

Sınıflandırıcı Topluluklarında Başarı Tabanlı Budama ve Ağırlıklı Oylama

Performance Based Pruning and Weighted Voting with Classification Ensembles

Mehmet Fatih AMASYALI¹, Okan ERSOY²

1. Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi
mfatih@ce.yildiz.edu.tr

2. Elektrik ve Bilgisayar Mühendisliği Okulu
Purdue University
ersoy@purdue.edu

ÖZETÇE

Topluluk algoritmaları elde ettikleri yüksek başarılar sayesinde makine öğrenmesi alanında oldukça aktif bir konuma gelmişlerdir. Bu çalışmada 3 topluluk algoritması – Bagging, Rastgele Altuzaylar, Random Forest- üzerinde budama ve kararların ağırlıklandırılması işlemlerinin etkileri 26 sınıflandırma veri kümesi ile incelenmiştir. Üyelerinin başarıları arasında en fazla farklılık bulunan algoritmanın Rastgele Altuzaylar olduğu, tüm algoritmalar içinde en başarılı algoritmanın kararları ağırlıklandırılmış Rastgele Altuzaylar olduğu görülmüştür.

ABSTRACT

Ensemble algorithms have been a very popular research topic because of their high performances. In this work, performance based ensemble pruning and decision weighting methods are investigated on 3 ensemble algorithms (Bagging, Random Subspaces, Random Forest) over 26 classification datasets. According to our experiments; the algorithm including most diversity among its base learners is Random Subspaces. The best performed ensemble algorithm is Random Subspaces with decision weighting.

1. GİRİŞ

Makine öğrenmesinde aktif alanlardan birisi de öğrenici topluluklarıdır. Temel fikir tek bir öğrenici eğitip kullanmak yerine çok sayıda öğreniciyi eğitip sonuçlarını birleştirmektir [1,2]. Topluluk içinde yer alan her bir öğreniciye temel öğrenici (base learner) denmektedir. Topluluklardaki temel öğrenici bir sınıflandırıcı, kümeleyici ya da bir regresör olabilir ve öğrenici toplulukları temel öğrenicilerinin türlerine göre isimlendirilirler. Örneğin temel öğrenicisi sınıflandırıcı olan topluluklara sınıflandırıcı toplulukları, temel öğrenicisi kümeleyici olanlara kümeleyici toplulukları denmektedir.

Öğrenici topluluklarının işe yaraması (tek bir öğrenici ile elde edilen sonuçlardan daha yüksek başarıya sahip olmaları) için temel öğrenicilerin birbirlerinden farklı sonuçlar üretmeleri gereklidir [2]. Tüm temel öğrenicilerin aynı sonucu üretmeleri durumunda bu aynı kararları birleştirmenin işe yaramayacağı açıktır. Bununla birlikte temel öğrenicilerin tekil performanslarının yüksek olması öğrenici toplulukların başarısında önemli bir etkidir. Dolayısıyla temel öğrenicilerin birbirlerinden farklı sonuçlar üretmelerini (diversity) ve temel öğrenicilerin tekil performanslarının yüksek olmasını istemek aslında birbirine karşıttır. Temel öğrenicilerin performansı arttıkça sonuçları arasındaki farklılık (diversity) azalacaktır. Örneğin uç noktada tüm temel

öğrenicilerin performansları yüzde 100 olursa farklılıkları sıfır olacaktır. Bu örnekle iki isteğimiz arasındaki karşıtlık açık bir şekilde ortaya çıkmaktadır.

Öğrenici topluluklarının tasarımında 2 temel mekanizma bulunmaktadır. Bunlardan ilki farklı sonuçlar üreten temel öğrenicilerin üretimidir. Bunun için ya farklı temel öğreniciler (heterojen öğrenici toplulukları) kullanılır (ki her bir temel öğrenici veri üzerinde farklı varsayımlara sahip olduğundan farklı sonuçlar üretecektir) ya da aynı öğreniciler (homojen öğrenici toplulukları) farklı hiper parametrelerle ya da farklı eğitim kümeleriyle eğitilir [3]. 2. bölümde bu mekanizmaların ayrıntılarına girilecektir.

Öğrenici topluluklarının tasarımındaki mekanizmalardan ikincisi ise temel öğrenicilerin kararlarının nasıl birleştirileceğidir. Genelde, temel öğrenicilerin kararları ya ağırlıklı ya da ağırlıksız olarak birleştirilir.

Öğrenici topluluklarında bunlara ek olarak son zamanlarda öğrenici topluluklarında budama adında bir işlemlerde kullanılmaya başlanmıştır [4]. Bu yöntemlerde T adet temel öğrenici eğitilir daha sonra bunların arasında K ($T > K$) tanesi belli bir kritere göre seçilir. Bu sayede T adet rastgele üretilmiş temel öğrenici yerine K adet seçilmiş temel öğrenicinin kullanımı daha başarılı olabilir. Ayrıca budama işlemi, öğrenici topluluklarının test sürecini test örnekleri daha az temel öğrenici kullanılarak sınıflandırıldığından kısaltmaktadır.

Bu çalışmada 3 popüler öğrenici topluluğu algoritmasında temel öğrenici performansına bağlı budamanın, farklı budama miktarlarıyla topluluk performansı üzerindeki etkileri ve temel öğrenicilerin kararlarının ağırlıklandırılarak birleştirilmesi incelenmiştir.

2. SINIFLANDIRICI TOPLULUKLARI

Farklı sonuçlara sahip temel öğrenicilerin üretiminde homojen (aynı tür temel öğrenicilerden oluşan) ve heterojen (farklı temel öğrenicilerden oluşan) öğrenici toplulukları kullanılmaktadır. Bu çalışmada homojen topluluklar incelendiğinden bu tür toplulukların üretim mekanizmaları detaylandırılmıştır. Giriş bölümünde homojen topluluklar üretmek için ya aynı temel öğreniciyi farklı hiper parametrelerle aynı eğitim kümesi üzerinde çalıştırmak gerektiği ya da aynı temel öğreniciyi farklı eğitim kümeleri üzerinde çalıştırmak gerektiği ifade edilmişti. Bu çalışmada ikinci yol ile üretilen öğrenici toplulukları üzerinde çalışılmıştır. Literatürde farklı eğitim kümeleri üretmek için çeşitli metotlar bulunmaktadır.

Bagging: Breiman tarafından önerilmiştir [5]. Bagging'de N adet örnekten oluşan eğitim setinden yine N örnekli bir eğitim seti yerine koymalı rastgele seçimle üretilir. Bu

durumda bazı eğitim örnekleri yeni eğitim kümesinde yer almazken (yaklaşık %33) bazıları birden fazla kez yer alırlar. Topluluktaki her bir temel öğrenci bu şekilde üretilmiş birbirinden farklı örnekler içeren eğitim kümeleriyle eğitilirler ve sonuçları çoğunluk oylaması ile birleştirilir.

Rastgele Altuzaylar (Random Subspaces) : Ho tarafından önerilmiştir [6]. Bu yöntemde temel öğrencilere eğitim kümesindeki tüm örnekler verilmekte ancak tüm özellikler değil özelliklerin bir rastgele alt kümesi verilmektedir. Bu sayede her bir temel öğrenci veriye farklı uzaylardan bakmakta ve farklı sonuçlar üretmektedirler. Temel öğrencilerin kararları çoğunluk oylaması ile birleştirilmektedir.

Boosting: Bu [7] algoritmada ise her bir temel öğrenci önceki temel öğrencilerin doğru sınıflandıramadığı örnekler ile eğitilmektedir. Temel öğrencilerin kararları kendi eğitim kümeleri üzerindeki başarılarıyla ağırlıklandırılarak birleştirilmektedir.

Random Forest: Breiman tarafından önerilmiştir [8]. Temel öğrencileri karar ağaçlarıdır. Bu yöntemde temel öğrenciler yine Bagging ile üretilmiş eğitim örnekleriyle eğitilirler ancak temel öğrencilerin (karar ağaçlarının) her bir düğümünde veriyi bölerken tüm özelliklerin incelenmesi yerine özelliklerin rastgele bir alt kümesi incelenir. Bu sayede hem karar ağacının üretim süresi azalmakta hem de ağaçların kararlarının farklılığı yeni bir rastgelelikle artırılmaktadır. Temel öğrencilerin sonuçları yine çoğunluk oylaması ile birleştirilmektedir.

Öğrenci topluluk algoritmalarının birbirleriyle karşılaştırılması üzerine literatürde birçok çalışma yer almaktadır [6]. Tüm veri kümelerinde en başarılı olan bir metot bulunmamakla birlikte diğer tüm metotların genelde Bagging'den daha başarılı sonuçlar ürettiği görülmüştür.

3. SINIFLANDIRICI TOPLULUKLARINDA BAŞARI TABANLI BUDAMA

Öğrenci topluluklarının üretimindeki temel yapının rastgelelik olması örnek ve özellik seçimlerin belli bir kriter gere yapılmadığı ifade etmektedir. Bu durumda acaba rastgele üretilmiş T adet temel öğrenciyi kullanmak yerine bir kriter gere seçilmiş daha azını kullanmanın topluluğun başarısı üzerindeki etkisi nasıl olur sorusu tartışılmaya başlanmıştır. T adet temel öğrenciden oluşan bir toplulukta 2^T adet olası alt topluluk bulunmaktadır. Bunların optimum alt topluluğun bulunabilmesi için tüm bu topluluk olasılıkların bir geçirme kümesi üzerinde denemesi gerekmektedir ki T'nin büyük değerleri için bu oldukça pahalı bir işlemdir.

Bunun yerine bu çalışmada oldukça basit bir fikir denenmiştir. Topluluktaki tüm öğrenciler yerine, temel öğrencilerin eğitiminde kullanılmayan (out of bag) örnekler üzerindeki başarısı en yüksek olan K tanesinin kullanımı. Bu yöntem başarısı az olan temel öğrencileri topluluk dışına attığından (budadığından) topluluk içindeki temel öğrencilerin ortalama başarısını arttırmakta ve dolayısıyla farklılığı azaltmaktadır. Örneğin en başarılı çok az tanesini seçmek ortalama başarıyı çok yukarıya çekmekle birlikte farklılığı çok düşürecektir. Bu durumda topluluğun başarısı için bir denge bulmak zorunludur. Bu fikir, 3 topluluk algoritması (Bagging, Rastgele Altuzaylar ve Random Forest) üzerinde farklı budama miktarlarıyla denenmiştir.

4. SINIFLANDIRICI TOPLULUKLARINDA KARARLARIN AĞIRLIKLIL BİRLEŞTİRİLMESİ

Topluluk algoritmalarında temel öğrencilerin kararlarının birleştirilmesinde en çok kullanılan yöntem çoğunluk oylamasıdır. Bu yöntemde tüm temel öğrencilerin kararları eşit öneme sahiptir ve bir test örneği temel öğrenciler tarafından en çok hangi sınıfla etiketlenmişse bu topluluk kararı olarak kabul edilir. Bir diğer karar birleştirme yöntemi ise ağırlıklı oylama. Bu yöntemde temel öğrencilerin kararlarının etkileri eşit değildir. Topluluğun kararı temel öğrencilerinin kararlarının ağırlıklı birleştirilmesiyle belirlenir. Kararların ağırlıklarının belirlenmesinde literatürde birçok yöntem bulunmaktadır [3]. Bu çalışmada ise yine temel öğrencilerin eğitimlerinde kullanılmayan örnekler üzerindeki başarıları kullanılmıştır. Bu yöntemin, aynı 3 topluluk algoritması üzerindeki etkileri incelenmiştir.

5. KULLANILAN VERİ KÜMELERİ

Makine öğrenmesi çalışmalarında herkesin üzerinde çalışabileceği ortak veri kümeleri oluşturulmuştur. Bu sayede farklı metotların performanslarının birbiriyle adil bir şekilde karşılaştırılabilmesi mümkün olmuştur. Metotların performanslarının karşılaştırılması için yapılan denemelerde kullanılan veri kümesi sayısının fazlalığı sonuçların genelleştirilebilirliğini arttırmaktadır. Bu nedenle bu çalışmada UCI [9] veri tabanından 26 veri kümesi kullanılmıştır. Bu veri kümelerinin isimleri, örnek, özellik ve sınıf sayıları Tablo 1'de verilmiştir.

Tablo 1: Denemelerde kullanılan 26 veri kümesi

Veri Kümesi	Özellik sayısı	Sınıf sayısı	Örnek sayısı
abalone	11	19	4153
audiology	70	5	169
autos	72	5	202
balance-scale	5	3	625
breast-cancer	39	2	286
colic	61	2	368
credit-a	43	2	690
credit-g	60	2	1000
diabetes	9	2	768
glass	10	5	205
heart-statlog	14	2	270
hepatitis	20	2	155
hypothyroid	32	3	3770
ionosphere	34	2	351
kr-vs-kp	40	2	3196
labor	27	2	57
letter	17	26	20000
lymph	38	2	142
primary-tumor	24	11	302
ringnorm	21	2	7400
sonar	61	2	208
soybean	84	18	675
splice	288	3	3190
vehicle	19	4	846
vowel	12	11	990
waveform	41	3	5000

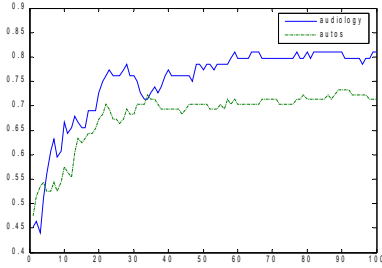
Kullanılan veri kümeleri incelendiğinde örnek, özellik ve sınıf sayılarının geniş bir yelpazede yer aldığı görülebilir. Bu yüzden seçilen veri kümelerinin makine öğrenmesi problemlerini iyi yansıttığı söylenebilir.

6. DENEYSEL SONUÇLAR

Topluluk algoritmalarının adil bir şekilde karşılaştırılabilmesi için hepsinde kullanılan temel öğrenici olarak bir karar ağacı olan CART [10] belirlenmiştir. Topluluk algoritmalarında budanmamış ağaçların başarıları genelde daha yüksek olduğundan karar ağaçları budanmadan kullanılmıştır. Tüm algoritmalarda Bagging boyutu, eğitim kümesiyle aynı boyutlu kullanılmıştır. Rastgele Altuzaylar'da altuzay boyutu orijinal boyutun (D) yarısı, Random Forest'larda $\log_2 D$ seçilmiştir. Sonuçların değerlendirilmesinde şans faktörünün etkisinin azaltılması için denemeler 5*2 çapraz geçişleme [11] ile yapılmıştır. Bir algoritmanın bir veri kümesi üzerindeki performansı 5*2 çapraz geçişlemeden elde edilen 10 değerlerin ortalaması olarak belirlenmiştir.

6.1. Topluluk Boyutunun Topluluk Başarımına Etkisi

Öğrenici topluluklarında, temel öğrenici sayısının artması genelde topluluğun başarısını artırır. Şekil 1'de "audiology" ve "autos" veri kümeleri üzerinde Random Forest algoritmasında temel öğrenici sayısının artışının topluluk performansına etkisi verilmiştir.



Şekil 1: Topluluk algoritmalarında topluluk boyutunun performans üzerindeki etkisi.

Şekil 1 incelendiğinde topluluk başarısının topluluk boyutuyla arttığı ve sonunda bir değere yakınsadığı görülmektedir.

6.2. Toplulukta Tümü Öğreniciler Yerine Bir Bölümünü Kullanmak (Budama)

Toplulukta üretilen tüm temel öğrenicilerin yerine bir kısmının kullanılmasının topluluk performansı üzerindeki etkisini incelemek için 3 topluluk algoritmasında da 100 ağacın üretilip, hepsinin, 75'inin 50'sinin ve 25'inin kullanımı denenmiştir. Üretilen 100 ağaçtan bir bölümünün seçiminde ağaçların eğitiminde kullanılmayan eğitim örnekleri üzerinde en başarılı olma kriteri kullanılmıştır.

6.3. Kararların Ağırlıklı Birleştirilmesi

Toplulukta üretilen temel öğrenicilerin kararlarının eşit olarak değilde ağırlıklandırılarak birleştirilmesinin topluluk performansı üzerindeki etkisini incelemek için aynı 3 topluluk algoritmasında 100 ağaç üretilip kararları eşit ve ağırlıklandırılarak birleştirilmiştir. Temel öğrenicilerin kararlarının ağırlıkları olarak eğitimlerinde yer almayan örnekler üzerindeki performansları kullanılmıştır.

6.4. Sayısal Sonuçlar

Üzerinde çalışılan 3 topluluk algoritmasında, başarı tabanlı budama ve ağırlıklandırmanın etkisi önce ayrı ayrı incelenmiştir. Tablo 2, 3 ve 4'de Bagging, Rastgele Altuzaylar ve Random Forest algoritmalarının tek bir ağaç, budama, ağırlıklandırma sonucunda elde edilen değerler verilmiştir. Bu tablolardaki değerler algoritmaların 6 algoritma içinde 26 veri kümesinde üstten ortalama kaçınıcı olduğunu göstermektedir. Küçük değerler başarıyı göstermektedir. Tablolarda algoritmaların yanlarında parantez içinde yer alan rakamlardan ilki kaç ağacın üretildiğini, ikincisi ise kaçının kullanıldığını göstermektedir.

Tablo 2: Random Forest sonuçları

Algoritma	Ortalama Başarı Sırası
Tek ağaç (CART)	5.5769
Random forest (100 , 25)	4.1154
Random forest (100 , 50)	3.1154
Random forest (100 , 75)	3.0385
Random forest (100 , 100) orijinal algoritma	2.3077
Random forest (100 , 100) Ağırlıklandırılmış	2.8462

Tablo 3: Bagging sonuçları

Algoritma	Ortalama Başarı Sırası
Tek ağaç (CART)	5.7308
Bagging (100 , 25)	4.2308
Bagging (100 , 50)	3.5385
Bagging (100 , 75)	3.2308
Bagging (100 , 100) orijinal algoritma	2.0769
Bagging (100 , 100) Ağırlıklandırılmış	2.1923

Tablo 4: Rastgele Altuzay sonuçları

Algoritma	Ortalama Başarı Sırası
Tek ağaç (CART)	5.6154
Rastgele Altuzay (100 , 25)	4.3462
Rastgele Altuzay (100 , 50)	3.3846
Rastgele Altuzay (100 , 75)	2.9231
Rastgele Altuzay (100 , 100) orijinal algoritma	2.4231
Rastgele Altuzay (100 , 100) Ağırlıklandırılmış	2.3077

Tablolar incelendiğinde tüm algoritmaların tek bir ağaca göre çok daha başarılı oldukları görülmektedir. Başarı tabanlı budama hiç bir topluluk algoritması üzerinde hiçbir budama miktarıyla orijinal algoritmalarından daha yüksek başarı vermemiştir. Üstelik budama miktarı arttıkça topluluk başarısı da düşmektedir. Bununla birlikte budamadan en az etkilenen (ortalama başarı sırasında en az düşme olan) Rastgele Altuzaylar'dır. En çok etkilenen ise Bagging'dir. Temel öğrenicilerin kararlarının ağırlıklandırılmasının orijinal algoritmaya göre daha iyi sonuç verdiği tek topluluk algoritması Rastgele Altuzaylar'dır.

Tüm algoritmalar birlikte değerlendirildiklerinde elde edilen ortalama başarı sıralamaları Tablo 5'te verilmiştir.

Tablo 5: Genel sonuçlar

Algoritma	Ortalama Başarı Sırası
Tek ağaç (CART)	14.9231
Random forest (100 , 25)	8.3077
Random forest (100 , 50)	6.8846
Random forest (100 , 75)	6.7692
Random forest (100 , 100) orijinal algoritma	5.6154
Random forest (100 , 100) Ağırlıklandırılmış	6.6154
Bagging (100 , 25)	12.2308
Bagging (100 , 50)	10.9231
Bagging (100 , 75)	10.4231
Bagging (100 , 100) orijinal algoritma	9.1538
Bagging (100 , 100) Ağırlıklandırılmış	9.2692
Rastgele Altuzay (100 , 25)	9.5769
Rastgele Altuzay (100 , 50)	7.1923
Rastgele Altuzay (100 , 75)	6.5769
Rastgele Altuzay (100 , 100) orijinal algoritma	6.0000
Rastgele Altuzay (100 , 100) Ağırlıklandırılmış	5.5385

16 algoritmanın 26 veri kümesi üzerindeki ortalama başarılarını içeren Tablo 5 incelendiğinde, topluluk algoritmalarının kendi içinde incelendiğinde elde edilen sonuçlarla benzer sonuçlar görülmektedir. Bunlara ek olarak, genel olarak en başarılı algoritmanın Ağırlıklandırılmış Rastgele Altuzay olduğu görülmektedir. Ardından orijinal Random Forest gelmektedir. Orijinal topluluk algoritmaları karşılaştırıldığında ise algoritmalar iyiden kötüye Random forest, Rastgele Altuzay, Bagging şeklinde sıralanmaktadır.

7. YORUMLAR VE TARTIŞMA

Topluluk algoritmalarının başarısı birbirine karşıt olan iki parametreden etkilenmektedir. Bunlardan ilki topluluk üyelerinin (temel öğrencilerin) başarılarının yüksek olması diğeri ise üyelerin birbirlerinden farklı sonuçlar üretmeleridir. Bu çalışmada, toplulukları üyelerinin başarısı tabanlı budamanın ve üyelerin kararlarını yine topluluk üyelerinin başarılarını kullanarak ağırlıklandırma yöntemlerinin topluluk başarısı üzerindeki etkileri 3 topluluk algoritması üzerinde 26 veri kümesi ile incelenmiştir. Bu incelemelerden elde edilen bulgular ve yorumları aşağıda özetlenmiştir.

-Başarı tabanlı budama hiç bir topluluk algoritması üzerinde hiçbir budama miktarıyla daha iyi başarı vermemiştir. Üstelik budama miktarı arttıkça topluluk başarısı da düşmektedir. Temel öğrenci başarısı tabanlı budama topluluk içindeki en başarılı üyeleri seçtiğinden, topluluk içindeki temel öğrencilerin ortalama başarısını arttırmakta ancak bununla birlikte farklılığı o kadar azaltmaktadır ki topluluğun başarısı da düşmektedir. Sonuçlara göre 3 topluluk algoritması içinde üyelerinin başarıları tabanlı budama uygun bir işlem değildir.

-Budama işleminden en az etkilenen (ortalama başarı sırasında en az düşme olan) Rastgele Altuzaylar'dır. En çok etkilenen ise Bagging'dir. 3 topluluk algoritmasının üyelerinin (temel öğrencilerinin) kullandıkları bilgiler incelendiğinde eğitim kümesinden en az bilgi kullanan üyelerin (Bagging üyeleri, örneklerin tüm özelliklerine ağacın her düğümünde sahiptir. Random forest üyeleri her düğümde özelliklerin rastgele bir alt kümesine sahip olsalar da ağacın genelde

eğitim kümesinin tüm özelliklerine sahip olduğu düşünülebilir. Rastgele Altuzaylar'da ise üyeler eğitim kümesinin sadece bir bölümüne sahiptir.) Rastgele Altuzaylar'da olduğu görülmektedir. Bunun sonucu olarak Random Altuzaylar'ın üyelerinin sonuçlarının birbirinden farklılığı diğer topluluk algoritmalarına göre daha fazla olacaktır. Rastgele Altuzaylar'da yer alan özellik gruplarının bazılarının sınıflandırmaya yardımcı olmayabileceği (dolayısıyla yararsız) düşünüldüğünde Rastgele Altuzay algoritmasının diğer topluluk algoritmalarına göre üyelerinin tümüne daha az ihtiyacı olduğu görülmektedir. Üyelerinin tümüne en çok ihtiyacı olan topluluk algoritması ise (üyeleri en çok bilgi kullanan, dolayısıyla temel öğrencilerinin kullandıkları bilgilerin kesişimi büyük) Bagging'tir.

- Temel öğrencilerin kararlarının ağırlıklandırılmasının daha iyi sonuç verdiği tek topluluk algoritması Rastgele Altuzaylar'dır. Üyelerinin kararlarının birbirlerinden farklılığı en yüksek olan Rastgele Altuzaylar olduğundan bu sonuç beklenen bir sonuçtur. Tüm üyelerin başarılarının birbirine çok yakın olduğu bir durumda kararların ağırlıklandırılması etkin bir işlem olmayacaktır. Ağırlıklandırmanın Bagging üzerindeki etkisizliği bunu kanıtlamaktadır.

Sonuç olarak topluluk algoritmalarının temel öğrencileri arasında ürettikleri farklılıkların (diversity) büyüktür küçüğe sıralaması Random Forest, Rastgele Altuzaylar, Bagging şeklindedir. Bununla birlikte topluluk algoritmaları tasarımında hem ortalama başarıları yüksek hem de ayrıklıkları fazla olan temel öğrenciler üretmek yerine; başarıları çok yüksek olmayan ancak yüksek ayrıklığa sahip öğrenciler üretmek ve kararlarını ağırlıklandırarak birleştirmek etkin bir alternatif olarak gözükmektedir.

8. KAYNAKÇA

- [1] Brown, G., "Ensemble Learning", *Encyclopedia of Machine Learning*, Springer Press, 2010.
- [2] Kuncheva, L., *Combining Pattern Classifiers Methods and Algorithms*, Wiley-Interscience, 2004.
- [3] Alpaydın, E., *Introduction to Machine Learning*, 2. Baskı, MIT Press, 2010.
- [4] Partalas, I., Tsoumakas, G., Vlahavas, I., "An ensemble uncertainty aware measure for directed hill climbing ensemble pruning", *Machine Learning*, 81(3), 2010.
- [5] Breiman, L., "Bagging predictors", *Machine Learning*, 24(2), 1996.
- [6] Ho, T. K., "The random subspace method for constructing decision forests", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8), 1998.
- [7] Schapire, R. E., "The boosting approach to machine learning An overview", *MSRI Workshop on Nonlinear Estimation and Classification*, 2002
- [8] Breiman, L., "Random Forests", *Machine Learning*, 45(1), 2001.
- [9] Blake, C. L., Merz, C. J., *UCI repository of machine learning databases*, 1998.
- [10] Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J., *Classification and regression trees*. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software, 1984.
- [11] Alpaydın, E., "Combined 5 x 2 cv F test for comparing supervised classification learning algorithms", *Neural computation*, 11(8), 1999.