinin\ sayısı}$

Karmaşıklık Matrisi

- Verilen veri kümesinde P pozitif ve N negatif örnek olsun.



$$dogruluk = \frac{TP + TN}{P + N}$$

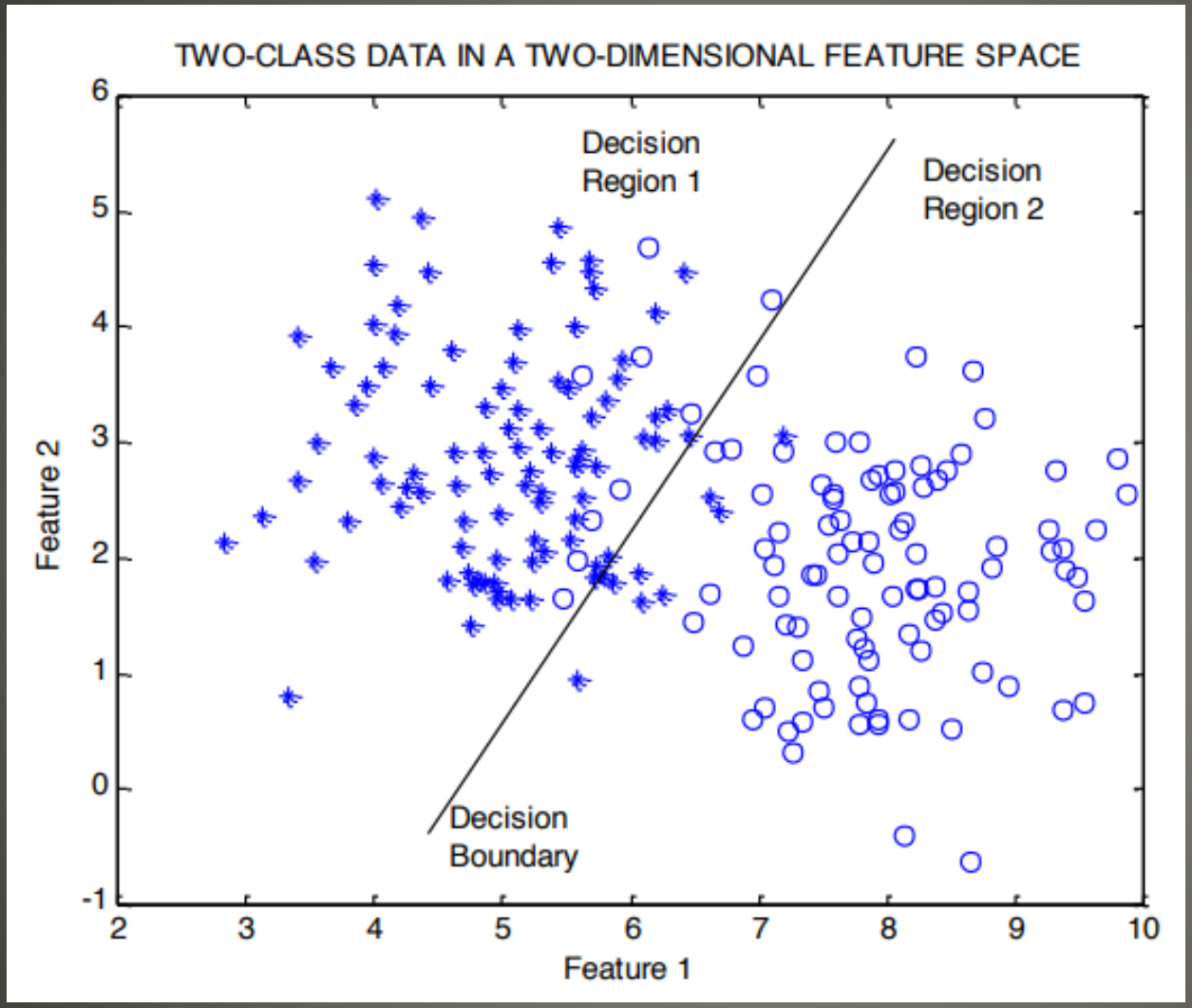
- Pozitif durumları tanımlamak için bir sınıflandırıcı kullandığımızı düşünelim(Örn. Bilgi edinimi):
- $Precision = \frac{TP}{TP+FP}$ $recall = \frac{TP}{TP+FN}$

Rastgele seçilen bir sonucun ilgili olma olasılığı

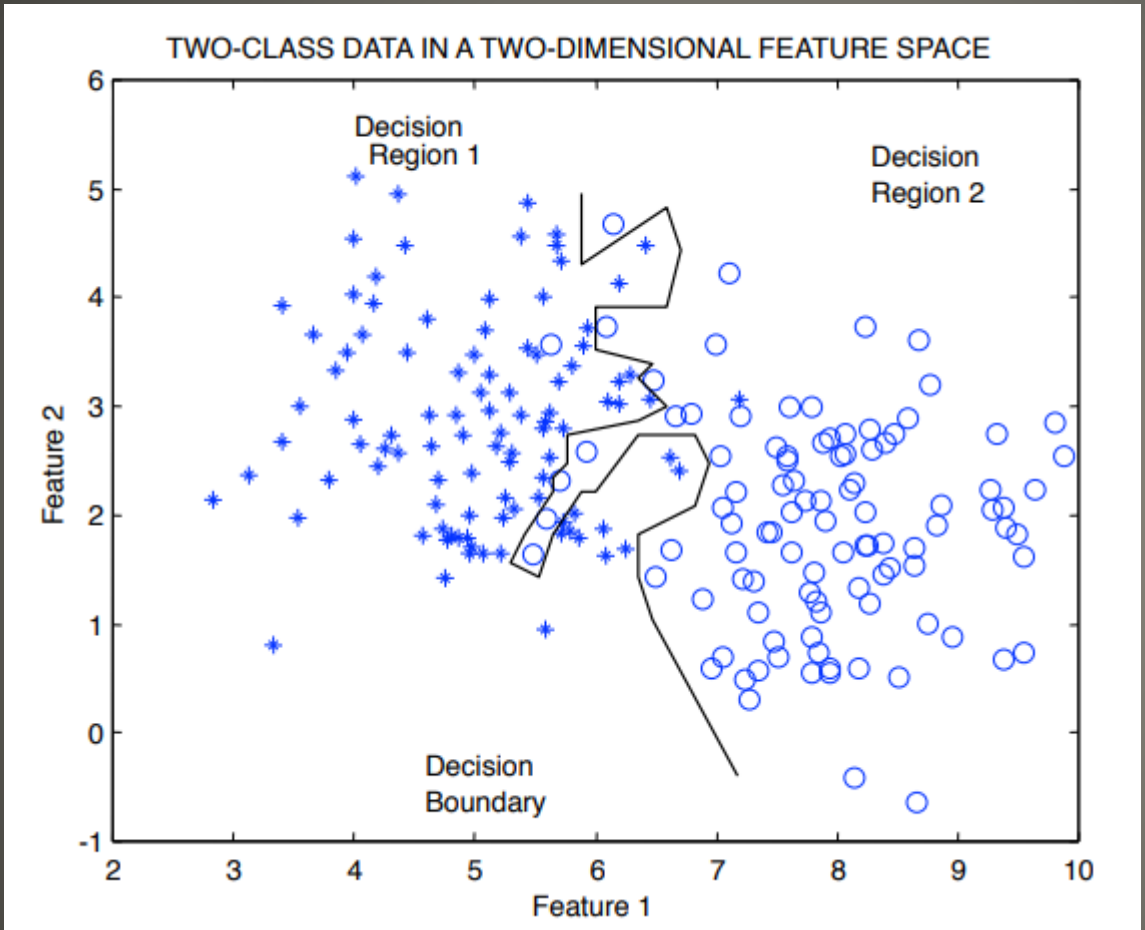
Rastgele seçilen ilgili bir belgenin alınma olasılığı

Eğitim ve Test Veri Seti

- Eğitim verileri: modeli oluşturmak için kullanılan veriler
- Test verileri: eğitim sürecinde kullanılmayan yeni veriler
- Eğitim performansı, genelleme performansının zayıf bir göstergesidir
- – Genelleştirme, ML'de gerçekten önemsediğimiz şeydir
- - Eğitim verilerini takmak kolay
- – Test verileri üzerindeki performans, genelleştirme performansının iyi bir göstergesidir
- – yani, test doğruluğu eğitim doğruluğundan daha önemlidir



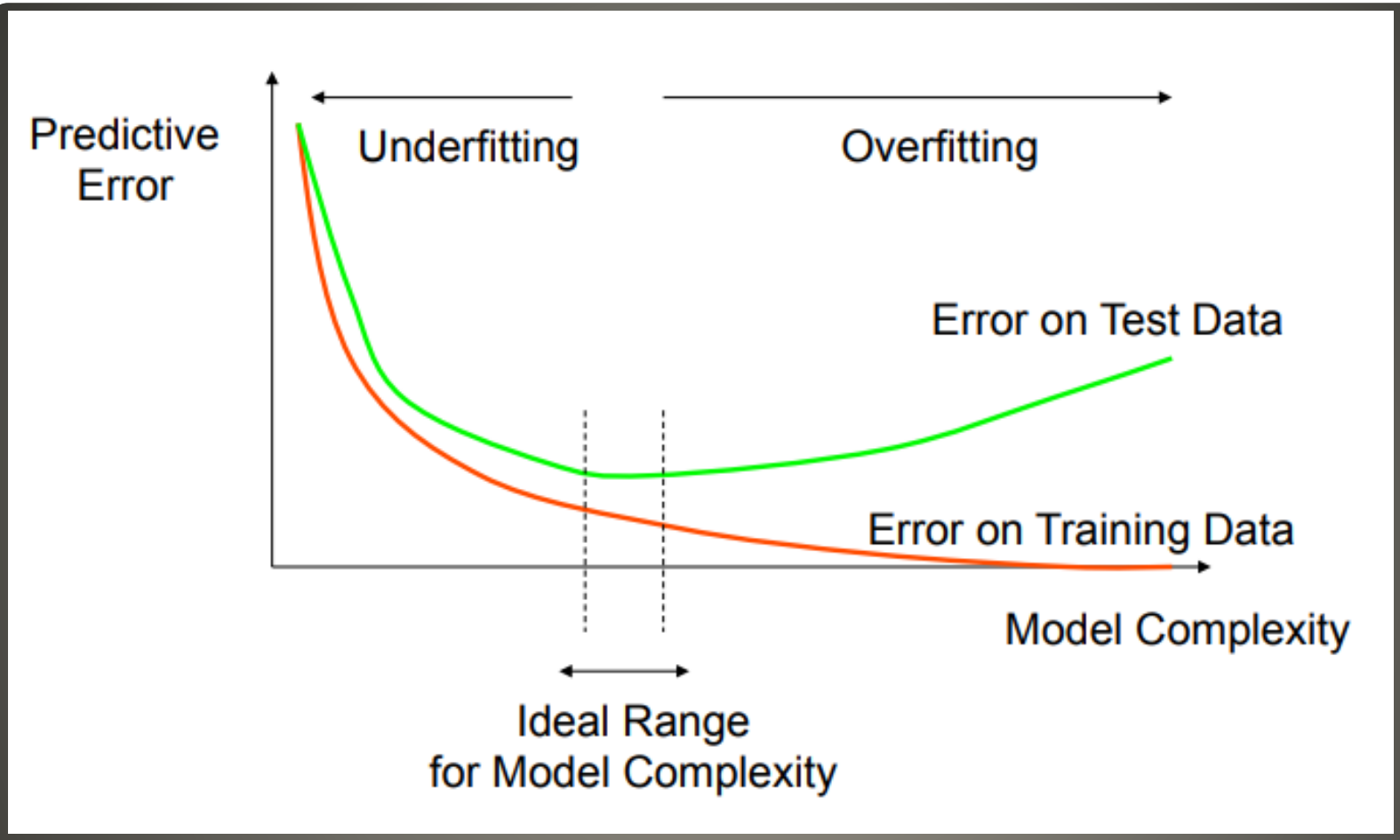
Basit Karar Sınırı



Daha Karmaşık Karar Sınırı

Aşırı Uyum Gösterme

- **Overfitting**
- Eğer modelimiz, eğitim için kullandığımız veri setimiz üzerinde gereğinden fazla çalışıp ezber yapmaya başlamışsa ya da eğitim setimiz tek düze ise **overfitting** olma riski büyük demektir.
- Eğitim setinde yüksek bir skor aldığımız bu modele, test verimizi gösterdiğimizde muhtemelen çok düşük bir skor elde edeceğiz. Çünkü model eğitim setindeki durumları ezberlemiştir ve test veri setinde bu durumları aramaktadır.
- **Underfitting**
- Aşırı öğrenmenin aksine, bir model yetersiz öğrenmeye sahipse, modelin eğitim verilerine uymadığı ve bu nedenle verilerdeki trendleri kaçırdığı anlamına gelir.
- Ayrıca modelin yeni veriler için genelleştirilemediği anlamına da gelir. Tahmin ettiğiniz gibi bu problem genellikle çok basit bir modelin sonucudur (yetersiz tahminleyici bağımsız değişken eksikliği).



Overfittingin Tahmin Üzerindeki Etkisi

Overfitting Çözüm

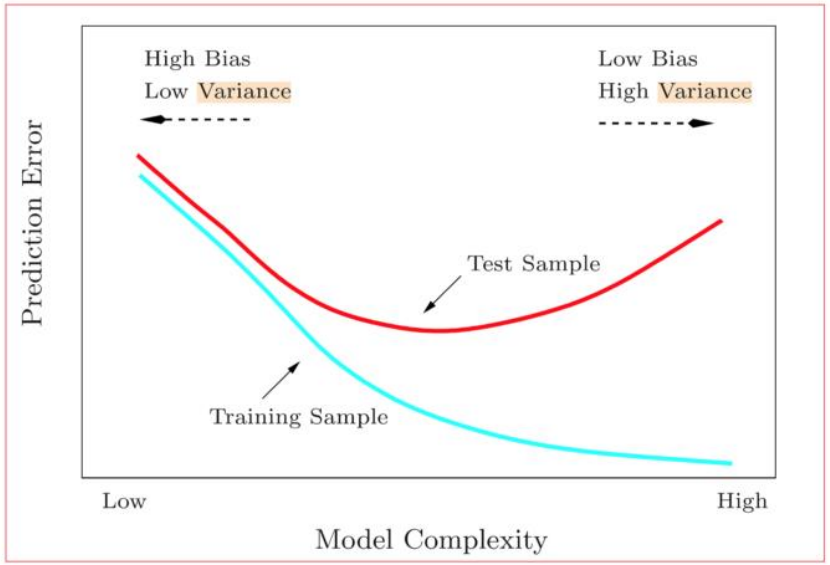
- **Öz nitelik sayısını azaltmak:** Birbirleriyle yüksek korelasyonlu olan kolonlar silinebilir ya da faktör analizi gibi yöntemlerle bu değişkenlerden tek bir değişken oluşturulabilir.
- **Daha fazla veri eklemek :** Eğer eğitim seti tek düze ise daha fazla veri ekleyerek veri çeşitliliği arttırılır.
- **Regularization (Düzenleme) :** Düzenleme, modelin karmaşıklığını azaltmak için bir kullanılan tekniktir. Bunu kayıp fonksiyonunu cezalandırarak yapar. Yani modelde ağırlığı yüksek olan değişkenlerin ağırlığını azaltarak bu değişkenlerin etki oranını azaltır.

Bias ve Varyans Problemi

- ***Varyans***, model eğitim veri setinde iyi performans gösterdiğinde, ancak bir test veri kümesi veya doğrulama veri kümesi gibi, eğitilmemiş bir veri kümesinde iyi performans göstermediğinde ortaya çıkar. Varyans, gerçek değerden tahmin edilen değerin ne kadar dağınık olduğunu söyler.
- **Bias**, gerçek değerlerden tahmin edilen değerlerin ne kadar uzak olduğudur. Tahmin edilen değerler gerçek değerlerden uzaksa, bias yüksektir.

Bias ve Varyans Problemi

- Aşağıdaki grafikten görüldüğü gibi model karmaşıklığı arttıkça eğitim seti üzerinde hatalı tahmin oranı azaltmakta ancak test veri seti üzerinde tahmin hatası artmaktadır.



Source: Elements of Statistical Learning by Trevor Hastie, Robert Tibshirani and Jerome Friedman

Bias ve Varyans Problemi

- Yüksek Bias Düşük Varyans: Modeller tutarlıdır, ancak ortalama hata oranı yüksektir.
- Yüksek Bias Yüksek Varyans : Modeller hem hatalı hem de tutarsızdır .
- Düşük Bias Düşük Varyans: Modeller ortalama olarak doğru ve tutarlıdır. Modellerimizde bu sonucu elde etmek için çabalamaktayız.
- Düşük Bias Yüksek Varyans: Modeller bir dereceye kadar doğrudur ancak ortalamada tutarsızdır. Veri setinde ufak bir değişiklik yapıldığında büyük hata oranına neden olmaktadır.

Bias ve Varyans Problemi

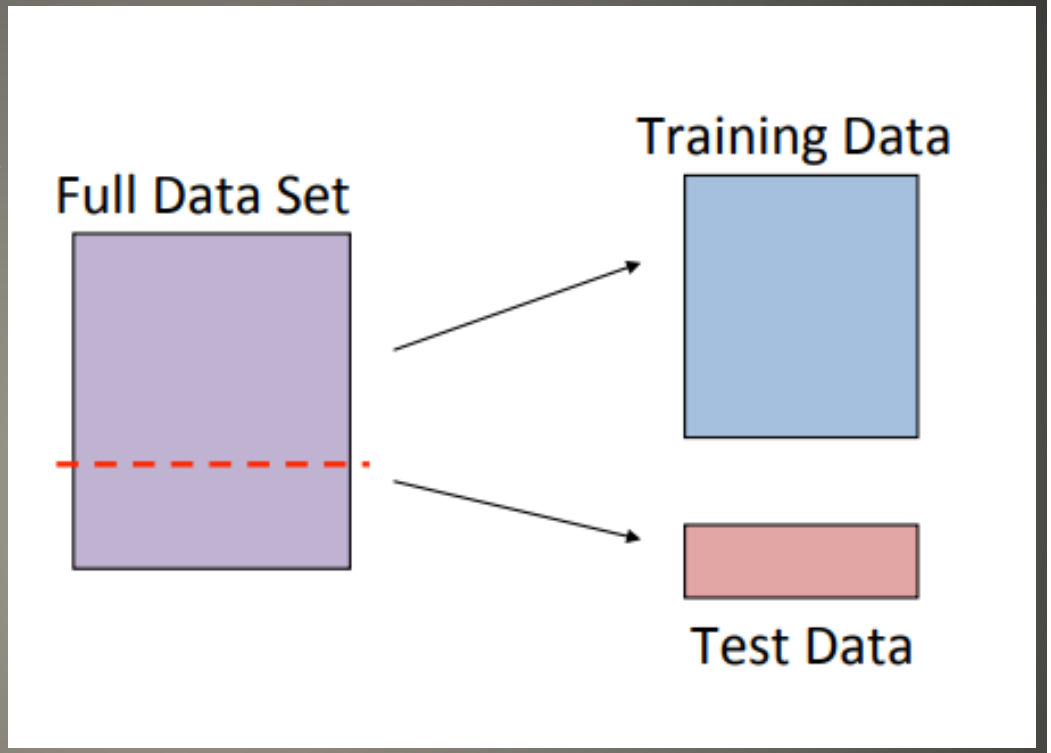
- Yüksek bias problemini çözmek için aşağıdaki yöntemleri uygulayabiliriz.
- **Daha fazla veri eklemek :** Daha fazla veri ekleyerek veri çeşitliliğini arttırmak gereklidir.
- **Daha fazla değişken eklemek :** Model karmaşıklığının artmasını sağlamaktadır.
- **Regularization (düzenleme) :** Değişkenlerin ağırlığını arttırmak için regularization değerini azaltın

Sınıflandırıcıları Karşılaştırma

- C1 ve C2 olmak üzere iki sınıflandırıcımız olduğunu ve gelecekteki tahminler için kullanmak üzere en iyisini seçmek istediğimizi varsayalım.
- Aralarında seçim yapmak için eğitim doğruluğunu kullanabilir miyiz?
- Hayır
- – örneğin, C1 = budanmış karar ağacı, C2 = 1-NN
- Eğitim doğruluğu(1-NN) = %100, ancak en iyi olmayabilir
- Bunun yerine, test doğruluğuna göre seçim yapın...

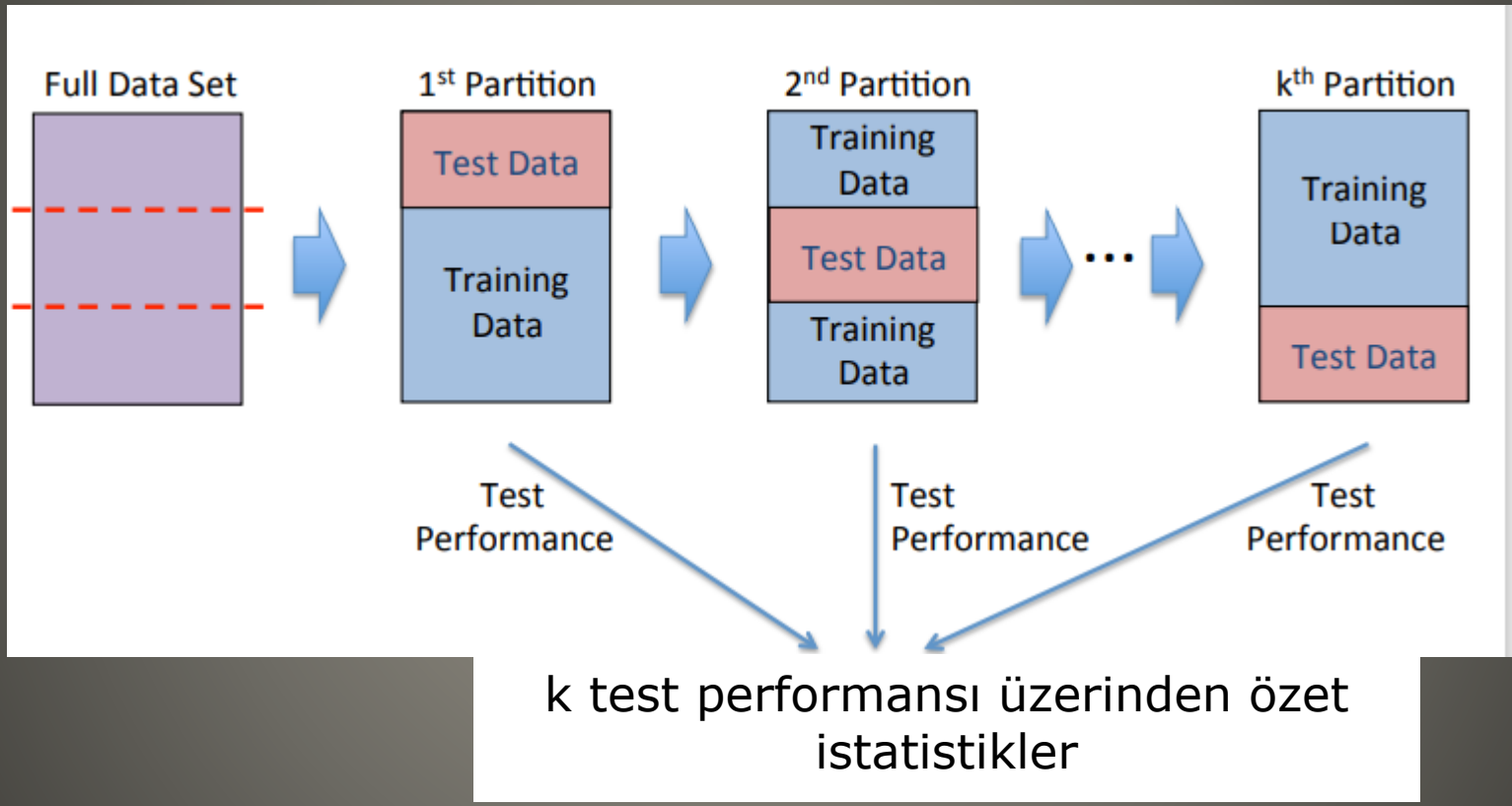
Eğitim ve Test Verisi

- Fikir:
- Her modeli "eğitim verileri" üzerinde eğitin...
- ...ve ardından her modelin doğruluğunu test verileri üzerinde test edin



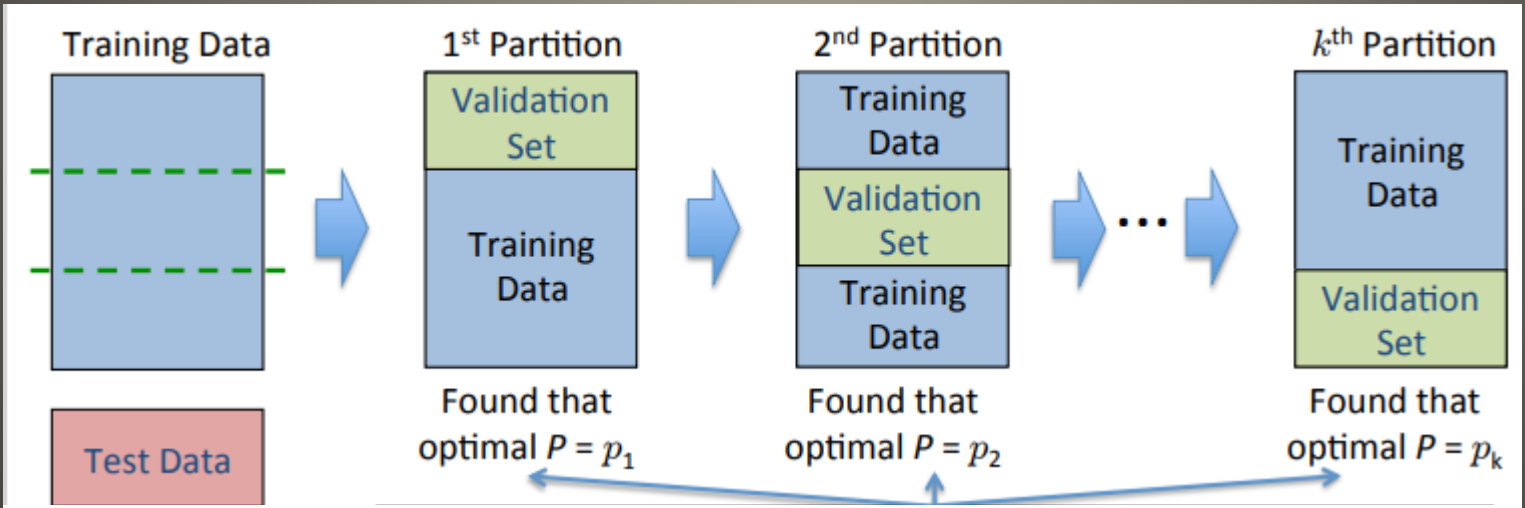
K-noktalı çapraz doğrulama

- Neden verilerin belirli bir "bölünmesini" seçelim?
- Prensip olarak, performans her bölüm için farklı olabileceğinden bunu birden çok kez yapmalıyız.
- k-Katlama Çapraz Doğrulama (ör. k=10)
- – n örneğinin tam veri kümesini rastgele bölümlere ayırın
- k ayrık alt küme (her biri kabaca n/k boyutunda)
- – Test seti olarak sırayla her katlamayı seçin; modeli diğer parçalar ile eğitin ve değerlendirin
- – k test performansı üzerinden istatistikleri hesaplayın veya k modelin en iyisini seçin
- – Ayrıca k = n olduğunda “biri devre dışı bırakılmış CV” de yapabilir



K-noktalı çapraz doğrulama

- P model parametresinin değerini seçmek için ÇD'yi de kullanabilir
- Parametre değerlerinin alanı üzerinde arama yapın
- – Doğrulama setinde P = p ile modeli değerlendirin
- En yüksek doğrulama performansına sahip p' değerini seçin
- Modeli tam eğitim setinde P = p' ile öğrenin



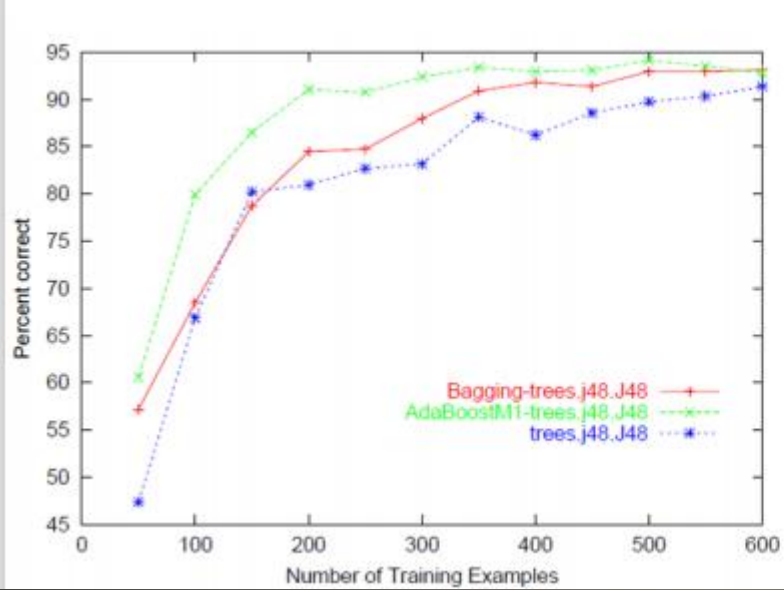
En iyi doğrulama performansına sahip modelin p değerini seçin

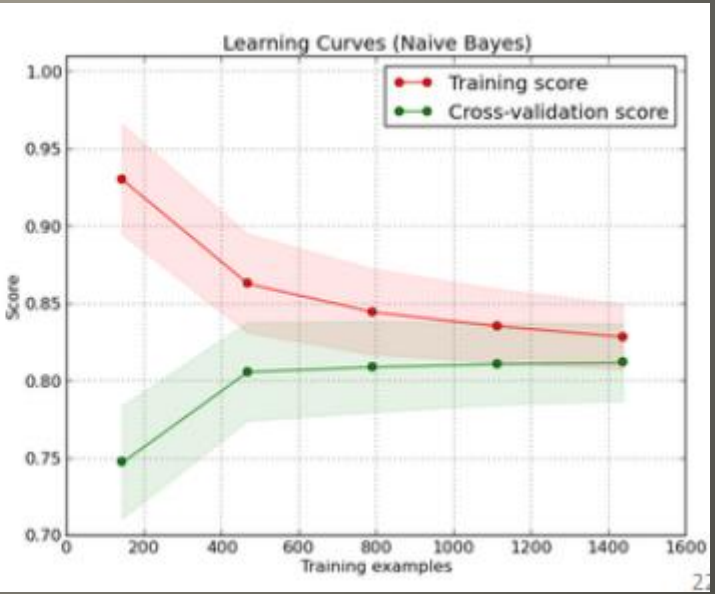
Model parametrelerini optimize etme

Çapraz doğrulamanın önemi

- Çapraz doğrulama, sınıflandırıcının "görünmeyen" veriler üzerinde ne kadar iyi performans göstereceğine dair yaklaşık bir tahmin oluşturur
- – k->n olarak model daha doğru eğitim verisi olur)
- – ...ama ÇD hesaplama açısından daha pahalı hale gelir
- – k < n seçimi yapılmalıdır
- • Farklı bölümler üzerinden ortalama almak, verilerin yalnızca tek bir eğitim/doğrulama bölümünden daha sağlamdır
- • Tekrar tekrar ÇD yapmak daha da iyi bir fikir!

- Eğitim örneğine karşı performansı gösterir
- – Tek bir eğitim/test bölümü üzerinden hesaplayın
- – Ardından, birden fazla CV denemesinin ortalaması





Eğitim Eğrisi

- Bir sınıflandırmanın doğruluğunu değerlendirmek için karışıklık matrisini hesaplayın.
- Tanım olarak bir karışıklık matrisi C şekildedir Ci,j grupta olduğu bilinen gözlemlerin sayısına eşittir i ve grupta olacağı tahmin ediliyor j.
- Böylece ikili sınıflandırmada, gerçek negatiflerin sayısı C0,0, yanlış negatifler C1,0, gerçek pozitifler C1,1 ve yanlış pozitifler C0,1.

```
from sklearn.metrics import confusion_matrix
y_true = [2, 0, 2, 2, 0, 1]
y_pred = [0, 0, 2, 2, 0, 2]
confusion_matrix(y_true, y_pred)
```

```
array([[2, 0, 0],
       [0, 0, 1],
       [1, 0, 2]])
```

```
from sklearn.metrics import confusion_matrix
y_true = ["cat", "ant", "cat", "cat", "ant", "bird"]
y_pred = ["ant", "ant", "cat", "cat", "ant", "cat"]
confusion_matrix(y_true, y_pred, labels=["ant", "bird"
, "cat"])
```

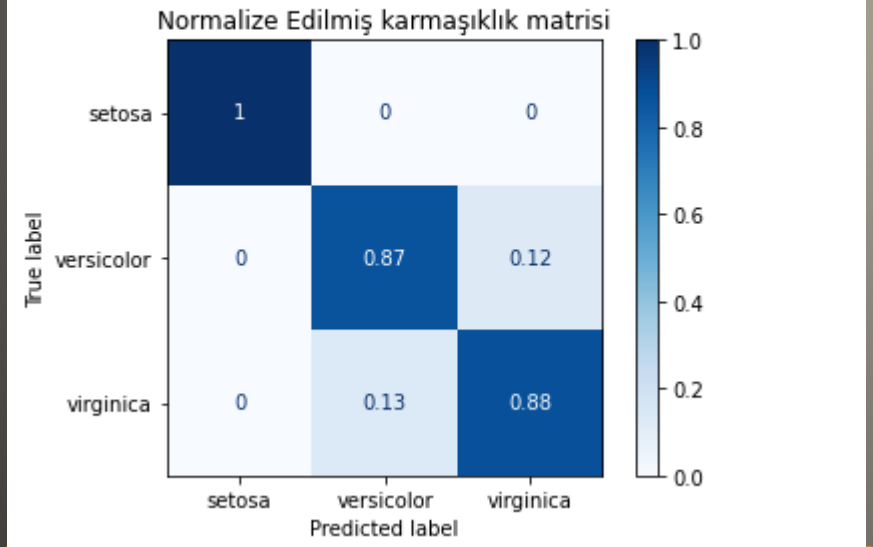
Scikit-Learn ile Performans Metrikleri- sklearn.metrics. confusion_matrix

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm, datasets
from sklearn.model_selection import train_test_split
from sklearn.metrics import plot_confusion_matrix
# Çalışmak için veri seti yüklenmesi
iris = datasets.load_iris()
X = iris.data
y = iris.target
class_names = iris.target_names

# Eğitim ve test veri kümelerini ayır
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3)

# Sınıflandırıcıyı çalıştır
classifier = svm.SVC(kernel='linear', C=.1).fit(X_train, y_train)

np.set_printoptions(precision=2)
```

Scikit-Learn ile Performans Metrikleri- sklearn.metrics. confusion_matrix

```
disp = plot_confusion_matrix(classifier, X_test, y_test,
                                 display_labels=class_names,
                                 cmap=plt.cm.Blues,
                                 normalize='pred')
disp.ax_.set_title('Normalize Edilmiş karmaşıklık matrisi')
print(disp.confusion_matrix)
plt.show()
```

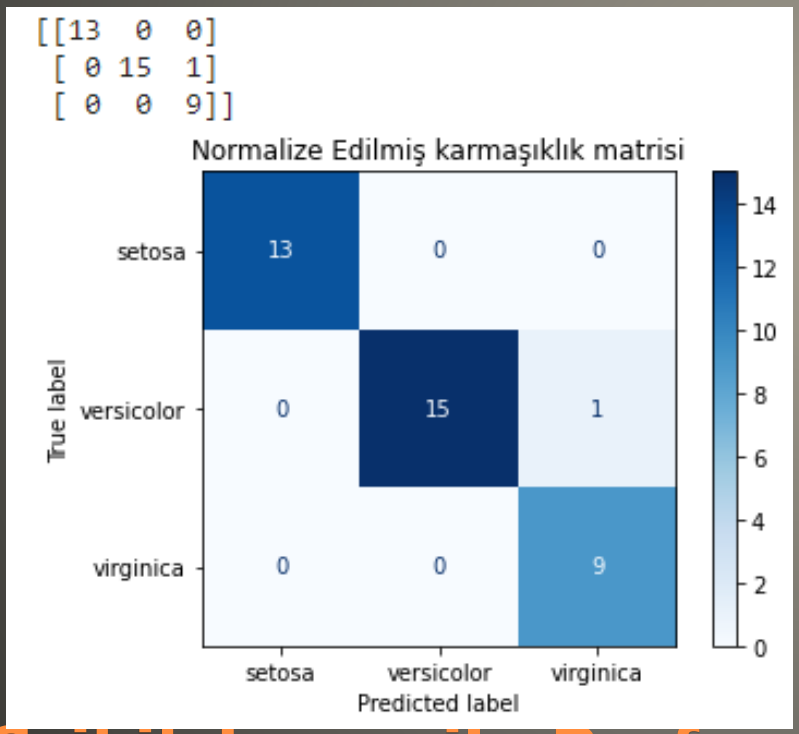
```
[[1.   0.   0.  ]
 [0.   0.87 0.12]
 [0.   0.13 0.88]]
```



Scikit-Learn ile Performans Metrikleri- sklearn.metrics. confusion_matrix

```
disp = plot_confusion_matrix(classifier, X_test, y_test,
                                 display_labels=class_names,
                                 cmap=plt.cm.Blues,
                                 normalize=None)
disp.ax_.set_title('Normalize Edilmiş karmaşıklık matrisi')
print(disp.confusion_matrix)
plt.show()
```

```
[[13  0  0]
 [ 0 15  1]
 [ 0  0  9]]
```



Scikit-Learn ile Performans Metrikleri- sklearn.metrics. confusion_matrix