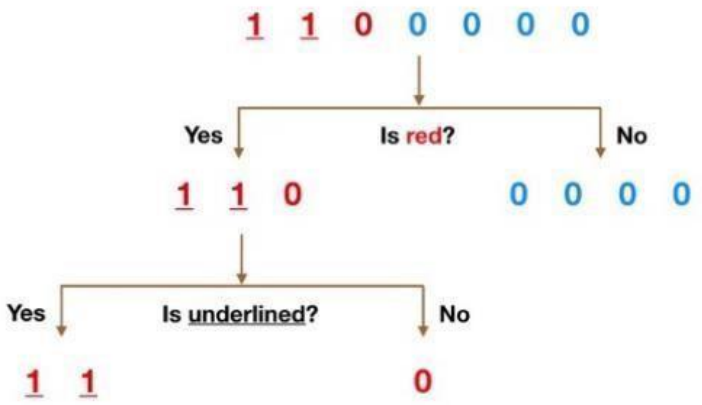
6.3.1. Rassal Orman Modeli Algoritması

Rastgele ormanlar, denetimli bir öğrenme algoritmasıdır. Hem sınıflandırma hem de regresyon için kullanılabilir.

Karar ağaçları:
Rastgele orman modelinin yapı taşları oldukları için karar ağaçlarının bilinmesi gerekmektedir. Oldukça sezgisel yaklaşımlar içerir. Çoğu insanın hayatlarının bir noktasında bilerek ya da bilmeyerek bir karar ağacı kullandığına bahse girerim.

Bir karar ağacının nasıl çalıştığını bir örnek üzerinden anlamak muhtemelen çok daha kolaydır.



Veri setimizin soldaki şeklin üstündeki sayılardan oluştuğunu hayal edin. İki 1 ve beş 0'ımız var (1'ler ve 0'lar sınıflarımızdır) ve özelliklerini kullanarak sınıfları ayırmak istiyoruz. Özellikler renklidir (kırmızıya karşı mavi) ve gözlemin altı çizili olup olmadığıdır. Peki bunu nasıl yapabiliriz?

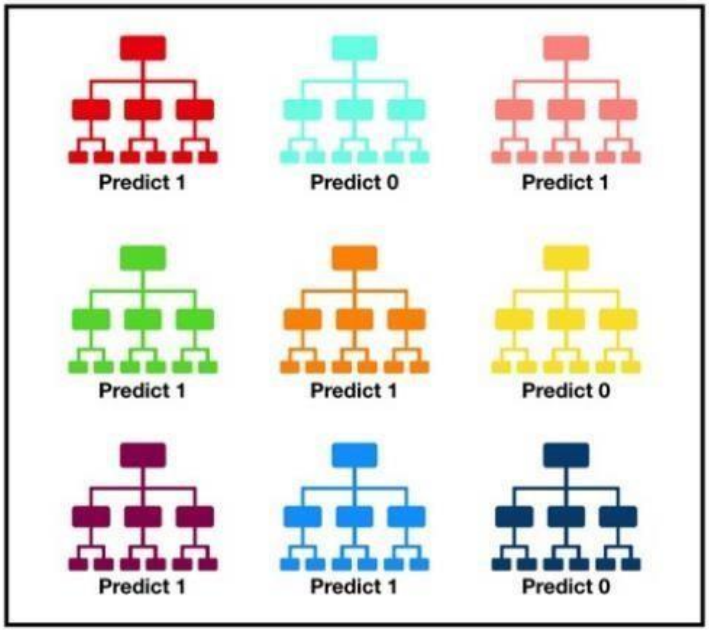
Renk, 0'lardan biri hariç tümü mavi olduğu için, ayrılması oldukça bariz bir özellik gibi görünüyor. Böylece "Kırmızı mı?" Sorusunu kullanabiliriz. ilk düğümümüzü ayırmak için. Bir ağaçtaki bir düğümü, yolun ikiye ayrıldığı nokta olarak düşünebilirsiniz - kriterleri karşılayan gözlemler Evet dalına ve Hayır dalına inmeyenler.

Hayır dalı (blues) artık 0'lardır, bu yüzden orada işimiz bitti, ancak Evet şubemiz yine de bölünebilir. Şimdi ikinci özelliği kullanıp "Altı çizili mi?" Diye sorabiliriz. ikinci bir bölme yapmak için.

Altı çizili iki 1, Evet alt dalına gider ve altı çizilmemiş 0, sağ alt daldan aşağı gider ve hepimiz işimiz biter. Karar ağacımız, verileri mükemmel bir şekilde bölmek için iki özelliği kullanabildi. Açıkçası gerçek hayatta verilerimiz bu kadar net olmayacak, ancak bir karar ağacının kullandığı mantık aynı kalacak. Her düğümde sorulacak - Hangi özellik, eldeki gözlemleri, ortaya çıkan grupların olabildiğince farklı olacağı şekilde (ve ortaya çıkan her alt grubun üyeleri mümkün olduğunca birbirine benzeyecek şekilde) bölmeme izin verir?

Rastgele Orman Sınıflandırıcısı:

Rastgele orman, adından da anlaşılacağı gibi, bir topluluk olarak çalışan çok sayıda bireysel karar ağacından oluşur. Rastgele ormandaki her bir ağaç bir sınıf tahmini verir ve en çok oyu alan sınıf, modelimizin öngörüsü haline gelir (aşağıdaki şekle bakın).



Tally: Six 1s and Three 0s

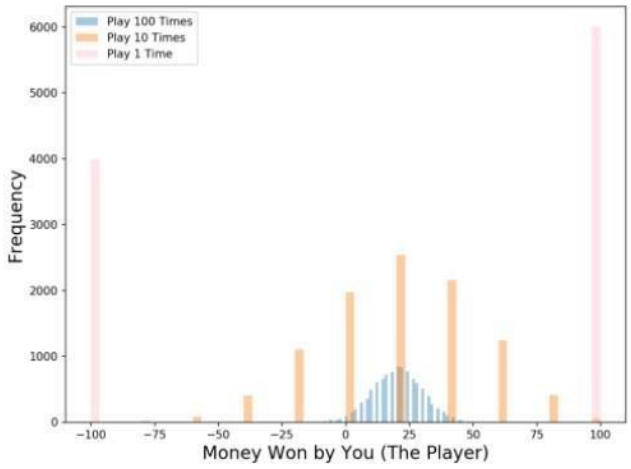
Rastgele ormanın ardındaki temel kavram basit ama güçlü bir kavramdır - kalabalıkların bilgeliği. Veri biliminde konuşursak, rastgele orman modelinin bu kadar iyi çalışmasının nedeni şudur: Bir komite olarak faaliyet gösteren çok sayıda görece ilişkisiz model (ağaç), münferit kurucu modellerin herhangi birinden daha iyi performans gösterecektir. Modeller arasındaki düşük korelasyon anahtardır. Tıpkı düşük korelasyonlu yatırımların (hisse senetleri ve tahviller gibi) bir araya gelerek parçalarının toplamından daha büyük bir portföy oluşturması gibi, ilişkisiz modeller, bireysel tahminlerin herhangi birinden daha doğru olan topluluk tahminleri üretebilir. Bu harika etkinin nedeni, ağaçların birbirlerini kendi hatalarından korumalarıdır (sürekli aynı yönde hata yapmadıkları sürece). Bazı ağaçlar yanlış olabilirken, diğer birçok ağaç haklı olacaktır, bu nedenle bir grup olarak ağaçlar doğru yönde hareket edebilecektir. Bu nedenle, rastgele ormanın iyi performans göstermesi için ön koşullar şunlardır: Özelliklerimizde bazı gerçek sinyaller olması gerekir, böylece bu özellikler kullanılarak oluşturulan modeller rastgele tahmin etmekten daha iyi sonuç verir. Tek tek ağaçların yaptığı tahminlerin (ve dolayısıyla hataların) birbirleriyle düşük korelasyonlara sahip olması gerekir.

İlişkisiz sonuçların neden bu kadar büyük olduğuna dair bir örnek:

İlişkisiz birçok modele sahip olmanın harika etkileri o kadar kritik bir kavramdır ki, şu oyunu oynadığımızı hayal edin: Bir sayı üretmek için tekdüze dağıtılmış rasgele sayı üreteci kullanıyorum. Oluşturduğum sayı 40'tan büyük veya ona eşitse, kazanırsınız (yani% 60 zafer şansınız olur) ve size biraz para öderim. 40'ın altındaysa ben kazanırım ve siz bana aynı miktarı ödersin. Şimdi size aşağıdaki seçenekleri sunuyorum. Şunlardan birini yapabiliriz:
Oyun 1 - 100 kez oynayın, her seferinde 1 $ bahis yapın.
Oyun 2 - 10 kez oynayın, her seferinde 10 $ bahis yapın.
Oyun 3 - bir kez oynayarak 100 $ bahis yapın.

Hangisini seçerdin? Her oyunun beklenen değeri aynıdır:
Beklenen Değer Oyunu 1 = (0.60 * 1 + 0.40 * -1) * 100 = 20
Beklenen Değer Oyunu 2 = (0.60 * 10 + 0.40 * -10) * 10 = 20
Beklenen Değer Oyunu 3 = 0.60 * 100 + 0.40 * -100 = 20

Dağılımlar ne olacak? Sonuçları bir Monte Carlo simülasyonu ile görselleştirelim (her oyun türü için 10.000 simülasyon çalıştıracağız; örneğin, 1. Oyundaki 100 oyunun 10.000 katını simüle edeceğiz). Soldaki tabloya bir göz atın - şimdi hangi oyunu seçerdiniz? Beklenen değerler aynı olsa bile, sonuç dağılımları, pozitif ve dardan (mavi) ikiliye (pembe) doğru büyük ölçüde farklıdır.



Outcome Distribution of 10,000 Simulations for each Game

Oyun 1 (100 kez oynadığımız yer), biraz para kazanmak için en iyi şansı sunuyor - yürüttüğüm 10.000 simülasyondan% 97'sinde para kazanıyorsunuz! Oyun 2'de (10 kez oynadığımız yerde) simülasyonların% 63'ünde para kazanırsınız, ciddi bir düşüş (ve para kaybetme olasılığınızda ciddi bir artış). Ve sadece bir kez oynadığımız 3. Oyun, beklendiği gibi simülasyonların% 60'ında para kazanıyorsunuz.

Dolayısıyla, oyunlar aynı beklenen değeri paylaşsa da, sonuç dağılımları tamamen farklıdır. 100 $ 'lık bahsimizi farklı oyunlara ne kadar çok bölersek, para kazanacağımıza o kadar

güvenebiliriz. Daha önce de belirtildiği gibi, bu işe yarar çünkü her oyun diğerlerinden bağımsızdır.

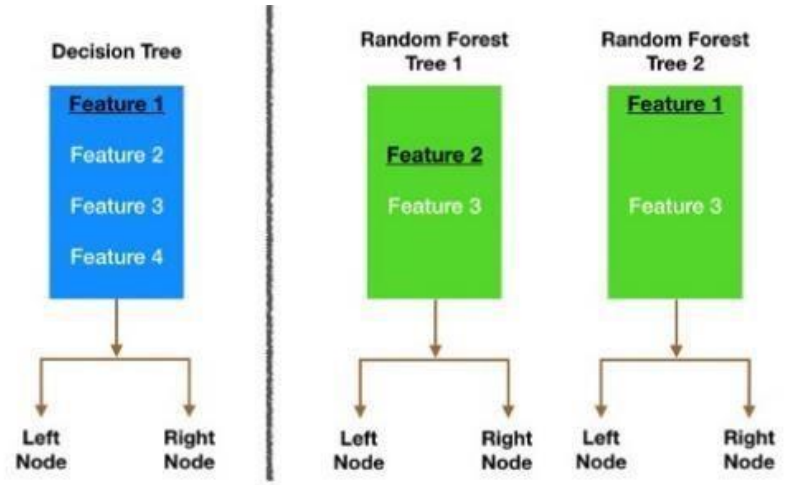
Rastgele orman aynıdır - her ağaç, önceki oyunumuzdaki bir oyun gibidir. Daha fazla oynadığımızda para kazanma şansımızın nasıl arttığını gördük. Benzer şekilde, rastgele bir orman modeliyle, modelimizdeki ilişkisiz ağaçların sayısı ile doğru tahminler yapma şansımız artar.

Modellerin birbirini çeşitlendirmesini sağlamak:
Öyleyse rastgele orman, her bir ağacın davranışının modeldeki diğer ağaçlardan herhangi birinin davranışıyla çok fazla ilişkili olmamasını nasıl sağlar?

Aşağıdaki iki yöntemi kullanır:
Torbalama (Bootstrap Aggregation) - Karar ağaçları, eğitildikleri verilere karşı çok hassastır - eğitim setinde yapılan küçük değişiklikler, önemli ölçüde farklı ağaç yapılarına neden olabilir. Rastgele orman, her bir ağacın veri kümesinden değiştirilerek rasgele örneklemesine izin vererek bundan yararlanır ve farklı ağaçlarla sonuçlanır. Bu işlem torbalama olarak bilinir.

Torbalama ile eğitim verilerini daha küçük parçalara ayırmadığımıza ve her ağacı farklı bir yığın üzerinde eğitmediğimize dikkat edin. Bunun yerine, N büyüklüğünde bir örneğimiz varsa, yine de her ağaca N boyutunda bir eğitim seti besliyoruz (aksi belirtilmedikçe). Ancak orijinal eğitim verileri yerine, değiştirilmiş N boyutunda rastgele bir örnek alıyoruz. Örneğin, eğitim verilerimiz [1, 2, 3, 4, 5, 6] ise, ağaçlarımızdan birine aşağıdaki listeyi verebiliriz [1, 2, 2, 3, 6, 6]. Her iki listenin de altı uzunluğunda olduğuna ve "2" ile "6" nın ikisinin de ağacımıza verdiğimiz rastgele seçilmiş eğitim verilerinde tekrarlandığına dikkat edin (çünkü değiştirme ile örnekleme yapıyoruz).

Özellik Rastgeleliği:
Normal bir karar ağacında, bir düğümü bölme zamanı geldiğinde, mümkün olan her özelliği göz önünde bulundururuz ve sağ düğümdekilerle sol düğümdeki gözlemler arasında en fazla ayrımı yaratanı seçeriz. Bunun aksine, rastgele bir ormandaki her ağaç yalnızca rastgele bir özellik alt kümesinden seçim yapabilir. Bu, modeldeki ağaçlar arasında daha fazla çeşitliliği zorlar ve sonuçta ağaçlarda daha düşük korelasyon ve daha fazla çeşitlilik ile sonuçlanır.



Node splitting in a random forest model is based on a random subset of features for each tree.

Görsel bir örnek üzerinden geçelim - yukarıdaki resimde, geleneksel karar ağacı (mavi), düğümü nasıl böleceğine karar verirken dört özelliğin tümünden seçim yapabilir. Verileri olabildiğince ayrılmış gruplara böldüğü için Özellik 1 (siyah ve altı çizili) ile devam etmeye karar verir. Şimdi rastgele ormanımıza bir göz atalım. Bu örnekte ormanın iki ağacını inceleyeceğiz. Rastgele Orman Ağacı 1'i kontrol ettiğimizde, düğüm bölme kararı için yalnızca Özellik 2 ve 3'ü (rastgele seçilir) dikkate alabileceğini görürüz. Geleneksel karar ağacımızdan (mavi olarak) Özelliğin bölme için en iyi özellik olduğunu biliyoruz, ancak Ağaç 1, Özellik 1'i göremediğinden, Özellik 2'ye (siyah ve altı çizili) gitmek zorunda kalır. Öte yandan Ağaç 2, yalnızca Özellik 1 ve 3'ü görebilir, bu nedenle Özellik 1'i seçebilir. Dolayısıyla rastgele ormanımızda, yalnızca farklı veri kümeleri üzerinde eğitilen (torbalama sayesinde) değil, aynı zamanda karar vermek için farklı özellikler kullanan ağaçlarla karşılaşıyoruz. Ve bu, sevgili okuyucum, birbirlerini tamponlayan ve hatalarından koruyan ilişkisiz ağaçlar yaratır.

Rastgele orman, birçok karar ağacından oluşan bir sınıflandırma algoritmasıdır. Komite tarafından tahmin edilmesi herhangi bir ağaçtan daha doğru olan ilişkisiz bir ağaç ormanı yaratmaya çalışmak için her bir ağacı inşa ederken torbalama kullanır ve rastgelelik özelliğini kullanır. Rasgele ormanımızın doğru sınıf tahminleri yapabilmesi için neye ihtiyacımız var? En azından bir miktar tahmin gücüne sahip özelliklere ihtiyacımız var. Sonuçta, eğer içine çöp koyarsak, o zaman çöpü dışarı atarız. Ormanın ağaçları ve daha da önemlisi tahminlerinin ilintisiz olması (veya en azından birbirleriyle düşük korelasyonlara sahip olması) gerekir. Algoritmanın kendisi özellik rastgeleliği aracılığıyla bu düşük korelasyonları bizim için tasarlamaya çalışırken, seçtiğimiz özellikler ve seçtiğimiz hiper parametreler nihai korelasyonları da etkileyecektir.