**20.05.2023**

**VERİ MADENCİLİĞİ YÖNTEMLERİYLE MÜŞTERİ KREDİ RİSKİNİN ANALİZİ**
**Veri Madenciliği Projesi Raporu**

***İshak KUTLU***
***Bilişim Enstitüsü / Bilgisayar Bilimleri***

## Özet

Finans alanında kredi riski değerlendirmesi, günümüzde giderek daha fazla önem kazanmaktadır. Finansal kuruluşların elinde, müşterilerin kredi riskini değerlendirmede kullanılabilecek önemli miktarlarda veriler bulunmaktadır. Ancak söz konusu verileri algoritmaların işleyebilmesi ve içlerinden faydalı bilginin (knowledge) çıkartılması sürecinde algoritmaların verimliliğinin arttırılabilmesi için veri madenciliği yöntemlerine ihtiyaç vardır. Veri temizleme, gürültü giderme, özellik seçimi, özellik çıkarımı, veri dengeleme, normalizasyon gibi veri madenciliği teknikleri, makine öğrenmesi yöntemlerinin çalışma ve sınıflandırma performansını arttırmada ve modellerin genelleme yapabilme kabiliyetini yükseltmede kilit rol oynamaktadır. Bu çerçevede müşterilerin kredi riskini sınıflandırmak amacıyla, veri madenciliği tekniklerinin uygulanmadığı ve uygulandığı veri setleri XGBoost, karar ağacı, rassal orman, extra ağaç, naive Bayes, KNN ve lojistik regresyon algoritmaları kullanılarak karşılaştırmalı olarak analiz edilmiştir. Yapılan deneysel çalışmada, veri madenciliği tekniklerinin uygulandığı senaryoda makine öğrenmesi algoritmalarının, daha az sayıda özellik kullanarak daha iyi sınıflandırma performansına ulaşmasının mümkün olup olamayacağı araştırılmıştır.

## 1. Giriş

Veri madenciliği algoritmaları, genellikle veri bütünlüğünün iyileştirilmesine, veri kalabalığının (gürültünün) azaltılmasına ve özellikler (feature/attribute) arasında korelasyonun en az olmasına odaklanır. Ticari bankalar, çok sayıda kredi kartı işlem verisine ve müşterilerine ilişkin temel verilere sahiptirler. Bu yüzden veri madenciliği yöntemleriyle söz konusu veriden değer yaratma süreci, modern bankaların gelişimi açısından önemli bir konu haline geldi. Ancak söz konusu gerçek dünya verileri genellikle gürültülü, düzensiz, eksik ve tutarsızdır. Bu yüzden veri madenciliği algoritmalarında onları doğrudan işlemek zordur. Gerçek dünya verilerinde gürültü gibi veri madenciliği algoritmalarının verimliliğini ciddi şekilde etkileyen çok sayıda önemsiz bilgi vardır. Bu nedenle ham veriler veri ön işleme yoluyla algoritmaların verimli bir şekilde işleyebilecekleri formatlara dönüştürülmelidir. Veri ön işleme, veri madenciliği için vazgeçilmezdir. İstatistikler, veri ön işlemenin veri madenciliği sürecinde zamanın yüzde 60'ını aldığını göstermektedir [1].

Bu çalışmada, müşteri segmantasyonu olarak da adlandırılan müşterilerin kredi riski, veri madenciliği yöntemleriyle araştırılmıştır. Açık kaynak bir veri platformu olan Kaggle'dan elde edilen veri setinde, müşterilerin kredi riski, iyi ve kötü şeklinde iki sınıftan oluşmaktadır. Söz konusu veri seti, Python programlama dili kullanılarak veri temizleme, dönüştürme, gürültü tespiti, özellik ve örnek seçimi, boyut indirgeme, özellik çıkarımı gibi veri ön işleme teknikleriyle modelleme aşamasında kullanılmaya hazır hale getirilecektir.

Müşterilerin kredi riskini etkileyebilen kredinin kullanım amacı, vadesi, müşterinin tasarruf durumu, geçmiş kredi kullanımı, konut sahipliği ve çek kullanımı gibi faktörler olabilir. Söz konusu özellikler veri madenciliği teknikleriyle analiz edilerek bunların müşteri kredi riski sınıfları üzerindeki etkileri değerlendirilecektir. Performans ölçüm kriterleri olarak "accuracy", "precision", "recall" ve "f1 score" metrikleri kullanılacaktır.

Tahmin modeli olarak, makine öğrenmesi (machine learning, ML) yöntemlerinden XGBoost, karar ağacı, rassal orman, extra ağaç, naive Bayes, KNN ve lojistik regresyon kullanılacaktır. Çalışmada boyut indirgeme

(reduce dimesion), özellik seçimi (feature selection), özellik çıkarımı (feature extraction) ve veri dengeleme (data balancing) yöntemlerinin kredi risklerini sınıflandırma üzerindeki etkileri değerlendirilecektir. Bu amaçla üç aşamalı bir süreç takip edilecektir. İlk aşamada algoritmaların verileri işleyebileceği asgari seviyede veri madenciliği teknikleri kullanılacaktır. İkinci aşamada ise gelişmiş veri madenciliği teknikleri kullanılarak modelleme yapılacaktır. Üçüncü aşamada ise her iki yöntemin sınıflandırma performansları, kullanılan veri madenciliği yöntemleri çerçevesinde değerlendirilecektir.

## 2. LİTERATÜR TARAMASI

Wang ve arkadaşları gelir seviyesi, aile üyesi sayısı, araç sahipliği vb. kriterlere göre yapılan geleneksel kredi puanlama analizinin büyük ölçekli iş ihtiyaçlarını karşılayamaması, subjektif olması ve yanıltmaya/hileye eğilimli olması gibi gerekçelerle verimsiz olduğunu savunur. Bunun yerine mevcut büyük veri çağında, kullanıcıların kişisel özellikleri ve sosyal kayıtları gibi verileri toplayarak, içerdiği kredi risklerinin özelliklerini yakamayı ve kullanıcıların gelecekteki temerrüt riskleri için makine öğrenimi algoritmalarına dayalı bir sınıflandırma modeli oluşturmayı amaçlar. Bu çerçevede kredi puanlama (risk) analizi için makine öğreniminde yaygın olarak kullanılan naive Bayes, lojistik regresyon, rassal orman, karar ağacı ve KNN algoritmalarının sınıflandırma performanslarının değerlendirilmesine odaklanılmıştır. Her sınıflandırıcının kendine göre güçlü ve zayıf yönleri olduğu belirtilmişitir. Bununla birlikte uygulama sonuçları, rassal ormanın precision, recall, eğri altındaki alanı işaret eden AUC (area under curve) ve accuracy açısından diğerlerinden daha iyi performansa sahip olduğunu göstermiştir [2].

Ziemba ve diğerleri kredi risk değerlendirmesi probleminde özellik seçme (feature selection) yöntemleriyle desteklenen sınıflandırıcıları analiz etmeye odaklanmıştır. Bir eğitim veri seti üzerinde sınıflandırıcı seçimi ve özellik seçme yöntemleri kullanmanın etkisi araştırılmış, yeni müşterileri sınıflandırmaya ve onlara kredi verip vermeme konuusnda karar vermeye yardımcı olan bir karar destek modeli oluşturulmuştur. Çoğu sınıflandırıcıda özellik seçimi, sınıflandırma sonuçlarının bir miktar bozulmasına neden olmuştur. Özellik seçiminde yararlanılan araçlardan CFS (correlation-based feature selection), SA (significance attribute), SU (symmetrical uncertainty), SUFCBF (fast correlation-based filter) gibi çeşitli filtreleme seçim yöntemleri arasında en iyi sınıflandırma sonuçlarını sunmuştur [3].

Ruyu ve arkadaşları, bir Amerikan P2P platformu olan Lending Club'ın kredi skorlamasıyla ilgili veri setinini kullanarak lojistik regresyon, naive Bayes, karar ağacı ve destek vektör makinesi algoritmalarının farklı açılardan performanslarını araştırmışlardır. Naive Bayes ve karar ağacının sınıflandırma doğruluğunun, lojistik regresyon ve destek vektör makinesinin doğruluğundan %40 daha yüksek olduğu; naive Bayes'in ve karar ağacının AUC değerinin, diğer iki algoritmadan %45 daha yüksek olduğu tespit edilmiştir. Ancak çalışmada farklı algoritmaların verimliliğini karşılaştırmak için farklı veri kümelerinin kullanılabileceği de belirtilmiştir. Diğer taraftan özellik seçimi için PCA gibi yöntemler uygulanmasının önemi üzerinde durulmuştur. Çoklu doğrusallık ve aşırı öğrenme gibi problemlerden kaçınmak için boyut indirgeme yaklaşımı önerilmiştir [4].

Yan-li ve Jia müşteri segmentasyonu, ilişkilendirme analizi ve risk tespiti alanında kredi kartı verilerinin, veri temizleme, dönüştürme, entegrasyon ve indirgeme gibi en temel veri ön işleme süreçlerini incelemiştir [1].

Wang, Li ve diğerleri hem yorumlanabilirliği hem de bireysel kredi riski sınıflandırma yeteneğini dikkate alan bir kredi riski değerlendirme modeli oluşturmuşlardır. Bu çerçevede bireysel kredi riski değerlendirmesinde, seçilen özelliklerin borçlunun kredi verilebilirlik riskiyle ilişkisinin yüksek olmaması durumunda, model seçiminden bağımsız bir şekilde, modelin sınıflandırma yeteneğinin güçlü olmayacağı tezinin işaret ettiği veri madenciliği yöntemlerini araştırmışlardır. Veri setinin özellik sayısının etkin bir şekilde azaltılmasının, gerçek veri madenciliği çalışmalarında çözülmesi gereken bir problem olduğunu ifade etmişlerdir. Çalışmada lojistik regresyon, AIC ve BIC özellik seçim yöntemleri karşılaştırılıp analiz edilmiş ve kişisel kredi riski değerlendirmesinde lojistik regresyona dayalı bir özellik seçim modeli oluşturulmuştur. KNN, karar ağacı ve

XGBoost sınıflandırma algoritmaları kullanılarak, bir bankanın gerçek kredi verileriyle gerçekleştirilen çalışmada boyut indirgemenin en yüksek başarımının XGBoost algoritmasında olduğunu göstermişlerdir [5].

Wang, Chen ve diğerleri Çin'in en popüler üçüncü parti online kredi skorlaması yapan Zhima Kredi Puanlama aracının, veri madenciliği yöntemleriyle performansının geliştirilip geliştirilemeyeceğini araştırmışlardır. Bulut hesaplama ve makine öğrenmesi gibi teknolojileri kullanan Zhima Kredi Puanlama, kullanıcının izniyle, kullanıcının internetteki çeşitli tüketim ve davranış verilerini esas alarak kredi verilebilirlik riskliliğini değerlendirir. Zhima Kredi Puanlama aracının puanlama performansının araştırıldığı çalışmada, (Zhima'nın algoritmalarından olan) C4.5, rassal orman, naive Bayes, K-en yakın komşu (KNN), destek vektör makineleri (SVM) ve geri yayılımlı yapay sinir ağı modelleri kullanılmıştır. Veri ön işleme aşamasında verilerde dengesizlik sorununu önlemek için SMOTE (Synthetic Minority Oversampling Technique) tekniği kullanılmıştır. SMOTE, azınlık örneklerine dayalı olarak yeni örnekleri yapay olarak sentezler ve sınıfları dengelemek için bunları veri kümesine ekler. 20 binden fazla online bireysel kredi bilgileri içeren bir veri seti üzerinde gerçekleştirilen deneysel çalışmanın sonuçları, Zhima Kredi Puanının gerçekten de sınıflandırma performanslarını iyileştirebileceğini ve iyileştirme derecesinin sınıflandırıcılara göre değiştiğini göstermiştir [6].

Bach, sınıflandırıcı performansını düşüren, makine öğrenimi görevlerinde yaygın bir sorun olan sınıf dengesizliğini araştırmıştır. Bu çerçevede alt örnekleme algoritması önermiştir. En yakın komşulara odaklanan önerilen algoritmanın ana fikri, çoğunluk sınıfındaki nesnelerin eşit şekilde ortadan kaldırılmasını garanti etmektir. Azınlık sınıflarına odaklana çalışma, azınlık sınıflarının özellikle ilginç olduğunu ve üyelerini olabildiğince doğru bir şekilde tanımanın gerekli olduğu vurgulanmıştır [7]. Diğer bir deyişle, azınlıktaki nesneleri yanlış sınıflandırmanın maliyetinin, çoğunluktaki nesneleri yanlış sınıflandırmanın maliyetinden tipik olarak çok daha yüksek olacağı gerçeğine işaret etmiştir.

Gupta ve diğerleri, yapay sinir ağları, karar ağacı, genetik algoritmalar, Bayesian ağlar, destek vektör makineleri, XGBoost gibi makine öğrenimi yaklaşımlarıyla kredi kartı sahtekarlığını tespit etmek amacıyla, birden fazla sınıflandırıcı ve veri dengeleme yöntemi kullanarak sürecin modellenebileceğini göstermiştir. Çalışmada dengesiz bir veri setlerinin, iyi bir performans göstermediği sonucuna varılmıştır. Dengesiz veriler üzerinde deneyler yapılmış ve XGBoost'un 0.91 kesinlik puanı ve 0.99 doğruluk puanıyla diğer sınıflandırıcılardan daha iyi performans verdiği gözlenmiştir. Deneysel çalışmada yüksek hızda rassal örnekleme (random oversampling), yüksek hızda örnekleme (oversampling), alt örnekleme (under sampling) ve SMOTE gibi veri dengeleme prosedürleri karşılaştırılmıştır. Dengesiz veriler üzerinde en uygun yöntem olduğu tespit edilen yüksek hızda rassal örnekleme (random oversampling) tekniği, deneysel çalışmada en iyi modele, yani XGBoost'a uygulandığında 0,99 kesinlik ve 0,99 doğruluk puanı verdiği gözlenmiştir [8]. Ancak veri setine göre en verimli veri dengeleme yönteminin farklılaşabileceği göz önünde bulundurulmalıdır. Bu yüzden en uygun veri dengeleme yöntemi, veri seti üzerinde deneme-yanılma yoluyla yapılan çalışmalara göre belirlenir.

## 3. METODOLOJİ

Makine öğrenmesi ve veri madenciliği teknikleriyle, müşteri kredi riskinin analizinde takip edilen bilimsel yöntemler aşağıda alt başlıklar şeklinde sunulmuştur.

### Veri Ön İşleme ve Analiz

Veri ön işlemenin amacı, orijinal iş verilerini yeni "iş modeli" ile düzenlemek, veri madeciliği kalitesini ve verimliliğini artırmak için ilgisiz özellikleri temizlemek suretiyle temiz, doğru ve basitleştirilmiş verileri elde etmektir.

Veri temizleme, verideki tutarsızlıkları gidermek amacıyla, eksik verileri tamamlamayı, gürültülü verileri düzleştirmeyi (smoothing), izole edilmiş verileri tespit etmeyi ve silmeyi gerektirir. Veri entegrasyonu, farklı veri kaynaklarına ait verileri veri tabanı, veri küpü veya sıradan dosyalar şeklinde veri ambarı gibi tutarlı bir

veri deposuna kaydetmeyi ifade eder. Veri dönüştürme, verilerin veri madenciliği için uygun bir forma dönüştürülmesini ifade eder. Veri azaltma, algoritmaların işleyeceği veri kümesini elde etmek için kullanılan azaltılmış veriler, orijinal verilerden çok daha küçük olmasına rağmen bütünlüğünü korur. Böylece veri madenciliği, yoğunlaştırılmış veri seti üzerinde daha fazla etkiye sahip olacak ve aynı (veya hemen hemen aynı) analiz sonucunu üretecektir [1].

Özellik seçimi (feature selection), belirli bir kritere göre bir problemi tanımlayan bir dizi özelliğin aranması olarak ifade edilebilir. Özellik seçme yöntemleri filtreler (filters), sarmalayıcılar (wrappers) ve gömülü (embedded) yöntemler olmak üzere üç türe ayrılır. Özellik seçimi için filtreleme tekniklerini kullanan prosedürler arasında, SA (significance attribute), SU (symmetrical uncertainty), SU-FCBF (fast correlation-based filter) ve CFS (correlation-based feature selection) sayılabilir.

SA (önem niteliği) yönteminde, nitelikler/özellikler ile nesneler (etiketler) arasındaki çift yönlü ilişki katsayıları kullanılır. Anlamlı bulunan özellikler, sınıfları doğru bir şekilde ayırma olasılığının yüksek olduğuna işaret eder. SU (simetrik belirsizlik) yönteminde, özelliklerin seçiminde korelasyonun ölçüsü olarak, nesnenin bireysel özellikleri tarafından sağlanan bilgi kazancı kullanılır. Bilgi kazancı, entropi temelinde, yani rassal değişken belirsizliğinin ölçüsüne göre hesaplanır ve bu belirsizliğin en aza indirilmesi amaçlanır. Bilgi kazancı simetriktir. Ancak bilgi kazancının değeri, daha büyük değere sahip olan özelliklere doğru sapmaktadır. SU bu sapmayı telafi eder ve bilgi kazancının değerinin [0,1] aralığındaki değerlerde normalleştirilmesine izin verir. 1 değeri, belirli bir özelliğin bilgisinin, bir nesnenin sınıfını tam olarak tahmin etmeye izin verdiği; 0 değeri ise belirli bir özelliğin, nesnenin belirli bir sınıfa ait olduğu hakkında herhangi bir bilgi taşımadığı anlamına gelir. SU-FCBF prosedüründe, her bir özelliğin SU'su hesaplanır ve sadece SU'su belirli bir eşikten büyük olan özellikler yeniden değerlendirilmek üzere seçilir. SU değerleri, azalan düzende özellikler kümesine yerleştirilir. Ardından özellik kümesinde yer alan fazla özellikler, buluşsal yöntemler kullanılarak özellik kümesinden kaldırılır. Geriye kalanlar ise modelde kullanılacak özellikler olarak seçilir. CFS prosedürü özellikler arasındaki korelasyonun incelenmesine dayanmaktadır. CFS prosedüründe kullanılan global korelasyon ölçüsü Pearson'un lineer korelasyonu iken, SU yerel korelasyon ölçüsü olarak kullanılır. Belirli bir nesne sınıfıyla güçlü bir şekilde ilişkili olan özellikler seçilir [3].

Verilerde dengesizlik (imbalance) sorununu önlemek için yüksek hızda rassal örnekleme (random oversampling) tekniğinden daha iyi performans gösteren SMOTE (Synthetic Minority Oversampling Technique) tekniği tercih edilmiştir. SMOTE, azınlık örneklerine dayalı olarak yeni örnekleri yapay olarak sentezler ve sınıfları dengelemek için bunları veri kümesine ekler. SMOTE yöntemi, sınıflandırma performanslarının iyileştirilmesinde kullanılır [6].

**XGBoost**

XGBoost (eXtreme Gradient Boosting) modeli, yüksek verimlilik ve tahmin doğruluğuna sahip ağaç tabanlı bir gradyan arttırma entegre modelidir. XGBoost, her biri zayıf öğrenen karar ağaçlarından oluşur. Boosting teknolojisi ise onu güçlü bir öğrenen ağaç konumuna yükseltir. Sınıflandırma probleminde zayıf sınıflandırıcıların global tahmin doğruluğu yüksek olmasa da verinin bazı yönlerinde çok yüksek tahmin doğruluğuna sahip olabilir. Özellik seçiminde lojistik regresyon, Akaike bilgi kriteri (Akakike information criteria, AIC), Bayesian bilgi kriteri (Bayesian information criteria, BIC) gibi yöntemler kullanılabilir [5].

XGBoost, sınıflandırma ve regresyon problemlerinin her ikisinde de en sık kullanılan makine öğrenimi algoritmasıdır. Diğer tüm makine öğrenimi algoritmalarından daha iyi performans gösterdiği iyi bilinmektedir. Gradyan arttırma (gradient boosting), bir hedef değişkeni tahmin etmek için birkaç küçük ve zayıf modelin tahminlerini birleştiren denetimli bir öğrenme tekniğidir. Kayıp (maliyet) fonksiyonu, daha iyi tahmin puanı elde etmek için gradyan azalma (descent) algoritması kullanılarak normalleştirilir [9].

**Karar Ağacı**

Karar ağacı (decision tree), değişkenlerin değerlerini bölerek sınıflandırma kurallarını belirleyen ağaç benzeri bir sınıflandırma modelidir. Karar ağacı algoritması, verileri bir dizi dikdörtgen bölüme ayırır. Ağacın tepesinde kök düğümler vardır. Kök düğümlerin her biri bölme/bölümleme düğümlerini temsil eder. Daha sonra "soru-cevap-soru" (question-judgment-question) ile ağaç benzeri bir sınıflandırma yol haritası oluşturmak için veriler dallar aracılığıyla diğer düğümlerle bağlanır. Düğümler, "safsızlık" (impurity) parametresi hesaplanarak bölünür [4].

Karar ağaçları (decision tree), kararı görselleştirmek için sezgiseldir. Regresyon ve sınıflandırma için kullanılan bir tür parametrik olmayan denetimli öğrenme yöntemi ve karar destek aracıdır. İnsan düşünme faaliyetini taklit eden bir akış şemasına sahiptir. Örneği girdi değişkenlerinin en temel ayırıcılarına göre mümkün olduğunca homojen küme gruplarına ayırmak için ağacın kökünden başlanır. Ağacın tüm dallarında yaprak düğümleri bulunana kadar bu işlem tekrarlanır.

Karar ağaçları özellik seçimine yardımcı olur. Çok az ön hazırlık gerektirdiği için yüksek boyutlu verileri doğru bir şekilde işleme yeteneğine sahiptir. Aykırı değerlerden çok fazla etkilenmez. Nicel ve nitel verileri işleyebilir. Ancak maksimum derinlik sınırlaması belirlenmediğinde, aşırı öğrenme (overfitting) problemine yatkındır. Bu sorun rassal orman (random forest) kullanılarak ortadan kaldırılabilir. Diğer taraftan veri kümelerindeki küçük değişiklikler tamamen farklı bir karar ağacının üretilmesine neden olabilir [2].

Karar ağacı, rassal orman, ve extra ağaç gibi ağaç tabanlı algortimalarda yaygın olarak kullanılan entropi, kök düğümü 0 ile 1 arasında olasılıklandıran bir belirsizlik ölçüsüdür. Entropi 0'a yakın değerler aldığında, tamamen homojen dağılıma sahip (belirsizliğin olmadığı) bir örnek anlamına gelir. Entropi 1'e yakın değerler aldığında ise homojen olmayan dağılıma sahip (belirsizliğin yüksek olduğu) bir örnek anlamına gelir [11]. Ağaç tabanlı algortimalar en iyi sınıflandırma performansına ulaşmak için entropiyi mümkün olduğu kadar 0 değerine yaklaştırmaya çalışır.

**Rassal Orman**

Rassal orman (random forest) büyük miktarda karar ağaçlarını bir araya getirir. Her karar ağacı bir sınıf tahminini temsil eder. Bu yöntem her bir karar ağacından tahminleri toplar ve en iyi tahmini yapan ağacıntahmini sınıf tahmini olarak kabul edilir. Eğitim sırasında her model verisi, rassal olarak seçilen farklı alt örneklerden oluşur. Alt örnekler, torbalama veya güçlendirilmiş önyükleme (bagging veya boot strapping) adı verilen işlemle alınır. Buna göre bazı örnekler tek bir karar ağacında birçok kez kullanılabilir. Bazı ağaçlar yüksek varyansa sahip olmasına rağmen, tüm orman düşük varyansa ancak yüksek yanlılığa sahip olacaktır. Test aşamasında, genel tahminleri elde etmek için her bir karar ağacının tahminlerinin ortalaması alınır ve süreç torbalama olarak bilinir. Bu yöntemin karar ağacından daha iyi performans göstermesinin sebebi, pek çok ilişkisiz karar ağacının kendi tahminlerini türetmek için birbirinin bireysel hatalarından korunabilmesidir. Böylece aşırı öğrenme (overfitting) probleminden kaçınmak mümkün olur. Ayrıca modelin çok sayıda veri kümesi üzerinde verimli bir şekilde çalışabilmesi ve sınıflandırmada hangi değişkenlerin önemli olduğunu tahmin edebilmesi gibi başka güçlü yanları da vardır. Rassal orman, nesne ile özellikler arasındaki doğrusal olmayan ilişkiyi yakalayabildiği için doğrusal modellerden daha iyi performans gösterir. Yöntemin dezavantajı, özellikler karar ağaçlarının yapı taşları olduğu için az sayıda özellikle verimli çalışamamasıdır [2].

**Ekstra Ağaç**

Denetimli sınıflandırma yöntemlerinden ekstra ağaç (extra tree), regresyon problemleri için ağaç tabanlı bir topluluk algoritmasıdır. Girdiler rassal seçilen bir ağaca sunulur. Böylece optimal budama noktaları, kendisinden türetilen ağacın varyansının büyük bir kısmını temsil edebilir. Standart ağaçların budama noktaları, hesaplama yüklerinde önemli ölçüde azalma nedeniyle doğruluğun artmasına yol açar. Örüntüleri öğrenme süreci, örneğin çıktı değerlerinden bağımsız olan rassal ağaçlar yoluyla gerçekleştirilir. Ekstra ağaç

ve diğer ağaç tabanlı kümeleme yöntemleri arasındaki iki temel fark, budama noktalarını rassal seçerek düğümleri ayırması ve tüm örneği kullanarak ağacı büyütmesidir. Sınıflandırma problemlerinde nihai tahminlerin üretilmesi için çoğunluk oyu birleştirilir. Ekstra ağaç, her temel tahminciyi eğitmek için rassal alt küme özelliklerini kullanır [10].

**Naive Bayes**

Naive Bayes, olasılıklara dayalı olarak en iyi sınıf tahmin edicisini seçme fikrine dayanır. Her bir X özelliğinin (feature) P(C|X) olasılığını, diğer bir ifadeyle özelliklerin tümünün birden sınıfı tahmin etme olasılığınının maksimize edilmesini amaçlar. Bunlar arasında, P(C) "önceki" olasılık sınıfıdır, P(X|C), etikete göre sınıf koşullu olasılığıdır ve P(X), normalleştirme için kullanılan "kanıt" faktörüdür. Naive Bayes sınıflandırıcısında tüm değişkenlerin bağımsız olduğu varsayılır [4].

Doğrusal bir sınıflandırıcı olan Naive Bayes, koşullu olasılığa odaklanır. Sınıflandırma tekniği, Bayes teoremine dayalı olarak geliştirilmiştir. Bayes teoremi etiketlerin (özniteliklerin) ve özelliklerin bağımsız olduğunu varsaydığından, yüksek boyutlu girdiler için uygundur. Bu varsayım hesaplamayı basitleştirdiği için “saf” (naive) olarak kabul edilir. Özellikle küçük örneklem büyüklükleri için genellikle alternatif sınıflandırıcılardan daha iyi performans gösterir.

Naive Bayes'in bir avantajı, gerekli sınıflandırma parametrelerini tahmin etmek için daha az eğitim verisi gerektirmesidir. Bununla birlikte, naive Bayes’in dezavantajı, bağımsız parametre varsayımıdır. Çünkü gerçek dünyada veri kümeleri, genellikle özellikler arasında ilişkilere sahip olduğundan bağımsızlık varsayımının sağlanması zordur. Bu durum, sınıflandırıcının etkinliğini büyük ölçüde azaltacaktır. Test veri setinde, kategorik değişkenin eğitim veri setinde bulunmayan bir kategorisi varsa, model sıfır olasılık atar. Buna “sıfır frekans” denir. Bu sorunu çözmek için Laplace tahmini gibi yumuşatma tekniği kullanılabilir [2].

**KNN**

KNN, yeni örneklerin veri noktalarına olan mesafelerini belirlemek için eğitim setinden örnekleri kullanılır ve her kategorideki en yakın komşuları analiz eder. Yeni örneklerin ait olduğu sınıfların tahmin edilmesi amacıyla, eğitim veri seti ile karşılaştırılması gerekir. Bu yüzden eğitim seti boyut (dimension) ve kapsam (size) olarak genişse, KNN maliyetli olabilir. Bununla birlikte KNN, eğitilmesi ve kesin sonuçlar alması kolay olduğu için yaygın olarak kullanılmaktadır. Ancak k (sınıf adedi) parametresinin dikkatli seçilmesi gerekmektedir [9].

KNN, bir veri noktasını kendisine en yakın (k değeri ile temsil edilen) veri kümesine atama yapmak suretiyle sınıflandırma yapar. En yaygın kullanılan uzaklık ölçüleri “minkowski”, “euclidean”, “manhattan” olarak sayılabilir. Aşağıdaki denklem, veri noktaları arasındaki uzaklığı manhattan yöntemine göre ölçer.

$$d = \sum |x_i - y_i| \tag{3.1}$$

Yukarıdaki denklemde d veri noktaları arasındaki uzaklığı, $x_i$ ve $y_i$ değişkenlerin değerlerini ifade eder. KNN seçilen bir k değerine göre uzaklık ölçüsünü kullanarak bir veri noktasını, k elemandan oluşan kendisine en yakın kümeye atar [12].

**Lojistik Regresyon**

Lojistik regresyon, sınıflandırma problemi için güçlü bir regresyon ve istatistiksel yöntemdir. ikili sınıflandırma problemlerinin temeli olarak kullanılır. Eğitim verilerinden, lojistik regresyon denkleminin katsayıları tahmin edilir. Optimize edilmiş katsayılar, varsayılan sınıfın değerini bire, diğer sınıfların değerini ise sıfıra yakın olarak tahmin eder. Bire yakın olanlar bir, sıfıra yakın olanlar ise sıfır değerini döndürür. Bu modelin diğer modellere göre en büyük avantajı, sadece bir sınıflandırma probleminin çözümünü değil, aynı zamanda olasılıkları da sağlar. Ancak modelin ağırlıklar toplamsal değil çarpımsal olduğundan yorumlamak zordur. Ayrıca bir özellik

iki sınıfı mükemmel bir şekilde ayırabiliyorsa, modelin eğitilmesi mümkün olmaz. Çünkü söz konusu özelliğin ağırlığı yakınsamaz ve optimum ağırlık sonsuz olur. Bu sorunu çözmek için ağırlıkların önceden bir olasılık dağılımı tanımlanabilir [2].

**Performans Metrikleri**

Karışıklık matrisi (confusion matrix), bir tür istatistiksel sınıflandırma değerlendirme yöntemidir. Matrisin sütunları sınıflandırıcının tahminlerini, satırları ise gerçek değerleri temsil eder. Makine öğreniminin örnek özelliklerini karıştırıp karıştırmadığını gözlemlemek amacıyla kullanıldığı için "karışıklık matrisi" olarak adlandırılır [4].

Modellerin sınıflandırma performanslarını kıyaslamak amacıyla “accuracy” metriğine başvurulmuştur. Accuracy metriği, Tablo 3.1’de gösterilen karışılık matrisi (confusion matrix) olarak bilinen bir tablodan türetilir [13].

| | | TAHMİN EDİLEN | | TOPLAM |
|---|---|---|---|---|
| | | $C^+$ | $C^-$ | |
| GERÇEK | $C^+$ | TP<br>True Pozitif<br>(Hits) | FN<br>False Negatif<br>(Miss) | $N^+$<br>Gerçek<br>Pozitif sayısı |
| | $C^-$ | FP<br>False Pozitif<br>(Miss) | TN<br>True Negatif<br>(Hits) | $N^-$<br>Gerçek<br>Negatif sayısı |
| TOPLAM | | $\tilde{N}^+$<br>Tahmin<br>Pozitif sayısı | $\tilde{N}^-$<br>Tahmin<br>Negatif sayısı | N<br>Toplam<br>Örnek sayısı |

Tablo 3.1. Karışıklık Matrisi

Tablo 3.1’de yer alan true positive (TP), gerçek değeri pozitif olup da pozitif değere sınıflandırılanların sayısını; false positive (FP), gerçek değeri negatif olup da pozitif değere sınıflandırılanların sayısını ifade eder. Gerçekte kovid bir insana kovid tanısının konması TP’ye, gerçekte kovid olmayan bir insana kovid tanısının konması ise FP’ye örnek gösterilebilir. Benzer şekilde true negative (TN), gerçek değeri negatif olup da negatif değere sınıflandırılanların sayısını; false negative (FN), gerçek değeri pozitif olup da negatif değere sınıflandırılanların sayısını temsil eder [14]. Gerçekte kovid olmayan bir insana kovid değil tanısının konması TN’ye, gerçekte kovid bir insana kovid değil tanısının konması ise FN’ye örnek gösterilebilir. $N^+$ ve $N^-$ ifadeleri ise sırasıyla, pozitif ve negatif gerçek değerlerin sayısını temsil eder. Accuracy, percision, recall ve f1-score metrikleri aşağıda gösterildiği şekilde hesaplanır. Çalışmada yararlanılan accuracy, percision, recall ve f1-score metrikleri aşağıda gösterildiği şekilde hesaplanır [15].

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{N}^+ + \text{N}^-} \tag{3.2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{3.3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{N}^+} \tag{3.4}$$

$$\text{f1-score} = \frac{2.\text{Precision}.\text{Recall}}{\text{Precision}+\text{Recall}} \tag{3.5}$$

Karışıklık matrisinden türetilen yukarıdaki metriklerin tümü test veri seti esas alınarak hesaplanır.

## 4. VERİ KÜMESİ

Müşterilerin kredi riskiyle ilgili verilerin yer aldığı veri setine, açık kaynak bir veri platformu olan Kaggle'dan, aşağı sunulan bağlantıdan erişilebilir.

https://www.kaggle.com/datasets/ppb00x/credit-risk-customers

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| **checking_status** | <0 | 0<=X<200 | no checking | <0 | <0 |
| **duration** | 6.0 | 48.0 | 12.0 | 42.0 | 24.0 |
| **credit_history** | critical/other existing credit | existing paid | critical/other existing credit | existing paid | delayed previously |
| **purpose** | radio/tv | radio/tv | education | furniture/equipment | new car |
| **credit_amount** | 1169.0 | 5951.0 | 2096.0 | 7882.0 | 4870.0 |
| **savings_status** | no known savings | <100 | <100 | <100 | <100 |
| **employment** | >=7 | 1<=X<4 | 4<=X<7 | 4<=X<7 | 1<=X<4 |
| **installment_commitment** | 4.0 | 2.0 | 2.0 | 2.0 | 3.0 |
| **personal_status** | male single | female div/dep/mar | male single | male single | male single |
| **other_parties** | none | none | none | guarantor | none |
| **residence_since** | 4.0 | 2.0 | 3.0 | 4.0 | 4.0 |
| **property_magnitude** | real estate | real estate | real estate | life insurance | no known property |
| **age** | 67.0 | 22.0 | 49.0 | 45.0 | 53.0 |
| **other_payment_plans** | none | none | none | none | none |
| **housing** | own | own | own | for free | for free |
| **existing_credits** | 2.0 | 1.0 | 1.0 | 1.0 | 2.0 |
| **job** | skilled | skilled | unskilled resident | skilled | skilled |
| **num_dependents** | 1.0 | 1.0 | 2.0 | 2.0 | 2.0 |
| **own_telephone** | yes | none | none | none | none |
| **foreign_worker** | yes | yes | yes | yes | yes |
| **class** | good | bad | good | good | bad |

Tablo 4.1. Veri Setinin Genel Görünümü

Veri seti 20 boyuttan/özellikten ve ("iyi" ve "kötü" şeklinde) ikili (binary) bir sınıftan oluşmaktadır (Tablo 4.1). Kategorik ve numerik değerlere sahip özelliklerden oluşan veri seti, 1000 örnek içermektedir. Örneklerin 700'ü "iyi", 300'ü ise "kötü" sınıfına aittir. Veri seti, veri madenciliği açısından değerlendirilirse, çok sayıda veri ön işleme süreci gerektirmektedir. Bu yönüyle, veri madenciliği pratiği için oldukça verimli bir veri seti olduğu söylenebilir.

Veri setinde bulunan kategorik özelliklerin sahip olduğu kategori sayıları aşağıda gösterilmiştir (Tablo 4.2).

```
checking_status: 4
duration: 33
credit_history: 5
purpose: 10
credit_amount: 921
savings_status: 5
employment: 5
installment_commitment: 4
personal_status: 4
other_parties: 3
residence_since: 4
property_magnitude: 4
age: 53
other_payment_plans: 3
housing: 3
existing_credits: 4
job: 4
num_dependents: 2
own_telephone: 2
foreign_worker: 2
class: 2
```

Tablo 4.2. Özelliklerin İçerdiği Kategori Sayıları

## 5. DENEYSEL ÇALIŞMA

Bu araştırma, kaggle platformu üzerinde, jupyter IDE'si kullanılarak python programlama dilinde gerçekleştirilmiştir. Veri ön işleme ve modelleme süreçlerinde pandas, numpy, sklearn, imblearn, re, xgboost; veri görselleştirme işlemlerinde ise matplotlib ve seaborn kütüphanelerinden yararlanılmıştır.

Çalışmanın yapıldığı bilgisayarın teknik özellikleri aşağıda listeleniştir.

- CPU: AMD Ryzen 5 3500X 6-Core Processor, 3600 Mhz, 6 Core(s), 6 Logical Processor(s)
- GPU: NVIDIA GeForce RTX 3060, VRAM 12 GB
- RAM: 16 GB
- İşletim Sistemi: Windows 10

Python ile yapılan çalışmalara aşağıdaki bağlantılardan ulaşılabilir.

https://www.kaggle.com/code/shakkutlu/pad-2-credit-risk-assessment-1

https://www.kaggle.com/code/shakkutlu/pad-2-credit-risk-assessment-2

Veri setinin null değerlere sahip örnekler içerip içermediği ve veri tipleri incelenmiş ve Tablo 4.3'te gösterildiği gibi herhangi bir null değere sahip satıra ya da uygun olmayan veri tipine rastlanmamıştır.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 21 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   checking_status         1000 non-null   object
 1   duration                1000 non-null   float64
 2   credit_history          1000 non-null   object
 3   purpose                 1000 non-null   object
 4   credit_amount           1000 non-null   float64
 5   savings_status          1000 non-null   object
 6   employment              1000 non-null   object
 7   installment_commitment  1000 non-null   float64
 8   personal_status         1000 non-null   object
 9   other_parties           1000 non-null   object
 10  residence_since         1000 non-null   float64
 11  property_magnitude      1000 non-null   object
 12  age                     1000 non-null   float64
 13  other_payment_plans     1000 non-null   object
 14  housing                 1000 non-null   object
 15  existing_credits        1000 non-null   float64
 16  job                     1000 non-null   object
 17  num_dependents          1000 non-null   float64
 18  own_telephone           1000 non-null   object
 19  foreign_worker          1000 non-null   object
 20  class                   1000 non-null   object
dtypes: float64(7), object(14)
memory usage: 164.2+ KB
```

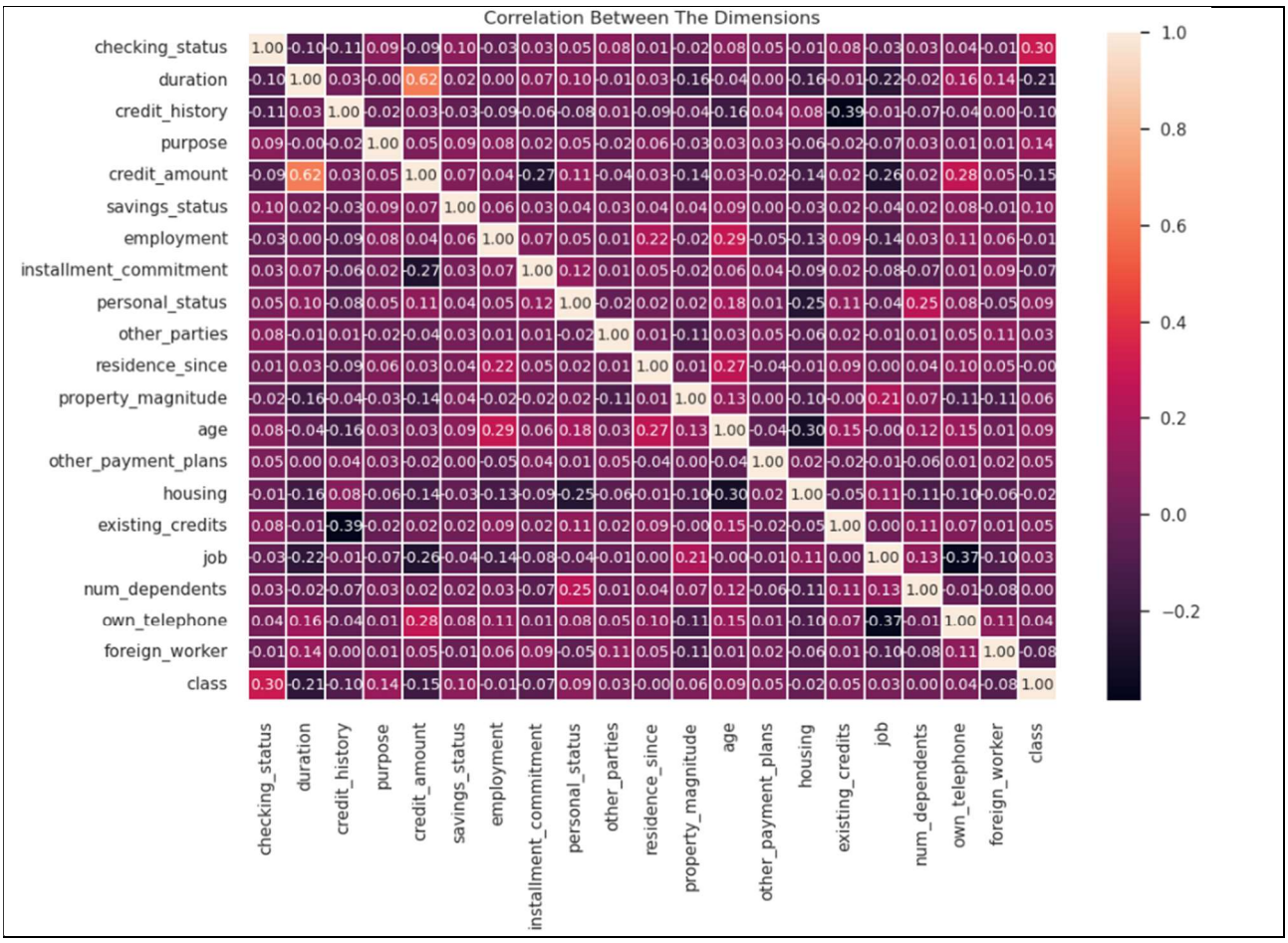
Tablo 4.3. Null Değerlerin Araştırılması ve Veri Tipleri

Veri setinin gürültülerden arındırılması amacıyla, Tablo 4.4'te gösterildiği gibi özelliklerin alabileceği değer aralıkları araştırılmış ve herhangi bir aykırı değere rastlanmamıştır.

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **duration** | 1000.0 | 20.903 | 12.058814 | 4.0 | 12.0 | 18.0 | 24.00 | 72.0 |
| **credit_amount** | 1000.0 | 3271.258 | 2822.736876 | 250.0 | 1365.5 | 2319.5 | 3972.25 | 18424.0 |
| **installment_commitment** | 1000.0 | 2.973 | 1.118715 | 1.0 | 2.0 | 3.0 | 4.00 | 4.0 |
| **residence_since** | 1000.0 | 2.845 | 1.103718 | 1.0 | 2.0 | 3.0 | 4.00 | 4.0 |
| **age** | 1000.0 | 35.546 | 11.375469 | 19.0 | 27.0 | 33.0 | 42.00 | 75.0 |
| **existing_credits** | 1000.0 | 1.407 | 0.577654 | 1.0 | 1.0 | 1.0 | 2.00 | 4.0 |
| **num_dependents** | 1000.0 | 1.155 | 0.362086 | 1.0 | 1.0 | 1.0 | 1.00 | 2.0 |

Tablo 4.4. Özelliklerin Alabileceği Değer Aralıklarının Araştırılması

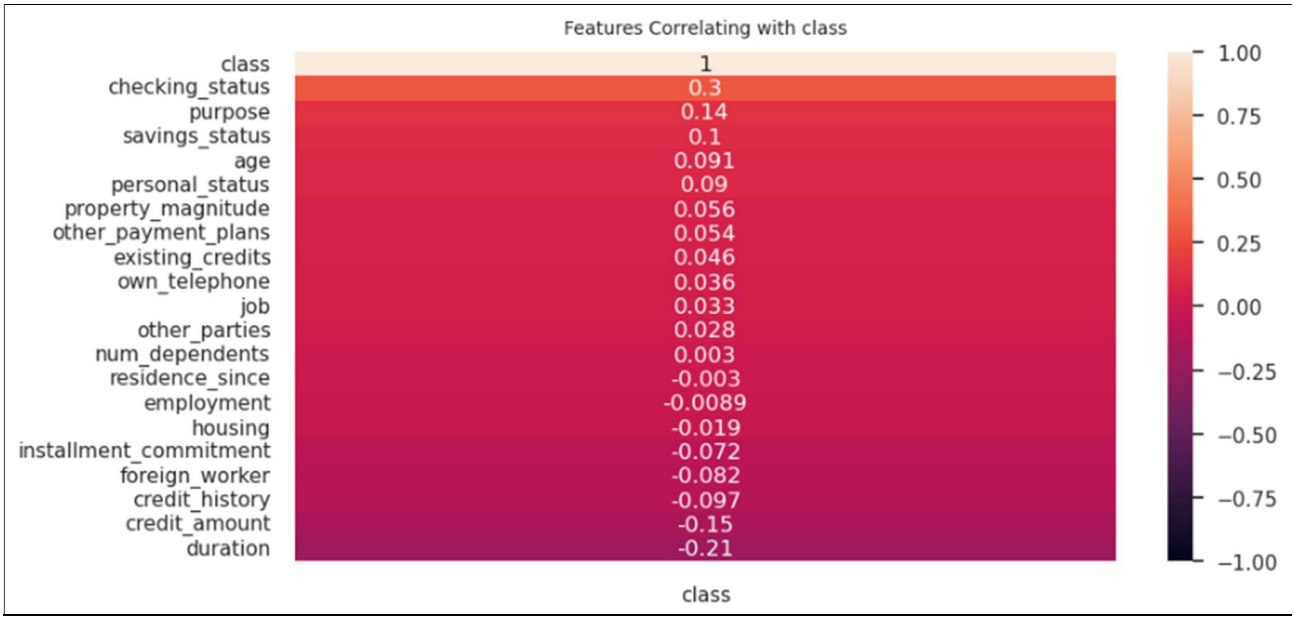
Aşağıda yer alan Şekil 4.1'de, keşifsel veri analizi kapsamında boyutların korelasyon matrisi oluşturulmuştur.




Şekil 4.1. Tüm Boyutların Korelasyon Matrisi

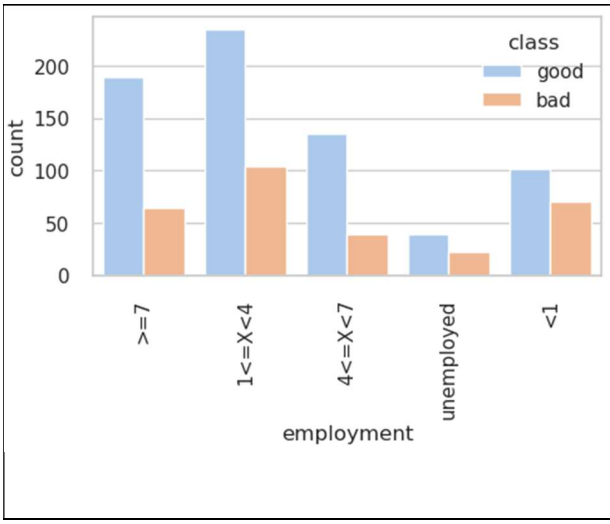
Özellik seçimi sürecinde bir fikir vermesi amacıyla, korelasyon matrisi dikkatlice incelenmiştir. Bu çerçevede "credit_amount" (kredi miktarı) ve "duration" (vade) özellikleri arasında 0,62 seviyesinde nispeten yüksek bir korelasyon vardır. Bu durum özellik seçiminde dikkate alınarak modelleme aşamasında bu iki özellikten sadece biri, diğer bir ifadeyle "class" sınıfıyla korelasyonu daha yüksek olan "duration" özelliği seçilmiştir.

Aşağıda bulunan Şekil 4.2'de ise tüm boyutların kredi riskini temsil eden "class" sınıfı ile korelasyonları gösterilmektedir. "class" sınıfıyla korelasyonu 0 veya 0'a yakın "own_telephone", "num_dependents" gibi birçok özellik modelleme sürecine dahil edilmemiştir.

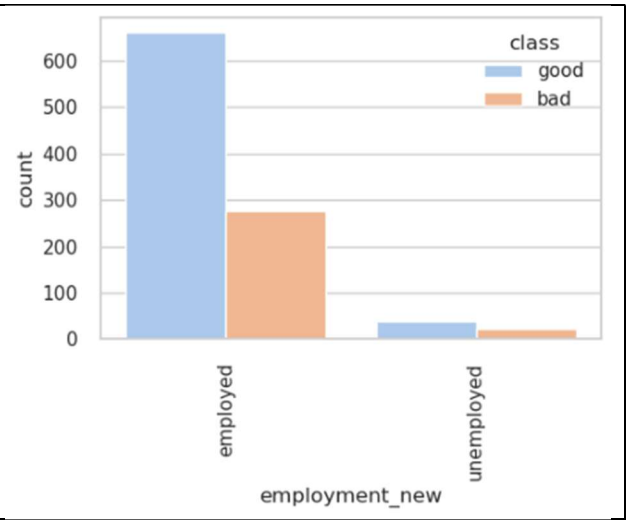



Şekil 4.2. Tüm Boyutların Özellik Sınıfı “class” ile Korelasyonu

Aşağıda Şekil 4.3.a’da gösterilen “employment” özelliğinin kategorileri Şekil 4.3.b’de gösterildiği gibi “employed” ve “unemployed” şeklinde kategoride birleştirilmiştir.

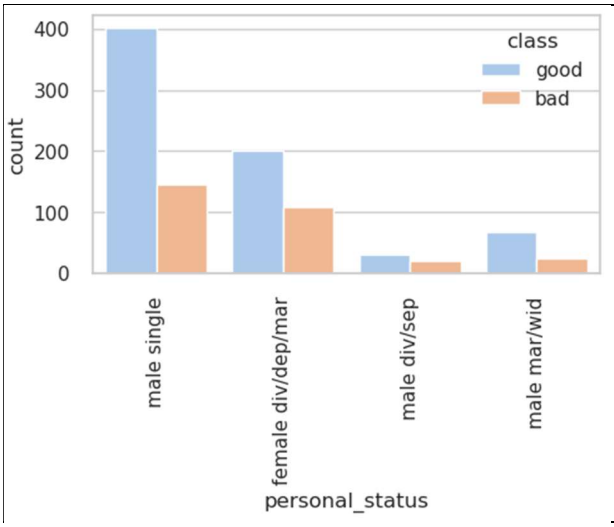



Şekil 4.3.a. Birleştirme Öncesi: “employment” Özelliği

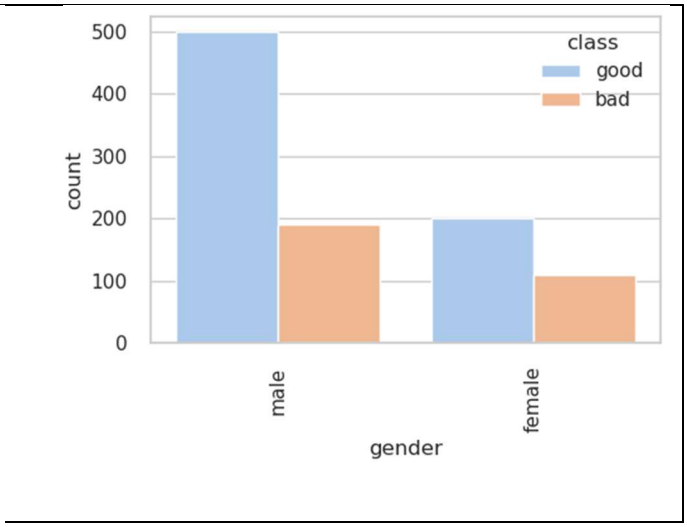



Şekil 4.3.b. Birleştirme Sonrası: “employment_new” Özelliği

Aşağıda Şekil 4.4.a’da gösterilen “personal_status” özelliğinin kategorileri Şekil 4.4.b’de gösterildiği gibi “male” ve “female” şeklinde iki kategoride birleştirilmiştir.

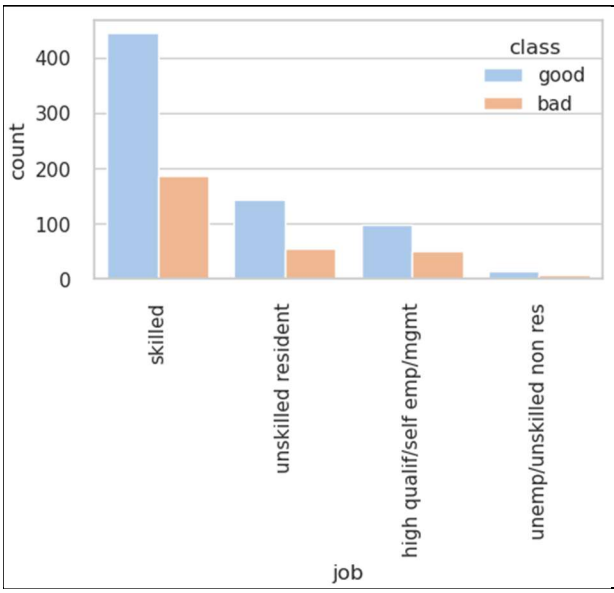



Şekil 4.4.a. Birleştirme Öncesi: “personal_status” Özelliği

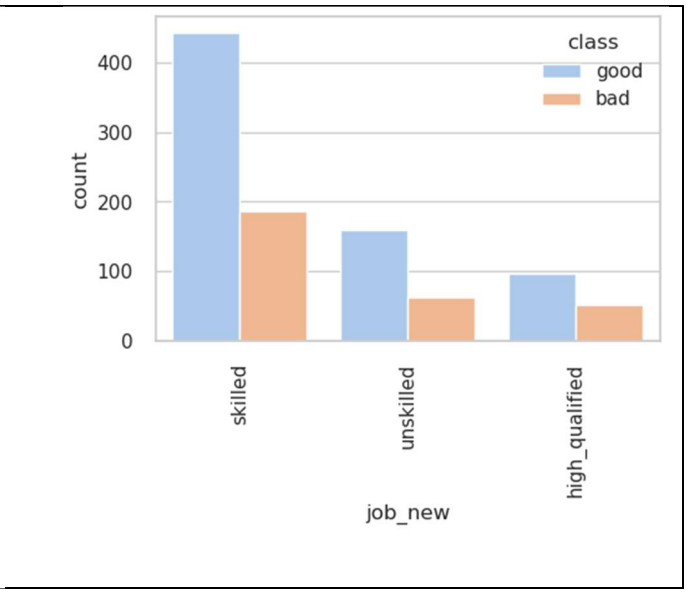



Şekil 4.4.b. Birleştirme Sonrası: “gender” Özelliği

Aşağıda Şekil 4.5.a’da gösterilen “personal_job” özelliğinin kategorileri Şekil 4.5.b’de gösterildiği gibi “skilled”, “unskilled“ ve “high_qualified” şeklinde üç kategoride birleştirilmiştir.

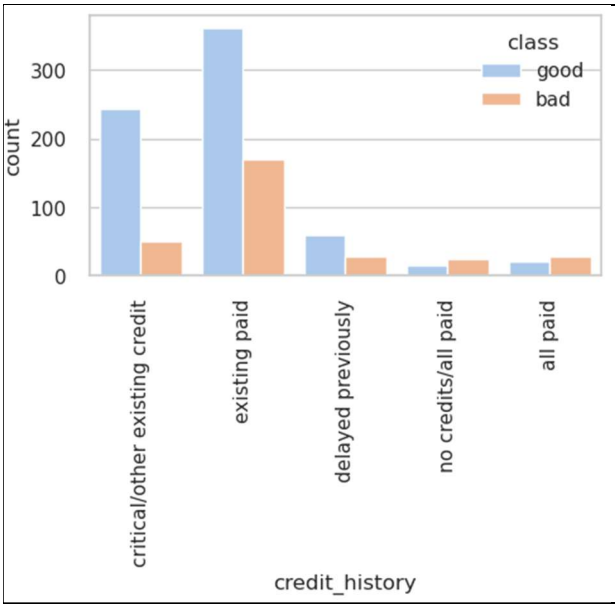



Şekil 4.5.a. Birleştirme Öncesi: “personal_job” Özelliği

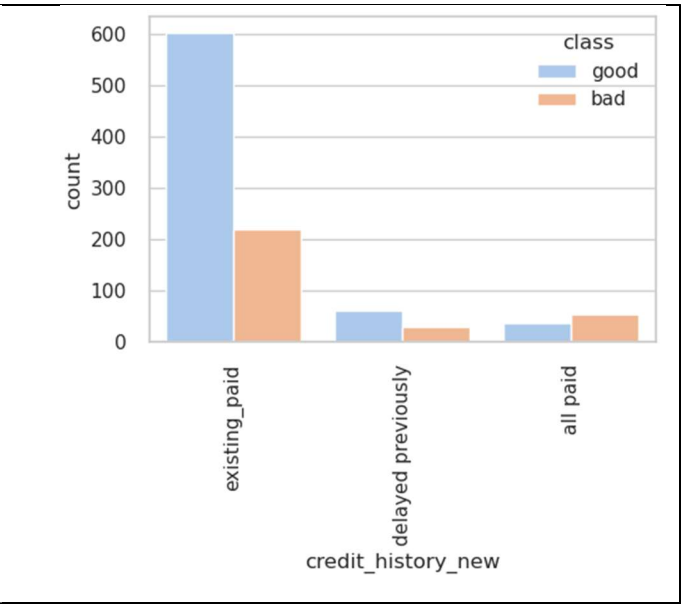



Şekil 4.5.b. Birleştirme Sonrası: “job_new” Özelliği

Aşağıda Şekil 4.6.a’da gösterilen “credit_history” özelliğinin kategorileri Şekil 4.6.b’de gösterildiği gibi “existing_paid”, “delayed previously“ ve “all paid” şeklinde üç kategoride birleştirilmiştir.



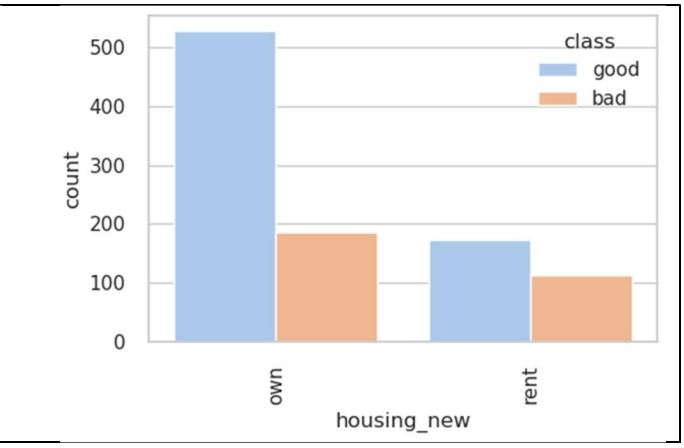

Şekil 4.6.a. Birleştirme Öncesi: “credit_history” Özelliği




Şekil 4.6.b. Birleştirme Sonrası: “credit_history_new” Özelliği

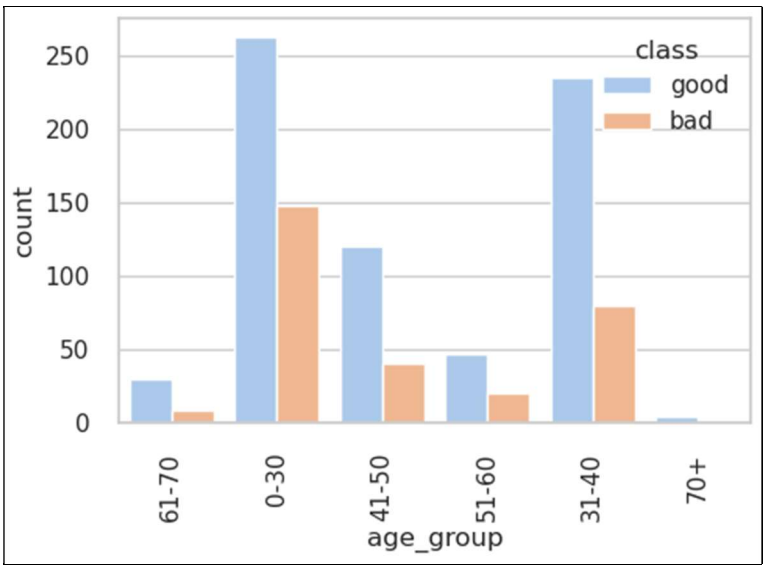
Aşağıda Şekil 4.7.a’da gösterilen “housing” özelliğinin kategorileri Şekil 4.7.b’de gösterildiği gibi “own” ve “rent” şeklinde iki kategoride birleştirilmiştir.


Şekil 4.7.a. Birleştirme Öncesi: “housing” Özelliği



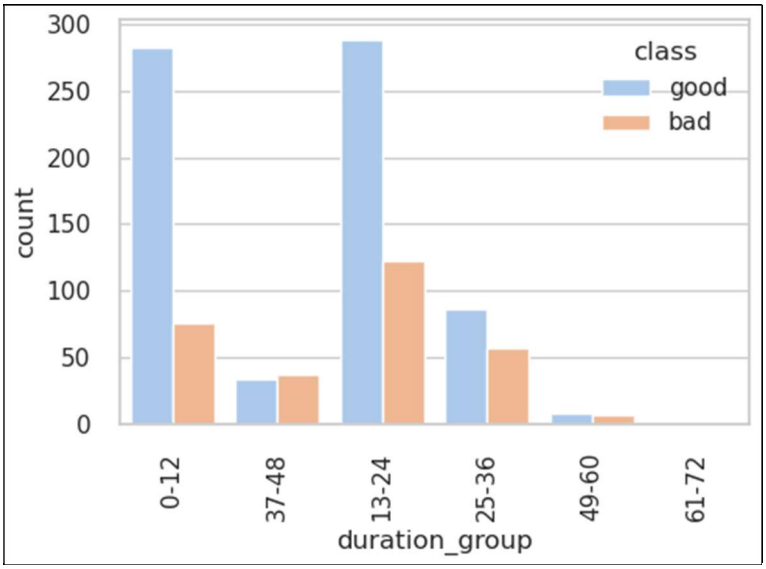

Şekil 4.7.b. Birleştirme Sonrası: “housing _new” Özelliği

Numerik “age” özelliği, aşağıda Şekil 4.8’de gösterildiği gibi yaş gruplarına ayrılarak kategorik değişkenlere dönüştürülmüştür. Böylece “age” özelliğinin kredi riskini temsil eden “class” sınıfı üzerinde toplulaştırılmış etkisinin elde edilmesi amaçlanmıştır.




Şekil 4.8. Numerik “age” Özelliğinin Kategorik Yapılması

Benzer şekilde numerik “duration” özelliği, aşağıda Şekil 4.9’da gösterildiği gibi vade gruplarına ayrılarak kategorik değişkenlere dönüştürülmüştür. “age” özelliğinde belirtildiği gibi “duration” özelliğinin de kredi riskini temsil eden “class” sınıfı üzerinde toplulaştırılmış etkisinin elde edilmesi amaçlanmıştır.



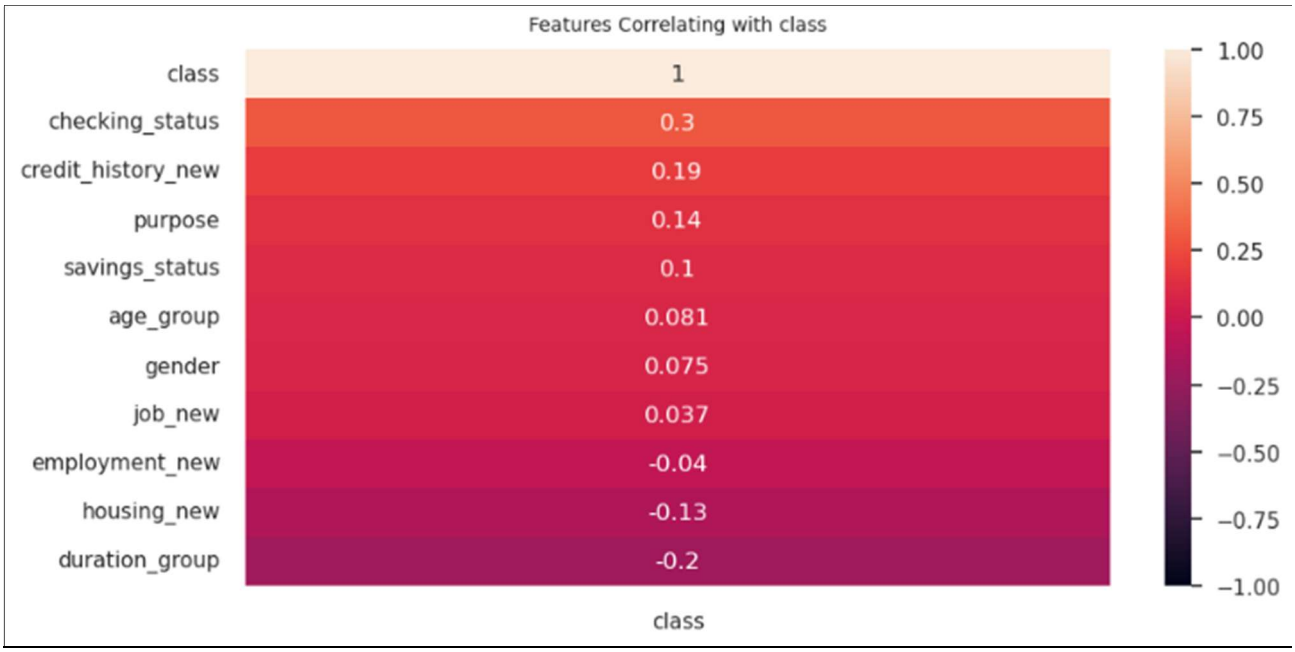

Şekil 4.9. Numerik “duration” Özelliğinin Kategorik Yapılması

Tablo 4.5’te gösterilen özellikler dışında, veri setinde yer alan diğer tüm özellikler kaldırılarak boyut indirgeme işlemi yapılmıştır.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   checking_status     1000 non-null   object
 1   purpose             1000 non-null   object
 2   savings_status      1000 non-null   object
 3   class               1000 non-null   object
 4   employment_new      1000 non-null   object
 5   gender              1000 non-null   object
 6   job_new             1000 non-null   object
 7   credit_history_new  1000 non-null   object
 8   housing_new         1000 non-null   object
 9   age_group           1000 non-null   object
 10  duration_group      1000 non-null   object
dtypes: object(11)
memory usage: 86.1+ KB
```

Tablo 4.5. Birinci Adım: Boyut İndirgeme Sonrası Özellikler

Şekil 4.10'da, "class" sınıfı ile yukarıda yeniden yapılandırılan özellikler de dahil modelde kullanılması planlanan özelliklerin korelasyonları yer almaktadır.




Şekil 4.10. Boyut İndirgeme Sonrası Boyutların Özellik Sınıfı "class" ile Korelasyonu

Ancak yeniden yapılandırılan "job_new" ve "employment_new" özellikleri ile "class" sınıfı arasında 0'a yakın bir korelasyon vardır. Bu yüzden "job_new" ve "employment_new" özellikleri modelleme aşamasında sürece dahil edilmemiştir. Boyut indirgeme kapsamında, problemin boyutu, yeniden yapılandırılan yapay özelliklerle birlikte, Tablo 4.6'da gösterildiği üzere 25'ten 8'e düşürülmüştür. Böylece accuracy metriğinde bir artış olsun ya da olmasın, problemin karmaşıklığının azaltılması ve modelin genelleme (generalization) yapabilme kabiliyetinin arttırılması amaçlanmaktadır.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 9 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   checking_status     1000 non-null   object
 1   purpose             1000 non-null   object
 2   savings_status      1000 non-null   object
 3   class               1000 non-null   object
 4   gender              1000 non-null   object
 5   credit_history_new  1000 non-null   object
 6   housing_new         1000 non-null   object
 7   age_group           1000 non-null   object
 8   duration_group      1000 non-null   object
dtypes: object(9)
memory usage: 70.4+ KB
```

Tablo 4.6. İkinci Adım: Boyut İndirgeme Sonrası Özellikler

Tümü kategorik değişkenlerden oluşan özellikler ve sınıf, algoritmalara girdi olarak verilmeden önce birkaç işlemden daha geçirilmiştir. İlk olarak tüm özellikler numerik değerlere dönüştürülmüş (encoding), sonra ise min-max yöntemi ile normalize edilmiştir. Daha sonra ise veri setindeki olası dengesizlikleri azaltmak için random oversampling vb. teknikler arasında en iyi performansı sunan SMOTE yöntemi (deneme-yanılma yoluyla) seçilmiştir. Son olarak, veri setinin yüzde 30'u test, yüzde 70'i ise eğitim için ayrılmıştır.

Modelleme aşamasında XGBoost, karar ağacı, rassal orman, extra ağaç, naive Bayes, KNN ve lojistik regresyon algoritmaları kullanılmıştır. Veri setine uygulanan modellerin kritik parametreleri deneme-yanılma yöntemiyle belirlenmiş olup Tablo 4.7'de sunulmuştur.

| | **XGBoost** | **Karar Ağacı** | **Rassal Orman** | **Extra Ağaç** | **Naive Bayes** | **KNN** | **Lojistik Regresyon** |
|---|---|---|---|---|---|---|---|
| **learning_rate** | 0,1 | - | - | - | - | - | - |
| **max_depth** | 5 | 6 | 9 | 11 | - | - | - |
| **n_estimators** | 100 | - | - | | - | - | - |
| **criterion** | - | entropy | entropy | entropy | - | - | - |
| **min_samples_split** | - | 2 | 7 | 3 | - | - | - |
| **metric** | - | - | - | - | - | minkowski | - |
| **n_neighbors** | - | - | - | - | - | 1 | - |
| **k** | - | - | - | - | - | 4 | - |

Tablo 4.7. Algoritmaların Optimize Edilen Kritik Parametreleri

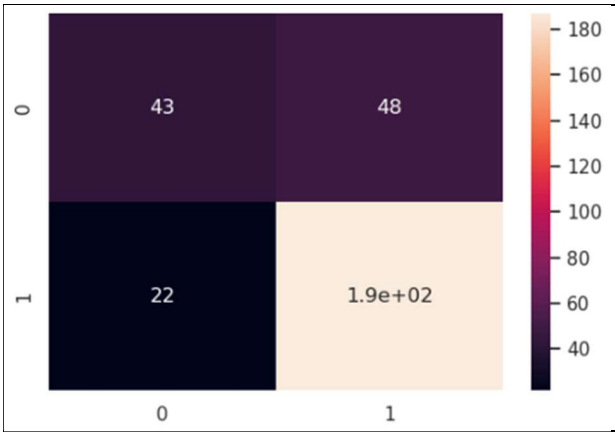
## 6. DENEYSEL ÇALIŞMA SONUÇLARI

Algoritmaların çalışmasını sağlayabilecek seviyede, asgari veri madenciliği teknikleri kullanılarak 20 özellik ve iki kategorik değere sahip bir sınıf ile çalıştırılan 7 farklı modelin performans metrikleri Tablo 6.1'de sunulmuştur.

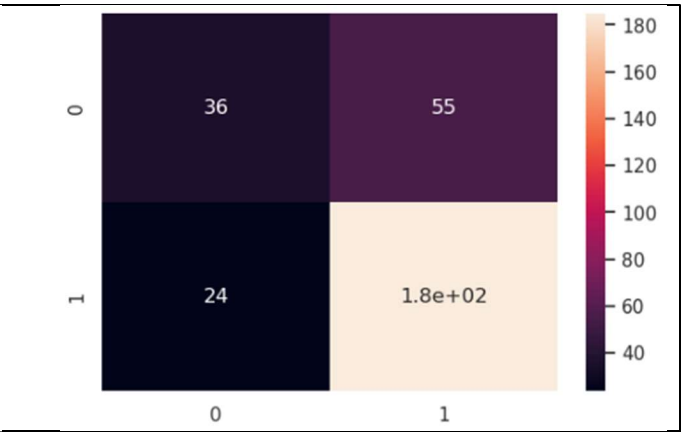
| | XGB | Decision Tree | Random Forest | Extra Trees | Naive Bayes | KNN | Logistic Regression |
|---|---|---|---|---|---|---|---|
| **Model** | XGB | Decision Tree | Random Forest | Extra Tree | Naive Bayes | KNN | Logistic Regression |
| **Accuracy** | 0.77 | 0.74 | 0.76 | 0.75 | 0.73 | 0.69 | 0.71 |
| **F1** | 0.84 | 0.82 | 0.85 | 0.84 | 0.81 | 0.8 | 0.81 |
| **Recall** | 0.89 | 0.89 | 0.94 | 0.95 | 0.82 | 0.91 | 0.9 |
| **Precision** | 0.8 | 0.77 | 0.77 | 0.76 | 0.8 | 0.72 | 0.74 |

Tablo 6.1. Veri Madenciliği Öncesi Algoritmaların Performansı

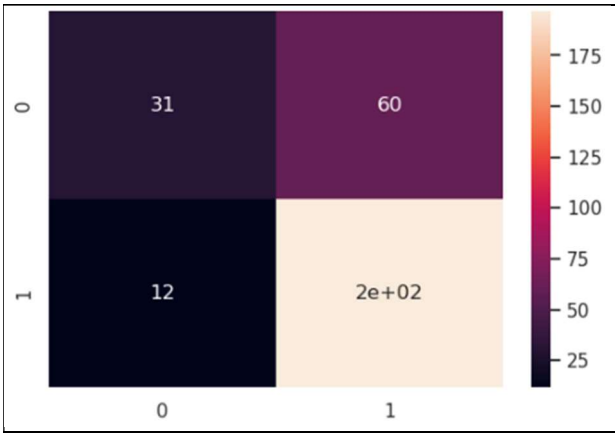
Veri seti üzerinde veri madenciliği teknikleri uygulanmaksızın çalıştırılan modeller arasında, Tablo 6.1'e göre yüzde 77 accuracy değeri ile en başarılı modelin XGBoost olduğu görülmektedir. Aşağıda yer alan şekilerde ise her bir modelin karmaşıklık matrisleri sunulmuştur (Şekil 6.1.a, b, c, d, e, f, g).



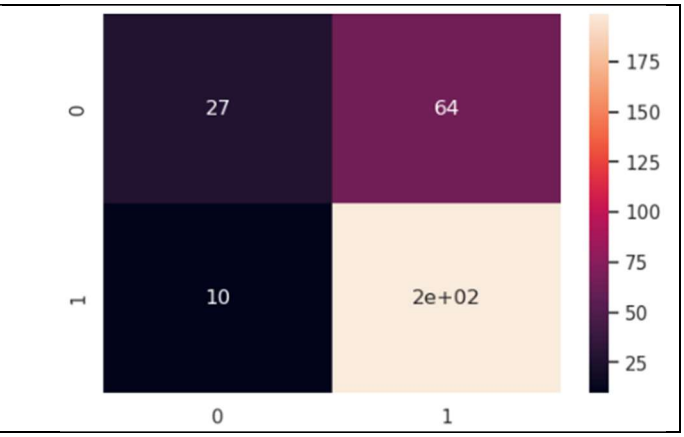

Şekil 6.1.a. XGBoost



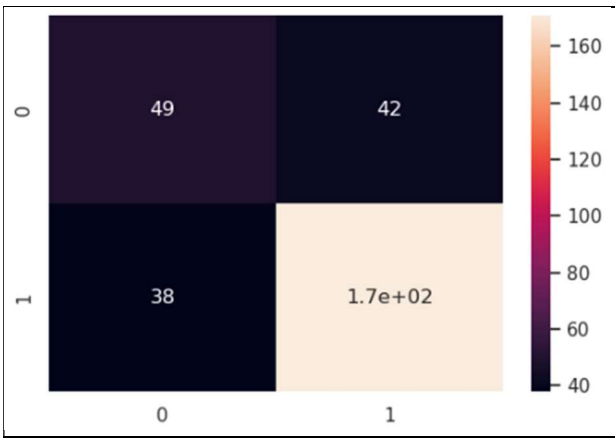

Şekil 6.1.b. Karar Ağacı



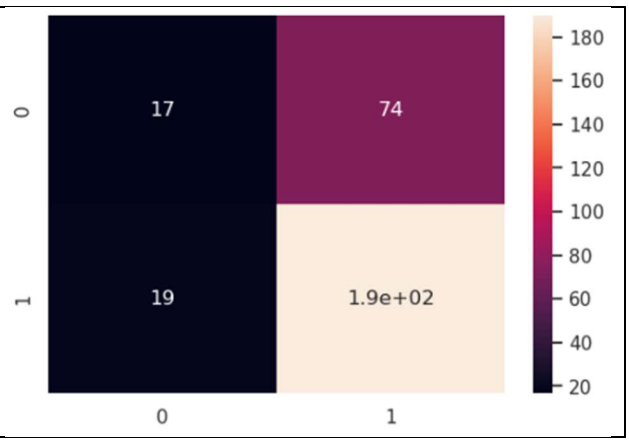

Şekil 6.1.c. Rassal Orman



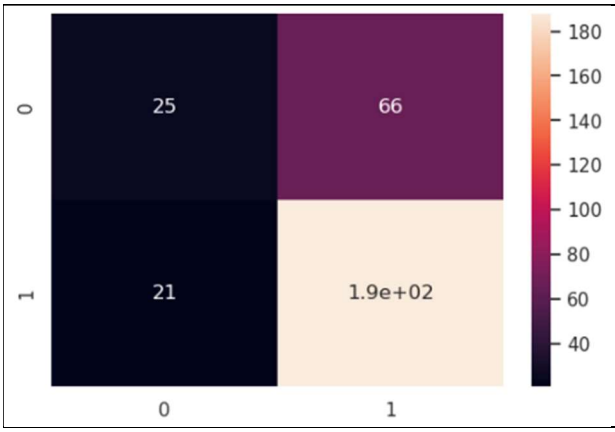

Şekil 6.1.d. Extra Ağaç




Şekil 6.1.e. Naive Bayes




Şekil 6.1.f. KNN




Şekil 6.1.g. Lojistik Regresyon

Rassal orman, extra ağaç, KNN ve lojistik regresyon algoritmalarının yanlış negatifleri, yanlış pozitiflerinden belirgin ölçüde fazla olduğu dikkat çekmektedir. Bunun anlamı söz konusu dört algoritma da kredi riski gerçekte iyi olan müşterileri kötü olarak yanlış sınıflandırmaya eğilimli olduğudur.

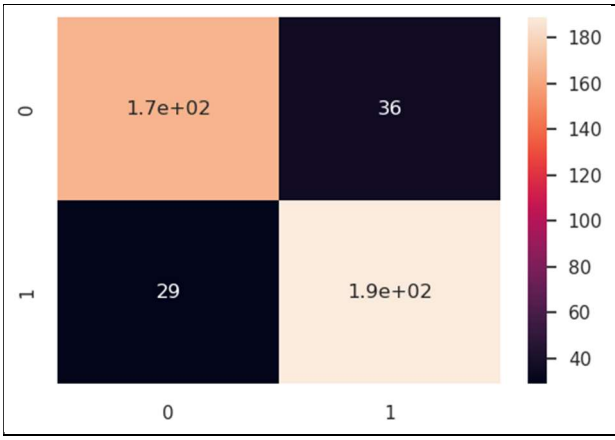
Veri madenciliği teknikleri kullanılarak normalizasyonun ve veri dengelemenin yapıldığı, yeni özelliklerin türetildiği, bazı özelliklerin kategorilerinin yeniden yapılandırıldığı, özellik seçimi ve boyut indirgeme yöntemleriyle 8 özelliğe düşürülen veri (eğitim) setiyle çalıştırılan 7 farklı modelin performans metrikleri Tablo 6.2'de sunulmuştur.

| | XGB | Decision Tree | Random Forest | Extra Trees | Naive Bayes | KNN | Logistic Regression |
|---|---|---|---|---|---|---|---|
| **Model** | XGB | Decision Tree | Random Forest | Extra Tree | Naive Bayes | KNN | Logistic Regression |
| **Accuracy** | 0.85 | 0.76 | 0.83 | 0.82 | 0.73 | 0.77 | 0.68 |
| **F1** | 0.85 | 0.76 | 0.84 | 0.82 | 0.74 | 0.77 | 0.69 |
| **Recall** | 0.87 | 0.72 | 0.85 | 0.79 | 0.75 | 0.75 | 0.67 |
| **Precision** | 0.84 | 0.81 | 0.82 | 0.85 | 0.73 | 0.79 | 0.71 |

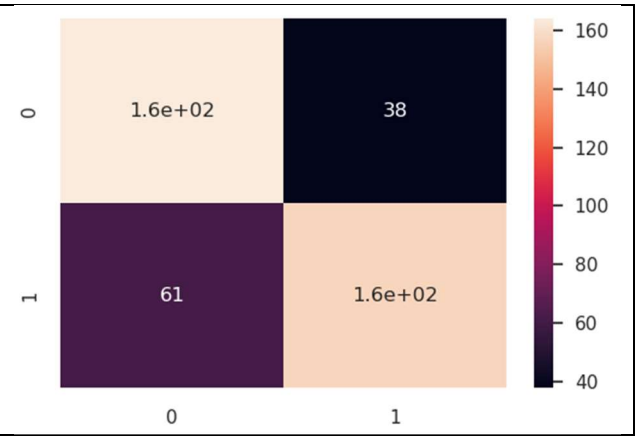
Tablo 6.2. Veri Madenciliği Sonrası Algoritmaların Performansı

Veri seti üzerinde veri madenciliği teknikleri uygulanarak çalıştırılan modeller arasında, Tablo 6.2'ye göre yüzde 85 accuracy değeri ile en başarılı modelin (bir önceki aşamada olduğu gibi) XGBoost olduğu görülmektedir. XGBoost özelinde ifade edilecek olursa, bir önceki aşamada 20 özellikle yapılan sınıflandırmanın accuracy değeri yüzde 77, veri madenciliği teknikleriyle sadece 8 özellik kullanılarak yapılan sınıflandırmanın accuracy değeri ise yüzde 85 olarak gerçekleşmiştir. Dolayısıyla veri madenciliği teknikleriyle

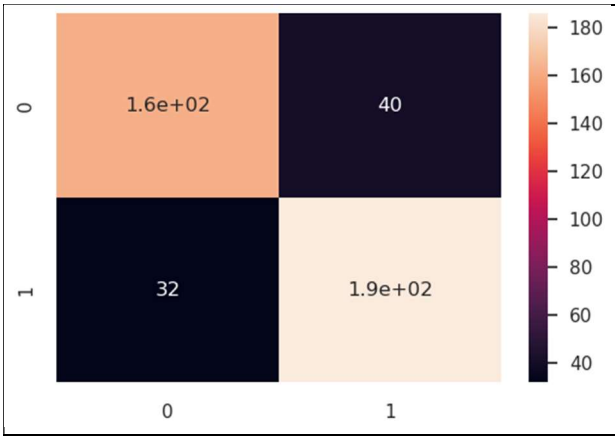
daha az sayıda özellik kullanılarak yapılan sınıflandırmanın daha iyi bir performansa sahip olduğu açıkça görülmektedir. Bu sonuç veri madenciliğinin modelleri sadeleştirmek suretiyle onların anlaşılabilirliğini arttırmanın yanında, algoritmaların performansı üzerinde önemli bir etkiye sahip olduğunu göstermektedir. Lojistik regresyon dışında diğer tüm algoritmaların sınıflandırma başarımlarının arttığı, özellikle de rassal orman, extra ağaç ve KNN algoritmalarının (bir önceki aşamaya göre) accuracy değerlerinin 7-8 puan arttığı görülmektedir. Aşağıda yer alan şekilerde ise veri madenciliği teknikleriyle işlenen verileri kullanan herbir modelin karmaşıklık matrisleri sunulmuştur (Şekil 6.2.a, b, c, d, e, f, g).



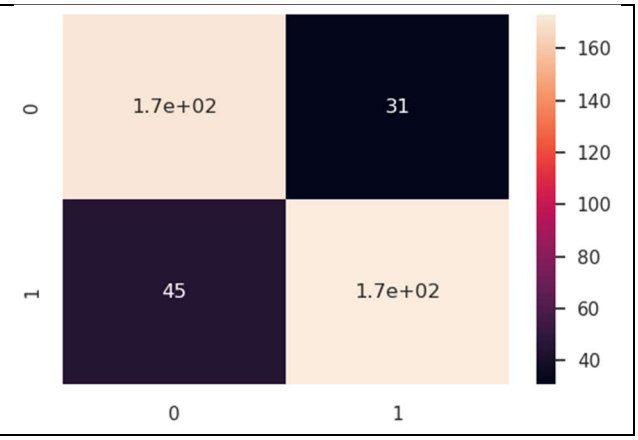

Şekil 6.2.a. XGBoost



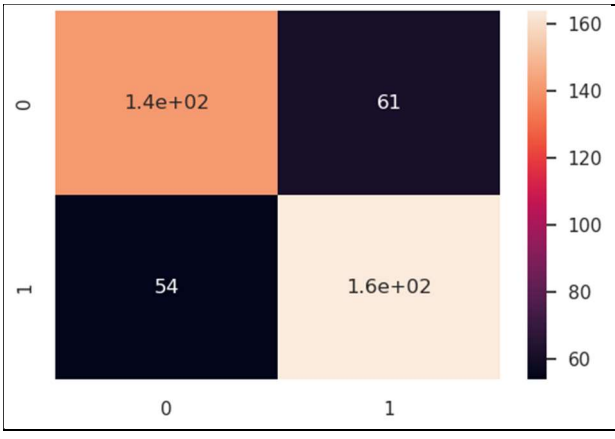

Şekil 6.2.b. Karar Ağacı



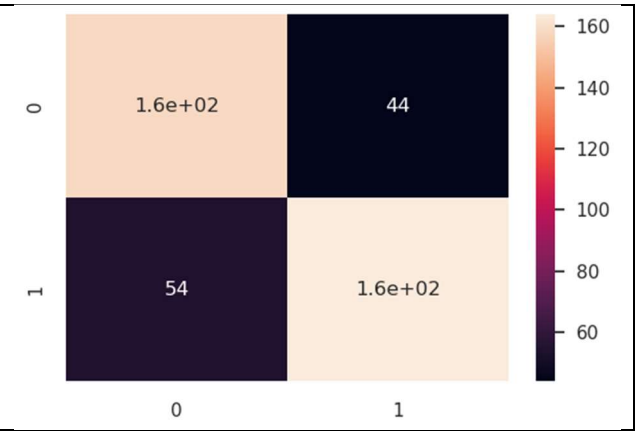

Şekil 6.2.c. Rassal Orman



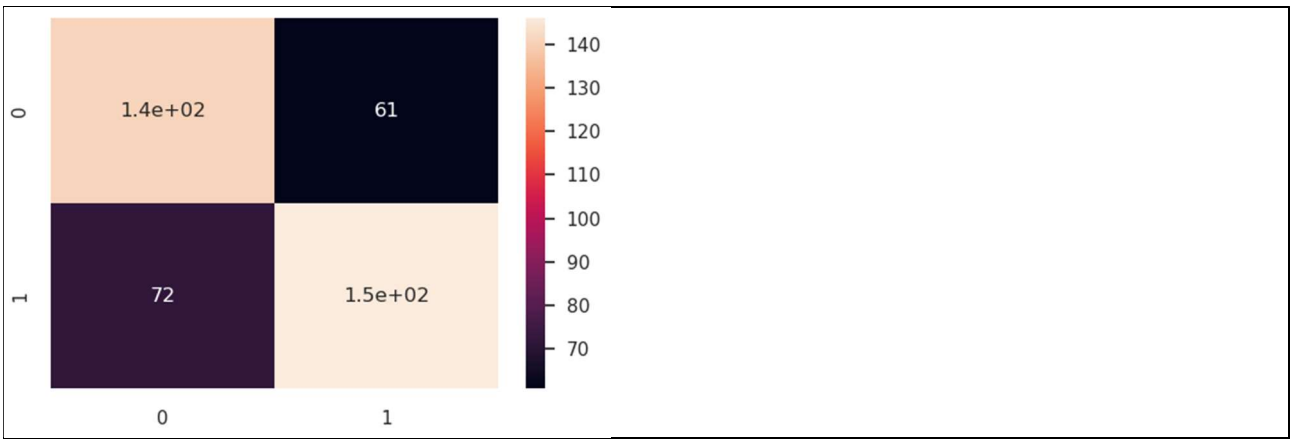

Şekil 6.2.d. Extra Ağaç




Şekil 6.2.e. Naive Bayes




Şekil 6.2.f. KNN




Şekil 6.2.g. Lojistik Regresyon

Veri madenciliği tekniklerinin kullanılmadığı duruma kıyasla, veri madenciliği tekniklerinin kullanıldığı durumda, karmaşıklık matrislerinde test setinin artan örnek sayısı, SMOTE yönteminin veri dengelemek amacıyla yapay örnekler oluşturmasından kaynaklanmaktadır.

Bir önceki aşamada yer alan karmaşıklık matrislerine (Şekil 6.1.a, b, c, d, e, f, g) göre veri madenciliği tekniklerine dayalı olarak elde edilen karmaşıklık matrislerinde (Şekil 6.2.a, b, c, d, e, f, g), yanlış negatif ve yanlış pozitiflerin dağılımının daha dengeli olduğu dikkat çekmektedir. Diğer taraftan yanlış negatif ve yanlış pozitif şeklinde sınıflandırılan eleman sayısı görünür de artmıştır. Ancak söz konusu mutlak artış SMOTE tekniğinin yapay örneklerinden kaynaklanmakta olup accuracy metriklerinde görülen yaklaşık 7-8 puana kadar olan iyileşme, modellerin başarımının oransal olarak iyileştiğine işaret eder.

## 7. SONUÇ VE DEĞERLENDİRME

XGBoost, karar ağacı, rassal orman, extra ağaç, naive Bayes, KNN ve lojistik regresyon modelleriyle yapılan çalışmada, “checking_status”, “purpose”, “savings_status”, “gender”, “credit_history”, “housing”, “age” ve “duration” şeklinde 8 özelliğin müşterilerin kredi riskini sınıflandırmada büyük ölçüde belirleyici olduğu tespit edilmiştir. XGBoost modeli yüzde 85 accuracy değeri ile en başarılı model olmakla birlikte, yüzde 83 ve 82 accuracy değerleri ile rassal orman ve extra ağaç algoritmalarının da oldukça başarılı olduğu ifade edilebilir.

Veri dengeleme, normalizasyon, özellik çıkarımı ve özellik seçimi gibi veri madenciliği teknikleriyle sadece 8 boyuta indirgenen veri setiyle yapılan sınıflandırmanın, 20 boyutlu veri setiyle yapılan sınıflandırmadan daha iyi bir performans gösterdiği gözlenmiştir. Korelasyon matrisi analiz edilerek çok sayıda gürültü (gereksiz özellikler) dışlanmıştır. Benzer şekilde birbiriyle korelasyonu yüksek olan özellikler tespit edilerek, bunlar arasında “class” sınıfıyla korelasyonu düşük olan özellikler veri setinden çıkarılmış, yüksek olan özellikler ise veri setinde bırakılmıştır. Diğer taraftan 8 farklı özellik üzerinde veri madenciliği teknikleriyle çalışılmış, içlerinden 5’inin sınıflandırma performansı üzerinde olumlu etkisi tespit edilmiştir. Min-max normalizasyon yöntemiyle algoritmaların yakınsama performansları (hızları) iyileştirilmiştir. Veri dengeleme (data balancing) tekniklerinden SMOTE tekniğinin (mevcut) müşteri kredi risk verileri üzerinde en başarılı veri dengeleme tekniği olduğu, dolayısıyla algoritmaların sınıflandırma performansları üzerinde önemli bir etkiye sahip olduğu sonucuna varılmıştır.

Sonuç olarak, veri madenciliği teknikleriyle boyutları düşürülen ve veri dengeleme işlemleri uygulanan veri setini kullanan modellerin accuracy performansında 8 puana kadar gözlenen önemli iyileşmeler olduğu görülmüştür. Dolayısıyla veri madenciliği teknikleriyle modellerin sınıflandırma performansları arttırılmış, problemin çözümünün anlaşılabilirliği ve sunumu kolaylaştırılmıştır. Bu çerçevede veri madenciliği tekniklerinin, algoritmaların potansiyel performanslarını açığa çıkarmada son derece önemli araçlar olduğu söylenebilir.

## KAYNAKLAR

[1] Yan-li, Z. ve Jia, Z. (2012). Research on Data Preprocessing In Credit Card Consuming Behavior Mining. Procedia Computer Science, 17 (2012), 638–643.

[2] Wang, Y., Zhang, Y., Lu, Y. ve Yu, X. (2020). A Comparative Assessment of Credit Risk Model Based on Machine Learning: A Case Study of Bank Loan Data. Procedia Computer Science, 174 (2020), 141–149.

[3] Ziemba, P., Radomska-Zalas, A. ve Becker, J. (2020). Client Evaluation Decision Models in the Credit Scoring Tasks. Procedia Computer Science, 176 (2020), 3301–3309.

[4] Ruyu, B., Mo, H. ve Haifeng, L. (2019). A Comparison of Credit Rating Classification Models Based on Spark-Evidence from Lending-club. Procedia Computer Science, 162 (2019) 811–818.

[5] Wang, K., Li, M., Cheng, J., Zhou, X. ve Li, G. (2022). Research on Personal Credit Risk Evaluation Based on XGBoost. Procedia Computer Science, 199 (2022) 1128–1135.

[6] Wang, H., Chen, W. ve Da, F. (2022). Zhima Credit Score in Default Prediction for Personal Loans. Procedia Computer Science, 199 (2022) 1478–1482.

[7] Bach, M. (2022). New Undersampling Method Based on the KNN Approach. Procedia Computer Science, 207 (2022) 3397–3406.

[8] Gupta, P., Varshney, A., Khan, M. R., Ahmed, R., Shuaib, M. ve Alam, S. (2023). Unbalanced Credit Card Fraud Detection Data: A Machine Learning-Oriented Comparative Study of Balancing Techniques. Procedia Computer Science, 218 (2023) 2575–2584.

[9] Murugan, M. S., T, S. K. ve Marappan, R. (2023). Large-Scale Data-Driven Financial Risk Management & Analysis Using Machine Learning Strategies. Journal Pre-proof, Measurement: Sensors (2023), doi: https://doi.org/10.1016/j.measen.2023.100756

[10] Okoro, E. E., Obomanu, T., Sanni, S. E., Olatunji, D. I. ve Igbinedion, P. (2022). Application of Artificial İntelligence in Predicting the Dynamics of Bottom Hole Pressure For Under-Balanced Drilling: Extra Tree Compared With Feed Forward Neural Network Model. Petroleum, 8 (2022) 227–236.

[11] Afriyie, J. K., Tawiah, K., Pels, W. A., Addai-Henne, S., Dwamena, H. A., Owiredu, E. O., Ayeh, S. A. ve Eshun, J. (2023). A Supervised Machine Learning Algorithm For Detecting and Predicting Fraud in Credit Card Transactions. Decision Analytics Journal, 6 (2023) 100163.

[12] Bhattacharjee, D., Ramesh, K., Jayaram, E. S., Mathad, M. S. ve Puhan, D. (2023). An Integrated Machine Learning and DEMATEL Approach for Feature Preference and Purchase Intention Modelling. Decision Analytics Journal, 6 (2023), 1–13.

[13] Mehrotra, D., Srivastava, R., Nagpal, R. ve Nagpal, D. (2021). Multiclass Classification of Mobile Applications as per Energy Consumption. Journal of King Saud University–Computer and Information Sciences, 33 (2021), 719–727.

[14] Zhu, X., Chu, Q., Song, X., Hu, P. ve Peng, L. (2023). Explainable Prediction of Loan Default Based on Machine Learning Models. Journal Pre-proof, Data Science and Management (2023), doi: https://doi.org/10.1016/j.dsm.2023.04.003

[15] Anshori, M. Y., Rahmalia, D. ve Herlambang, T. (2021). Comparison Backpropagation (BP) and Learning Vector Quantification (LVQ) on Classifying Price Range of Smartphone in Market. Journal of Physics: Conference Series, 1836 (2021), 1-10.

[16]https://www.kaggle.com/datasets/ppb00x/credit-risk-customers, Erişim Tarihi: 11.05.2023.

[17]https://www.kaggle.com/code/dimitriosthomaidis/xgboost-feature-engineering, Erişim Tarihi: 11.05.2023.

[18]https://www.kaggle.com/code/raphaelmarconato/credit-risk-eda-and-machine-learning-73, Erişim Tarihi: 11.05.2023.

[19]https://www.kaggle.com/code/subhranilmondal12/data-visualization-mlmodel-withcross-validation, Erişim Tarihi: 11.05.2023.