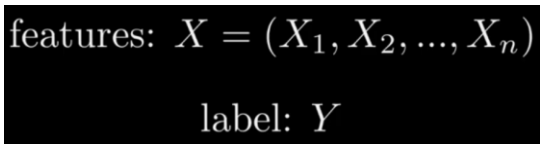
Youtube – Normalized Nerd

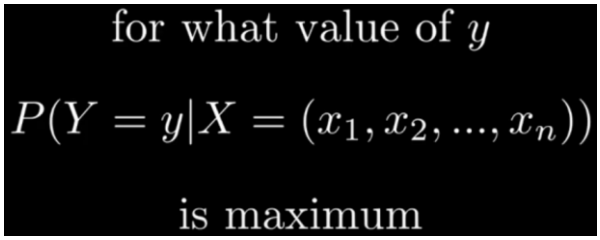
The Math Behind Bayesian Classifers

The Math Behind Bayesian Classifiers

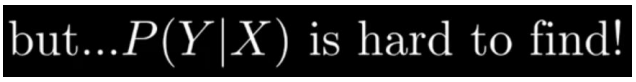
Bu videoda Bayesian sınıflandırıcılardan bahsedeceğim ve Naïve Bayes sınıflandırıcının nasıl çalıştığını açıklayacağım. Öncelikle sınıflandırma problemimizi tanımlayalım. X_1, X_2,…. Xn'den oluşan n sayıda özniteliğe sahibiz ve Y ile temsil edilen doğru etiketi bulmamız gerekiyor diyelim. Şimdi bu sorunu olasılıksal bir bakış açısıyla ele almak için bu Y ve X'i random değişkenler olarak ele almalıyız.



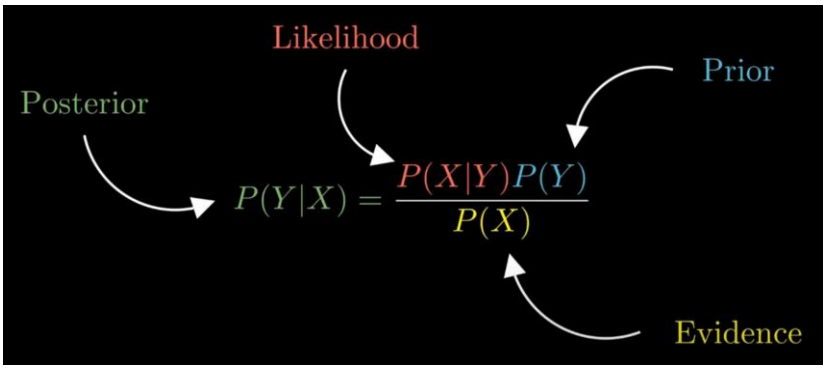
Büyük Y'nin küçük harf y değerini aldığını ve büyük X'in n'e kadar X_1, X_2 değerini aldığını varsayalım. Şimdi doğru etiketi bulmak için, Y'nin tüm olası değerleri için bu ifadeyi bulmamız gerekiyor. Bu ifade aslında koşullu bir olasılıktır, bu da basitçe, büyük X'in bu kümeye eşit olduğu göz önüne alındığında, büyük Y'nin olasılığının küçük y'ye eşit olduğu anlamına gelir. Daha kesin olmak gerekirse, bu koşullu olasılığın maksimum olduğu küçük y harfinin belirli değerini bulmamız gerekir. Neden? Çünkü bu durumda, y'nin bu özel değeri için 'tamam, ifade maksimum olur, yani sınıf etiketi bu olmalıdır' diyebileceğiz.



Ancak sorun şu ki, X verildiğinde Y'nin olasılığını doğrudan bulmak zor. Bu sorunu çözmek için Bayes teoremini kullanıyoruz. Evet, bayes'in devreye girdiği kısım burası.



Bayes teoremi; X verildiğinde Y'nin olasılığının *(Y verildiğinde X'in olasılığı) X (Y'nin olasılığı) / (X'in olasılığı)* ifadesini bulmakla aynı olduğunu söylüyor. Şimdi bu denklemin sağ tarafında görebileceğimiz her şey veri setimizden bulunabilir. Bulmak istediğimiz sol taraftaki şeye *posterior*, sağ taraftaki Y olasılığına ise *prior* denir. Neden bu tür bir isim? ***Prior***, herhangi bir kanıtı dikkate almadan önce bir olaya karşılık gelen olasılık anlamına gelir ve ***posterior***, bir kanıtın dikkate alınmasından sonra o olayın olasılığı anlamına gelir. Kanıt, özellikler kümesinden başka bir şey değildir. Yani burada X'in olasılığı kanıt (***evidence***) olarak adlandırılır. Y verildiğinde X'in olasılığı P(X|Y) ise ***likelihood*** olarak adlandırılır. Şimdi burada ilginç bir şey var. Paydanın değeri, paya koyduğumuz Y değerinden bağımsız olarak aynı kalır. Bu, farklı sınıf etiketleri için bu koşullu olasılığın değerini karşılaştırabileceğimiz anlamına gelir. Paydayı görmezden gelebiliriz, çünkü kanıtlar aynı kalır.



Konsepti gerçekten anlamak için bir örnek üzerinden gitmemiz gerekiyor. Bu küçük veri setini ele alalım. X_1 ve X_2 olmak üzere iki özelliğimiz olduğunu ve etiket değişkeninin Y olduğunu görebilirsiniz. X_1 ve X_2, 0, 1 veya 2'den değerler alabilir; bu, hem X_1 hem de X_2'nin kategorik değişkenler olduğu anlamına gelir. Ve Y, 0 veya 1 olmak üzere iki değer alabilir, bu nedenle temelde sadece binary classificationdır (ikili sınıflandırmadır).

X_1	X_2	Y
0	0	0
0	1	1
1	2	1
0	0	1
2	2	0
1	1	0
0	2	1
2	0	0
2	1	0
1	0	0

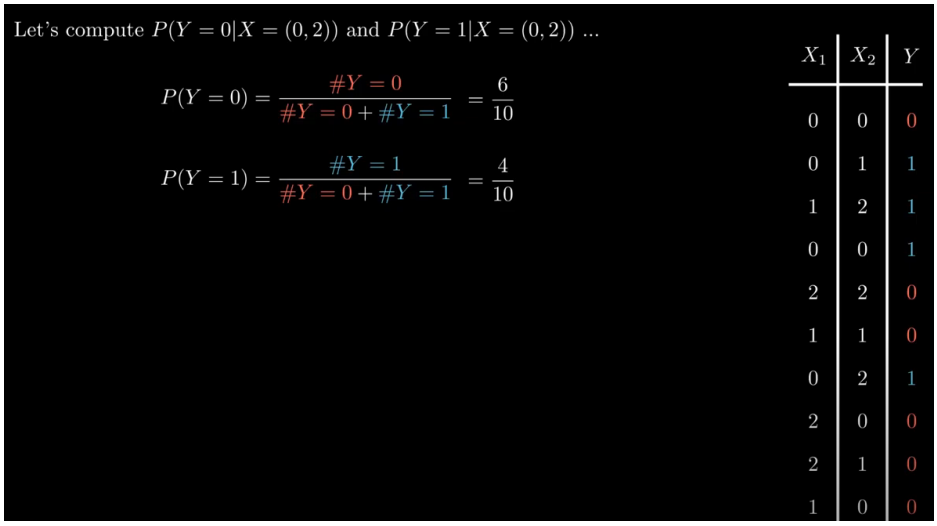
$X_1, X_2 \in \{0, 1, 2\}$
$Y \in \{0, 1\}$

Şimdi birisinin, X'in (0, 2)'ye eşit olduğu göz önüne alındığında, Y'nin değerini tahmin etmemizi istediğini hayal edin. Bu, birisinin 0 ve 2 olan X_1 ve X_2 değerini verdiği ve bu özellik kümesi için doğru etiketi bulmamızı istediği anlamına gelir.

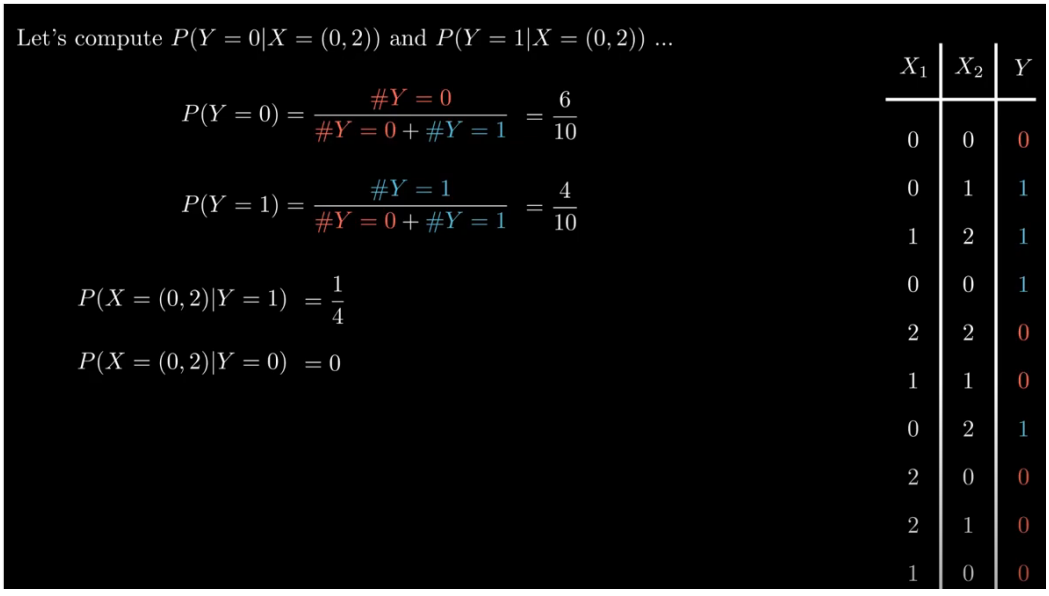
$$\text{Estimate the value of } Y \text{given that } X = (0, 2)$$

Her iki etiket için de Y=0 ve Y=1 koşullu olasılıkların değerini hesaplayalım. Her şeyden önce, Y'nin olasılığının 0'a ve Y'nin olasılığının 1'e eşit olduğu anlamına gelen *prior*ı hesaplayacağız.

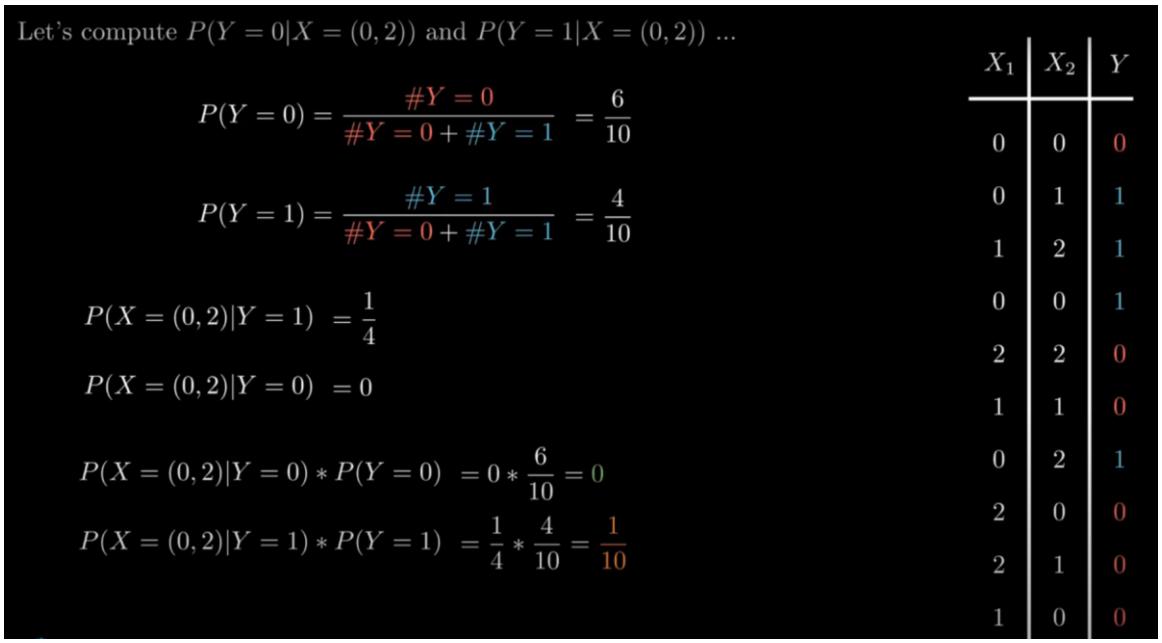
Formül çok basit. Paya Y'nin frekansı sıfıra eşit yazacağız ve paydaya toplam gerçekleşme sayısını yazacağız. Buradaki very setine baktığımızda Y'nin altı oluşumu sıfıra eşittir ve dört Y oluşumu 1'e eşittir. Dolayısıyla olasılığını hesaplamak için P(Y=0) değeri 6/10 olacaktır. P(Y=1) olasılığı için payı Y eşittir 1'in frekansıyla değiştirmemiz gerekiyor. Bu da 4/10 olacak.



Şimdi likelihoodları hesaplayalım. İlk önce, Y=1 olması durumunda X'in olasılığını hesaplayacağız. Sadece Y'nin değerinin 1 ve özellik kombinasyonunun 0, 2 olduğu tüm satırlara bakmamız gerekiyor. Veri setine baktığımızda bu ifadenin değeri 1/4 olacaktır. Benzer şekilde Y=0 için likelihoodu hesaplayalım ve göreceğiz ki veri setimizde Y'nin değerinin 0 olduğu ve özellik kombinasyonunun (0,2) olduğu tek bir gözlem yok. Yani bu likelihoodun değeri 0'dır.



Şimdi, posterior olasılığı maksimize eden sınıf etiketini bulmak için daha önce gösterdiğim payı hesaplamamız yeterlidir. Yani bunu yapmanın yolu, likelihood prior ile çarpmaktır. Hesaplamadan sonra, sınıf etiketi 0 için payın değerinin 0, sınıf etiketi 1 için 1/10 olduğunu görüyoruz. Yani açıkça sınıf etiketi 1, posterior olasılığımızı maksimize eder.

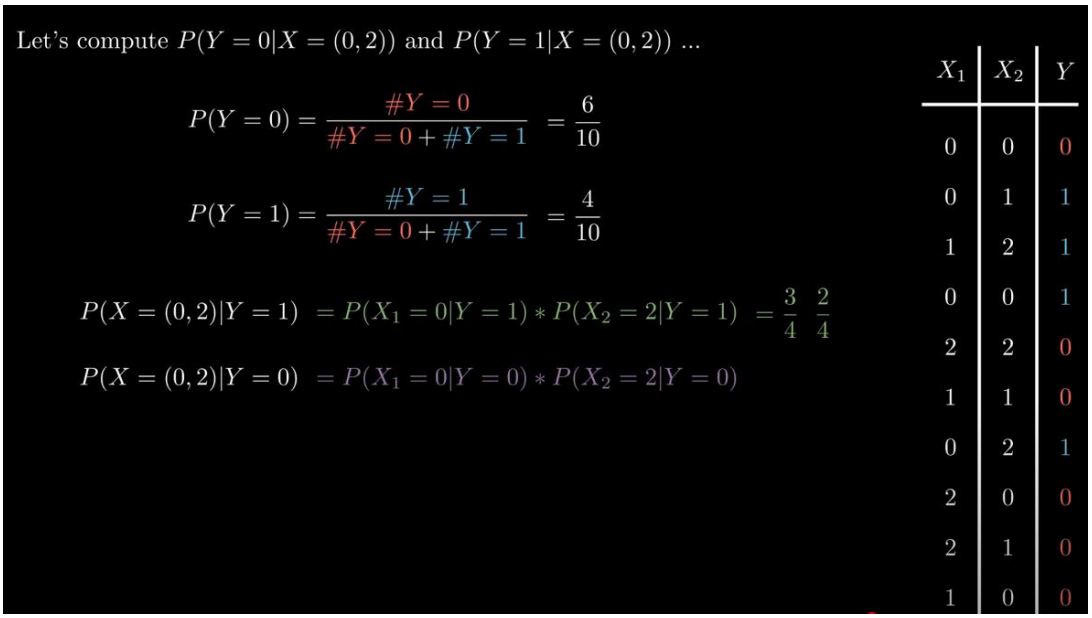


Tamam, yani bir cevabımız var, bu yüzden bunda sorun yok, değil mi? Hayır. Bu yöntemle ilgili büyük bir sorun var. Sorun şu ki, veri setimizde bu X_1 ve X_2 kombinasyonunu bulmak zor. Gördüğünüz gibi, Y'nin 1'e eşit olduğu ve X_1'in 0 ve X_2'nin 2 olduğu örneğin yalnızca bir örneğini bulduk ve Y'nin 0'a ve X_1'in 0'a ve X_2'nin 2'ye eşit olduğu tek bir gözlem bulamadık ve bu sadece iki feature içindi.

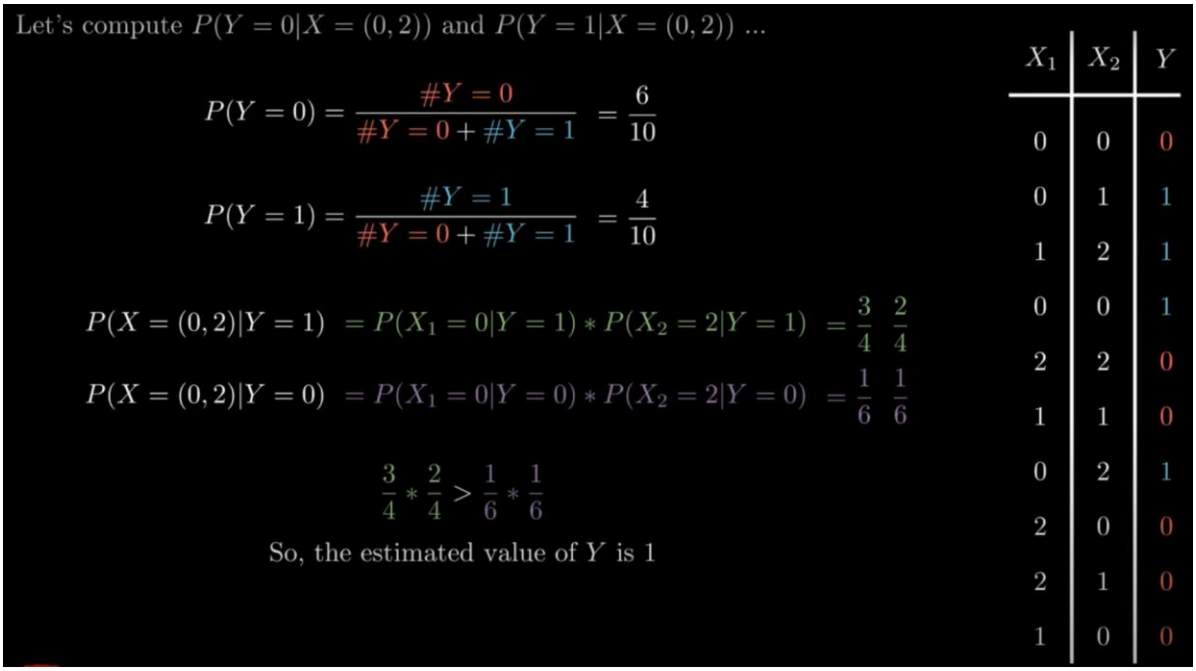
Diyelim ki 50 feature'ımız var. Veri setimizde 50 feature'ın belirli bir kombinasyonunu bulmanın ne kadar zor olacağını hayal edin. Belirli bir özellik kombinasyonuyla asla karşılaşmamamız çok olası değil mi? Birden fazla arama kombinasyonunun tek bir oluşumunu bile bulamazsak, olasılık değeri her zaman 0 olacaktır ve birden fazla 0 değerimiz varsa gerçekten karşılaştıramayız değil mi? Bu problemin üstesinden gelmek için Naive Bayes sınıflandırıcısını kullanıyoruz.

Konsept çok basit, sadece X_1 ve X_2'nin birbirinden bağımsız olduğunu düşünmemiz gerekiyor. Şimdi bu neden yardımcı olabilir? Eğer X_1 ve X_2'nin birbirinden bağımsız olduğunu düşünürsek, o zaman veri setimizde o X_1 ve X_2 kombinasyonunu gerçekten bulmamız gerekmez. Y verildiğinde X'in olasılığını P(X|Y) bir çarpım olarak yazabiliriz. Feature'lar bağımsız değilse bunu yazamayacağımızı bilin. Naive Bayes'in yaptığı büyük varsayım bu. Dürüst olmak gerekirse, gerçek dünyada özellikler aslında bağımsız olmayabilir. Bu yüzden soruna naif bir yaklaşım deniyor. Ama işimizi kolaylaştırıyor ve gerçekten iyi sonuçlar veriyor.

Öyleyse, Y'nin 1'e eşit olduğu durumda X_1'in 0'a eşit olma olasılığını hesaplayalım. Bunu yapmak için, Y'nin 1'e ve X_1'in 0'a eşit olduğu girişleri saymamız gerekiyor. Böyle 3 durum olduğunu ve Y'nin 1'e eşit olduğu oluşumların sayısının 4 olduğunu görüyoruz, dolayısıyla olasılık 3/4 olacaktır. Benzer şekilde, Y'nin 1'e eşit olduğu durumda X_2'nin 2'ye eşit olma olasılığını hesaplarsak, 2/4 elde ederiz.



Şimdi bir sonraki satıra gelelim. Yani burada ilk önce Y'nin 0'a eşit olduğu yerde X_1'in değerinin kaç kez 0'a eşit olduğunu bulmamız gerekiyor. Sadece bir gözlem var ve Y'nin 0'a eşit olduğu durumların sayısı 6'dır. Olasılık 1/6 olur. Benzer şekilde, X_2'nin 2'ye ve Y'nin 0'a eşit olduğu durumların sayısını hesaplarsak, burada da 1/6 olacaktır. Şimdi bunları karşılaştıralım. Açık bir şekilde birinci olasılık ikincisinden daha büyük. Böylece, 1. sınıf etiketinin posterior olasılığımızı maksimize ettiğini açıkça görebiliriz. Yani tahmini sınıf etiketi 1'dir. Naive Bayes'in nasıl çalıştığını öğrenmiş olduk.



Şimdi sürekli özelliklerle nasıl başa çıkabileceğinizden bahsedeceğim. Şimdiye kadar kategorik değişkenlerle uğraşıyorduk. X_1 ve X_2'nin ikisi de kategorikti ve bu durumda aslında frekansları hesaplayabilirdik ama sürekli özellikler olması durumunda frekansları hesaplayamayız. Peki bununla nasıl başa çıkılır? Birçok yolu var, iki yolu konuşacağız. Bunlardan ilki discretization

(ayrıklaştırmadır). Yaş adında sürekli bir değişken olduğunu varsayalım ve yaş 11 ile 50 arasında olduğunu düşünelim. Şimdi tüm aralığı, sürekli değişkenin kategorik bir değişken olacağı şekilde birkaç gruba ayırabiliriz. Örneğin, dört gruba ayıralım. Bunların ilki 11 ile 20 arasındaki yaş grubuna karşılık geliyor, ikincisi 21 ila 30 arasındaki yaş grubuna tekabül ediyor gibi yorumlayabiliriz değil mi? İkinci yöntem ise bilinen bir dağılımı özelliklerimize fit etmektir. Bu dağılım, verilerin ihtiyacına göre normal dağılım olabilir. Ama bu sorunu nasıl düzeltir? Bilinen bir dağılımımız olduğunda bunun olasılık fonksiyonu olduğunu biliyoruz. Bu nedenle, Y bir etikete eşit olarak verildiğinde X'in olasılık değerini (likelihood) hesaplamak için sadece bilinen dağılımın olasılık yoğunluk fonksiyonunu kullanabiliriz. Burada f bilinen bir dağılımın olasılık fonksiyonunu gösterir. Baştaki çarpma işareti burada da özelliklerin birbirinden bağımsız olmasını düşündüğümüzü söylüyor. Çünkü ancak o zaman olasılıkları çarpabiliriz. Sürekli özelliklerle bu şekilde başa çıkıyoruz.