ENSEMBLE LEARNİNG

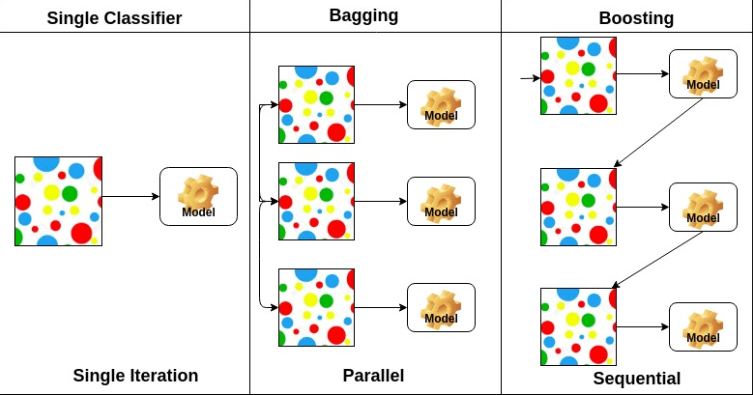
Random Forest bir Topluluk Modelleri (ensemble models) algoritmasıdır. Peki Topluluk Model (ensemble model )nedir? Bunu şöyle bir örnekle açıklamak istiyorum:

metin, kişi içeren bir resim

Açıklama otomatik olarak oluşturuldu

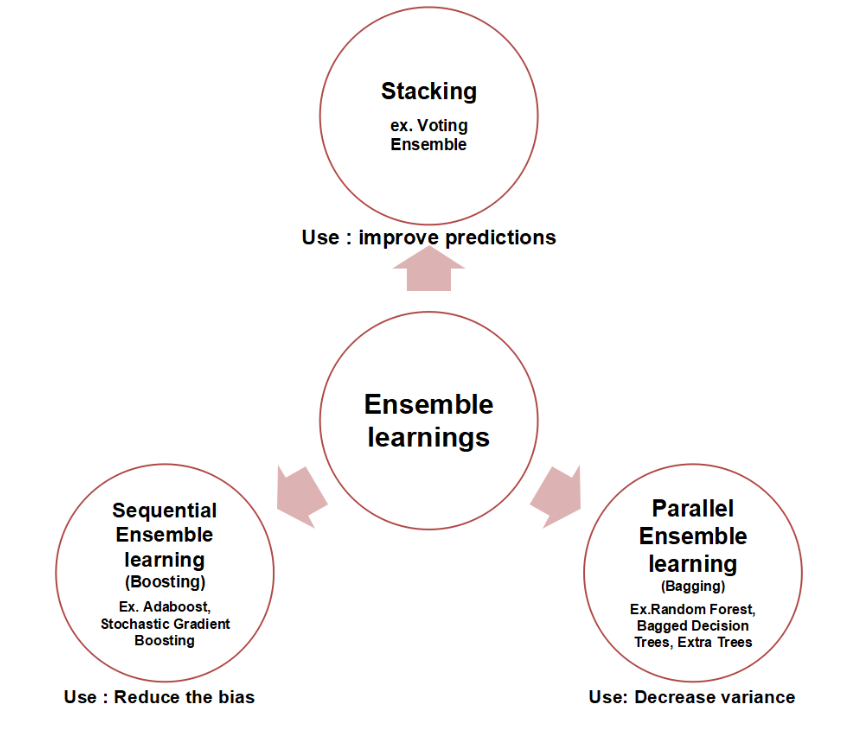
İnternetten bir telefon almak istediğimizi düşünelim. Telefonu alırken dikkat ettiğimiz başlıca şey özellikleridir. Özelliklerine baktık ve 2 telefonu beğendik. Hangisini seçeceğiz? Bu süreçte dikkat edeceğimiz şey **yorumlardır**. Çünkü fotoğraflar gerçeği yansıtmayabilir ya da ilerleyen süreçlerde telefonla ilgili beklenmedik bir şeyler olabilir. Yani başkalarının bu telefon hakkındaki düşünceleridir. Bu yorumlardan yola çıkarak kendi terazimize koyarak bir seçim yaparız.

Genel olarak Makine Öğrenmesindeki topluluk modelleri (ensemble models) yukarıda örneklediğimiz gibi çalışır. Genel performansı iyileştirmek için birden fazla modelden alınan kararları birleştirirler.



Topluluk Modelleri 3 yönteme ayrılır:

1. Bagging / Bootstrap Aggregation
2. Boosting
3. Stacking



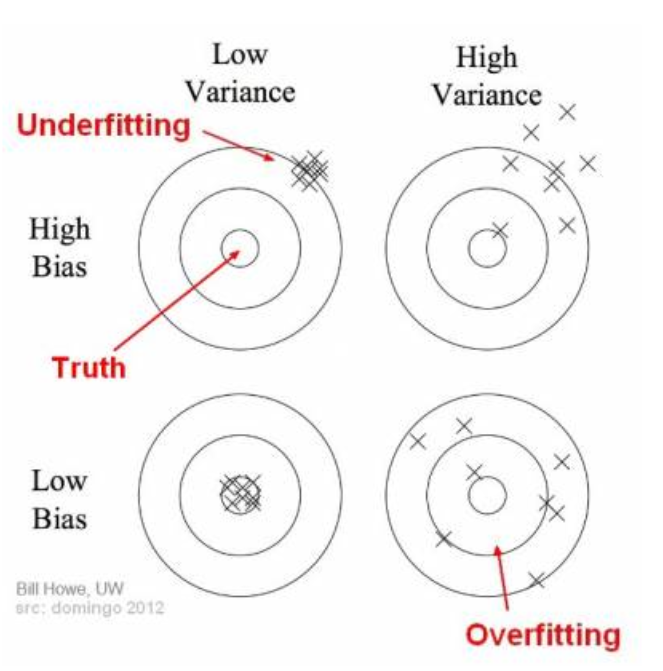
**1.Bagging**

Random Forest, Bagging yöntemine dahil olan bir algoritmadır.

Bagging yöntemi Paralel olarak çalışan bir yöntemdir. Orijinal veri setinden elde edilen önyükleme örneklerine tahminciler uygulanarak bir topluluk oluşturulur. Burada önyükleme uygulaması, iadeli rastgele

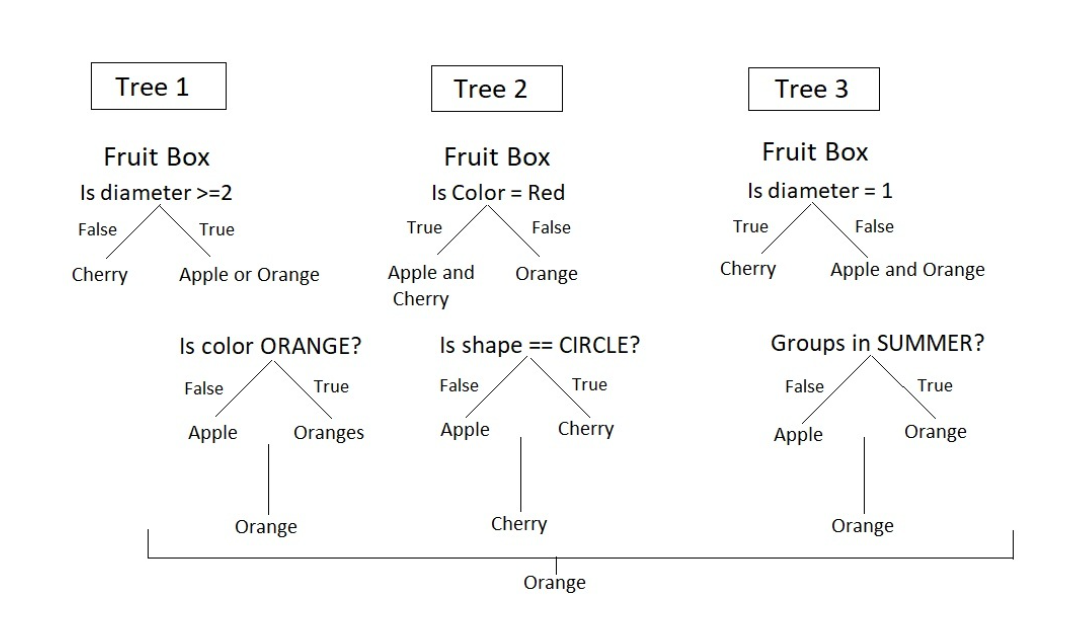
ler oluşturmak için kullanılır. Oluşturulan alt örneklemlerin her birindeki gözlem ile orijinal veri setindeki gözlem sayısı aynı olacaktır. Bu nedenle örneklemlerde bazı gözlemler hiç yokken bazı gözlemlerden 2 veya daha fazla defa bulunabilir.

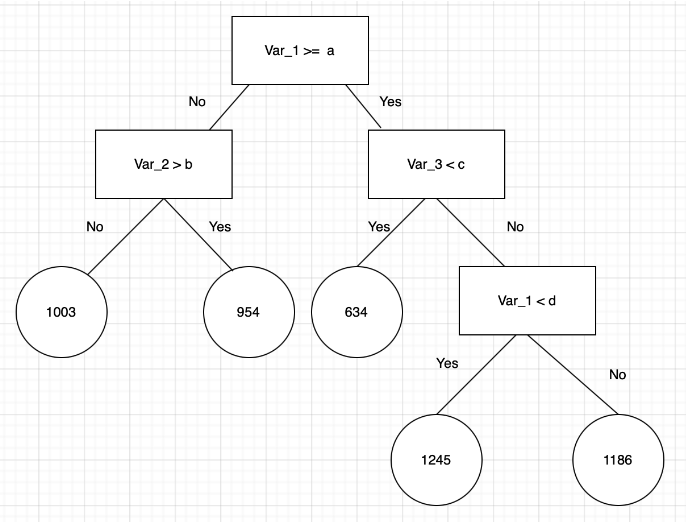
Tahminlerin birleştirilmesi aşamasında regresyon ağaçları için ortalama alınırken sınıflandıra ağaçlarında sonuçlar oylama ile belirlenir. Varyansı düşürmeyi amaçlar.



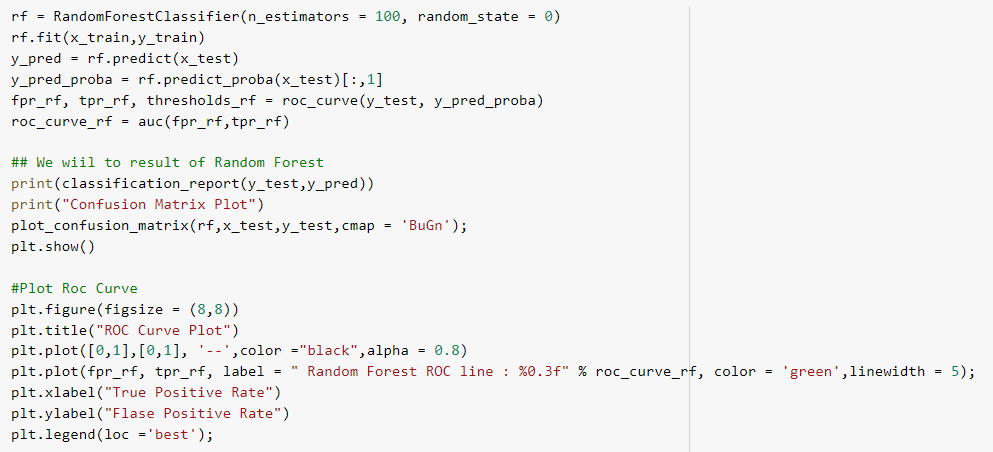
Random Forest algoritması birbirinden bağımsız olarak çalışan karar ağaçlarını bir araya getirerek eğer regresyon tahmini yapılacaksa karar ağaçları tahminlerinin ortalamasını alarak, sınıflandırma tahmini yapılacaksa tahminler arasından en çok oy alan tahmin seçilir. Aslında yukarda anlatılan örnekte bir sınıflandırma yapısı mevcuttur. Random Forest farklı decision tree üzerinden bir sonuç çıkarma odaklı olduğu için overfitting’i azaltmaktadır.

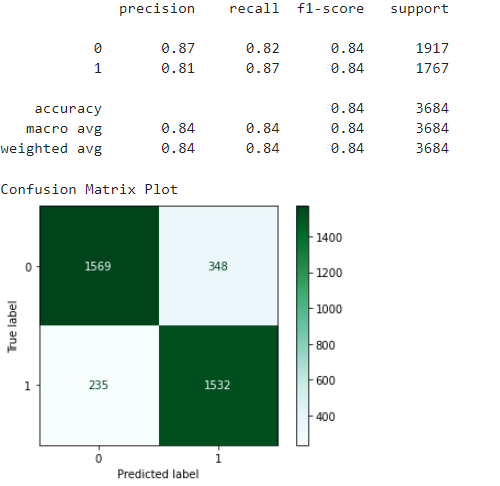
Biz iki telefon arasından bir seçim yapacaktık ve buna karar vermek için yorumlardan yararlanırız. Bundan dolayı her bir yorum bir karar ağacıdır diyebiliriz. Sonunda bizim yorumlardan yararlanarak verdiğimiz karar da Random Forest algoritmasına dahildir diyebiliriz.

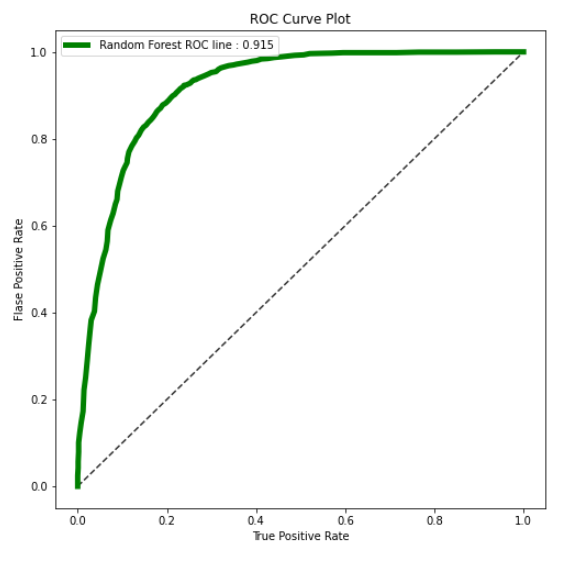




Şimdi Random Forest algoritmasına Python da göz atalım.

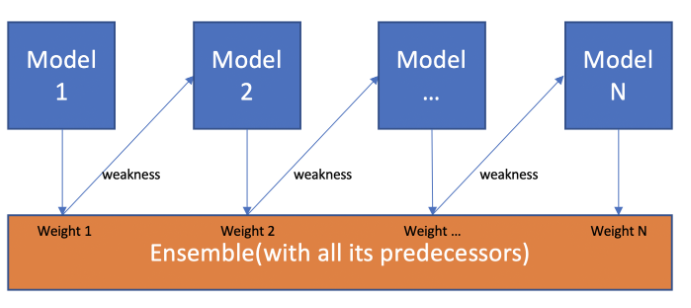






**2.Boosting**

Boosting, tek bir model tarafından yapılan yüksek kaliteli tahmine odaklanan birçok ML modelinden farklı olarak, güçlendirme algoritmaları, her biri önceki modellerin zayıflıklarını telafi eden bir dizi zayıf modeli eğiterek tahmin gücünü iyileştirmeye çalışır. Boost'u anlamak için, boost'un belirli bir modelden ziyade genel bir algoritma olduğunu anlamak çok önemlidir. Güçlendirme, zayıf bir model belirlemenizi gerektirir.

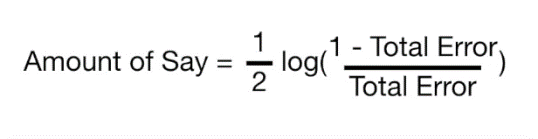


<https://towardsdatascience.com/boosting-algorithms-explained-d38f56ef3f30>

Boosting için iki ana algoritmadan bahsedeceğim: Adaptive Boost (AdaBoost) ve XGBoost.

1. **Adaptive Boost**

Adaptive Boost her bir iterasyonda yanlış sınıflandırılmış veri noktalarını tanımlar. Bir sonraki iterasyon için ağırlıklarını arttırır. Doğru olanların için ise bir sonraki iterasyon için ağırlıklarını azaltır. Böylece bir sonraki sınıflandırıcı yanlış olanları doğru yapmak için ekstra dikkat etmiş olur.



metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

<https://www.youtube.com/watch?v=LsK-xG1cLYA&t=1057s>

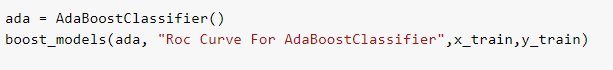
Şimdi Adaptive Boost’a python tarafında bakalım

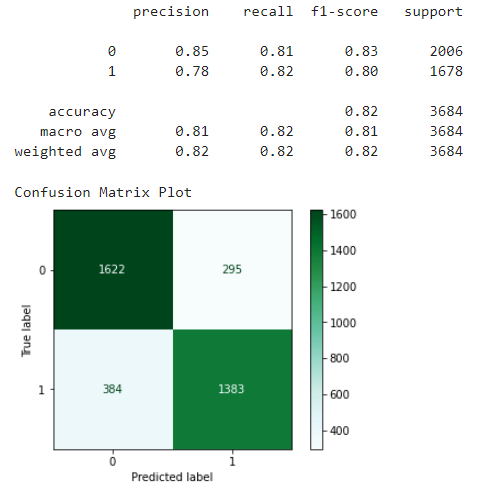
İlk başta birden çok boost modeli deneyeceğim için temiz kod prensibinde “boost\_models” adında bir metot tanımlıyorum.

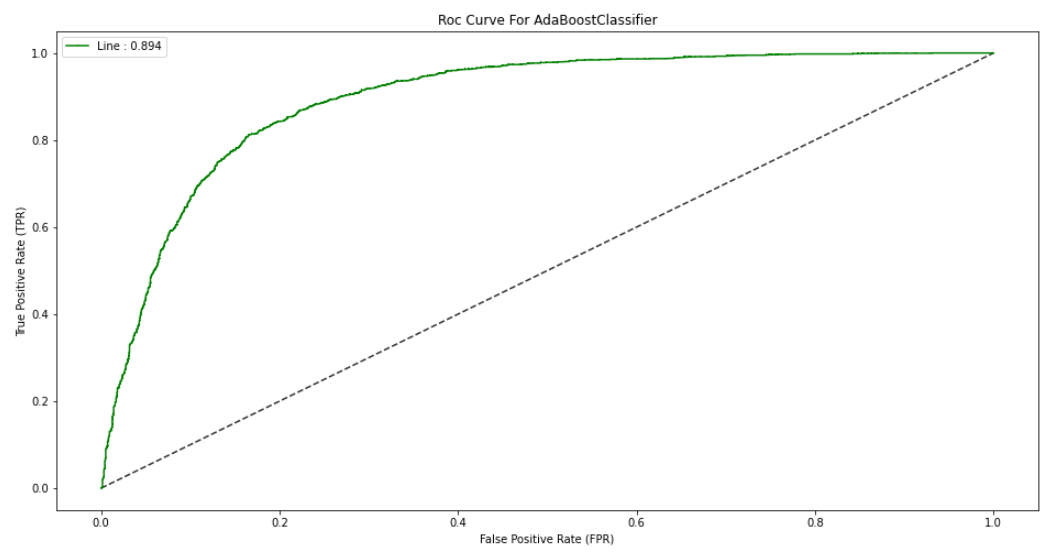
metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

“boost\_model” metodunu çağırarak Adaptive Boost Classifier Algoritmasını uygulamış oluyorum.







1. **XGBoost**

eXtreme Gradient Boosting (XGBoost), etkinlik, hesaplama hızı ve model performansı için tasarlanmış gradyan artırma algoritmasının (terminoloji uyarısı) ölçeklenebilir ve geliştirilmiş bir sürümüdür. Açık kaynaklı bir kitaplıktır ve Dağıtılmış Makine Öğrenimi Topluluğunun bir parçasıdır.

Regresyon tarafında herhangi bir eşik çizgisi belirliyoruz (örnekte 0.5 alınmıştır) ve verilerimizin buna göre uzaklıklarına bakıyoruz . Bu uzaklılara dayanarak bir “similarity score” elde ediyoruz. Buradaki Lambda parametresi bizim regularization için kullandığımız parametredir.

**Graphical user interface, application

Description automatically generated**

Bundan sonra dikey kesitler alarak decision tree’de root belirliyoruz.

Chart

Description automatically generatedYandaki dikey kesitte kesitin değeri sağındaki ve solundaki en yakın root’ların değer ortalamasından bulunmaktadır. Bu kesiti örnek alarak

Yukarıdaki formüle göre bir Gain değeri buluyoruz. Sonrasında bunu diğer kesit için de yaparak Gain değeri en büyük olan kesiti baş kesitimiz olarak alıyoruz. Bundan sonra ise bu şekilde devam ederek diğer alt kesitleri de buluyoruz. Sonrasında da belirli parametreler eşliğinde çıkan outpular ile çıktımızı elde ediyoruz.

Yukarıda tanımladığım “boost\_models” fonksiyonunu yine çağırıyorum.

Graphical user interface

Description automatically generated

Chart, waterfall chart

Description automatically generated

Chart, line chart

Description automatically generated