**DÜZCE ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BETİK DİLLER PROJE RAPORU**

**MELİSA AK**

**BAGGİNG REGRESSOR TEKNİĞİ**

**OCAK 2021**

**İÇİNDEKİLER DİZİNİ**

[ŞEKİLLER DİZİNİ 3](#_Toc61174668)

[BÖLÜM1 4](#_Toc61174669)

[GİRİŞ 4](#_Toc61174670)

[ENSEMBLE LEARNİNG 4](#_Toc61174671)

[BAGGİNG 4](#_Toc61174672)

[BÖLÜM 2 5](#_Toc61174673)

[BAGGİNG REGRESSOR 5](#_Toc61174674)

[Gerekli Kütüphanelerin Yüklenmesi 7](#_Toc61174675)

[Veri Setinin Yüklenmesi 7](#_Toc61174676)

[Veri Ön İşleme 8](#_Toc61174677)

[Veriler Eğitim İçin Hazırlanması 8](#_Toc61174678)

[Karar Ağacı Modeli Eğitilmesi 8](#_Toc61174679)

[Random Forest Modelinin Oluşturulması 9](#_Toc61174680)

[Bagging Modelinin Oluşturulması 9](#_Toc61174681)

[ROC Eğrisinin Çıkarılması 10](#_Toc61174682)

[Karar Ağacının ROC Eğrisinin Çıkarılması 10](#_Toc61174683)

[Rastgele Orman Ağacının ROC Eğrisinin Çıkarılması 11](#_Toc61174684)

[Bagging Regressor ROC Eğrisinin Çıkarılması 11](#_Toc61174685)

[BÖLÜM 3 12](#_Toc61174686)

[SONUÇ 12](#_Toc61174687)

# **ŞEKİLLER DİZİNİ**

[Şekil 1- Ensemble Learning 4](#_Toc61174692)

[Şekil 2-Bagging Regressor Blok Şeması 5](#_Toc61174693)

[Şekil 3- Akış Diyagramı 6](#_Toc61174694)

[Şekil 4- Veri Setindeki 1 ve 0'ların Durumu 7](#_Toc61174695)

[Şekil 5- ROC Eğrisi 10](#_Toc61174696)

[Şekil 6-Karar Ağacının ROC Eğrisi 10](#_Toc61174697)

[Şekil 7-Rastgele Orman Ağacının ROC Eğrisi 11](#_Toc61174698)

[Şekil 8-Bagging Regressor ROC Eğrisi 11](#_Toc61174699)

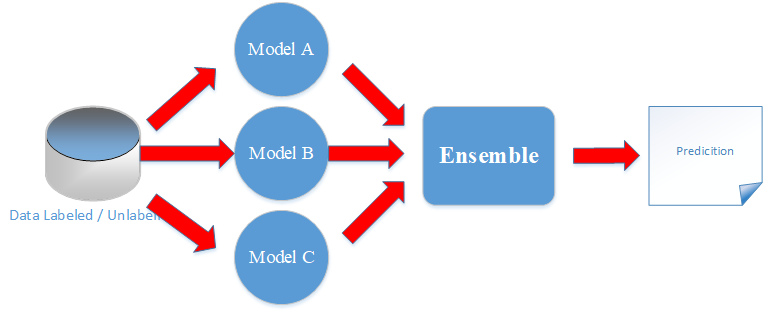
[Tablo 1-Veri Seti 7](#_Toc60582532)

# **BÖLÜM1**

# **GİRİŞ**

## **ENSEMBLE LEARNİNG**

Topluluk öğrenmesi, birden fazla modelin aynı sorunu çözmek için eğitildiği ve daha iyi sonuçlar elde etmek amacıyla birleştirildiği bir makine öğrenimi modelidir. Sınıflandırma algoritmaları ile bir verinin hangi sınıfa ait olduğunu tahmin edildiğinde, bunun için probleme en uygun sınıflandırma yöntemini seçilir, gerekli optimizasyonlar yapılır ve yüksek doğruluk oranı elde etmeye çalışılır. Ancak bazı veri setlerinde yüksek varyans ve düşük doğruluk değerine sahip sınıflandırmalar gerçekleşmektedir. Sınıflandırma ve tahmin işlemlerindeki bu soruna çözüm üretmek ve performans kaybını önlemek için farklı metotlar öne sürülmüştür. Bu metotlardan biri de topluluk öğrenme yöntemleridir.



Şekil 1- Ensemble Learning

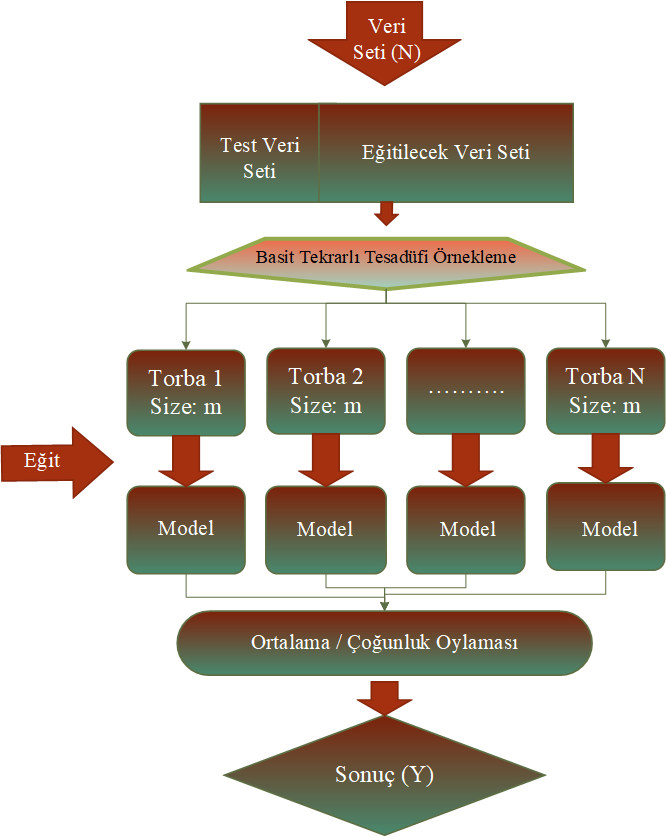
### **BAGGİNG**

Bagging, “bootstrap aggregation” ifadesinin kısaltılmışıdır. Bu ne anlama gelir? Yeniden örneklem yöntemi ile elde edilen ağaçların bir araya getirilmesi anlamına gelir. Bu açıdan tek ağaç yapılarına bambaşka bir bakış açısı getirmiş ve makine öğrenmesi dünyasını ileri taşımıştır. Bu yöntemde örnek veri seti içerisinden yerine koyacak şekilde tekrar tekrar örnekler çekilerek yeni ağaçlar oluşturulur ve bu ağaçlardan ortaya bir topluluk çıkar. Ağaç topluluğu. Bu sebeple topluluk öğrenme yöntemleri olarak anılır. Ortaya çıkan ağaçların hepsi modelleme için kullanılarak her birisine fikri sorulur. Her birisinin fikri değerlendirilerek bir oylama/ortalama fikir alınarak tahmin işleminin sonucu tek bir ifade olarak ortaya sunulur. Bagging bir metodoloji olarak görülmelidir.

# **BÖLÜM 2**

# **BAGGİNG REGRESSOR**

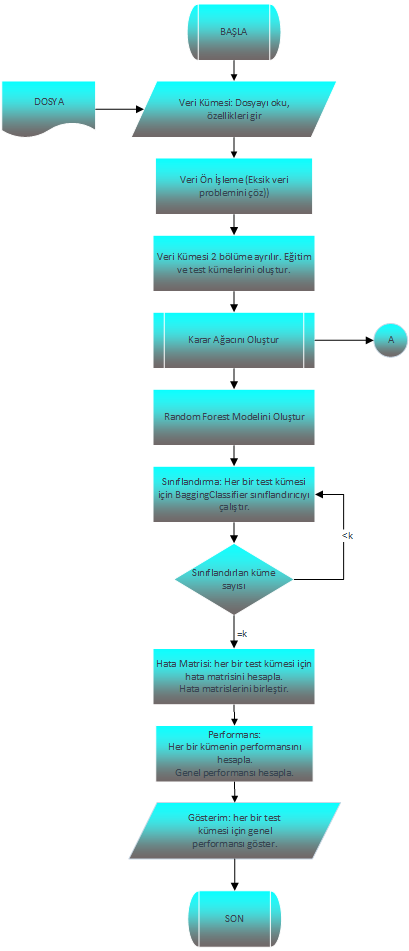
Bagging, var olan bir eğitim setinden yeni eğitim setleri türeterek temel öğrenciyi yeniden eğitmeyi amaçlayan bir yöntemdir.



Şekil 2-Bagging Regressor Blok Şeması

Şekil 2 ‘den görüldüğü üzere öncelikle mevcut veri seti eğitim ve test olarak ayrılır. Burada eğitilecek (n tane örnekten oluşan) veri setinden, basit tekrarlı tesadüfi örnekleme yoluyla torbalama (bagging) (1’den n’ e kadar) örneklem birimleri atanır ve ortaya bir topluluk çıkar.

Seçilen her örneklem tekrar eğitim setine geri koyulur. Bu durumda bazı eğitim örnekleri yeni eğitim kümesinde yer almazken bazıları birden fazla yer alır. Ortaya çıkan örneklem birimleri modelleme için kullanılır ve eğitilir. Her bir model değerlendirilerek regresyon problemi çözüyor ise ortalama, sınıflandırma problemi çözüyor ise çoğunluk oylaması sonuçlarına göre tahmin işleminin sonucu tek bir ifade ile ortaya çıkar. Bunun sonucunda gözlem ve değişken bazında rassallık sağlanır, aşırı örenme sorunu hafifler ve tahmin başarısı artar.



Şekil **3**- Akış Diyagramı

## **Gerekli Kütüphanelerin Yüklenmesi**

Bagging Regresyon için gerekli olan numpy, sklearn, seaborn ve pandas kütüphaneleri yüklenmiştir.

## **Veri Setinin Yüklenmesi**

Veri seti yüklenmiştir. Toplam 5 özellik ve 173913 gözlem mevcuttur. Bunlar; id\_assessment, id\_student, date\_submitted, is\_banked ve score’ dur. Burada hedef değişkeni Banka durumudur. Hedef değişkeni eğer öğrencinin bankada parası varsa 1, yoksa 0 olarak kodlanmıştır. Diğer 4 nitelik ise bağımsız değişkenlerdir. Sınıflandırma modeli bağımsız değişkenleri kullanarak bağımlı değişkeni (bankada para olup olmadığını) tahmin edecektir.

1

0

173911

173913

173912

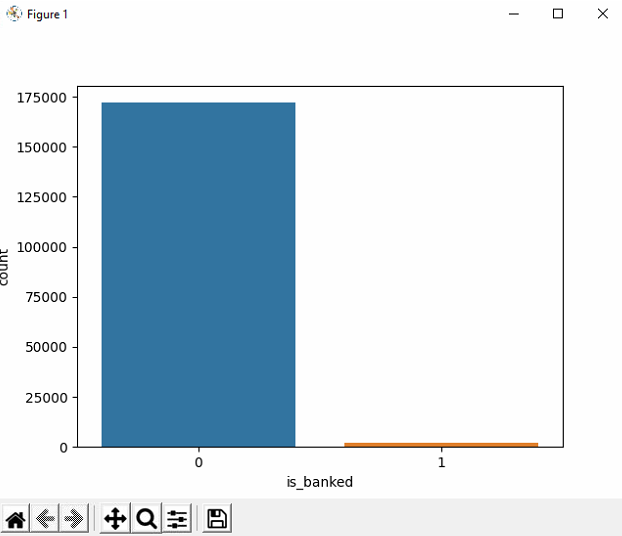
….

2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id\_assessment | id\_student | date\_submitted | Is\_banked | Score |
| 1752 | 11391 | 18 | 0 | 78 |
| 1752 | 28400 | 22 | 0 | 70 |
| 1752 | 31604 | 17 | 0 | 72 |
| .  . | .  . | .  . | .  . | .  . |
| 37443 | 546286 | 215 | 0 | 80 |
| 37443 | 546724 | 230 | 0 | 100 |
| 37443 | 558486 | 224 | 0 | 80 |

Tablo 1-Veri Seti

Veri seti incelenmiştir. Buna göre bankada parası olan 1909, parası olmayan 172003 kişi vardır.



Şekil **4**- Veri Setindeki 1 ve 0'ların Durumu

## **Veri Ön İşleme**

İnfo () komutu verinin ne türde olduğunu, kaç sütundan oluştuğu, değişkenlerinin ne tipte olduğu, veride boş yani NaN değer olup olmadığı hakkında bilgi verir.

Bu serinin türüne bakıldığında DataFrame olduğunu, 5 sütundan oluştuğunu, 173913 satırdan oluştuğunu bunlardan hiçbir satırın boş olmadığını görülmektedir.

Ayrıca score değişkenin bir object, diğer değişkenlerin ise integer değerler olduğu görülür.

## **Veriler Eğitim İçin Hazırlanması**

Veriler eğitim için hazırlanmıştır. Burada id\_assessment ve id\_student Değerleri bağımsız değişkenlerdir ve X’ atanmıştır. Banka durumu değişkeni de bağımlı değişkeni de bağımlı değişken olan y’ye atanmıştır.

Daha sonra veri seti eğitim ve test seti olarak ayrılmıştır. Yüzde 70’i eğitim seti yüzde 30’u da test seti olarak ayrılmıştır.

## **Karar Ağacı Modeli Eğitilmesi**

Bu adımda karar ağacı modeli eğitilmiştir ve karar ağacına doğru sınıflandırma oranı yaklaşık 0,992 olarak bulunmuştur.

[[51437 183]

[190 364]]

Confusion matrisine bakıldığında ise 52174 veri içerisinden 51801 tanesini doğru tahmin ettiği görülmektedir.

## **Random Forest Modelinin Oluşturulması**

Burada sınıflandırma için sklearn.enseble kütüphanesinin RandomForestClassifier sınıfı kullanılmıştır. Bu model parametre olarak n\_estimators alır ve bu parametre rastgele ormandaki ağaçların sayısını tanımlar.

Bir topluluk öğrenmesi yöntemi olarak Random Forest’ın 0.987 ile karar ağacına göre daha düşük bir doğruluk oranı yakaladığı görülür.

[[51425 195]

[479 75]]

Confusion matrisine bakıldığında ise 52174 veri içerisinden 51500 tanesini doğru tahmin ettiği görülmektedir.

## **Bagging Modelinin Oluşturulması**

Bagging modelinin test sonuçlarına bakıldığında yaklaşık 0. 991’lik bir doğruluk oranına sahip olduğu görülür.

[[51571 49]

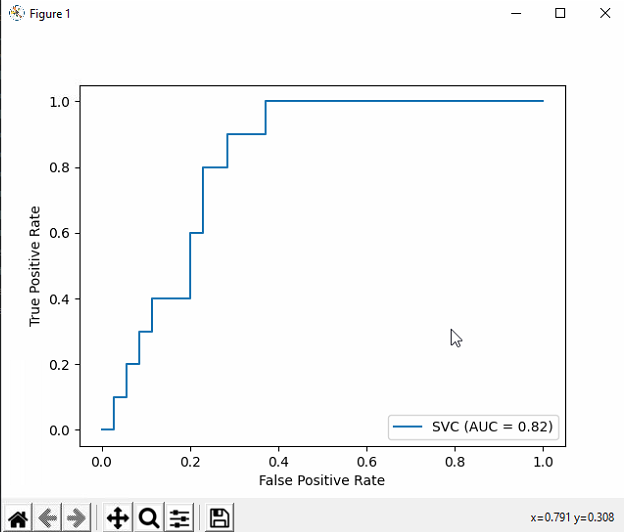
[408 146]]

Confusion matrisine bakıldığında ise 52174 veri içerisinden 51717 tanesini doğru tahmin ettiği görülmektedir.

## **ROC Eğrisinin Çıkarılması**

Bir sınıflandırma probleminin performansını değerlendirilmesi için AUC-ROC eğrisinden yararlanılır. Veri modelinin tahminde ne kadar iyi olduğunu açıklar. Tipik bir ROC eğrisinde X ekseninde yanlış pozitif oran (FPR) ve Y ekseninde gerçek pozitif oran (TPR) vardır.

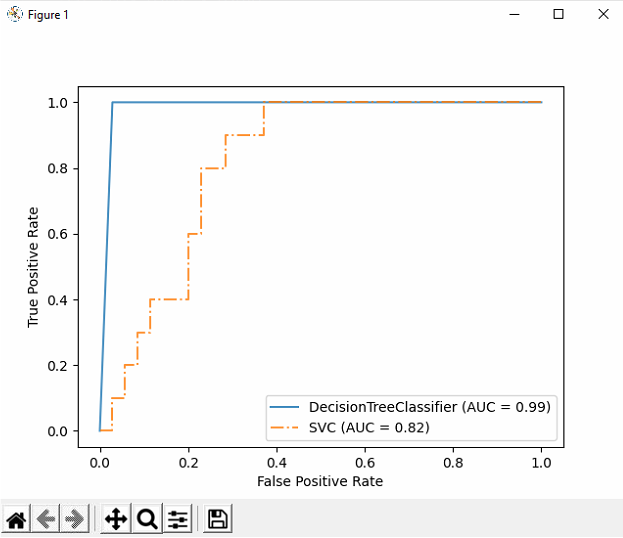
Test için ayrılan veriler ile ROC eğrisi oluşturulmuştur. AUC (Area Under Curve) değeri 0,82 olarak bulunmuştur.



Şekil 5- ROC Eğrisi

## **Karar Ağacının ROC Eğrisinin Çıkarılması**

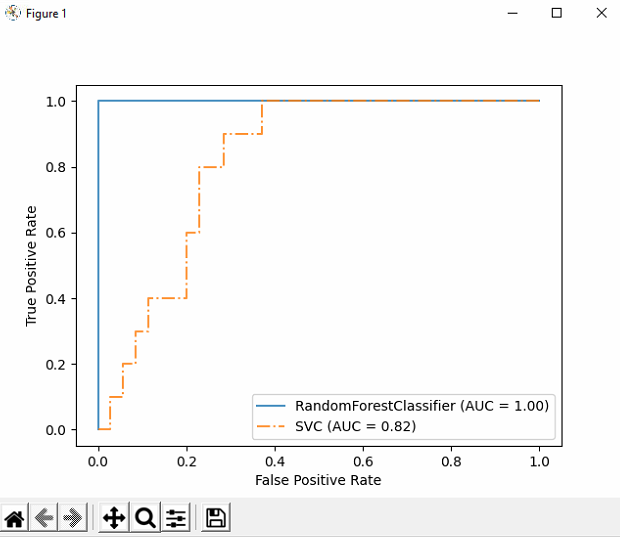
Karar ağacının ROC eğrisi oluşturulmuştur. Buna göre Karar ağacının performansı (AUC) .99 olarak bulunmuştur.



Şekil **6**-Karar Ağacının ROC Eğrisi

## **Rastgele Orman Ağacının ROC Eğrisinin Çıkarılması**

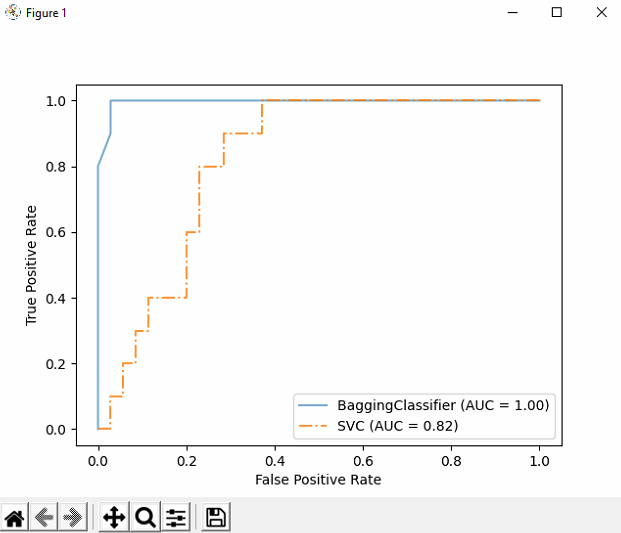
Rastgele orman ağacının ROC eğrisi oluşturulmuştur. Buna göre rastgele orman ağacının performansı (AUC) 1,00 olarak bulunmuştur.



Şekil 7-Rastgele Orman Ağacının ROC Eğrisi

## **Bagging Regressor ROC Eğrisinin Çıkarılması**

Bagging Regressor’un ROC eğrisi oluşturulmuştur. Buna göre Bagging Regressor’un performansı (AUC) 1,00 olarak bulunmuştur.



Şekil 8-Bagging Regressor ROC Eğrisi

# **BÖLÜM 3**

# **SONUÇ**

Bu çalışmada regresyon (regression) probleminin çözümüne yönelik topluluk öğrenme algoritmalarından biri olan torbalama (bagging) algoritması kullanılarak bir çalışma yapılmıştır. Ortaya konulan bulgular, regresyon probleminin torbalama (bagging) algoritması ile çok başarılı bir şekilde, diğer bir ifadeyle düşük hata oranlarıyla çözüme kavuşturulduğunu göstermektedir.

Öncelikle Bagging regressor için gerekli kütüphaneler eklendi, öğrencilerin parası olup olmaması durumuna göre bir sonuç çıkarıldı. Veri için ön işleme gerçekleşti bu sayede elimizdeki veri seti türünün dataframe olduğu görüldü. Veriler eğitim seti için hazırlandıktan sonra karar ağacı modeli eğitildi. Random forest ile karar ağacına göre daha düşük bir oran yakalandı. Elimizdeki sonuçlara göre bagging modeli oluşturuldu.

Test sonuçlarına bakıldığında yaklaşık 0. 991’lik bir doğruluk oranına sahip olduğu görüldü. Bu verilerden yola çıkarak Bagging regressor tekniğinin her zaman tam sonuç vermese bile ortalama bir sonuç verdiği görüldü. Bagging büyük veri setlerinde yapılan işlemleri kolaylaştırır. Bu yüzden tercih edilen bir problem çözme metotudur.