Karar Ağaçları Metot Tanıtımı ve Kredi Borcu Verisi Üzerinde Uygulamaları

Ezgi Cinkılıç, 201301012

Yapay Zeka Mühendisliği 4. Sınıf

YAP101: Veri Bilimine Giriş

5 Nisan 2025

İçindekiler

Özet

Karar ağaçları, veriyi en fazla bilgi kazandıran özniteliklerine göre dallara ayırarak hiyerarşik sınıflandırma veya regresyon yapan denetimli makine öğrenmesi modelidir. Modelin teorik yapısı entropi ve bilgi kazanımı gibi anlaşılır hesaplamalara dayanmaktadır. İç işleyişi ve karar verme sürecinin anlaşılması kolaydır. Sınıflandırma ve regresyon problemlerinde kullanılabilmeleri ve hem sayısal hem de kategorik verilerle uyumlu olmaları sayesinde yaygın olarak tercih edilir. Tek başlarına kullanılabildikleri gibi, topluluk yöntemlerinde yapı taşı olarak da kullanılmaktadır. Öznitelik seçiminin algoritma tarafından yapılması nedeniyle başka makine öğrenmesi modellerinde öznitelik seçiminde kullanılabilmektedir. Aşırı öğrenme riski, yüksek duyarlılık ve eksenlere hizalı bölme en büyük dezavantajlarıdır.

Bu projede, karar ağaçları ve regresyon ağaçlarının temel prensipleri, tarihsel gelişimi ve algoritmik yapısı incelenmiştir. Karar ağaçlarının hem kategorik hem de sayısal veriler üzerinde nasıl çalıştığı detaylandırılmış, öznitelik seçiminde bilgi kazancı ve Gini indeksi gibi metrikler açıklanmıştır. Pratik bir uygulama olarak, bir kredi veri seti üzerinde borç ödeme tahmini (sınıflandırma) ve kredi miktarı tahmini (regresyon) modelleri eğitilmiştir. Eğitilen modellerin performansları karşılaştırılmış ve yorumlanmıştır. Bu çalışma hem teorik altyapıyı hem de pratik uygulamaları bir araya getirerek, karar ağacı tabanlı modellerin avantaj ve sınırlılıklarını kapsamlı şekilde ele almaktadır.

GİRİŞ

Karar ağaçlarını seçmemin etmemin nedeni, teorik yapılarının sade ve anlaşılır olmasıdır. Entropi ve bilgi kazanımı gibi temel kavramlar kolaylıkla açıklanabilir ve model sonuçlarının nedenlerinin açık bir şekilde görülebilir. Ayrıca karar ağaçlarının kullanım alanı çok geniştir, hem sınıflandırma hem de regresyon problemlerinde yaygın olarak kullanılmaktadır. Eski bir tarihe sahip karar ağaçları Random Forest ve XGBoost gibi topluluk öğrenme yöntemlerinin de temelini oluşturmaktadır.

Karar ağaçları ilk olarak 1963 yılında Morgan ve Sonquist tarafından, sosyal koşulları etkileyen faktörleri analiz etmek amacıyla geliştirilmiştir. Klasik regresyon modellerinin yetersiz kaldığı durumlarda, karar ağaçlarının veriyi daha etkili gruplandırdığını ve açıklayıcılığını arttırdığını gösterilmiştir. Yöntemin teorik temeli, 1936'da Ronald Fisher'ın yayınladığı Diskriminant Analizi makalesine dayanmaktadır.[1] 1980'lerde Ross Quinlan tarafından geliştirilen ID3 ve ardından gelen C4.5 algoritmaları, karar ağaçlarının gelişiminde önemli rol oynamıştır. C4.5, sürekli ve eksik verilerle çalışabilme, budama ve özellik ağırlıklandırma gibi yetenekleriyle karar ağaçlarının veri madenciliği ve makine öğrenmesinde yaygınlaşmasını sağlamıştır. Aynı dönemde Breiman ve Stone tarafından geliştirilen CART algoritması da sınıflandırma ve regresyon problemlerine yönelik temel bir yaklaşım olarak öne çıkmıştır. Gini endeksine dayalı yapısıyla özellikle scikit-learn gibi kütüphanelerde yaygın kullanılmaktadır. [5] Zamanla Bagging (Torbalama) ve Boosting (Yükseltme) gibi topluluk öğrenme yöntemleriyle birlikte geliştirilerek, sınıflandırma ve regresyon problemlerinde daha güçlü modellerin oluşmasına zemin hazırlamıştır. Rastgele Orman ve XGBoost gibi metotların temelinde karar ağaçları vardır.

Karar ağaçları, kolay yorumlanabilir ve görselleştirilebilir olması, ön işleme gereksiniminin düşük olması, öznitelik seçiminin algoritma tarafından yapılması ve hem sayısal hem de kategorik verilerle çalışabilmesi gibi avantajlara sahiptir. Hızlı çalışması ve hesaplama verimli olması özellikle büyük veri setlerinde önemli bir avantaj sağlar. Ayrıca, insan kararlarına yakın tahminler sunması ve açıklanabilirliğinin yüksek olması tıbbi, finansal ve hukuki alanlarda tercih edilme nedenleri arasındadır. Dezavantajları ise aşırı öğrenme (overfitting) riskinin yüksek olması, dengesiz verilerde yanlılık gösterebilmesi ve küçük değişikliklere duyarlı olması, büyük verilerde bölünme kararlarının hesaplama maliyetinin yüksek olabilmesi, sadece eksenlere hizalanmış bölmeler yapabilmesi ve diğer regresyon veya sınıflandırma yöntemlerine kıyasla öğrenme gücünün ve tahmin doğruluğunun genellikle daha düşük olmasıdır. [2]

Metodun Ayrıntıları

Karar ağaçları, veriyi özelliklerine göre dallara ayırarak hiyerarşik sınıflandırma veya regresyon yapan bir makine öğrenmesi modelidir. Kök düğümü tüm veriyi temsil ederken, karar düğümleri veriyi alt düğümlere ayırır ve bölünmeyen düğümler yaprak düğümler olarak adlandırılır. Modelin aşırı öğrenmesini önlemek için budama işlemi uygulanabilir. Bir düğümün alt düğümleri varsa ebeveyn, alt düğümler ise çocuk düğümler olarak tanımlanır.

Modelin performansını belirleyen temel parametreler arasında maksimum derinlik, bir düğümün bölünebilmesi için gereken minimum örnek sayısı ve yaprak düğümde bulunması gereken minimum örnek sayısı yer alır. Ayrıca, maksimum yaprak ve öznitelik sayısı, modelin genelleştirme kapasitesini dengelemek için kullanılır.

En iyi ağacı bulmak NP-hard bir problem olduğundan, sezgisel yöntemler kullanılır. Aç gözlü böl ve fethet algoritması şu şekilde çalışır: Başlangıçta tüm eğitim verileri kök düğümde yer alır ve ağaç yukarıdan aşağıya, özyinelemeli olarak oluşturulur. Veriler, en iyi özellik temelinde dallara ayrılır; kategorik veriler doğrudan bölünür, sürekli veriler için uygun eşik değerler belirlenerek bölme yapılır. Bilgi kazancı, Gini katsayısı veya varyans azaltma gibi metrikler kullanılarak en iyi ayrımı sağlayan öznitelik seçilir ve düğüm oluşturulur. Ağaç büyütme, tüm örnekler aynı sınıfa ait olduğunda, kullanılabilecek öznitelik kalmadığında veya örnek kalmadığında durur. Bu süreç, karar ağacının öğrenme aşamasını oluşturur, yeni veri geldiğinde ilgili düğümler boyunca tahmin yapılır. [3]

Öznitelik seçimi, veri kümesinin en iyi şekilde bölünmesini sağlayacak özelliklerin belirlenmesidir. Kategorik veriler için entropi, veri kümesinin dağınıklığını, Gini katsayısı ise alt kümenin saflık derecesini belirtir. Veri saflaştıkça entropi ve Gini katsayısı azalmaktadır. Bilgi kazancı, entropi değişimini kullanarak bir özelliğin veri kümesini ne kadar iyi böldüğünü ölçer. En iyi ayrımı sağlayan özelliği belirlemek için bilgi kazancı hesaplanır ve en yüksek bilgi kazancına sahip özellik seçilir. Gini katsayısı CART, bilgi kazancı ise ID3 ve C4.5 algoritmalarında kullanılır. Sürekli veriler için ise, veri değerleri artan sırada sıralanır ve bitişik değerler arasındaki her nokta eşik değer olarak alınır. Bu eşik değerleri, bilgi kazancı veya kazanç oranı gibi metriklerle değerlendirilir ve en yüksek kazancı sağlayan eşik seçilir. Formüller Ek3'te yer almaktadır.

Aşırı öğrenme (overfitting) sorununu önlemek için karar ağaçlarında budama (pruning) işlemi yapılır. Budama, ağacın karmaşıklığını azaltarak genelleştirilebilirliği artırır ve iki ana yöntemi vardır. Ön budamada ağacın büyümesi tamamlanmadan durdurulurken, sonradan

budamada ağacın tamamen büyütülmesinin ardından gereksiz dallar kaldırılır. Sonradan budama genellikle doğrulama seti kullanılarak yapılır ve aşırı öğrenme riskini azaltır.

Regresyon ağaçları, sınıflandırma ağaçlarından farklı olarak yapraklarında sayısal değerler barındırır ve bu değer, o yapraktaki örneklerin ortalamasıdır. Ağaçların bölünmesi, yapraklar arasındaki varyansı en aza indirecek şekilde yapılır, böylece her bölme daha homojen ve doğru tahminler sağlar. Öznitelik seçimi her bölme için en yüksek standart sapmaya sahip özellikler seçilerek yapılır. Ağaç büyümesi, yapraklardaki varyans belirli bir eşik değerin altına inince veya veri sayısı belli bir değerin altına düşerse durdurulur. Son olarak, her yaprağa yazılan ortalama değer, o yaprağın tahmin değeri olarak kullanılır. [3]

Karar ağaçları hem sürekli hem de kategorik veriler üzerinde sınıflandırma yapabilen esnek bir yöntemdir. Kategorik değişkenler doğrudan kullanılabilirken, sürekli değişkenler belirli eşik değerlerine göre bölünerek modele dahil edilir. Regresyon ağaçları ise yalnızca sürekli değişkenler ile çalışabilir. Kategorik değişkenler, modelde kullanılmadan önce sayısal hale getirilmelidir. Ayrıca, regresyon ağaçlarının bağımlı değişkeninin mutlaka sayısal olması gerekmektedir.

ÖRNEK UYGULAMALAR

Bu çalışmanın temel amacı, bireylerin kredi borçlarını zamanında ödeyip ödemeyeceklerini tahmin etmek için karar ağacı kullanarak sınıflandırmak ve bireyin talep ettiği kredi miktarını regresyon ağacı kullanarak tahmin etmektir.

Veri

Veri seti, 2025 yılında Kaggle platformuna yüklenen ve makine öğrenmesi uygulamaları için sentetik olarak oluşturulmuş bir set olup kişisel bilgiler, kredi ve kredi geçmişi ve hedef değişken olan kredi geri ödeme durumundan oluşmaktadır. [4] Toplam 14 öznitelik bulunmaktadır. 8 öznitelik sayısal, 6 öznitelik kategoriktir. Veri setinde 45.000 satır veri bulunmaktadır. Boş değer bulunmamaktadır. Özniteliklerin açıklamaları ve veri yapıları Tablo 1'de yer almaktadır.

Tablo 1: Öznitelik Açıklamaları ve Veri Yapıları

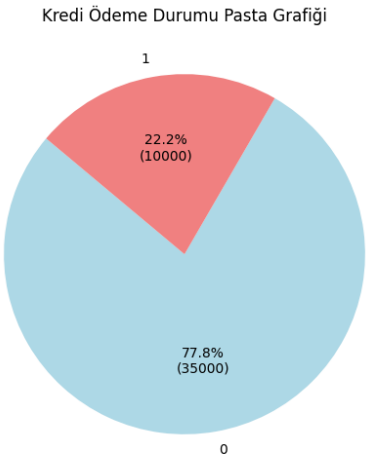
Veri Adı	Veri Yapısı
person_age (Kişinin yaşı)	Sayısal (Kesikli)
person_gender (Kişinin cinsiyeti)	İki sınıflı kategorik (Nominal)
person_education (Kişinin eğitim durumu)	Çok sınıflı kategorik (Ordinal)
person_income (Kişinin yıllık geliri)	Sayısal (Sürekli)

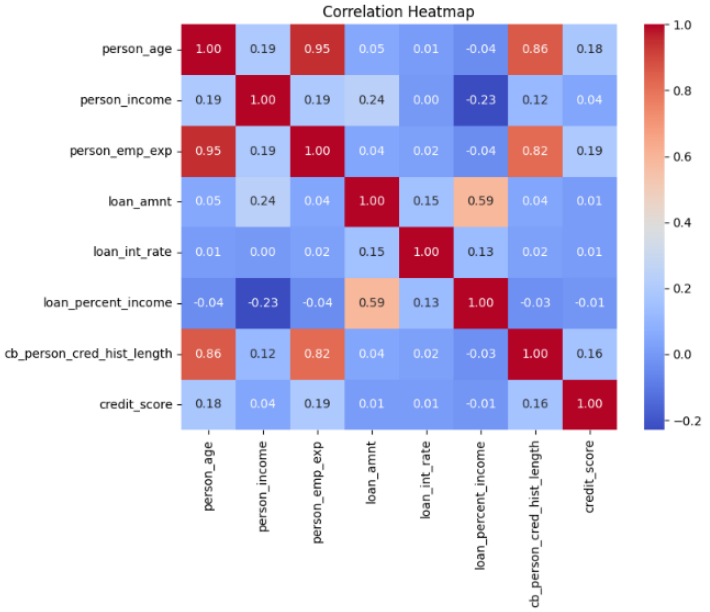
person_emp_exp (Kişinin toplam çalışma deneyimi)	Sayısal (Sürekli)
person_home_ownership (Kişinin ev sahipliği durumu)	Sayısal (Sürekli)
loan_amnt (Talep edilen kredi miktarı)	Sırasız çok sınıflı kategorik
loan_intent (Kredinin amacı)	Çok sınıflı kategorik (Nominal)
loan_int_rate (Kredi faiz oranı)	Sayısal (Sürekli)
loan_percent_income (Talep edilen kredinin gelire oranı)	Sayısal (Sürekli)
cb_person_cred_hist_length (Kredi geçmişinin uzunluğu)	Sayısal (Kesikli)
credit_score (Kredi skoru)	Sayısal (Sürekli)
previous_loan_defaults_on_file (Geçmiş Borç Bilgisi)	İki sınıflı kategorik (Nominal)
loan_status (Kredinin ödenip ödenmediği)	İki sınıflı kategorik (Nominal)

Veri Analizi

Hedef değişken, kredinin zamanında ödenip ödenmediği bilgisidir. Şekil1'deki pasta grafiğinde borcunu zamanında ödemeyenler 0, ödeyenler 1 ile gösterilmektedir. Zamanında ödemeyenlerin sayısı ödeyenlerden çoktur, bu da verilerin eşit dağılmadığını göstermektedir.

Şekil 1 ve 2. Kredi Geri Ödeme Durumu Pasta Grafiği ve Sayısal Öznitelikler Korelasyon Matrisi

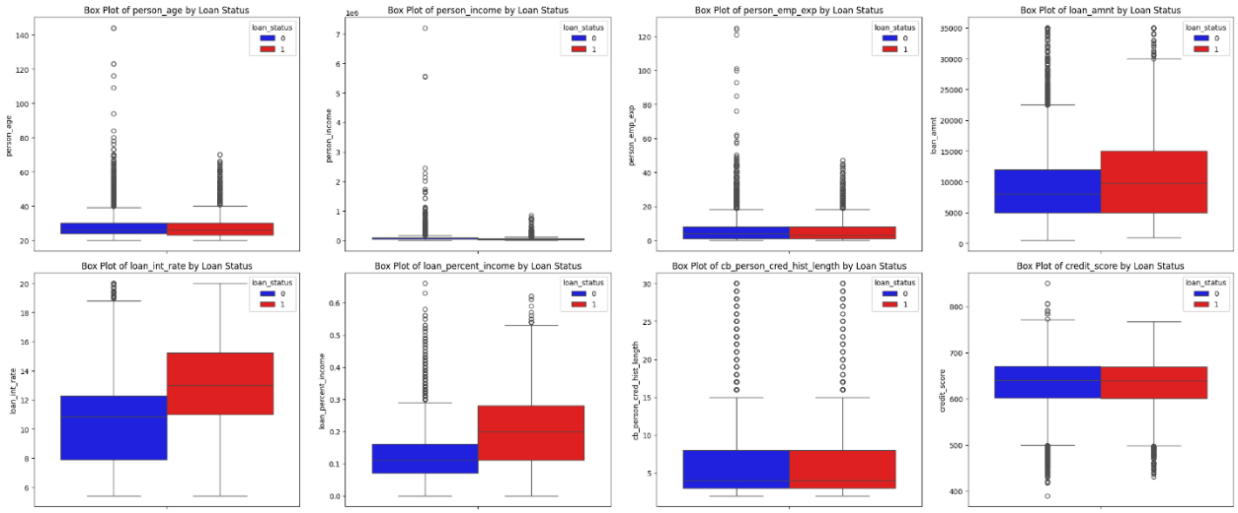




Şekil2'de yer alan sayısal özniteliklerin korelasyon matrisi incelendiğinde, kişinin yaşı ile çalışma deneyimi (0.95) ve kredi geçmişi (0.86) arasında güçlü pozitif korelasyonlar gözlemlenmiştir, bu da yaş arttıkça çalışma geçmişi ve kredi geçmişinin arttığını göstermektedir. Kredi miktarı ile kredinin gelire oranı arasındaki pozitif korelasyon (0.59), gelir arttıkça kredi

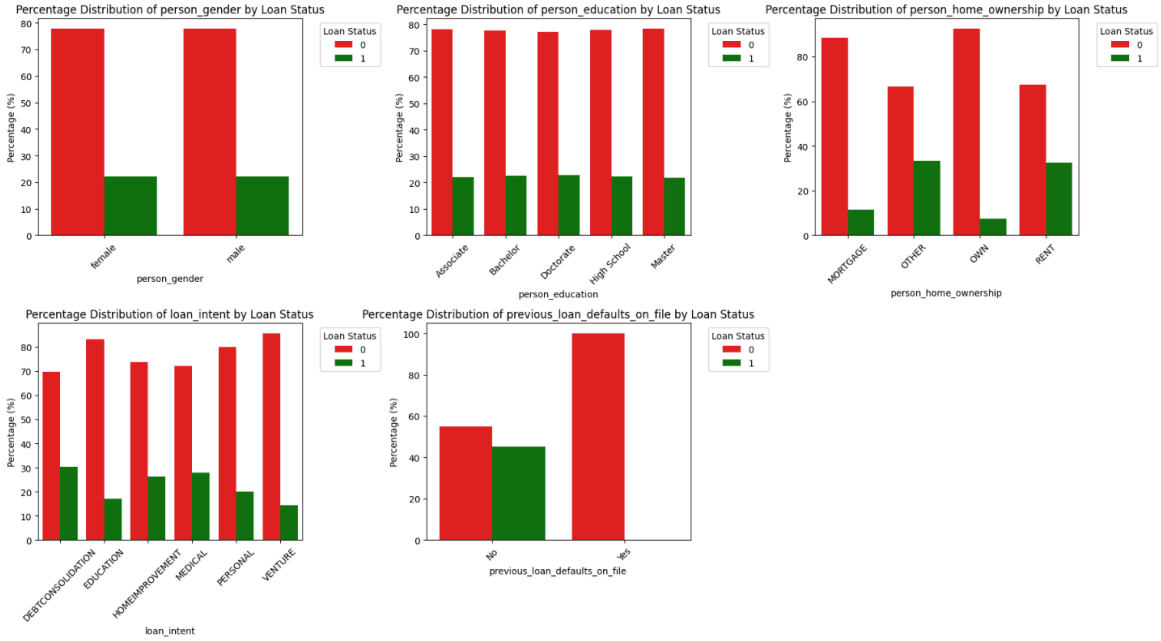
miktarının arttığını, gelir ile kredi oranı arasındaki negatif korelasyon (-0.23) ise yüksek gelirli bireylerin daha düşük kredi yükü taşıdığını ortaya koymaktadır. Kredi skoru ile diğer değişkenler arasında ise güçlü bir ilişki gözlemlenmemektedir. Histogram ve kredi geri ödeme durumu için kutu grafikleri (Şekil3) kullanılarak sayısal özniteliklerin dağılımları, uç değerler ve ayırıcı özellikler analiz edilmiştir. Yaş, 20 ile 144 arasında değişmekte olup, çoğunluk 24-30 yaş arasında yer almaktadır ve uç değerler fazladır. Kutu grafiği incelendiğinde, yaşların çakışması nedeniyle ayırıcı bir özellik olmadığı söylenebilir. Gelir ve çalışma yılı özelliklerinde de uç değerler bulunmakta olup, bu özellikler de ayırıcı özellikler olarak değerlendirilememektedir. Kredi faizi ortalama %11 olup, kredi skorları normal dağılıma yakın bir şekilde dağılmıştır. Kredi faiz oranı ve kredinin gelire oranı ise kutu grafiklerinden hareketle ayırıcı özellikler olarak değerlendirilebilir.

Şekil 3. Sayısal Özniteliklerin Kutu Grafikleri



Kategorik verilerin sütun grafikleri incelendiğinde (Şekil4), cinsiyet ve geçmiş ödenmemiş kredi borcu bilgilerinin uniform dağılıma yakın olduğu gözlemlenmiştir. Ev sahipliği durumu çoğunlukla kirada oturanlar ve mortgage ile ev sahibi olanlardan oluşmaktadır. Kişilerin çoğunluğu lise, ön lisans ve lisans mezunudur. Kredi ödeme durumu için ayrı çizilen grafiklerde, geçmiş ödenmemiş kredi bilgisinin en etkili ayırıcı olduğu görülmektedir. Kredi ödeme durumu dengesiz dağıldığı için birimler yüzdeye çevrilmiş ve ev sahipliği durumunun etkisi yüksek bulunmuştur. Eğitim seviyeleri ise ayırıcı özellik açısından etkili değildir. Kategorik özniteliklerde veri dönüşümü yapılmadan önce, her bir kategorinin eşsiz değerleri incelenmiştir. Cinsiyet, eğitim durumu ve geçmiş ödenmemiş kredi bilgisi Label Encoding ile dönüştürülürken, ev sahipliği durumu ve kredi çekme amacı One-Hot Encoding yöntemiyle dönüştürülmüştür.

Şekil 4. Kategorik Veriler İçin Kredi Ödeme Durumuna Göre Yüzde Sütun Grafikleri



Karar Ağacı ile Sınıflandırma Uygulaması

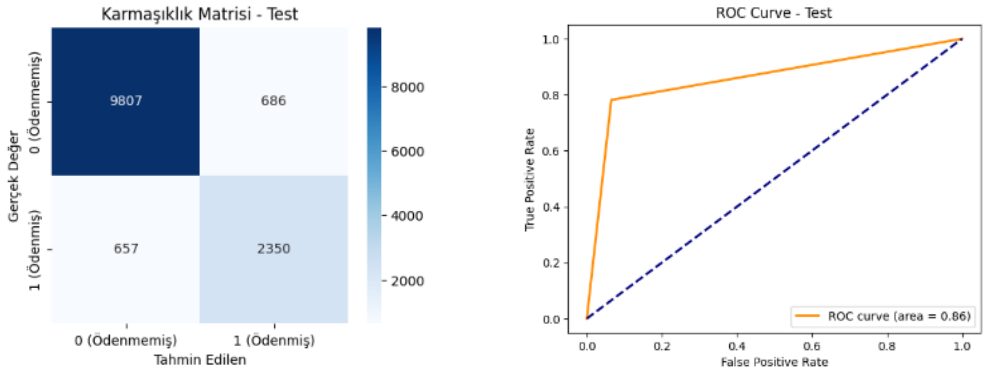
Bu çalışmada, Scikit-Learn kütüphanesi kullanılarak karar ağacı eğitilmiş ve kredi başvurusunda bulunan bireylerin borçlarını zamanında ödeyip ödemeyeceği tahmin edilmiştir. Veri seti eğitim/test verisi olarak %70- %30 oranında bölünmüştür. Üç uygulama gerçekleştirilmiştir. İlk uygulamada aşırı öğrenme eğilimini gözlemlemek amacıyla parametreler üzerinde kısıtlama yapılmadan model eğitilmiştir. Eğitim verisini ezberlediği için genelleme yeteneği düşük, modelin ayrım gücü zayıftır. İkinci aşamada, modelin genelleme yeteneğini artırmak amacıyla en uygun ağaç derinliği aranmıştır. Ağacın maksimum derinliği 10 ile kısıtlanarak model eğitilmiştir. Bu modelde aşırı öğrenmeye rastlanmamıştır, performans metrikleri ve model ayrım gücü iyileşmiştir. Üçüncü ve son aşamada, Rastgele Arama yöntemi kullanılarak hiperparametre optimizasyonu yapılmıştır. Bir düğümün bölünebilmesi için gereken minimum örnek sayısı 42, ağacın maksimum derinliği 25 ve maksimum yaprak sayısı 130 olarak belirlenmiştir. Modelin performansını güvenilir bir şekilde değerlendirebilmek için 3 katlı çapraz doğrulama uygulanmış ve elde edilen sonuçlar Tablo2'de karşılaştırılmıştır. Kullanılan performans metrikleri doğruluk, F1 skoru, kesinlik, duyarlılık ve ROC eğrisinin altında kalan alandır (AUC). Örnek karar ağacı Ek1'de yer almaktadır.

Sonuç olarak, en iyi performans, Rastgele Arama ile optimize edilen modelde elde edilmiştir. Yapılan çalışmalar sonucunda parametre seçiminin aşırı öğrenmeyi engellediği ve modelin genelleme kapasitesini arttırdığı görülmüştür. Eğitim ve test verisinin birbirine yakın olması ve yüksek performans metriklerine sahip olması modelin başarılı olduğunu göstermektedir.

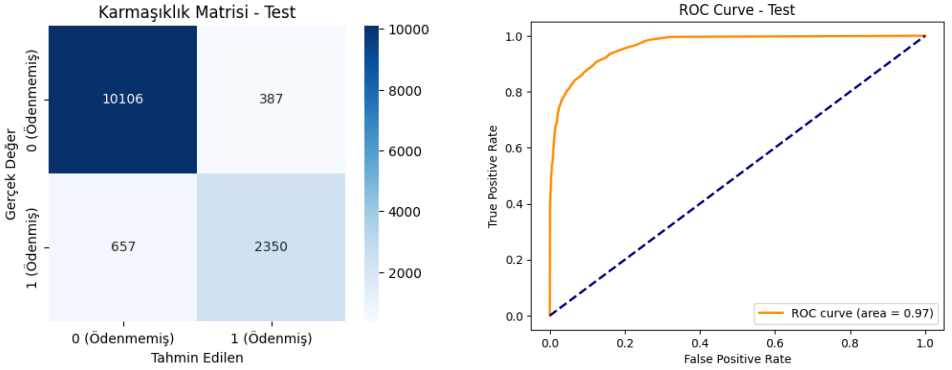
Tablo 2: Karar Ağacı Uygulamaları Performans Metrikleri

	1. Uygulama		2. Uygulama		3. Uygulama	
	Test	Eğitim	Test	Eğitim	Test	Eğitim
Doğruluk	0.90051	1	0.91940	0.93295	0.92266	0.93660
F1	0.90068	1	0.91684	0.93050	0.92134	0.93536
Kesinlik	0.90086	1	0.91765	0.93255	0.92105	0.93550
Duyarlılık	0.90051	1	0.91940	0.93295	0.92266	0.93660

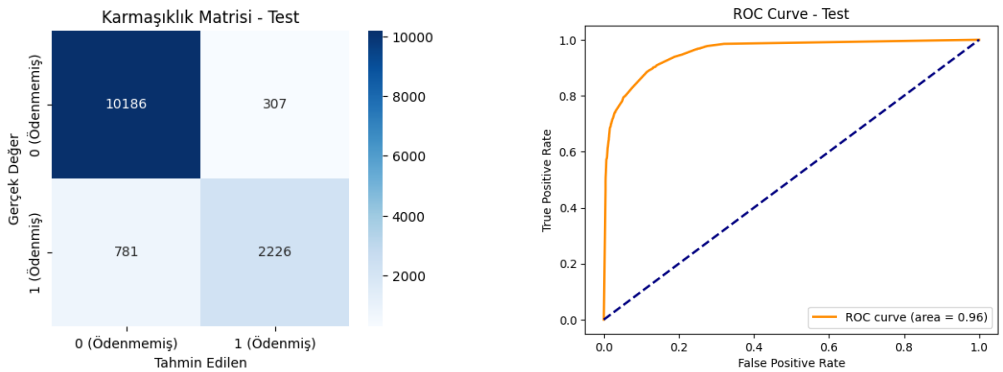
Şekil 5 ve 6. Karar Ağacı 1. Uygulama Test Verisi İçin Karmaşıklık Matrisi ve ROC Eğrisi



Şekil 7 ve 8. Karar Ağacı 2. Uygulama Test Verisi İçin Karmaşıklık Matrisi ve ROC Eğrisi



Şekil 9 ve 10. Karar Ağacı 3. Uygulama Test Verisi İçin Karmaşıklık Matrisi ve ROC Eğrisi



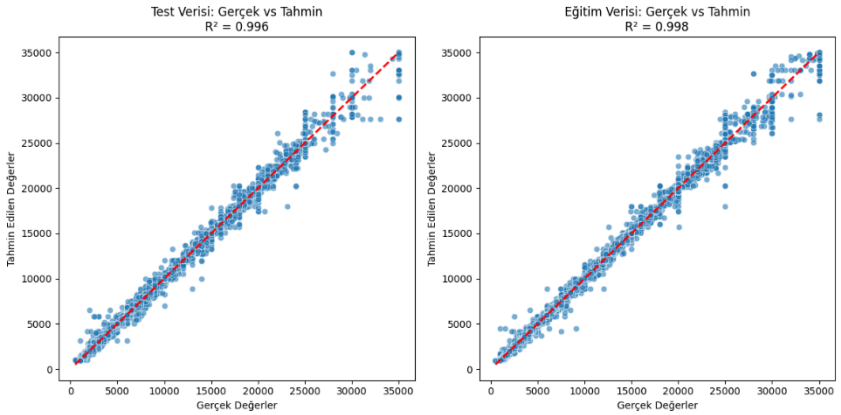
Karar ağacı analizi sonucunda özelliklerin önem dereceleri belirlenmiş olup, bu değerler her bir özelliğin sınıflandırma performansına olan katkısını ortaya koymaktadır. Yapılan değerlendirmede en yüksek ayırt edici güce sahip beş özellik müşterinin daha önce kredi temerrüdüne düşüp düşmediği, talep edilen kredi tutarının yıllık gelire oranı, kredi faiz oranı, müşterinin yıllık gelir düzeyi ve konut durumunu gösteren kirada oturma bilgisi. Bu özelliklerin modelin karar verme sürecinde en belirleyici faktörler olduğu görülmektedir. Öznitelik önem derecelerinin belirlenmesi diğer makine öğrenmesi modelleri için öznitelik seçimi ve veri boyutunun azaltılması gibi amaçlarla etkili bir şekilde kullanılabilir.

Regresyon Ağacı

Bu çalışmada, bireylerin çektikleri kredi miktarını tahmin etmek amacıyla Scikit-Learn kütüphanesindeki DecisionTreeRegressor modeli kullanılmıştır. Modelin doğruluğunu artırmak ve aşırı öğrenmeyi önlemek için Rastgele Arama yöntemiyle hiperparametre optimizasyonu yapılmıştır. Veri seti, regresyon problemine uygun hale getirilerek, yeni hedef değişken (talep edilen kredi miktarı) doğrultusunda %70- %30 oranında eğitim ve test setlerine ayrılmıştır. Modelin performansı, ortalama kare hatası (MSE), kök ortalama kare hatası (RMSE), ortalama mutlak hata (MAE), R-kare ve açıklanan varyans gibi metriklerle değerlendirilmiştir. Örnek regresyon ağacı Ek2'de yer almaktadır.

Model hem eğitim hem de test verisi üzerinde iyi performans sergilemektedir. R^2 ve açıklanan varyans değerlerinin 0.99'un üzerinde olması, modelin genelleme yeteneğinin güçlü olduğunu ve aşırı öğrenme probleminin olmadığını gösterir. MSE değerinin büyük olması metrik büyük hataları daha fazla cezalandırdığı için test verisindeki uç değerlerden ve normalizasyon yapılmamasından kaynaklanmaktadır. Elde edilen performans metrikleri Tablo3'te yer almaktadır.

Şekil 11 ve 12. Regresyon Ağacı Gerçek-Tahmin Kredi Miktarları ve Regresyon Çizgisi



Tablo 3: Regresyon Ağacı Uygulaması Performans Metrikleri

	Test	Eğitim
MSE	154138	78974
RMSE	392.604	281.023
MAE	187.460	121.802
R^2	0.996183	0.998008
Açıklanan Varyans	0.996183	0.9980086

SONUÇLAR

Tüm sonuçlar incelendiğinde, seçilen veri setinin ayrıştırılabilir olduğu ve karar ağacında modelin ayrıştırıcı gücünün %97 gibi yüksek bir değerle başarılı olduğu görülmüştür. Regresyon ağacında ise R^2 kare değeri %99'un üzerinde olarak mükemmele yakın bir performans sergilemiştir. Bu sonuçlar, veri setinin doğrusal olarak ayrıştırılabildiğini ve karar ağacı ile regresyon ağacının yüksek performans gösterdiğini göstermektedir. Diğer sınıflandırma ve regresyon metotlarının da benzer sonuçlar vermesi beklenmektedir. Ancak, daha karmaşık ve doğrusal ayrıştırılamayan veri setlerinde, eğitim performansı düşük çıkabilirdi. Bu durumda, öğrenmeyi artırmak için daha karmaşık modellere geçiş yapılabilir.

Karar ağaçları zaman ve hafıza karmaşıklığı açısından orta düzeyde performans gösterir. Küçük ve orta ölçekli veri setlerinde hızlı ve etkili bir seçenek iken, büyük veri setlerinde eğitim süresi ve bellek kullanımı artar, tahmin süresi ise hızlıdır. Karşılaştırma yapmak gerekirse, lojistik regresyon daha hızlı çalışırken ve daha az bellek kullanırken, SVM, KNN, YSA gibi metotlar daha yavaş çalışır ve daha fazla bellek kullanır. Eğitim ve tahmin için zaman ve hafıza karmaşıklığı:

- Eğitim zaman karmaşıklığı $O(m * n * \log n)$
- Tahmin zaman karmaşıklığı $O(d)$
- Eğitim hafıza karmaşıklığı $O(m * n)$
- Model saklanması hafıza Karmaşıklığı $O(2d)$

Sonraki çalışmalarda, modelin öğrenme kapasitesini artırmak için öznitelik seçimi ve özellik mühendisliği yöntemleri uygulanabilir. Ayrıca, sınıflar arası dengesizliği gidermek için veri dengeleme teknikleri kullanılabilir, bu sayede modelin her sınıfı eşit derecede öğrenmesi sağlanabilir. Bunun yanında, topluluk yöntemlerinden bagging ve boosting gibi yaklaşımlar denenerek modelin performansına katkısı gözlemlenebilir.

KAYNAKÇA

[1] Clark, W. A. V., & Deurloo, M. C. (2005). Categorical Modeling/Automatic Interaction Detection. Encyclopedia of Social Measurement.

[2] Protopapas, P., Rader, K., & Pan, W. (n.d.). CS 109A/AC 209A/STAT 121A Data Science [Ders Materyali]. Harvard University.

[3] Liu, B. (n.d.). CS 583: Data Mining and Text Mining [Ders Materyali]. University of Illinois at Chicago.

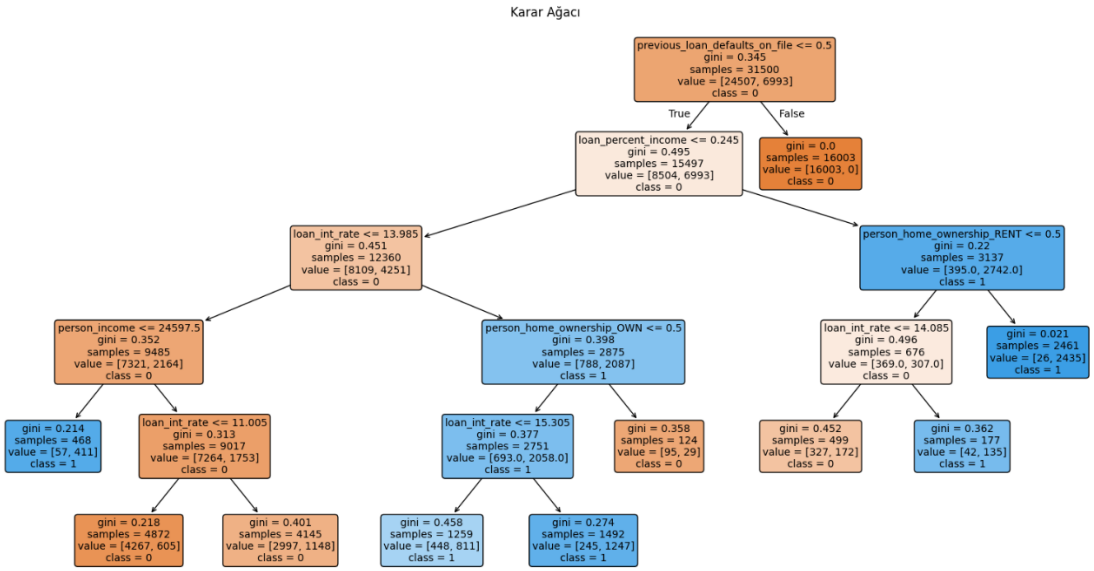
[4] Uday. (2025). bank_loan_data. Kaggle. [Online] [Erişim Tarihi: 20.03.2025] https://www.kaggle.com/datasets/udaymalviya/bank-loan-data/data

[5] Akca, M. F. (2020). Karar Ağaçları (Makine Öğrenmesi Serisi-3). Deep Learning Türkiye. [Online] [Erişim Tarihi: 15.03.2025] https://medium.com/deep-learning-turkiye/karar-a%C4%9Fa%C3%A7lar%C4%B1-makine-%C3%B6%C4%9Frenmesi-serisi-3-a03f3ff00ba5

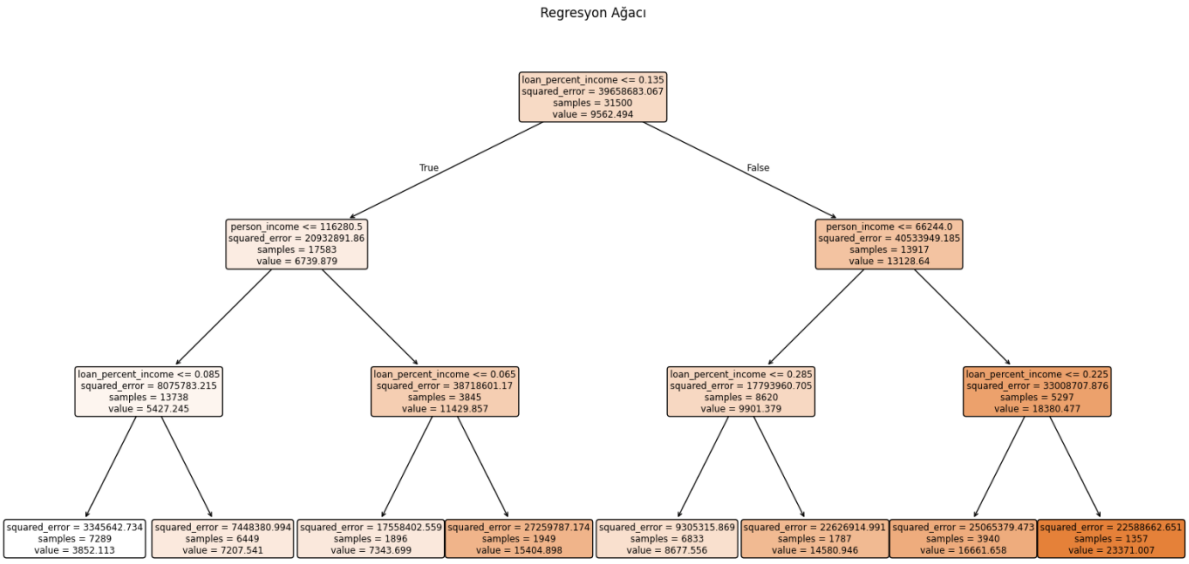
Kodlar aşağıdaki bağlantıya sahip Colab dosyasında yer almaktadır: https://colab.research.google.com/drive/1mxeH66uB6il4szgvVK6Q3HYDw0jbjLmt?usp=sharing

EKLER

Ek1: Örnek 6 Derinlikli Karar Ağacı



Ek2: Örnek 4 Derinlikli Regresyon Ağacı



Ek3: Entropi, Gini Katsayısı ve Bilgi Kazanımı Formülleri

$$H(S) = -\sum_{i=1}^{c} P_i \log_2 P_i$$

$$Gini = 1 - \sum_{j} p_j^2$$

$$IG(S, A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v)$$

$H(S)$, başlangıçtaki entropidir.

A, belirlenen özellik (değişken).

S_v, özellik A'nın belirli bir değerine sahip olan alt veri kümesi.

$H(S_v)$, alt veri kümesinin entropisidir.