FEN BİLİMLERİ ENSTİTÜSÜ

İŞLETMELERİN İFLAS TAHMİNİNDE MAKİNE ÖĞRENMESİ ALGORİTMALARININ KARŞILAŞTIRMALI ANALİZİ

Gizem DİLKİ

Danışman
Prof. Dr. Özlem DENİZ BAŞAR

YÜKSEK LİSANS TEZİ
İSTATİSTİK ANABİLİM DALI
İSTANBUL - 2021

KABUL VE ONAY SAYFASI

Gizem DİLKİ tarafından hazırlanan "**İşletmelerin İflas Tahmininde Makine Öğrenmesi Algoritmalarının Karşılaştırmalı Analizi**" adlı tez çalışması 24/02/2021 tarihinde aşağıdaki jüri üyeleri önünde başarı ile savunularak, İstanbul Ticaret Üniversitesi Fen Bilimleri Enstitüsü **İstatistik Anabilim Dalı**'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman Prof. Dr. Özlem DENİZ BAŞAR .....
İstanbul Ticaret Üniversitesi

Jüri Üyesi Prof. Dr. Münevver TURANLI .....
İstanbul Ticaret Üniversitesi

Jüri Üyesi Doç. Dr. Selay GİRAY YAKUT .....
Marmara Üniversitesi

Onay Tarihi : 15/03/2021

İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsünün 15.03.2021 tarih ve 2021/308 numaralı Yönetim Kurulu Kararının 1. maddesi gereğince, ders yüklerini ve tez yükümlülüğünü yerine getirdiği belirlenen "Gizem DİLKİ" (TC:30574641460) adlı öğrencinin mezun olmasına oy birliği ile karar verilmiştir.

Prof. Dr. Necip ŞİMŞEK

Enstitü Müdürü

AKADEMİK VE ETİK KURALLARA
UYGUNLUK BEYANI

İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

15/03/2021

Gizem DİLKİ

İÇİNDEKİLER

ÖZET

Yüksek Lisans Tezi

İŞLETMELERİN İFLAS TAHMİNİNDE MAKİNE ÖĞRENMESİ ALGORİTMALARININ KARŞILAŞTIRMALI ANALİZİ

Gizem DİLKİ

İstanbul Ticaret Üniversitesi
Fen Bilimleri Enstitüsü
İstatistik Anabilim Dalı

Danışman: Prof. Dr. Özlem DENİZ BAŞAR
2021, 116 sayfa

İflas, işletmelerin finansal açıdan işlevini yerine getirememesi olarak tanımlanmaktadır. İflas sürece yaygın olarak gerçekleşir, bu durumda iflasın çeşitli modellemeler yardımıyla tahmin edilmesi mümkün olabilmektedir.

Bu çalışmada, öncelikle iflas tanımı ve iflas sürecine değinilmiştir. Ardından, iflas tahmin modellerinin tarihsel gelişimine yer verilmiştir. Gelişen teknoloji ile birlikte, veri saklama, saklanan veriyi işlemeye verilen önem üzerinde durulmuş, bu bağlamda makine öğrenmesi disiplininden bahsedilmiştir. Makine öğrenmesi, iflas tahmin problemine sınıflandırma algoritmaları kullanılarak uyarlanmıştır. Bu amaçla çalışmada Kaliforniya Üniversitesi veri tabanından alınan Polonyalı şirketler veri seti kullanılmıştır. Sınıflandırma algoritmaları olarak denetimli makine öğrenmesi algoritmalarından olasılık tabanlı Naive Bayes, tembel öğrenici k En Yakın Komşuluk ve istatistiksel öğrenme teorisi temelli Destek Vektör Makinesi kullanılmıştır. Veri ön işleme aşamasında Kanıt Ağırlığı ve Bilgi Değeri kriterleri yardımıyla değişken seçimi yapılmıştır. Veri setindeki dengesizliği azaltmak amacıyla SMOTE aşırı örnekleme yöntemi kullanılmıştır. Çalışmada duyarlılık, keskinlik, F puanı ve doğruluk değerleri ile ROC eğrisi ve AUC değeri hesaplanmıştır. İlgili performans ölçütleri ile karşılaştırılan üç algoritma arasından en başarılı sınıflama sonucunu veren algoritma Destek Vektör Makinesi algoritması olmuştur.

Anahtar Kelimeler: iflas tahmini, k en yakın komşuluk, naive bayes, destek vektör makineleri, smote, kanıt ağırlığı, bilgi değeri

ABSTRACT

M.Sc. Thesis

COMPARISON ANALYSIS OF MACHINE LEARNING ALGORITHMS IN BANKRUPTCY PREDICTION

Gizem DİLKİ

Istanbul Commerce University
Graduate School of Applied and Natural Sciences
Department of Statistics

Supervisor: Prof. Dr. Özlem DENİZ BAŞAR
2021, 116 pages

Bankruptcy is defined as the inability of companies to perform their functions financially. Bankruptcy widely occurs in the process; with this, it is possible to predict bankruptcy with the help of various modeling.

In this study, first of all, the definition of bankruptcy and the bankruptcy process are discussed. Then, the historical development of bankruptcy prediction models are included. With the developing technology, the importance given to data retention and processing stored data is emphasized and machine learning discipline is mentioned in this context. Machine learning is adapted to the bankruptcy prediction problem using classification algorithms. For this purpose, a data set of Polish companies taken from the University of California (UCI) database is used in the study. As classification algorithms from supervised machine learning algorithms, probability-based Naive Bayes, lazy learner k Nearest Neighborliness and Support Vector Machine based on statistical learning theory is used. Variable selection is made with the help of Evidence Weight and Information Value criteria during the data pre-processing phase. The SMOTE over-sampling method is used to reduce the imbalance in the data set. In the study, sensitivity, specificity, F score and accuracy values and ROC curve and AUC value are calculated. Of the three algorithms compared with the relevant performance criteria, the algorithm that gave the most successful classification result is the Support Vector Machine algorithm.

Keywords: bankruptcy prediction, k nearest neighbour, naive bayes, support vector machines, smote, weight of evidence, information value

TEŞEKKÜR

Lisans hayatımın ilk dersinden bugüne kendim ile özdeşleştirdiğim, disiplini ve mükemmeliyetçiliğini örnek aldığım, bana İstatistik bilimini sevdiren, hayatımın birçok evresinde önemli dokunuşları bulunan, danışmanım olmasından onur duyduğum değerli hocam Prof. Dr. Özlem DENİZ BAŞAR'a, bana olan inancı ve tez çalışmam boyunca gösterdiği destek için en içten teşekkürlerimi sunarım.

Ayrıca gerek lisans gerek yüksek lisans eğitimim boyunca değerli desteklerini hiçbir zaman esirgemeyen hocalarım Prof. Dr. Münevver Turanlı, Prof. Dr. Ünal Halit ÖZDEN ve Doç. Dr. Seda BAĞDATLI KALKAN'a teşekkürlerimi iletirim.

Hayatımın her evresinde olduğu gibi bu zorlu tez sürecim boyunca da beni yalnız bırakmayan sevgili aileme ve hem manevi hem de teknik desteğini hiçbir zaman esirgemeyen EMRE KURT'a gösterdikleri sonsuz destek ve sabır için teşekkür ve minnetlerimi sunarım.

Gizem DİLKİ

İSTANBUL, 2021

ŞEKİLLER

Sayfa

TABLOLAR

Sayfa

SİMGELER ve KISALTMALAR

AUC	Eğri Altında Kalan Alan (Area Under Curve)
D&B	Dun and Bradstreet
DRM	Deneysel Risk Minimizasyonu
DVM	Destek Vektör Makinesi
IV	Bilgi Değeri (Information Value)
kNN	k En Yakın Komşuluk (K-Nearest Neighborhood)
LA	Logit Analizi
MDA	Çoklu Diskriminant Analizi (Multiple Discriminant Analysis)
NB	Naïve Bayes
PA	Probit Analizi
RBF	Radyal Tabanlı Fonksiyon (Radial Basis Function)
ROC	Alıcı Çalışma Özellikleri Eğrisi (Receiver Operating Characteristic)
SMOTE	Sentetik Azınlık Aşırı Örnekleme (Synthetic Minority Over-sampling Technique)
VC	Vapnik - Chervonenkis
WOE	Kanıt Ağırlığı (Weight of Evidence)

1. GİRİŞ

Geçmişten günümüze kar amacıyla kurulan tüm organizasyonların ortak amacı, bulundukları pazarda yer edinmek ve devamlılık sağlamaktır. Bu amaçla firmalar, varlıklarını sürdürebilmek ve sektörde uzun dönemler boyunca faaliyetlerini devam ettirebilmek amacıyla çeşitli çalışmalar yapmaktadır. Firma sürekliliğini sağlamak amacıyla gerçekleştirilen çalışmaların büyük bir bölümünü firmaların operasyon ömrünü direkt olarak etkileyecek başlıklardan biri olan finansal sorunlar oluşturmaktadır.

Firmaların başarılı olmak ve uzun vadeli sürekliliklerini sağlamak için finansal sağlıklarını düzenli olarak değerlendirmeleri, olumsuz bir durum öngörüldüğünde hızlıca müdahale etmeleri gerekmektedir. Şirketlerin finansal koşullarını olumsuz yönde ve derinden etkileyen durumların sonucunda şirket iflasa sürüklenebilmektedir. Bu düşünceden hareketle, firmanın iflas etme potansiyelinin tahmini, şirketlerin geleceğe yönelik adımlarında en etkin rolü oynayan konulardan biri haline gelmiştir.

Maddi kayıplar, şirketlerin başta yöneticileri, yatırımcıları, hissedarları, ardından alacaklıları ve çalışanları hatta şirketin büyüklüğüne göre ülke ekonomisi için bile önemli bir yere sahiptir (Onan, 2015). Bu nedenle, iflas tahmini firmalar için bir endişe kaynağı haline gelmiş ve uzun dönemler boyunca konuyla ilgili akademik düzeyde çalışmalar yapılmıştır.

Çalışmanın ilk amacı, iflas tahmini problemini bir makine öğrenmesi problemine uyarlayarak ilgili dönemde firmaların iflas edip etmeyeceğini tespit etmektir. İkinci amaç ise, birden fazla makine öğrenmesi algoritması ile (Naive Bayes, k En Yakın Komşu ve Destek Vektör Makinesi) tahminde bulunarak çeşitli ölçütler üzerinden algoritmaların tahmin performanslarını karşılaştırmaktır.

Bu motivasyonla hazırlanan çalışmanın ilk bölümünde, literatürde yer alan çalışmalar incelenmiştir. İflas tahmini probleminde makine öğrenmesi algoritmalarının karşılaştırma analizi yapılan akademik çalışmalarda hangi algoritmanın daha iyi sonuç verdiğine değinilmiştir. Çalışmanın ikinci bölümünde, iflas tahmin modellerinin tarihsel gelişimine yer verilmiştir. Öncelikle iflas tanımı ve iflas süreci üzerinde durulmuştur. Ardından, 1930'lu yıllardan başlayarak gelişen iflas tahmin modelleri özelinde, teorik tahmin yöntemlerinden makine öğrenmesi algoritmaları yardımıyla çözümlenen iflas tahmin problemlerine uzanan gelişim süreci anlatılmıştır.

Çalışmanın üçüncü bölümünde, makine öğrenmesi kavramı detaylı olarak irdelenmiştir. Veri madenciliği motivasyonundan yola çıkılarak ortaya çıkan ve istatistik bilimi ile kesişen yapay zeka ile onun bir alt kümesi olan makine öğrenmesi problemleri aktarılmıştır. Makine öğrenmesi problemi özelinde kullanılabilecek algoritmalar tanıtılmıştır. Bölümün devamında, bu çalışma özelinde seçilen üç farklı algoritma Naive Bayes, k En Yakın Komşu ve Destek Vektör Makineleri teorik alt yapıları ile birlikte verilmiştir.

Uygulama bölümünde, araştırmada kullanılan veri seti tanıtılmış ardından araştırmanın metodolojisine yer verilmiştir. 5 farklı dönem için firmaların iflas tahminine yer verilen problem için WOE ve IV kriterleri ile değişken seçimi adımları anlatılmıştır. Veri setindeki hedef değişken oranından ötürü aşırı örnekleme çalışması yapılmış, bu adım sonrasında veri seti, eğitim ve test verisi olarak ayrılmıştır. Veri seti tüm sınıflandırma algoritmaları için %70 eğitim seti ve %30 test seti olacak şekilde bölünmüştür.

Teorik altyapıları aktarılan makine öğrenmesi algoritmalarının iflas tahmin problemine uygulanması ile ilgili adımlara yer verilmiştir. Çalışma öncesinde tanıtılan Doğruluk, Keskinlik, Duyarlılık, F1 Puanı, ROC eğrisi ve eğri altında kalan alan AUC değeri performans ölçütleri ile Naive Bayes, k En Yakın Komşu ve Destek Vektör Makinesi algoritmaları sınanmıştır. İlgili metrikler üzerinden analiz bulguları paylaşılmıştır.

Çalışmanın son bölümünde ise algoritmaları karşılaştırma işlemi gerçekleştirilmiştir. Algoritmalar karşılaştırıldıktan sonra, en yüksek doğru sınıflama performansına sahip makine öğrenmesi algoritması seçilmiş ve ileriki çalışmalar için öneriler sunulmuştur.

2. LİTERATÜR ÖZETİ

Son yıllarda makine öğrenmesi algoritmalarının üzerinde çokça durulması, modelleri eğitecek yeni açık kaynak kodlu ya da paket programların geliştirilmesiyle, önceleri teorik tabanlı çalışmaların yapıldığı iflas tahmini problemi makine öğrenmesi algoritmaları ile tahmin edilmeye başlanmıştır. Çalışmalar göstermektedir ki farklı veri setleri ve farklı algoritma, parametre seçimleri ile doğru tahmin etme oranı değişebilmektedir. İflas tahmini problemi üzerinden makine öğrenmesi algoritmalarının karşılaştırılması araştırıldığında öne çıkan bazı çalışmalar Tablo 2.1'de verilmiştir.

Tablo 2. 1. Literatür Özeti

Çalışmanın Sahibi, Yılı	**Karşılaştırılan Algoritmalar**	**En İyi Tahmini Sağlayan Algoritma**
(Chen vd., 2011)	• k En Yakın Komşu ve Parçacık Sürümü Optimizasyonu Hibrit Modeli • Destek Vektör Makineleri k En Yakın Komşu ve Çok Katmanlı Algılayıcı Hibrit Modeli	Yaklaşık eşit sonuçlar vermiştir.
(Olson vd., 2012)	• Yapay Sinir Ağları • Destek Vektör Makineleri • Karar Ağaçları	Karar Ağaçları
(Arieshanti vd., 2013)	• k En Yakın Komşu • Bulanık En Yakın Komşu • Destek Vektör Makineleri • En Yakın Komşu Destek Vektör Makineleri • Çok Katmanlı Algılayıcılar • Torbalama • Çok Katmanlı Algılayıcılar ve Çoklu Doğrusal Regresyon Hibrit Modeli	Bulanık En Yakın Komşu
(Zhou vd., 2014)	• Destek Vektör Makineleri • Genetik Algoritmalar	Destek Vektör Makineleri

(Gepp ve Kumar, 2015)	• Lojistik Regresyon • Diskriminant Analizi • CART Karar Ağacı • Cox Modelleri	CART Karar Ağacı
(Lu vd., 2015)	• Genetik Algoritma • Parçacık Sürüsü Optimizasyonu • Anahtarlama Parçacık Sürüsü Optimizasyonu ve Destek Vektör Makinesi Hibrit Modeli	Anahtarlama Parçacık Sürüsü Optimizasyonu ve Destek Vektör Makinesi
(Onan, 2015)	• Farklı Karar Ağacı Algoritmaları	Rastgele Orman
(Klepáč ve Hampel, 2016)	• Lineer Çekirdek Fonksiyonlu Destek Vektör Makineleri • Polinominal Çekirdek Fonksiyonlu Destek Vektör Makineleri • Radyal Tabanlı Çekirdek Fonksiyonlu Destek Vektör Makineleri • Rastgele Orman • Karar Ağaçları	Rastgele Orman ve Karar Ağaçları
(Zięba vd., 2016)	• Yapay Sinir Ağları • Destek Vektör Makineleri • Lojistik Regresyon • Rastgele Orman • Extreme Gradient Boosting	Extreme Gradient Boosting
(Alaka vd., 2018)	• Çoklu Ayırım Analizi • Lojistik Regresyon • Yapay Sinir Ağları • Destek Vektör Makineleri • Kaba Setler • Vaka Tabanlı Sınama • Karar Ağacı • Genetik Algoritmalar	Yapay Sinir Ağları ve Destek Vektör Makineleri
(Staňková ve Hampel, 2018)	• Lojistik Regresyon • Destek Vektör Makineleri • Sınıflandırma Ağaçları	Lojistik Regresyon

(Le vd., 2018)	• Rastgele Orman • Karar Ağaçları • Çok Katmanlı Algılayıcı • Destek Vektör Makinesi	Rastgele Orman
(Alexandropoulos vd., 2019)	• Yoğun Derin Sinir Ağları • Naive Bayes • Lojistik Regresyon • CART Karar Ağacı • Çok Katmanlı Algılayıcı	Yoğun Derin Sinir Ağları
(Mai vd., 2019)	• Lojistik Regresyon • Destek Vektör Makineleri • Rastgele Orman	Rastgele Orman
(Korol, 2019)	• Bulanık Kümeler • Çok Katmanlı Yapay Sinir Ağları • Tekrarlayan Yapay Sinir Ağları • Karar Ağaçları	Bulanık Kümeler
(Fernández-Gámez vd., 2019)	• AdaBoost • Naive Bayes • C4.5 • Çok Katmanlı Algılayıcılar • Destek Vektör Makineleri	Naive Bayes
(Horak vd., 2020)	• Destek Vektör Makinesi • Yapay Sinir Ağları	Sinir Ağları

3. İFLAS TAHMİN MODELLERİ

İflas, firmaların faaliyetlerinin durmasına ve bulundukları sektörde varlık gösterememesine neden olan negatif bir durumdur. Firmaların iflası zaman içerisinde gerçekleşmektedir. Bu nedenle iflas, farklı etmenlere bağlı olarak gelişen, farklı türlerde karşımıza çıkabilen bir durum halini almıştır.

3.1. İflas Tanımı

İşletmeler, ürettikleri mal ya da sundukları hizmetler ile gerek yerli gerek ulusal gerekse küresel boyutta ekonomiye yön veren önemli organizasyonlardan biridir. Geçmişten günümüze tüm şirketlerin hizmet verilen sektör ya da şirket büyüklüğü gözetilmeksizin ekonomiye katma değer sağlayabilecek yapılardan biri olduğu düşünülmektedir. Bu nedenle firmalar, varlıklarını devam ettirebilmek, piyasada uzun dönemler boyunca yer edinebilmek amacıyla birtakım stratejik çalışmalar yapmaktadır. Çalışmaların büyük bir bölümünü firmaların yaşam süresini direkt etkileyecek konulardan biri olan finansal konular oluşturmaktadır. Firmalar, mali durumlarını düzenli olarak gözden geçirme ihtiyacı hissetmekte, finansal sağlıklarını ölçümlemektedir. Finansal sağlıklarıyla en çok ilgilenenler genelde organizasyon içinde hissedarlar, memurlar, yöneticiler, çalışanlar ve iç denetçiler; organizasyon dışında ise bankalar, müşteriler, alacaklılar ve tedarikçiler olmuştur (Shariq, 2016).

Şirketlerin olumsuz finansal durumları finansal başarısızlık, tasfiye, iflas gibi çeşitli terimlerle ifade edilmektedir. Bu terimler iç içe olmakla birlikte farklı anlamlara gelmektedir. Bu tanımlar arasından, şirketin finansal olarak negatif durumunu nitelendirirken kullanılabilecek en ağır terim iflastır. İflas farklı şekillerde tanımlanabilir.

Finansal sıkıntıların kesin ayrımı henüz yapılamamakla birlikte teorik olarak bakıldığında finansal sıkıntı farklı derecelere sahiptir. Örnek olarak, nakit akışını yönetmekte zorluk yaşayan firmalar hafif finansal sıkıntı yaşayan firmalar

olarak kabul edilir; ancak borçlarını ödeyememe, temerrüt gibi durumların konu olduğu finansal sıkıntılar yüksek finansal sıkıntılardır ve iflas olarak adlandırılır (Shi ve Li, 2019). Literatürde iflas tahmin modeli çalışmaları başlatıldığından bugüne, şirketlerin finansal sağlığına yönelik araştırmalar temelde iflas tanımına ve finansal tahminine odaklanmıştır. Çoğu zaman, çalışmalarda görev alan araştırmacılar başarısız ve başarısız olmayan firmaları ayırt eden çizgi olarak nihai başarısızlık terimi kullanarak iflas etme durumunu referans vermişlerdir.

Finansal başarısızlık terimi, uzun yıllar önce, tatmin edici olmayan iş şartlarını tanımlamak için işletmelerle ilgili istatistikler sağlayan Dun and Bradstreet (D&B) firması tarafından benimsenmiştir. D&B'ye göre "atama veya iflas sonrasında faaliyeti durdurulan işletmeler; bu tür eylemleri icra, haciz veya sonra alacaklılara zarar ile sona erenler; gönüllü olarak geri çekilenler, ödenmemiş yükümlülükleri terk edenler veya alıcılık, iflas yeniden yapılanması veya düzenleme gibi mahkeme işlemlerine dahil olanlar, alacaklılarla gönüllü olarak uzlaşmaya çalışanlar" iş başarısızlıkları yaşayan firmalar arasında yer alır. Hem yeniden yapılandırma hem de tasfiye dünyanın birçok ülkesinde mevcut eylem durumlarıdır ve şu önermeye dayanmaktadır: Bir varlığın içsel veya ekonomik değeri mevcut tasfiye değerinden büyük ise, hem bir kamu politikası hem de taraf mülkiyeti açısından, firmanın yeniden örgütlenmeye devam etmeye çalışmasına izin verilmelidir. Ancak, eğer firmanın varlıkları "canlıdan daha fazla ölü değerinde" yani ekonomik gidiş-endişe değeri tasfiye değerini aşıyorsa tercih edilen bir alternatiftir (Altman ve Hotchkiss, 2006).

İflas, şirket sahiplerinin borçlarını belirlenen süre zarfında alacaklılara ödeyememesi ile meydana gelen hukuki bir süreç olarak tanımlanabilir. İflas kendi içinde birden fazla türe ayrılmaktadır. İlk tip iflas, firmanın öz sermaye yani toplam borç ile alacaklar arasındaki fark durumuna göre değerlendirilir. Daha gözlemlenebilir olan ikinci tip, şirketin bölge mahkemesinde yaptığı resmi beyanla, mal varlığını tasfiye etmek ya da bir kurtarma programına teşebbüs etmek için bir dilekçe sunmasını ifade eder (Altman ve Hotchkiss, 2006).

Literatürde yer alan bir diğer çalışmaya göre iflas türleri üç ana başlığa ayrılmıştır. Bunlardan ilki olan yasal iflas, şirketin iflas beyanı için mahkemeye gittiği durumda oluşan iflastır. Şirketin anapara ve faizi geri ödemek için sözleşmeyi zamanında yerine getirememesi durumunu açıklayan türe ise teknik iflas denir. Üçüncü olarak, muhasebesel iflas olarak tanımlanan, şirket muhasebe kayıtlarına sadece negatif varlıklar girişi yaptığı durumu anlatan bir iflas türü bulunmaktadır (Ross vd., 2010).

Bir şirket kısa vadeli borcunu yerine getiremezse, teknik olarak iflas etmiş sayılır. Teknik iflas, likidite eksikliği gösterir ancak şirketin iflas ettiğini belirtmez. Yükümlülükleri yerine getirememe durumu geçici olabilir; firma borçlarını bir süre sonra gidererek şirket faaliyetlerine devam edebilir. Altman'a göre iflas, şirketin toplam borcunun tutulan tüm varlıkların değerini aşması halinde ortaya çıkan uzun vadeli bir durum olarak nitelendirilir. İflas sadece önemli zorunlu ödemeleri geri ödeyememek değil, aynı zamanda bir işletmenin toplam yükümlülüklerinin muhasebe görünümünden toplam varlıklarını aştığı anlamına gelen negatif net varlık değerinin durumunu da içerir (Zopounidis ve Doumpos, 1999).

3.2. İflas Süreci

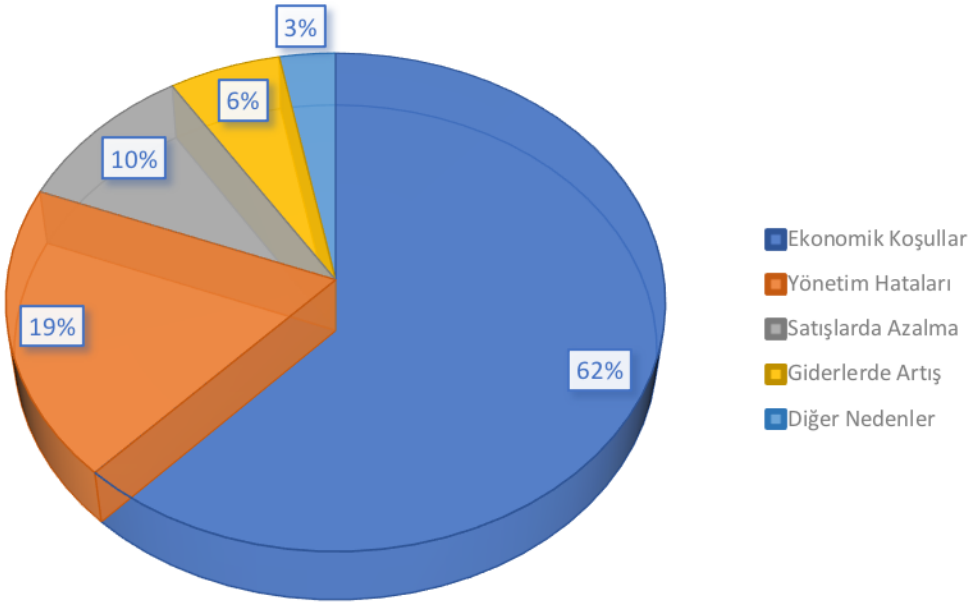
Şirketlerin iflas süreci temel olarak dört adımdan oluşmaktadır. İlk aşamada şirketi başarısızlığa götüren nedenler baş gösterir. İkinci aşamada bu başarısızlık artık ölçülebilir ya da hissedilebilir evreye gelir. Üçüncü evre ise iflasın tam olarak yaşandığı gelişim evresidir. Son aşama olarak tanımlanabilecek kurtarma evresinde ise iflastan dönüş ya da tamamen yok olma ihtimali bulunmaktadır.

Bu dört aşamadan oluşan şirket başarısızlığı süreci arasında yönetimin attığı yanlış adımlar, şirket politikasındaki yanlış adımlar ve dış etkenlerin iflasın

varlığı ve tespiti açısından büyük önemi vardır (Burksaitiene ve Mazintiene, 2011).

3.2.1. İflasın nedenleri

İflas başvuruları genellikle yönetim ve şirket politikasındaki hatalardan kaynaklanır. Bu hatalar yatırımcılar, çalışanlar, ortaklar ve hatta toplum üzerinde büyük bir etkiye yol açabilir. İflas sonucunda ortaya çıkan tablo genellikle yüksek maliyetli olmaktadır (Ooghe ve De Prijcker, 2008).



Şekil 3. 1. İflasın Nedenleri (Zhou ve Elhag, 2007)

Şekil 3.1'de D&B'nin 1987'de yayınladığı çalışmaya göre iflas nedenlerinin oransal dağılımları verilmiştir. Yapılan çalışmaya göre iflas nedenleri beş başlıkta incelenebilmektedir. %62'lik oran ile ekonomik koşullar iflas nedenleri arasında ilk sırada yer almaktadır. İflasa neden olan ikinci problem başlığı %19 oran ile yönetimden kaynaklı durumlar, üçüncüsü %10 oran ile satışların azalmasından kaynaklı yaşanabilecek sıkıntılardır. Dördüncü olarak %6 pay ile giderlerdeki artış ve geriye kalan %3'lük kısım ise diğer nedenleri

kapsamaktadır. Bu durumda ekonomik ve yönetici kaynaklı nedenler iflas olasılığının temelini oluştururlar (Zhou ve Elhag, 2007).

Şirketlerin başarısızlık nedenleri incelendiğinde, ilk olarak karşılaşılan durum ekonomik nedenler, diğer bir deyişle kaynak sıkıntısıdır. İşletme, kıt kaynaklarla, en çok üretimi gerçekleştirmeyi amaçlayan bilim dalı olarak bilinmektedir. Şirket başarısızlığı, bir şirketin kaynakları mikro ve makro ortam gereksinimlerine yanıt vermek için yeterli olmadığında başlar. Bu nedenle şirket değerli bir stratejik konum yaratmada veya sürdürmede başarılı olamaz (Thornhill ve Amit, 2003). Şirket kaynakları operasyonun devamı için yeterli olmadığı durumlarda stratejik bir konum oluşturamaz veya mevcut statülerini koruyamazlar.

Firmaları iflasa sürükleyen ikinci temel neden de yöneticilerin ve dış çevrenin etkisidir. Şirket yöneticileri şirketin kaynaklarının nasıl bölüştürüleceğinin belirlenmesinde önemli rol oynar (Keats ve Bracker, 1988). Yöneticilerin bilgi birikim ve karar vermedeki yetersizliği şirketi mevcut iç ve dış durumlara adapte edememe, stratejik kararlar alamama, pazarlama ve operasyonel anlamda yetersizlikler, muhasebe ve finans alanında kısıtlı bilgi ve beceriler, faaliyet ve maliyetlerin kontrol edilememesi gibi durumları beraberinde getirir. Diğer taraftan, yöneticinin şirkete bağlı olmaması, (yöneticinin kişisel çıkarları ile şirket çıkarlarının örtüşmemesi) şirketi başarısızlığa götüren diğer bir neden olarak gösterilebilir. Kişisel çıkarlar ve işletme çıkarları örtüşmediği durumlarda yöneticiler motivasyon kaybı yaşamakta ve koordinasyon yetilerini kaybetmektedirler. Yönetim tarzı, yöneticinin çok yüksek risk alma veya esnek olmayan karar almaya yönlendiren kişiliği, ortaklarla ilişkide güçlü etkisi olan diğer faktörlerdir.

Şirketleri iflasa sürükleyen dış nedenler arasında kriz de gösterilebilir. Kriz, kaynakların (özellikle zamanın) durumla başa çıkmak için yetersiz olduğu durum olarak tanımlanabilir (Starbuck ve Hedberg, 2001). Bu nedenle krizler şirket organizasyonları için büyük bir tehdit oluşturur. Kriz, büyüklüğü,

etkilediği sektör, ticari yayılım alanı gibi değişkenlere göre şirketlere çeşitli zararlar verebilir (Mıshra, 1996). Bu nedenle finansal krizler de iflas sayısının artması durumunda ülkenin ekonomik durumunun nasıl etkilenebileceğini gösteren en önemli örneklerden biri olmuştur. Özellikle 2008 yılındaki dünya çapındaki finansal krizden sonra, araştırmacılar krizi iflasın bir nedeni olarak görmeye başlamışlardır (Shi ve Li, 2019).

3.2.2. İflas tespiti

Şirketin finansal durumunun bozulması gözle görülür bir aşamaya geldiğinde iflas sürecinin ikinci aşamasına geçilmiş olur. İlk etapta iflas sinyallerini almak amacıyla finansal parametreler üzerinden değerlendirmeler yapılır. Kritik sorunlara çözüm bulmak için düzeltici önlemler alınmazsa, şirketin durumu daha da kötüleşir. Şirket ve çevresi arasındaki büyük uyumsuzluk olarak, şirketin stratejik konumu daha yoksul olur ve şirket sürecin ikinci aşamasına girer (Argenti, 1976).

Literatürde firmanın iflas edip etmeyeceğini tahmin etmek amacıyla geliştirilmiş birçok model bulunmaktadır. Modeller çoğunlukla bilanço kalemleri ile çalışmaktadır. Bunun dışında yönetici ve şirket üst düzey yetkililerinin şirket içi durumları değerlendirmesi de iflasın belirlenmesinde etkin rol oynayabilir.

Aşağıdaki durumlar iflas için erken uyarı sinyali rolü taşıyabilir.

- Yönetim kurulu üyelerinin istifaları
- Kredi limitinin azaltılması
- Adi hisse senedi depresif bir pazarda veya defter değerinden daha düşük bir değere satılması
- Şirket yöneticilerinin hisse senedi satması
- Varlıklarda azalma
- Şirketin, mali durumdaki düşüşü göz ardı etmesi
- Yeni ürünler ile rekabet pazarına girmesi
- Eski moda ürünlerin satılması

- Araştırma ve geliştirme bütçesinin orantılı olarak rakiplerinden daha düşük olması
- Aşırı stok

3.2.3. Kurtarma

Şirketler aynı nedenlerle iflasa sürüklenmedikleri gibi aynı süreçlerle de iflastan kurtulamazlar. Başarısız şirketlerin bu durumda başvurabilecekleri iki yol vardır: iflas etmek veya toparlanmak. İflas ani bir olay değildir, engellenebilir. Ancak, iflas sürecinden finansal durumlar ile yönetim ve genel organizasyon unsurları kritik derecede etkilenirse süreç şirketin iflası ile sonuçlanır. Öte yandan, iflas sürecinde yer alan şirketler iflas, tasfiye veya birleşme gibi farklı şekillerde ortadan kaybolabilir (Balcaen ve Ooghe, 2006). Bununla birlikte, iflas sürecinin erken aşamalarında meydana gelen temel sorunlara çözümler üretilerek düzeltici eylemler ile durumun kurtarılması mümkündür (Argenti, 1976). Düzeltici eylemler krizi istikrarla yönetmek, paydaşların desteği, strateji odaklı hareket etme, finansal yapılanma olarak sıralanabilir. Ancak unutulmamalıdır ki iflasın temel nedenlerini gideren yalnızca uzun vadeli düzeltici eylemler kalıcı bir iyileşmeye yol açabilir. Aksi halde mevcut durumu kurtarma çözümleri başarılı olmadığında iflas kaçınılmazdır.

Bu bilgilerin ışığında, gelecekte bir süre içinde bir şirketin finansal sağlığını başarılı bir şekilde tahmin edebilme ihtiyacı açıkça görülmektedir. Erken uyarı, paydaşların zararı en aza indirmesine, hatta iflası tamamen önlemede yardımcı olabilir. Firmaların finansal durumlarını incelemede genel olarak bilanço kalemleri, finansal oranlar incelenmektedir. Gelir tablosu bilgileri firma faaliyetleri hakkında bilgi verirken, bilanço belirli bir zamanda işletmenin varlık ve yükümlülüklerini ortaya koyar. Yatırımcılar hisse, tahvil gibi araçlarla işlem yapmak istediklerinde, alacaklılar ise borç verme kararları vermek için muhasebe bilgilerini kullanırlar. Diğer bir deyişle, işletmelerin finansal kapasitesini değerlendirme finansal sağlamlığın ölçülmesi kararlarını şekillendirir (Shariq, 2016).

3.3. İflas Tahmin Yöntemleri

İflas tahmini, 1932'de Fitzpatrick tarafından yapılan çalışmalardan bu yana, finansal tahminleme çalışmaları arasında en zorlu çalışmalardan biri olmuştur. 1930'lardan günümüze, neredeyse yaklaşık bir asırdır devam eden bu araştırma konusunda birçok farklı yaklaşım sunulmuştur

Şirketlerin iflas tahminlerinin araştırılmaya başlandığı 1930'lu yıllarda araştırmacılar için gelişmiş metotlar ya da bilgisayar programları bulunmamaktaydı. Genelde, iflas etmiş ve iflas etmemiş firmaların finansal oranları birbirleri ile karşılaştırılmaktaydı. Bu karşılaştırmalar sonucunda, iflas eden firmaların finansal oranlarının değerlerinin iflas etmemiş firmalara göre daha kötü oldukları ortaya çıkarılmıştır.

1966'da Beaver'in öncü çalışması tek değişkenli analiz ve 1968'de Altman'ın çok değişkenli analizi ile iflas konulu çalışmalarda tahminlemeye yönelik istatistiksel çalışmaların devri başlamıştır. Uzun yıllar boyunca iflas tahminlemelerinde istatistiksel tabanlı modellere yer verilmiştir ancak, istatistiksel yöntemlerin kullanımında sağlanması gereken varsayımların göz ardı edilmesi nedeniyle tahminlerden sapmalar meydana gelmiştir.

İstatistiksel modellerin katı varsayımları olması, elde edilen verilerin bu varsayımları karşılayamaması nedeniyle farklı metotlara yönelim başlamıştır. Paralelinde teknolojide gelişmeler, verilerin saklama biçimi ve boyutlarının değişmesi, yazılım alanındaki gelişmeler ile istatistiksel yöntemler yerini yapay zekaya dayalı modellere bırakmıştır.

Kronolojik sıralaması ile bakıldığında iflas tahmini modelleri teorik modeller, istatistiksel modeller ve yapay zeka tabanlı modeller olarak üç gruba ayrılmaktadır. Bu modellerin temel özellikleri ile Tablo 3.1'de verilmiştir (Aziz ve Dar, 2006).

Tablo 3.1. İflas Tahmin Modellerinin Karşılaştırılması (Aziz ve Dar, 2006)

Teorik Tabanlı İflas Tahmin Yöntemleri
• Başarısızlığın nitel nedenlerine odaklanır. • Teorinin önerdiği başarısızlık argümanını tatmin edebilecek bilgiler ile şekillenir. • Çok değişkenli yapıda bulunur. • İstatistiksel teknikleri nicel yöntemleri destekleyecek şekilde kullanır.
İstatistiksel Tabanlı İflas Tahmin Yöntemleri
• Başarısızlık belirtilerine odaklanır. • Şirket hesaplarından çekilen bilgiler ile şekillenir. • Tek değişkenli veya çok değişkenli (daha yaygın) olabilir. • Klasik standart modelleme prosedürlerini izler.
Yapay Zeka Tabanlı İflas Tahmin Yöntemleri
• Başarısızlık belirtilerine odaklanır. • Şirket hesaplarından çekilen bilgiler ile şekillenir. • Genellikle çok değişkenli yapıda bulunur. • Teknolojik ilerleme ve sonuç üretmenin sonucu bilgisayar teknolojisine bağlıdır.

3.3.1. Teorik tabanlı iflas tahmin yöntemleri

İflas tahmininde kullanılan teorik modeller, bu alanda yapılan çalışmalarda ilkleri oluşturmaktadır. Teorik modellerin amacı, finansal pozisyonların nedenini belirlemeye çalışmaktır. Bu modeller genel olarak çok değişkenlidir ve iflasın nedenini matematiksel oranlar ile öngörmek amacını taşımaktadır (Klepáč ve Hampel, 2018).

1932'de FitzPatrick'in 13 farklı finansal oranı kullanarak başarılı olan ve iflas eden firmaları karşılaştırması ile teorik iflas tahmin modelleri devri başlamıştır. Teorik modeller arasında en yaygın olarak kullanılan yöntem Oran Analizi (Ratio Analysis) olarak bilinmektedir. Oran analizi, nakit akışı (likidite-varlık akışı) temeline dayanmaktadır. Uygulamada nakit akışına yer verilmesinin amacı modeli oranları ile optimal bir dizilim geliştirmektir (Beaver, 1966). Elde edilen diziler incelenerek iflas etme potansiyeli olan firmaların finansal oranlarının benzer olup olmadığı araştırılmıştır. Bu yöntemde, deneysel

sonuçların maruz kaldığı örüntüyü açıklamak yerine, her deneysel sonucun kendi içinde değerlendirilmesi gerektiği savunulmaktadır

İflas tahmininde kullanılan başlıca teorik modeller aşağıda verilmiştir.

- **Nakit Yönetimi Teorisi**

Geçmiş ve mevcuttaki nakit akışlarının finansal durumu açıkça gösterebilmesi durumunda, gelecek dönemlerdeki finansal durumu belirleyerek iflas riskini tahmin edilebilmesi düşüncesi ile ortaya çıkmıştır. Nakit yönetimi teorisi, pozitif nakit akışı olan firmaların sermayelerini yükseltip sermaye piyasasından borç alabildiği; negatif veya yetersiz nakit girişi olan firmaların ise borç alamadığı ve bu nedenle iflas riski ile karşı karşıya olduğu durumları açıklayan iflas tahmin modelidir. Buradan hareketle, firmanın cari yıl kârı, nakit akışı borç yükümlülükleri veya cari yıllık kârının toplamı ile beklenen öz kaynak değerinin negatif olduğunda iflas ettiği varsayılmaktadır. Firmanın gelecekteki nakit akışları sermaye arttırmak için öz kaynak piyasasına girme becerisini etkileyebilir; bu nakit akışları doğrudan temettü şeklinde ödenmez ve karlı projelere yeniden yatırılabilir. Hissedarlar yöneticilerin nakit tutmalarına izin verirken, kârsız veya negatif projelere yatırım yapabilir (Scott, 1981);(Zeitun vd., 2007).

- **Likidite, Karlılık ve Varlık Analizi**

Bu teori, finansal oranların bir firmanın sağlığının göstergeleri olarak algılanmasını temel alır. Göstergelerin üç ana kategorisi likidite, karlılık ve varlıklardır. Firmanın göstergeleri "iyi" olduğunda sağlıklı olarak algılanır, ancak göstergeler kötüyse sağlıksız ve iflas riski olarak algılanır. Ancak bu analiz genel bir analiz olduğundan tek başına kullanıldığında zayıf bir tahmin edici olarak karşımıza çıkar, bu nedenle farklı teoriler ile birlikte kullanılır (Lim vd., 2012)

- **Bilanço Kompozisyonu Ölçümü**

Firmanın tüm varlık ve borçlarını tek seferde bir bütün halinde gösteren en güvenilir kaynak bilançodur. Bu nedenle bilançoda meydana gelen değişiklikleri

gözlemlemek firmanın finansal gidişatı konusunda bilgi verebilmektedir. Bir firmanın finansal tabloları bilançodaki varlık ve borçlarının bileşiminde önemli değişiklikler yansıtıyorsa, denge halinde kaymalar meydana gelebilir. Bu değişikliklerin gelecekte kontrol edilemez hale gelme potansiyeli taşıyor ise firmanın olası bir iflas riski ile karşı karşıya kalabileceği öngörülebilmektedir (Aziz ve Dar, 2006).

- **Gambler's Ruin Teorisi**

Gambler (kumarbaz) teorisi, belirli stokastik süreçlerin sonucunu tahmin etmek için kullanılan matematiksel araçlar olan rastgele yürüyüşlerin klasik bir örneğidir (Harik ve Goldberg, 1999). Teorinin arkasındaki motivasyon şu şekilde açıklanabilir: kumarbaz olarak atfedilen oyuncunun p olasılıkla 1 dolar kazandığı 1-p olasılıkla da 1 dolar kaybettiği bir dünyada, elinde bir miktar (X) dolar ile oyuna devam ederse, elindeki para N (N>X) dolara çıkarana kadar ya da hiç parası kalmayana kadar oyuna devam edecektir. Kumarbazın hedeflediği N değerine ulaşmadan oyunu bitirmeyeceği bilindiğine göre, kumarbazın hedeflediği parayı (N) yakalama olasılığı hesaplanmaya çalışılır (Sigman, 2009). Kumarbaz teorisinin iflas tahminine uygulanmasında ise iflas olasılığı hesaplanmak istenmekte ve bu olasılık likit kaynakların giriş (p) ve çıkışlarına (1-p) dayanmaktadır (Scott, 1981).

- **Merton Modeli**

Merton modelinde firmanın, tahvil sahiplerine vade t'de B ödemeyi vaat ettiği varsayılır. Bu ödeme yerine getirilmezse, yani firmanın değeri B'den küçükse, tahvil sahipleri şirketi devralır ve hissedarlar herhangi bir pay alamaz. Bu durumda firmanın varlık değerinin bilinmesi tahvil paylaşımı açısından büyük önem taşımaktadır; Varlıkların değeri belirli bir eşiğin (varsayılan nokta) altına düştüğünde, firma temerrüde düşer (Tudela ve Young, 2003). Modele göre, firmanın gelecekteki varlık değeri, beklenen değeri ve standart sapması ile karakterize bir olasılık dağılımına sahiptir. Varlıkların gelecekteki değerinin standart sapma sayısı varsayılan noktadan uzaktır ve temerrüt olasılığı daha düşüktür (Lim vd., 2012).

- **Kredi Riski Teorisi**

Kredi riski teorileri çoğunlukla finansal firmalara ve Basel I-Basel II gibi anlaşmalara bağlıdır. Kredi riskinde gözetilen durum herhangi bir borçlunun, herhangi bir nedenle, temerrüde düşecek riskidir. Basel II yönergelerine uygun olarak, kredi riski tarafında iç değerlendirme modelleri geliştirmek için bir dizi girişimde bulunulmuştur. Bu modeller ve bunların risk tahminleri kurumsal finansın ekonomik teorilerine dayanmaktadır ve topluca kredi riski teorileri olarak adlandırılmaktadır (Aziz ve Dar, 2006).

3.3.2. İstatistik tabanlı iflas tahmin yöntemleri

İstatistiksel teknikler iflasın öngörülmesi için kullanılan en güncel tekniklerdendir. Bu modellerde klasik modelleme vizyonu kullanılmıştır; tek değişkenli ve çok değişkenli istatistiksel modellerin iki gruba ayrılan finansal yetersizlik belirtilerini konu alır. Çok değişkenli diskriminant analizi, Logit ve Probit modeller bu gruba aittir (Klepáč ve Hampel, 2018).

- **Tek Değişkenli İstatistiksel Modeller**

İflas tahmininde istatistiksel yöntemleri kullanan ilk isimlerden biri 1967 yılında yayınladığı çalışması ile Beaver olmuştur (Balcaen ve Ooghe, 2006). Şirket iflasını tahmin etmek için iflas etmiş ve etmemiş firmaların finansal oranları ile tek değişkenli bir model olan tek değişkenli diskriminant analizi üzerinde çalışmıştır. Tek değişkenli tahmin modelinde, modelde yer alan her değişken için ayrı ayrı bir sınıflandırma işlemi gerçekleştirilir. Bir firma sınıflandırılırken, her parametre ayrı ayrı analiz edilir ve yanlış sınıflandırmaların yüzdesinin en aza indirildiği noktada model sınıflandırma işlemini gerçekleştirir. Bu tür sınıflandırmalarda sınıflandırma doğruluğu toplam yanlış sınıflandırma oranı ile Tip I (gerçekte iflas etmemiş olup iflas etmiş gruba sınıflama) ve Tip II (gerçekte iflas etmiş olup iflas etmemiş gruba sınıflama) hatalarının yüzdesi ile ölçülebilir.

- **Çok Değişkenli Diskriminant Analizi**

1968'de Altman istatistiksel çok değişkenli analiz tekniğini şirket başarısızlığı tahmini sorununa dahil etmiş ve 'Z-skor modeli' adı verilen bir model kurgulamıştır (Balcaen ve Ooghe, 2006). Kullandığı yönteme Çoklu Diskriminant Analizi (MDA) denir. Kısaca MDA, gözlemi gözlemin bireysel özelliklerine bağlı birkaç gruptan birine sınıflandırmak için kullanılan istatistiksel bir tekniktir. MDA modeli, başarısız olan şirketler grubu ile başarısız olmayan firmalar grubu arasında en iyi ayrımı sağlayan doğrusal bir değişken bileşiminden oluşur.

Altman, Çoklu Diskriminant Analizi çerçevesinde bir model geliştirmek amacıyla ABD işletmelerini temsil eden örnek için aşağıda verilen 5 parametre kullanılmıştır:

- Çalışma Sermayesi / Toplam Varlıklar,
- Kazançlar / Toplam Varlıklar,
- Faiz ve Vergi Öncesi Kazançlar / Toplam Varlıklar,
- Piyasa Kapitalizasyonu / Toplam Borçlar
- Satışlar / Toplam Varlıklar

1960'larda metodolojik ve teknik sınırlar ile mevcut veri örneklerinin sınırlamalarına rağmen, model hem tahmin hem de beklenti örneği için yüksek doğruluk oranları sergilemiş ve 1970'lerde ABD'de kurumsal şirketler üzerinde uygulamasıyla gerçek koşullarına ilişkin yüksek düzeyde tahminlerde bulunulmuştur (Bod'a ve Úradníček, 2016).

1980'lerden sonra gelişen diğer tekniklerle birlikte kullanımı azalsa da MDA yöntemi karşılaştırmalı çalışmalar için sıklıkla temel yöntem olarak kullanılmaktadır (Altman ve Narayanan, 1997). MDA için doğrusal ayrımcı fonksiyon aşağıdaki gibidir (Lachenbruch ve Goldstein, 1975).

D_i, i firması için hesaplanan diskriminant puanı (-∞,+∞),

x_{ij} = $x_j$parametresinin değeri (∀ i için, j = 1, ..., n))

D_j diskriminant katsayıları (j = 0, 1,..., n.) olmak üzere,

$$D_{\mathrm{i}} = D_0 + D_1 x_{i1} + D_2 x_{i2} + \cdots + D_n x_{in} \qquad (3.\ 1)$$

MDA modelinde, bir şirketin birkaç (çoğunlukla finansal) özelliği veya öznitelikleri tek birçok değişkenli ayrımcı puan D_i olarak birleştirilir. Bu ayrımcı skor -∞ ve +∞ arasında bir değere sahip olan ve firmanın finansal durumunu gösteren tek boyutlu bir ölçüdür. Bu nedenle MDA sürekli puanlama sistemi olarak adlandırılır. Çoğu çalışmada, düşük bir ayrımcı puan, kötü bir finansal duruma işaret eder. Birkaç değişkenin tek bir performans ölçüsüne veya ayrımcı skora entegrasyonu, 'bütünün parçaların toplamından daha değerli olması' prensibine dayanır (Taffler ve Agarwal, 2003). Tek değişkenli olarak önemsiz görünen değişkenlerin çok değişkenli bir bağlamda önemli bilgiler sağlaması mümkündür (Altman, 1968). Ek olarak, bazı katsayıların MDA'nın çok değişkenli karakterinden kaynaklanan beklenmeyen, sezgisel olmayan bir işarete sahip olması mümkündür (Ooghe ve Verbaere, 1985).

- **Logit ve Probit Modelleri**

Çoklu Diskriminant Analizinin açıkça baskın olduğu dönemden sonra bu yöntem yerini Logit Analizi (LA), Probit Analizi (PA) ve doğrusal olasılık modelleme gibi daha az talepkar istatistiksel tekniklere bırakmıştır. Logit Analizi, ikili veya ordinal yanıt olasılığı ile açıklayıcı değişkenler arasındaki ilişkiyi araştırmak için kullanılır. Yöntem, ikili veya ordinal yanıt verileri için doğrusal lojistik regresyon modeline maksimum olasılık yöntemiyle uyarlanır. İflas tahmininde Logit Analizinin ilk kullanıcıları arasında Ohlson yer almıştır (Back, 1996).

İkili bir sonucun olasılığı durumunda yaygın olarak kullanılan bir teknik olan Logit Analizi, kümülatif olasılık işlevine dayanır, bağımsız değişkenlerin normal olmasını gerektirmeden belirli bir sınıfa ait bir gözlemin koşullu olasılığını sağlar ve aynı anda çözülen bir problemdeki tüm perspektif faktörlerini dikkate alır. Bu tür modellerin özelliği, diskriminant analizinde aranan çok değişkenli normallik kriterinin aranmamasıdır (Chi ve Tang, 2006);(Zhou ve Elhag, 2007).

Logit Analizi, tahmin edilen sonucun 0 veya 1 olmasını sağlayan doğrusal olmayan bir regresyon modelidir. Bağımlı değişken 0 veya 1 değerlerini alan ikili bir değişken olarak kurgulanır. Logit Analizi, bağımlı değişkenin y = 1 olma olasılığını tahmin etmektedir. Tahmin edilecek bağımlı değişken Y için (Park, 2013);

$x_1, x_2 \dots x_k$; bağımsız değişkenler olmak üzere,

$$P(y = 1 \,|x_{1,} x_2, \dots, x_k = F(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k) \quad (3.\ 2)$$

$$P(y = 1 \,|x_{1,} x_2, \dots, x_k = \frac{1}{1 + e - \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k} \quad (3.\ 3)$$

Probit modeller ise kullandıkları fonksiyonlar noktasında Logit modellerden ayrılmaktadır. Logit birikimi standart lojistik dağılımı (F) Probit ise birikimli standart normal dağılım(Φ) kullanır.

$$P(y = 1\ x_{1,} x_2, \dots, x_k = \Phi(\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k) \quad (3.\ 4)$$

İstatistiksel Tahmin Modellerinin Dezavantajları

Tek değişkenli modellerin önemli bir avantajı basitliğidir, her oran için firmanın oran değerini bir eşik noktası ile karşılaştırır ve sınıflandırmaya buna göre karar verir. Tek değişkenli analizin uygulaması basit olsa da her uygulamada sadece bir finansal oran için sınıflama yaptığından, farklı finansal oranları sınıflaması ile karşılaştırıldığında tutarsız ve kafa karıştırıcı sonuçlar verebilmektedir (Altman, 1968). Ayrıca, tek değişkenli modellerde doğrusallık varsayımı bulunduğundan ve genelde bu varsayım göz ardı edildiğinden çoğunlukla modellerde şüpheli sonuçlar elde edilmiştir (Keasey ve Watson, 1991). Bu nedenle değişkenlerin önemini tespit etmek bu analizde zor kabul edilmektedir.

MDA hata tahmininde en sık kullanılan modelleme tekniği olmasına rağmen, bazı ciddi dezavantajları, temel varsayımların ihlali ile ilgili sorunları bulunmaktadır. İlk olarak, MDA sınıflandırma yönteminin doğrusal olduğu varsayılır, bu da belirli bir kesme noktasının üstünde veya altında ayrımcı bir puanın otomatik olarak iyi veya kötü bir finansal durumu işaret ettiği anlamına gelir. Aynı şekilde, MDA sınıflandırma kuralı sezgisel olarak bazı değişkenlerin finansal sağlıkla doğrusal bir ilişki göstermemesi ile çelişmektedir: bazı değişkenler hem çok düşük hem de çok yüksek bir değere sahip olduklarında finansal sorunları gösterir (Balcaen ve Ooghe, 2006).

Ancak, modeller çeşitlendikçe istatistiksel yöntemlere olan güven de azalmaya başlamıştır. Çok değişkenli diskriminant analizi, Logit ve Probit modelleri gibi istatistiksel iflas tahmin modellerinin etkinliğinin değerlendirilmesi ile ilgili olarak modellerin sınıflandırılmasının sonuçlarını en üst düzeye çıkarmak için bu modellerin eşiklerini manipüle etme becerisi gündeme gelmiştir. İstatistiksel yöntemler, modelin etkinliğini artırmak için eşiğin manuel ayarlama kolaylığı nedeniyle güvenilir sonuçlar sağlamaz (Nwogugu, 2005).

İflas modelleri genellikle bilanço ve gelir tablosundan elde edilen verilerle hesaplanan finansal oranları ile tahmin edilmektedir. Finansal oranların kullanımı, uygunlukları ve standardizasyonu kadar iflas eden ve iflas etmeyen firmalar arasında iyi bir ayrımcılık sağladıkları için de kullanılmaktadır (Jardin, 2016). Fakat, finansal oranların hesaplanma biçimi temelde onları güçsüz kılmaktadır; modellerin çoğunluğu belirli bir zaman için finansal oranları statik değerlere dayalı olarak geliştirilmiştir. Finansal oranlar için dinamik bir yaklaşım olumsuz finansal durumu olan işletmelerin iflas riski şirketleri ayırmada yardımcı olabilir (Korol, 2019).

Modellerin varsayımları dışında probleme yol açan nedenler; bağımlı ve bağımsız değişken seçimi, örnekleme yöntemleri ve yıllık verilerin kullanılmasıdır.

3.3.3. Yapay zeka tabanlı iflas tahmin yöntemleri

İflas tahmininde kullanılan en yaygın teknikler istatistiksel tabanlı modeller olsa da bu modellerin birçok çalışmada tanımlanan değişkenler arasında doğrusallık, normallik ve bağımsızlık gibi istatistiksel varsayımlarla ilgili pek çok dezavantajı bulunmaktadır.

1980'lerin sonlarında Logit ve Probit modellerin popülaritesi yerini Sinir Ağları tekniklerine bırakmıştır. 1988'de Messier ve Hansen'in çalışmaları ile başlayıp, 1990'larda iflas tahmini çalışmalarında kullanılan birincil yöntem haline gelen Sinir Ağları ile, insan beynini taklit eden bir örüntü tanıma işlevi yardımıyla karar verme yeteneğine sahip modeller geliştirilmesi hedeflenmiştir. Sinir Ağları çalışmalarını takiben 1993 yılında Theodossiou'nun Kümülatif Toplamlar algoritması ile literatür çeşitlilik kazanmaya başlamıştır (Gissel vd., 2007). Böylece istatistiksel iflas tahmin yöntemlerinin popülaritesi geri planda kalmaya başlamış sinir ağları, genetik algoritmalar, destek vektör makineleri, bulanık mantık gibi yöntemlere ait çalışmalar çoğalmıştır (Korol, 2019).

Modellerin varsayımlarının sağlanamaması ile birlikte, Basel Bankacılık Denetleme Komitesi tarafından 2004 yılında yayınlanan tavsiyelerden sonra, finans kurumları hesaplamalı öngörü tekniklerine dayalı daha karmaşık sistemler kullanma ihtiyacı artmıştır. İstatistiksel modellerin aksine, bu yöntemler herhangi bir ön bilgi kabul etmeyen, varsayımlara takılmayan ancak otomatik şekilde geçmiş gözlemlere bakarak gelecek değerleri tahmin eden model arayışına girilmiştir. Böylece iflas tahmini üzerine yapılan çalışmalarda yapay zekaya dayalı yöntemlere eğilim artmıştır.

Literatürde yapay zekaya dayalı iflas tahmin yöntemleri olarak yer alsa da temelde analizler yapay zekanın bir alt dalı olan makine öğrenmesi algoritmaları kullanılarak yapılmaktadır. İflas tahmini problemi, makine öğrenmesi problemleri içinde ikili sınıflama problemi olarak değerlendirilir. Söz konusu şirket "iflas etti" ya da "iflas etmedi" şeklinde etiketlenir. Eldeki veriler

makine öğrenmesi doğası gereği eğitim ve test verisi olarak ayrılır. Eğitim verileri kullanılarak sınıflanması tamamlanan problem test verileri üzerinden doğruluk oranları hesaplanarak bir tahmin sonucu elde edilir.

Buradan hareketle makine öğrenmesi metodolojisi altında kullanılan başlıca algoritmalar şu şekilde sıralanabilir:

- Bagging ve Boosting Yöntemleri
- Çok Katmanlı Algılayıcı
- Cox Sağkalım Analizi
- Destek Vektör Makinesi
- Bulanık Kümeler
- Genetik Algoritmalar
- k En Yakın Komşu
- Karar Ağacı
- Naive Bayes
- Parçacık Sürü Optimizasyonu
- Rastgele Orman
- Kaba Setler
- Sıralı Minimal Optimizasyon
- Vaka Tabanlı Muhakeme
- Yapay Sinir Ağları

4. MAKİNE ÖĞRENMESİ

İlk olarak bilgisayarların ulaşılabilir bir aygıt haline dönüşmesi, akabinde bilgisayar biliminin de daha geniş kitlelere ulaşabilir hal alması ile problemleri insanlardan daha hızlı çözen, karar alma noktasında öğretilen tüm olasılıkları göz önünde bulunduran sistemler geliştirilmeye başlanmıştır. Gelişmelerin biyoteknoloji, medikal, eğitim, işletme, bankacılık ve finans gibi farklı bilim dallarındaki problemlere çözüm sunmaya başlamasıyla birlikte bu alandaki çalışmalar hızlıca artmış ve bugün makine öğrenmesi ismi verilen kavram ortaya çıkmıştır. Makine öğrenmesi, temelinde veri madenciliği ve yapay zekaya dayanan bir disiplindir.

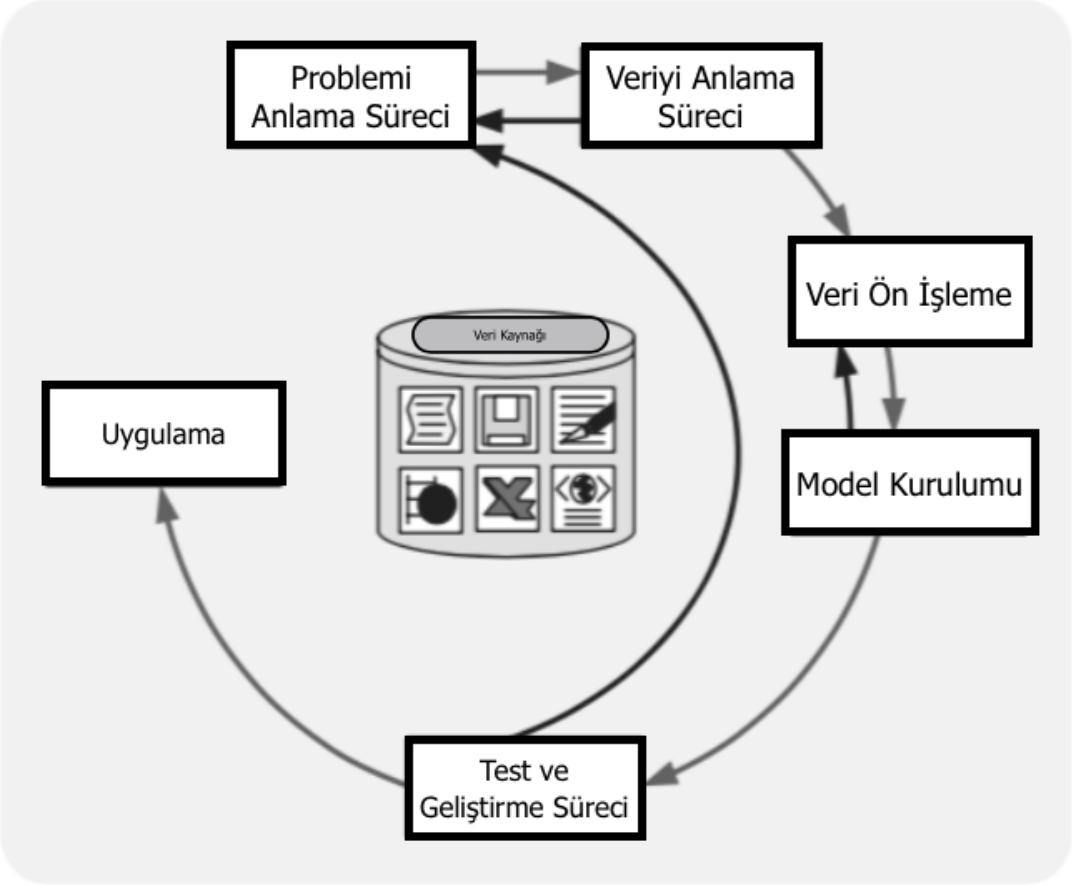
4.1. Veri Madenciliği

Veri madenciliği, veritabanlarında tutulan bilgilerin daha önceden belirlenmemiş örüntülerini tanıma, verinin eğilimini bulma ve bu bilgileri tahmine dayalı modeller oluşturmak için kullanma süreci olarak tanımlanabilir. Bir başka tanım ile veri madenciliği, önceden keşfedilmemiş örüntüleri ortaya çıkarmak için veri seçimi ve modelleme sürecidir. İnsanların algılayamayacağı kadar karmaşık örüntüleri anlaşılır hale getirmek büyük veri kümelerini tarayarak verilerden yeni ve geçerli çıkarımlar yapmayı amaçlamaktadır (Koh ve Tan, 2005).

Veri madenciliğinde ana fikir, program ve kullanıcı davranışını belirleyen sistem özelliklerinin tutarlı ve yararlı kalıplarını keşfetmek ve ilgili sistem özelliklerini anomalileri ve bilinen özellikleri tanıyabilen sınıflandırıcıları hesaplamak için kullanmaktır (Lee ve Stolfo, 1998).

Veri madenciliği süreçlerinde ilk adım, Şekil 4.1'de belirtildiği üzere, problemin tanımlanmasıdır. Çözüme kavuşturulmak istenen durum netleştirilir ve buradan hareketle proje hedefleri, gereksinimleri belirlenir.

İkinci adım olarak, problem özelinde veri toplanacak kaynaklar belirlenir. Çalışılan verinin boyutuna göre değişkenlik göstermekle birlikte, veri, birden fazla veri tabanında tutulabilir. Kaynaklar belirlendikten sonra kullanılacak veri bir araya getirilir. Analize başlanmadan önce, veride temizlik yapılır; örneğin, analize uygun olmayan tarihteki verilerin çıkarılması, kullanıcı tarafından manuel girilen ve hata yapılan verilerin düzeltilmesi, çıkarılması gibi. Verilere daha yakından bakıldığında, verinin, çözüme kavuşturulmak istenen durumu ne kadar iyi yansıttığı üzerinde durulur, duruma göre yeni öznitelik ekleme ya da çıkarma işlemi yapılabilir.



Şekil 4. 1. Veri Madenciliği Süreci (Olson ve Delen, 2008)

Araştırılmak istenen konuya uygun olarak hazırlanan veri seti doğrultusunda çeşitli modelleme teknikleri seçilir ve uygulanır. Algoritma sonuçlarına göre, modeli iyileştirmek amacıyla parametreler üzerinde değişiklikler yapılabilir. Gerekli görülen yerlerde veri dönüşümleri, kimi zaman veri sadeleştirmeleri

yapılabilir. Veri madenciliğinin bu aşamasında, hedeflenen sonuca ne kadar ulaşılabildiği önem arz etmektedir.

Veri madenciliği sürecinin son aşamasında, elde edilen yeni bilgilerin gücüyle, belirlenen problemi çözümlemek amacıyla birtakım geliştirmeler yapılır. Bir önceki aşamada karar verilen model uygulaması ile, yeni bilgiler türetilir ve sorun çözüme kavuşturulur. Böylece, veri tabanlarında saklanan verilerin cevherleri ortaya çıkarılır.

Veri madenciliği bilgi keşif ruhu, yeni ve yararlı bilgiler öğrenme gayesi olarak tanımlanır. Bu nedenle, veri madenciliği tek başına düşünülmemelidir. Eldeki verilerden yeni bilgiler edinme amacı doğrultusunda farklı disiplinlerden yardım alan, nihai olarak bu disiplinleri tek bir noktada birleştiren ana neden olarak düşünülmelidir. Veri madenciliği, her durumda olmamakla birlikte genellikle büyük veri kümelerine uygulandığından yapay zeka ve dolayısıyla makine öğrenmesi tekniklerinden yararlanır (Olson ve Delen, 2008).

4.2. Klasik İstatistik, Yapay Zeka ve Makine Öğrenmesi

İstatistik, verilerden bilgi elde etmek amacıyla veri toplama, derleme, verilerin analizi ve yorumlanması olarak tanımlanabilir. Bu tanımla birlikte istatistik, veri madenciliğiyle aynı disiplin olarak düşünülse de birtakım farklar mevcuttur. Klasik istatistik, varsayımlar istatistiği olarak bilinir. Analize başlanmadan önce, problem doğrultusunda varsayımlar belirlenir ve bu varsayımlar test edilir. Klasik istatistikte varsayımı doğrulama ya da yanlışlama üzerinde durulur; kurulan hipotez ile daha önceden belirlenmiş parametreler sınanır. Klasik istatistikte hesaplanan son değerin yansız, tutarlı ve sapmasız olduğu varsayılır.

Yapay zeka, akıllı makineler, özellikle de bilgisayar programları geliştirmek için türetilmiş bir bilim ve mühendislik dalıdır. Tablo 4.1'de yer verildiği üzere, 1931 yılında Kurt Gödel tarafından ortaya konulan "hesaplanabilir sistemler" düşüncesiyle başlayan yapay zeka süreci, 1937 yılında Alan Turing'in akıllı

makinelerin sınırları üzerine yaptığı çalışmalar ile devam etmiştir. McCulloch ve Pitts tarafından 1943 yılında sinir ağlarını modelleyerek önerme mantığıyla bağlantı kurmuş, 1950'de yine Alan Turing'in çalışmasıyla, makinelerin düşünme yeteneğinin olup olmadığı üzerinde durulmuş, bu sayede yapay zeka popülerlik kazanmıştır.

Tablo 4. 1. Yapay Zekanın Gelişimi (Ertel, 2011)

Yıl	Gelişme
1931	Avusturyalı Kurt Gödel, birinci dereceden yüklem mantığında tüm doğru ifadelerin türetilebilir olduğunu göstermiştir. Üst düzey mantıkta ise, kanıtlanamayan doğru ifadeler vardır.
1937	Alan Turing durdurma sorunu ile akıllı makinelerin sınırları üzerinde durmuştur.
1943	McCulloch ve Pitts nöral ağları modellemiş ve önerme mantığı ile bağlantı kurmuştur.
1950	Alan Turing, Turing testi ile makine zekasını incelemiş ve öğrenen makineleri ile genetik algoritmalar hakkında makaleler yayınlamıştır.

Yapay zeka ile problem çözme, insan zekası ve muhakeme bilgisi ile temelde aynı amaca hizmet etmek için geliştirilmiştir. Yapay zeka konulu birçok araştırmada bilgi ediniminin rolüne odaklanılmış; özellikle değişen denetim koşullarında makine öğrenimi üzerinde durulmuştur. Yapay zeka dilinde öğrenme, önceki durumlardan tecrübe ederek problem çözebilmeyi geliştiren bir sistemdir ve makine öğrenimi yöntemleri iflas tahmini de dahil olmak üzere çeşitli problemlerde başarıyla uygulanmıştır (Aziz ve Dar, 2006).

Makine öğrenmesi ise, yapay zekanın bir alt dalı olarak düşünülebilir. Örnek verileri veya geçmiş deneyimleri kullanarak bir performans ölçütü optimize etmek için bilgisayarları programlamaktır (Alpaydin, 2010). Yapay zekanın gelişimini takiben 1950'li ve 1960'lı yıllarda popülerleşen makine öğrenmesinin

öncüleri yine Alan Turing, John McCarthy, Arthur Samuels, Alan Newell ve Frank Rosenblatt gibi araştırmacılar olmuştur (Alzubi vd., 2018).

Makine öğrenimi ve istatistik, yöntemler açısından yakından ilişkili alanlardır, ancak temel hedefleri bakımından farklıdır: istatistik örneklemden ana kütleye dair çıkarımlar yapmayı hedeflerken, makine öğrenmesi genellenebilir modeller oluşturma amacı taşır (Bartlett, 2019). Klasik istatistik, genel olarak verileri bir doğru etrafında şekillendirme üzerine kuruludur. Burada amaç gelecek hakkında tahmin yapmaktan çok bağımlı ve bağımsız değişkenler arasındaki ilişkiyi tespit etmektir. Makine öğrenmesi uygulamalarında bu süreçten farklı olarak eğitim ve test seti adı verilen iki farklı veri seti bulunur. Model çıkarımları eğitim seti üzerinden gerçekleştirilir ve devamında test kümesi ile model değerlendirilir.

4.3. Makine Öğrenmesi Türleri

Günümüz sorunlarına aktif olarak cevap arayan makine öğrenmesi algoritmaları doğal dil işleme, tıbbi tanı koyma, kredi kartı dolandırıcılığı, konuşma ve el yazısı tanıma, nesne tanıma, müşteri kaybı tahmini borsa tahmini gibi birçok alanda kullanılmaktadır. 1990'lardan itibaren tek başına bir alan olan makine öğrenmesi zamanla kendi içerisinde farklı öğrenme tiplerine ve farklı problemlere ayrılmıştır.

- **Denetimli öğrenme**

Denetimli öğrenme, algoritmaya mümkün durumlar ve sonuçların en başta aktarıldığı makine öğrenmesi çeşididir. Bu tür algoritmalarda veri eğitim verisi ve test verisi olarak parçalanır. Eğitim veri setindeki tüm girdiler ve onun mevcut sonuçları algoritma tarafından işlenerek bir örüntü yakalanmaya çalışılır.

Denetimli makine öğrenimi otomatik olarak bir tahmin işlevi oluşturmak için $F: X \rightarrow Y$ fonksiyonunu kullanır. Burada X (öznitelikler) Y (hedef değişkeni) ile

eşlenerek (X_i, Y_i) tarafından temsil edilen eğitim kümesi çiftlerini oluştururlar (Fabris vd., 2017).

- **Denetimsiz Öğrenme**

Denetimli öğrenmede algoritmayı öğrenmeye yönlendiren bir eğitim seti bulunur denetimsiz öğrenmede bu eğitim seti bulunmamaktadır.

Öğretmensiz öğrenme şeklinde düşünebileceğimiz denetimsiz öğrenme sürecinde, sistemin somut veri kümeleri yoktur ve sorunların çoğunun sonuçları büyük ölçüde bilinmemektedir. Diğer bir deyişle, otomatik öğrenme algoritması yaratma hedefi operasyon başladığı anda kör bir haldedir. Sistem süreç boyunca rehberlik edebilmek için birtakım mantıksal işlemlere sahiptir, ancak uygun giriş ve çıkış algoritmaları eksikliği süreci zor hale getirir. Bu nedenle denetimsiz öğrenme, tüm bilgisayar sistemlerinde bulunan giriş verileri ve ikili mantık mekanizması aracılığıyla sınırsız miktarda veriyi hiç referansı yokken yorumlama ve çözüm bulma yeteneği ile problemi sonuca kavuşturur (Hastie vd., 2009).

- **Yarı Denetimli Öğrenme**

Denetimli ve denetimsiz öğrenme algoritmalarının birleşimden oluşan bir öğrenme şeklidir. Denetimli öğrenmede eğitim seti oluşturmak kaynak kısıtı açısından zordur, buna karşılık denetimsiz öğrenmede algoritmayı girdi verisi olmadan eğitmek ve yorumlamak zordur. Yarı denetimli makine öğrenmesinde büyük hacimli veri setlerinin küçük bir alt kümesi olacak düzeyde girdi ve çıktı sonuçları algoritmaya öğretilerek hem denetimli öğrenmedeki maliyet yükünden hem de denetimsiz öğrenmedeki yorumlama zorluğundan kurtulma hedeflenir.

- **Takviyeli Öğrenme**

Takviyeli öğrenmeyi anlamanın en kolay yolu, denetimli öğrenmeyle karşılaştırmaktır. Denetimli öğrenmede ajana (agent) eğitim kümesi ile verilen duruma nasıl tepki verileceği öğretilir. Takviyeli öğrenmede ise ajana nasıl tepki

verileceği öğretilmez, "özgür seçim" yapma durumu öğretilir. Algoritma, ajan bir kez aksiyon olduktan sonra aksiyonun iyi ya da kötü olduğunu söyler Ödül mekanizması olarak nitelendirilen bu teknikte olumlu bir ödül iyi davranış ve olumsuz bir ödül kötü davranıştır. Algoritma buradan yola çıkarak gelecek aksiyonlarında nasıl davranması gerektiğini öğrenir (Ghory, 2004).

4.4. Makine Öğrenmesi Problemleri

Makine öğrenmesinde, algoritmanın eğitilebilmesine bağlı olarak öğrenme türleri yer aldığı gibi, probleme konu olan verinin türüne göre de problem türleri alt başlıklara ayrılmaktadır.

- **Regresyon**

Regresyon, hakkında çıkarımlar yapılmak istenen bir değişkeni (bağımlı değişken) bir grup bağımsız değişken ile matematiksel olarak açıklama işlemidir. Makine öğrenmesinde genellikle parametrik olmayan regresyon modelleri üzerinde durulur. Makine öğrenmesinde kullanılan başlıca regresyon modelleri Destek Vektör Regresyonu, Karar Ağacı Regresyonu, Rastgele Orman Regresyonu olarak sıralanabilir.

- **Sınıflama**

Yanıt değişkeninin kategorilere ayrıldığı problem türüdür. Yanıtlar sürekli değişkenler yerine Evet/Hayır, Doğru/Yanlış gibi kesikli değişkenler olarak kategorilendirilir. Sınıf sayısına göre problem, 2 sınıfa sahipse ikili sınıflandırma problemi, ikiden fazla sınıfa sahipse çok sınıflı sınıflama problemi olarak belirlenir. Makine öğrenmesinde en çok kullanılan sınıflama algoritmaları k En Yakın Komşu, Naive Bayes, Rastgele Orman, Gradient Boosting algoritmalarıdır.

- **Kümeleme**

Kümeleme problemi, veri kümesini belli niteliklerin benzerliklerini göz önünde bulundurarak gruplama işlemidir. Çeşitli uzaklık ölçütleri kullanılarak veri

noktalarının birbirlerine olan uzaklıkları ölçümlenir (Leskovec vd., 2011). Kümelemenin amacı, ilgi alanı olan bir veri kümesi içinde benzer nesnelerin deseni veya gruplarını belirlemektir (Kassambara, 2017). Kümeleme denetimsiz öğrenme algoritmaları kategorisine girer. Bu algoritmalar, veri içindeki yapıları öğrenmeye çalışır ve verilerin yapısındaki benzerliğe göre kümeler yapmaya çalışır. Farklı sınıflar veya kümeler daha sonra etiketlenir. Algoritma, eğitildiğinde, kümelerden birine görünmeyen yeni veriler koyar. En çok kullanılan kümeleme algoritmaları, k Ortalama Kümeleme, Hiyerarşik Kümeleme, Beklenti Maksimizasyonu ile Kümeleme olarak sıralanabilir.

- **Aykırılık Tespiti**

Belirli bir örüntü analiz ve desen değişiklikleri veya anomali algılama sorunları aykırılık tespiti problemi olarak tanımlanmaktadır. Örneğin, kredi kartı şirketleri, istemcilerinin olağan işlem davranışından sapma bulmak için anormallik algılama algoritmaları kullanır ve olağan bir işlem olduğunda uyarılar yükseltir. En çok kullanılan aykırılık tespiti algoritmaları En yakın komşu, Yapay Sinir ağları, Bulanık Mantıktır (Killourhy ve Maxion, 2009).

- **Boyut İndirgeme**

Makine öğrenmesinde boyut kavramı veri seti içerisinde bulunan öznitelik (değişken, parametre) sayısını ifade eder. Veri seti içinde değişken sayısı fazla olduğunda genelde modeller şişme (gerçek dışı doğruluk) meydana gelebilir. Bu nedenle boyut indirgeme teknikleri kullanılarak birbiriyle yakın özelliklere sahip olan parametrelerde modele katkısı daha fazla olan tutulur, diğer değişken analizden çıkarılır. Bu amaçla kullanılan algoritma genelde Temel Bileşenler Analizidir.

4.5. Naive Bayes Algoritması

Naive Bayes (NB) algoritması, üzerinde çalışılan veri setindeki değerlerin sıklığını ve birleşimlerini sayarak muhtemel bir sınıf yaratan olasılık sınıflayıcısıdır. Algoritma, Bayes teoremi üzerine kuruludur; tüm değişkenlerin

sınıf değişkeninin değeri dikkate alınarak bağımsız olduğu varsayılır. Bu varsayıma sınıf koşullu bağımsızlığı denir. Bu, söz konusu hesaplamayı basitleştirmek için yapılır ve bu algoritma naive (naif) olarak ifade edilir (Sarıtaş, 2019)

4.5.1. Bayes teoremi

Bayes teoremi, adını 18. Yüzyılda bu konu üzerinde çalışmalar yapmış Thomas Bayes'ten alır. Bayes teoremini anlamak için önce birkaç terim üzerinde durulmalıdır;

Marjinal Olasılık: Başka bir değişkenin sonucuna bakılmaksızın bir olayın olasılığıdır. P(X=A)

- Ortak Olasılık: Aynı anda meydana gelen iki olayın olasılığıdır. P(A, B)
- Koşullu Olasılık: İkinci bir olayın varlığında meydana gelen bir olayın olasılığıdır. P(A|B)

Bu teoreme göre, E'nin veri seti içinde bir nokta olduğunu düşünelim. E bir delil (evidence) olarak kabul edilir. H değişkenin de bir hipotez olduğunu düşünelim; E'nin C sınıfına ait olması gibi. Sınıflama problemlerinde, E değeri bilindiğinde H hipotezinin gerçekleşme olasılığı yani P(H|X) üzerinde durulur. Diğer bir deyişle, X'in özellikleri bilindiğinde, X değerinin C sınıfına ait olma olasılığı bulunmaya çalışılır. Buradan hareketle Bayes Teoremi

$$P(H|E) = \frac{P(E|H)}{P(E)}.P(H) \qquad (4.\ 1)$$

şeklinde ifade edilir.

Her bir verisi sınıf etiketiyle verilmiş bir T eğitim seti düşünelim. Bu eğitim setinde k tane sınıf olsun ($C_1, C_2 \dots C_k$). Her bir örnek n boyutlu vektör x

($x_1, x_2, \dots x_n$) ve n tane sınıflayıcı ($A_1, A_2, \dots A_n$)) ile gösterilsin. Herhangi bir x verildiğinde, sınıflandırıcı x'in x'e koşullanmış en yüksek sonsal olasılığına sahip sınıfa ait olduğunu tahmin eder. Yani x'in C_i sınıfına ait olduğu tahmin edilir.

$$P(C_{\mathrm{i}}|x) > P\left(C_j\middle|x\right) \;\; 1 \le j \le \frac{m}{j} = i \tag{4. 2}$$

Böylece, $P\left(C_j\middle|x\right)$ değerini maksimum yapan sınıf bulunur.

$$P(C_i|\, x) = \frac{P(x|C_i).P(C_i)}{P(B)} \tag{4. 3}$$

Veri setinde birçok niteleyici verildiğinde $P(x|C_i)$ olasılığını hesaplamak oldukça maliyetlidir. Bu nedenle, $P(x|C_i)$ ve $P(C_i)$ değerlendirilirken hesaplamadan kaçınmak için sınıfların koşullu bağımsızlığı adı altında bir naif varsayım geliştirilmiştir. Böylelikle, niteleyici değerlerinin, sınıf etiketi göz önüne alındığında, birbirinden koşullu olarak bağımsız olduğu varsayılır.

$$P\left(x|\, c_i\right) \approx \prod_{k=1}^{n} P(\, x_k|\, C_i) \tag{4. 4}$$

Buradan hareketle, değerlerin Gaussian dağılımına uyduğu varsayılır ve μ ortalama ve σ standart sapma olmak üzere $P(x_1|C_i), P(x_2|C_i), \dots, P(x_n|C_i)$ olasılıkları eğitim setinden kolaylıkla tahmin edilebilir.

$$P(x\,|c_i) = x_k, \mu_{c_i}\sigma_{c_i} \tag{4. 5}$$

X'in sınıfını tahmin etmek için, $P(x|C_i)P(C_i)$ olasılığı her sınıf için değerlendirilir. Sınıflandırıcı, X'in sınıf etiketini $P(x|C_i)P(C_i)$ olasılığının maksimum olduğu noktada tahmin eder (Lewis, 1998).

Naive Bayes algoritması, temelleri yüzyıllar önce atılmış olsa da günümüzde sıkça kullanılmaya devam eden bir algoritmadır. K en yakın komşu algoritması gibi tembel öğreniciler ile karşılaştırıldığında sınıflama işlemini daha kısa sürede tamamlar; hızlıdır. Sınıflama performansı ilgisiz değişkenler çıkarıldığında yükselir. Ancak bu algoritmayla görece iyi sonuçlar almak için veri setinin büyük olması gerekmektedir (Jadhav ve Channe, 2016).

Naive Bayes algoritmasının avantajları:

- Etkisiz değişkenleri dışarıda tutar.
- Yüksek performansa sahiptir.
- Algoritma işlem süresi kısadır.

Naive Bayes algoritmasının dezavantajları:

- Naive Bayes algoritmasında algoritmanın performansı gözlem sayısına bağlıdır.
- Tüm eğitim kümesi hafızada yer tuttuğu için örnek tabanlı ve tembel öğrenicidir (Archana ve Elangovan, 2014).

4.6. k En Yakın Komşuluk Algoritması

k En Yakın Komşuluk (kNN) algoritması denetimli öğrenme algoritmalarından biridir. Parametrik yapıda olmayan bu algoritma basit ve uygulanması kolay olduğu için en yaygın olarak kullanılan sınıflandırma algoritmalarından biri olarak kabul edilmektedir. K en yakın komşu algoritması hem regresyon hem de sınıflandırma problemlerinde etki alanına sahip bir algoritmadır.

k En Yakın Komşu algoritması, ilk olarak 1950'lerin başında ortaya atılmıştır. 1965 yılında N. J. Nilsson tarafından uzaklığı en küçükleyen sınıflandırıcılar çalışmaları ile bu algoritma üzerinde ilerleme kaydedilmiştir. Nilsson'ı destekleyen çalışmaları ile T. Cover ve P. Hart' 1967'de algoritmanın çalışma prensiplerini günümüzde kullanılan haline dönüştürmüşlerdir (Hu vd., 2016).

kNN algoritmasının çıkış noktası, benzerlik gösteren verilerin aynı sınıfta bulunacak olma olasılığıdır. Bir uzaklık ölçütü yardımıyla verilerin birbirlerine olan uzaklıkları hesaplanır. Buradan hareketle, henüz bir sınıfa atanmamış veriye en yakın sınıflanmış veri tespit edilir. Çalışmalarda uzaklık hesaplamalarında temel kabul edilen, en fazla kullanılan uzaklık metriği, Euclidean uzaklığıdır (Hu vd., 2016).

Uzaklığı hesaplanacak veri noktaları x_1 ve x_2 olarak düşünüldüğünde Euclidean uzaklığı,

$x_1 = (x_{11}, x_{12} \dots x_{1n})$ ve $x_2 = (x_{21}, x_{22} \dots x_{2n})$ olmak üzere,

$$dist_{euclidean}(x_1 x_2) = \sqrt{\sum_{i=1}^{n} (\boldsymbol{x_{1i}} - \boldsymbol{x_{2i}})^2} \quad (4.\ 6)$$

Uzaklık ölçütü olarak Manhattan, Minkowski, ve Chebyshev uzaklıkları da kullanılan diğer uzaklık ölçütleri olarak sayılabilir (Prasath vd., 2019). Uzaklık ölçütlerine göre optimal k değeri değişmekte, böylece sınıflandırma doğruluk oranlarında farklı sonuçlar gözlenebilmektedir.

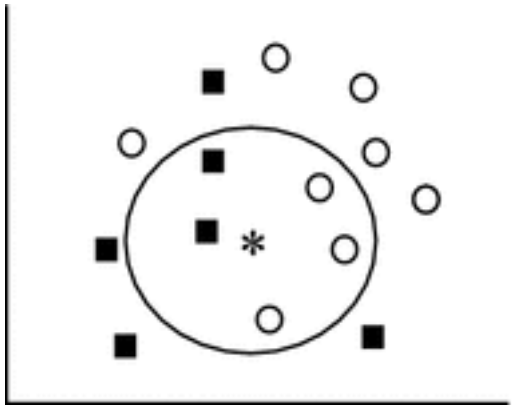
$$dist_{minkowski}(x_1 x_2) = \sqrt{\sum_{i=1}^{n} |\boldsymbol{x_{1i}} - \boldsymbol{x_{2i}}|^2} \quad (4.\ 7)$$

$$dist_{manhattan}(X_1 X_2) = \sum_{i=1}^{n} |\boldsymbol{x_{1i}} - \boldsymbol{x_{2i}}| \quad (4.\ 8)$$

$$dist_{chebyshev}(X_1 X_2) = \max_i |\boldsymbol{x_{1i}} - \boldsymbol{x_{2i}}| \quad (4.\ 9)$$

k En Yakın Komşu algoritmasının doğası gereği, dikkat edilmesi gereken, sınıflamanın seyrini değiştirebilecek en önemli parametre k komşu değeridir. k komşu değeri sınıflama işlemi başlamadan önce belirlenen kullanıcı tanımlı bir parametredir. k=1'den başlayarak veri adedi kadar k değeri seçmek mümkündür. Eğer k=1 ise algoritmanın adı "en yakın komşu algoritması"

olarak anılabilir. k komşu değerinin optimum düzeyde seçilmemesi birtakım sorunlara yol açabilir. k değeri olması gerektiğinden büyük verilir ise birbiri ile benzerlik göstermeyen veriler aynı sınıf içerisinde bulunmaya zorlanır. Bu durumda hesaplanacak doğruluk oranı düşecektir. Diğer bir durumda eğer k değeri optimum düzeyin altında seçilir ise aynı sınıfta bulunması gereken veriler birbirinden koparılacaktır. Bu şartlar altında yine aynı şekilde hesaplanacak doğruluk oranı düşecektir.



Şekil 4. 2. k En Yakın Komşu Sınıflaması (Hu vd., 2016)

Şekil 4.2'de gösterildiği üzere, * ile gösterilen veri, k = 1 ise, nokta kare sınıfa aittir; k = 5 ise, en yakın beş noktanın çoğunluk sınıfı olan daire sınıfına aittir (Hu vd., 2016).

İkili sınıflandırma problemlerinde, birbirleri ile bağlantılı oylardan kaçınmak amacıyla k değerini tek sayı olarak belirlemek kurtarıcı bir alternatiftir (Archana ve Elangovan, 2014).

Sınıflandırma modellerinde temel yapı modelin kendi içinde bir artık sınıflandırıcı üretmesi ve veri eklendikçe bu artık sınıflandırıcıyı kullanması üzerine kuruludur. kNN algoritmasında ise artık sınıflandırıcı işleyişi yer almamaktadır. Sınıflandırılmak istenen her yeni veri için en yakın komşuların oluşturduğu küme tekrar tekrar hesaplanır. Bu nedenle sınıflandırma için kullanılan sürenin uzun olduğu kNN algoritması tembel öğrenici olarak tanımlanır (Khan vd., 2002). kNN algoritmasının bu özelliği veri setinin sürekli

güncellendiği durumlarda algoritmayı doğruluk oranı açısından daha başarılı yapar; veriler güncellendikçe sınıflandırma sonuçları da güncellenir. Ancak veri seti hacminin yüksek olduğu durumlarda algoritmanın öğrenme işlemi görece uzun sürdüğünden olumsuz bir durum oluşturabilir.

kNN algoritmasının avantajları:

- Anlaşılması ve uygulanması kolay bir sınıflandırma tekniğidir.
- Eğitim süresi kısadır.
- Gürültülü (bozuk) eğitim verilerine dayanıklıdır.
- Çoklu sınıflandırma problemleri için uygundur.

kNN algoritmasının dezavantajları:

- Verilerin yerel yapısına duyarlıdır.
- Sınırlı bir bellek ile çalışır.
- Denetimli öğrenme algoritmaları içinde yer almasına rağmen tembel öğrenici olduğu için yavaş çalışmaktadır (Archana ve Elangovan, 2014).

4.7. Destek Vektör Makinesi Algoritması

Destek Vektör Makinesi (DVM), etiketlenmiş eğitim verilerinden, girdi ve çıktı verilerini göz önünde bulundurarak eşleme işlemi yapan bir denetimli öğrenme algoritmasıdır. Bu işlem bir sınıflandırma ya da bir regresyon problemi için kullanılabilir.

Sınıflandırma ve regresyon problemlerinde sıkça başvurulan yöntemlerden biri olan DVM, 1963 yılında Vladimir Vabnik ve Alexey Shervonenikis tarafından temelleri atılmış, 1995 yılında ise geliştirilmesi tamamlanmış bir makine öğrenmesi algoritmasıdır (Satapathy vd., 2019);(Talabani, 2019).

Sınıflama problemi özelinde algoritma ardındaki ana fikir, eğitim verileri doğrusal ayrılabiliyor ise bir hiperdüzlem yardımı ile sınıflamaktır. Eğitim verileri doğrusal ayrılamıyor ise, çekirdek fonksiyonları yardımı ile veriyi çok boyutlu

bir uzaya haritalandırarak elde edilen yeni alanda doğrusal bir hiperdüzlem elde ederek sınıflamaktır (Onel vd., 2018). Daha sonra, maksimum marjlı hiperdüzlemlerin eğitim verilerindeki sınıfları en iyi şekilde ayırmaları için eş zamanlı olarak eğitilir. İki paralel hiperdüzlem arasındaki mesafeyi en üst düzeye çıkararak verileri ayıran hiperdüzlemin her iki tarafına iki paralel hiperdüzlem oluşturulur. Bu paralel hiperdüzlemler arasındaki kenar boşluğu veya mesafe ne kadar büyükse sınıflandırıcının genelleme hatasının o kadar iyi olacağı varsayımı yapılır. Tıbbi tanı, biyoinformatik, yüz tanıma, görüntü işleme ve metin madenciliği gibi çok sayıda gerçek dünyadaki uygulamada, bilgi keşfi ve veri madenciliği için en popüler, en son teknoloji araçlardan biri olan DVM, istenilen sayıda değişken ile eğitilerek doğrusal olmayan veriler üzerinde çalışabilir. Bu nedenle, doğrusal olmayan, karmaşık sistemleri ve süreçleri modelleme konusunda oldukça yeteneklidir (Olson ve Delen, 2008). Destek Vektör Makinesi algoritmasının altyapısı istatistiksel öğrenme teorisi ve yapısal risk minimizasyonuna dayanmaktadır.

4.7.1. İstatistiksel öğrenme teorisi

İstatistiksel öğrenme teorisi makine öğrenimi algoritmalarının çoğu için teorik bir temel oluşturur ve genel olarak yapay zekanın gelişmiş dallarından biri olduğu söylenebilir. 1960'larda Rusya'da ortaya çıkmış ve farklı algoritmaların geliştirilmesinin ardından 1990'larda geniş bir popülerlik kazanmıştır (Luxburg ve Schoelkopf, 2008).

İstatistiksel öğrenme teorisi, bilgi sağlama, çıkarım yapma, karar verme problemlerini incelemek için geliştirilmiş bir teoridir. Denetimli öğrenme problemleri içerisinde kullanılan istatistiksel öğrenme teorisi, hedef alanı en iyi biçimde temsil edecek düzlemi belirlemeyi amaçlamaktadır (Bousquet vd., 2004).

$R^n x\ R$ boyutlu uzayda $\{(x_1, y_1),\ldots,(x_i, y_i)\}$ eğitim seti mevcut olduğu varsayıldığında, bilinmeyen bir olasılık dağılımı ile örneklenen P(x,y); hatayı

ölçen kayıp fonksiyonu V(y,f(x)); belirli bir x için, f(x) gerçek değer y yerine tahmin değeri olarak kabul edilir. Teori, yeni verilerdeki hata beklentisini en aza indiren fonksiyon f(x)'i aşağıdaki gibi ifade eder (Jakkula, 2006).

$$\int \mathrm{V}\big(\mathrm{y}, \mathrm{f}(\mathbf{x})\big)\mathrm{P}(\mathbf{x}, \mathrm{y})\mathrm{d}\mathbf{x}\,\mathrm{dy} \qquad (4.\ 10)$$

En temel haliyle istatistiksel öğrenme teorisi, verilerin öncelikle bir dağılıma uygun olmadığını, yapılan analizler sonucunda veriye en uygun modelin oluşturulacağını savunur; klasik istatistiğin getirdiği verilerin belli bir dağılıma uygunluğu önceden varsayma ve bu varsayımlar üzerine hareket etme metoduna karşı duruş sergiler (Çomak, 2008).

Risk Minimizasyonu Problemi

Risk Minimizasyonu probleminde, yanıt değişkenine en uygun yaklaşımı bulabilmek amacıyla bir kayıp ya da tutarsızlık olarak adlandırabilen L değişkeni L(y, (x, w)) tanımlanır. Kayıp değişkeninin beklenen değeri aşağıdaki eşitlik ile belirlenir.

$$R(w) \int \mathrm{L}\big(\mathrm{y}, \mathrm{f}(\mathbf{x}, \mathbf{w})\big)\mathrm{dP}(\mathbf{x}, \mathrm{y}) \qquad (4.\ 11)$$

Burada ulaşılmak istenen, x ve w sınıf değişkenlerini göz önünde bulundurarak risk fonksiyonunu R(w)'yi minimize etmektedir. Ancak, ortak olasılık dağılımı P(x, y) = P(y|x)P(x) bilinmemektedir ve bu bilgi sadece eğitim veri setinde bulunmaktadır (Vapnik, 1992).

Deneysel Risk Minimizasyonu

Risk minimizasyonu problemindeki ortak olasılık dağılımı problemini çözümlemek için risk fonksiyonu R(w) deneysel risk fonksiyonu ile değiştirilir.

$$E(w) = \frac{1}{\ell}\sum_{i=1}^{\ell} \mathrm{L}\big(y_i, \mathrm{f}(\boldsymbol{x_i}, \mathbf{w})\big) \qquad (4.\ 12)$$

Deneysel Risk Minimizasyonu (DRM) E(w) problemini çözümlemek amacıyla, w ϵ W kümesi üzerinde E(w) değerini en aza indiren f(x,w_l^*) hesaplamasının en yakın hesaplama olduğu varsayılır. Tümevarım prensibi olarak adlandırılan bu prensip, en küçük kareler, en çok olabilirlik gibi yöntemlerde de kullanılmaktadır.

Bu noktada DRM için tümevarım ilkesinin tutarlılığı, diğer bir ifadeyle R(w_l^*)) değerinin w ϵ W kümesi özelinde ℓ değeri sonsunuza giderken minimum olma özelliğini koruyup koruyamadığına dair bir durum aranır. Bu amaçla, Deneysel Risk E(w), gerçek riske R(w), tüm f(x,w), w ϵ W seti üzerinden uniform (tekdüze) yakınsanır.

$$Prob\left\{\sup_{w\,\epsilon\,W} |R(w) - E(w)| > \varepsilon\right\} \to 0 \quad \ell \to \infty \qquad (4.\ 13)$$

DRM probleminde ikinci olarak, ℓ değeri arttıkça hesaplanacak yakınsamanın ne kadar hızlı olacağı önem taşır. Burada geçerli koşul tüm fonksiyonların tekdüze yakınsanmasıdır.

4.7.2. VC boyutu

Deneysel riskin gerçek riske tekdüze yakınsaması teorisi, yakınsama hızına uygun, yeterli koşulların ve sınırların tanımlanmasını içermektedir. P(x,y) dağılım fonksiyonundan bağımsız olan bu sınırlar, öğrenme makinesi tarafından uygulanan fonksiyonlar kümesinin kapasitesinin nicel ölçüsüne yani kümenin VC (Vapnik - Chervonenkis) boyutuna dayanır.

İkili bir problemi ele aldığımızda, y ϵ {0, 1} ve f(x, w), w ϵ W gösterge fonksiyon sınıfı olarak kabul edilir. Kayıp fonksiyonu bu durumdan iki değer alabilir: y =f(x, w) ise, L(y, f(x, w)) =0 ve diğer durumlarda L(y, f(x, w)) = 1. Böylece gerçek risk, P(w) tarafından belirlenen hata olasılığı olarak karşımıza çıkar. Deneysel Risk ise, v(w) tarafından belirlenen eğitim setindeki hataların sıklığıdır.

Gösterge fonksiyonları kümesinin alabileceği maksimum VC boyutu sayısı h olarak belirlendiğinde, 2^{h} mümkün durum ile parçalara ayrılabilir. Örnek olarak, n boyutlu uzayda, doğrusal durumda en fazla n +1 noktaya bölünebildiğinden, h= n+1 olarak belirlenir.

$$Prob\left\{\sup_{w\,\epsilon\,W}|P(w)-v(w)|>\varepsilon\right\}<\left(\frac{2le}{h}\right)^{h}exp\{-\varepsilon^{2}l\} \qquad (4.\ 14)$$

1 – η olasılığı ile tüm w W değerleri için:

$$P(w)-v(w)+C_{0}\left(\frac{l}{h},\eta\right) \qquad (4.\ 15)$$

Güven aralığı ile,

$$C_{0}\left(\frac{l}{h},\eta\right)=\sqrt{\frac{h\left(\frac{ln2l}{h}+1\right)-ln\,\eta}{l}} \qquad (4.\ 16)$$

4.16'da verilen eşitlik tüm w ϵ W için, deneysel riski v(w) minimize eden w^{*}ile birlikte gerçek risk P(w)'nin sınırlarını oluşturur.

4.14'te verilen $|\mathrm{P(w)}-\mathrm{v(w)}|$ sapmasının P(w) fonksiyonu için maksimum olması ve ½'ye yakın olması beklenir; çünkü bu değer hata sapmasını maksimum yapan $\sigma(\mathrm{w})=\sqrt{\mathrm{P(w)}\left(1-\mathrm{P(w)}\right)}$ değeridir. Böylelikle güven aralığındaki 4.16'da en kötü sınır, en kötü karar koşulu tarafından belirlenir. Tekdüze yakınsanan P(w) için,

$$Prob\left\{\sup_{w\,\epsilon\,W}\frac{P(w)-v(w)}{\sigma(w)}>\varepsilon\right\} \quad (4.\ 17)$$

Böylece, ilgili sapmanın varyansı (P(w)-v(w))/(σ(w), w parametresinden bağımsızlaşır. 4.17'de verilen olasılık için bir sınır oluşursa, bu sınır P(w) için tekdüze bir sınır olur. Bu sınır henüz oluşturulmamıştır, ancak P(w)<<1 için $\sigma(\mathrm{w}) \cong \sqrt{\mathrm{P(w)}}$ yaklaşımı doğrudur ve aşağıda verilen eşitsizlik söz konusudur:

$$Prob\left\{\sup_{w\,\epsilon\,W}\frac{P(w)-v(w)}{\sqrt{P(w)}}>\varepsilon\right\}<\left(\frac{2le}{h}\right)^{h}exp\left\{-\frac{\varepsilon^{2}l}{4}\right\} \quad (4.\ 18)$$

1 – η olasılığı ile tüm w ϵ W değerleri için:

$$P(w)<v(w)+C_1\left(l/_h, v(w), \eta\right) \quad (4.\ 19)$$

Güven aralığı:

$$C_1\left(l/_h, v(w), \eta\right)=2\left(\frac{h(\frac{ln2l}{h}+1)-ln\eta)}{l}\right)+\left(1+\sqrt{1+\frac{v(w)l}{h(\frac{ln2l}{h}+1)-ln\eta)}}\right) \quad (4.\ 20)$$

Güven aralığı artık v(w)'ye bağlıdır ve v(w) = 0 için aşağıdaki (4.21) halini alır.

$$C_1(l/h, 0, \eta)=2C_0^2(l/h, 0, \eta) \quad (4.\ 21)$$

4.7.3. Yapısal risk minimizasyonu

Deneysel Risk Minimizasyonu teorisi 4.15 ve 4.19'da verilen eşitsizliklere göre şekillenir. ℓ/h değeri büyüdükçe, güven aralığı olan C_0 ve C_1 küçülür hatta göz ardı edilebilir. Devamında, gerçek risk deneysel risk ile sınırlandırılır ve

eğitim setindeki hatanın frekansı küçük ise test setindeki hata olasılığının küçük olması beklenir. ℓ/h değerinin küçük olduğu durumlarda ise güven aralığı göz ardı edilemez ve v(w) = 0 durumu hata olasılığının küçük olduğunu garanti etmez. Bu durumda P(w)'nin en aza indirilmesi, v(w) ve güven aralığının eşzamanlı olarak en aza indirilmesi ile mümkündür. Böylelikle, VC boyutu kontrol altında tutulur.

Bu amaçla Sp = {f(x, w), w ∈ Wp} kümesinin iç içe geçmiş alt kümeleri kullanılır.

$$s_1 \subset s_2 \subset \cdots \subset s_n$$

Alt kümelere karşılık aşağı verilen VC boyutları kullanılır.

$$h_1 < h_2 < \cdots < h_3$$

Diğer bir deyişle, yapısal riski en aza indirmek için öncelikle deneysel risk en aza indirilmelidir. Daha sonra deneysel riskleri ve güven aralığını en aza indirecek en uygun S ögesi seçilir. Bu süreç h değerinin arttığı, minimum deneysel riskin azaldığı buna karşılık güven aralıklarının büyüdüğü bir dengeleme mantığı paralelinde gerçekleşir.

4.7.4. Destek vektör makinesi ile sınıflandırma

Destek Vektör Makineleri istatistiksel öğrenme teorisi ve yapısal risk minimizasyonu metodolojilerinin yardımıyla, sınıfları birbirlerinden ayıracak en optimum hiperdüzlemi (hyperline) üretmeyi amaçlamaktadır, bu hiperdüzlem çizilirken sınırlar (marjin) düzleme en yakın verilere destek vektör makineleri adı verilir (Erdal, 2015).

Bir hiperdüzlemden sınırlarının bir tarafına en kısa mesafe ile, diğer tarafına giden en kısa mesafenin eşit olduğunu söyleyebiliriz. Sınırlar hesaplanırken bu mesafe aslında her iki sınıfın en yakın eğitim grubuna en kısa mesafesidir.

Ayırıcı bir doğrusal hiperdüzlem şu şekilde ifade edilebilir:

$w_x + b = 0$

Burada $w = w_1, w_2 \dots w_n$ n tane değişkeni olan vektör ağırlığını, b ise sabit kaysayıyı ifade etmektedir. İki adet değişken olduğunu düşünülürse formül aşağıdaki şekilde tekrar yazılabilir:

$$w_0 + w_1 x_1 + w_2 x_2 = 0 \qquad (4.\ 22)$$

Böylece, hiperdüzlemin üzerindeki herhangi bir nokta;

$w_0 + w_1 x_1 + w_2 x_2 > 0$

Benzer olarak, hiperdüzlemin altındaki herhangi bir nokta:

$w_0 + w_1 x_1 + w_2 x_2 < 0$

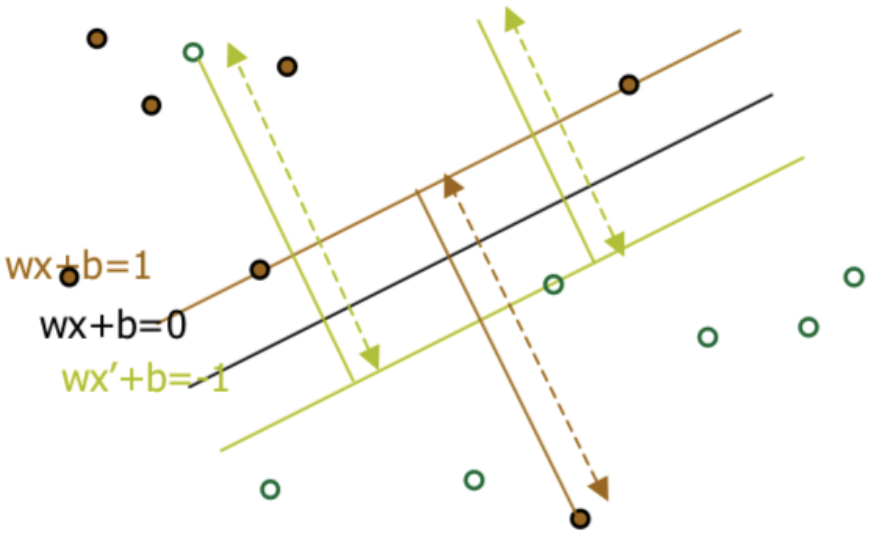
şeklinde gösterilebilir. Hiperdüzlemin sınırları ise aşağıda belirtildiği gibidir.

$w_0 + w_1 x_1 + w_2 x_2 \geq 1 \quad y_i = +1$

$w_0 + w_1 x_1 + w_2 x_2 \leq -1 \quad y_i = -1$

Verilen eşitsizlikler birleştirilir ve Şekil 4.3'te verilen düzlemi oluşturur:

$$y_i(w_0 + w_1 x_1 + w_2 x_2) \geq 1\ , \forall_i \qquad yi(wxi + b) \geq 1\ i,\dots,I \qquad (4.\ 23)$$



Şekil 4. 3. Hiperdüzlem Çizimi (Jakkula, 2006)

Optimal hiperdüzlemin oluşturulabilmesi için, hiperdüzlem sınıfları (d) maksimum olmalıdır. Bu amaçla w ağırlık vektörü normu olan $\|w\|$'i minimize etmek için Lagrange çarpanı yöntemi kullanılır (Erdal, 2015).

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i \left[y_i(wx+b) - 1\right] \tag{4. 24}$$

Lagrange çarpanları yöntemiyle, w ve b kısıtlarına göre $L(w,b,\alpha)$ fonksiyonunun ekstremum noktaları bulunur.

$$\sum_{i=1}^{l} \alpha_i \, y_i \, , \alpha_i \geq 0, i = 1, \dots l \tag{4. 25}$$

$$w = \sum_{i=1}^{l} \alpha_i \, y_i x_i \, , \alpha_i \geq 0, i = 1, \dots l \tag{4. 26}$$

$\alpha \neq 0$ noktaları destek vektörleri olarak anılır. Karush Kuhn Tucker (KKT) koşulları denklemi kullanılarak $\alpha \neq 0$ sağlanmadığı noktalarda aşağıdaki eşitliğe dönüşür (Erdal, 2015).

$\alpha_i[y_i(\mathrm{wx}+\mathrm{b})-1]$

$$L(\alpha)\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{J}\alpha_i\alpha_j y_i y_j\left(x_i x_j\right) \quad (4.\ 27)$$

$L(\alpha)$ Fonksiyonu maksimize edilerek problem çözüme kavuşturulur.

4.7.5. Yumuşak (soft) marjin

Gerçek hayatta her veri setinde veriler doğrusal olarak ayrılmaz. Bu durumda hesaplanan hiperdüzlemler ile ilgili birtakım sorunlar ortaya çıkar.

- X negatif sınıftan ise, $w_t x+b=-1$ x pozitif sınıfa ait ise $w_t x+b=1$ kullanılacağından denkleme göre w ve b yok olur.
- W ve b parametreleri yok olduğunda marjin oluşturulamaz, bu nedenle marjini maksimize etme amacı yok olur. Böylece, hiperdüzlemi hesaplayan formül geçersiz kılınır (Murty ve Raghava, 2016).

Doğrusal olarak ayrılan veri setlerine günlük hayatta çok sık karşılaşılmadığı için maksimum marjine sahip sınıflandırıcıyı seçmek aşırı öğrenme problemine neden olabilir. Bu sorunu çözmek için uygulanabilecek ilk yöntem, sınıflandırma hatasına izin vermektir (Rossi ve Villa, 2006).

Eğer $w_t x+b=wx+b>0$ ise X pozitif sınıfa atanır.
Eğer $w_t x+b=wx+b<0$ ise X negatif sınıfa atanır.

Pozitif sınıfın örüntüsü 1'den büyük ise e_1 hatası ile adlandırılır. Negatif sınıf ise, e_2 (<1) hatasıyla ilişkilidir, ancak burada yanlış sınıflama yoktur. Bu tür hataları minimize etmek adına, ölçme işleminde bu tür hataların toplamına karşılık gelen bir hata terimi (e_i) eklenir. $e_i=0$ ise X_i sınıflandırma hatası yoktur. Ek olarak, e_i negatif olamaz ($e_i=\geq 0\ \forall i$).

Benzer şekilde, kısıtlara hata terimi eklenir:

$$w'x_i + b \geq -1 + e_i, y_i = -1 \qquad (4.\ 28)$$
$$w'x_i + b \leq 1 - e_i, y_i = 1$$

İlgili eşitlik için 3 olasılık bulunur:

- e_i= 0 olabilir. Bu durumda $x_i$destek düzlemindedir, ($w'x_i + b = 1$). Böylece doğru bir şekilde sınıflandırılır.

- $e_i < 1$ve $w'x_i + b \geq 1 - e_i > 0$ ise, x_i doğru bir şekilde sınıflandırılmaz. Ancak, x_i yine de marjin üzerindedir.

- $e_i \geq 1$ ve $w'x_i + b \leq 0$, x_i yanlış sınıflandırılır.

$$Minimize \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} e_i \qquad (4.\ 29)$$

Hata terimi ile birlikte veri noktalarının yanlış sınıflandırılmasına izin verilir ve nesnel fonksiyona göre marjin maksimum yapılırken yanlış sınıflandırma miktarı en aza indirilir. Formüle C (cost) parametresi, yumuşak marjin maliyet fonksiyonunun parametresi olarak eklenir. C, marjin boyutu ile eğitimdeki hata miktarı arasındaki dengeyi belirleyen bir parametredir (Yu ve Kim, 2012).

C parametresinin değeri ne kadar büyürse, sınıflandırma işlemi sert marjin sınıflandırmasına o kadar yaklaşır. Bu nedenle C parametresi sınıflandırma doğruluğu üzerinde güçlü bir etkiye sahiptir (Tuba vd., 2016).

Veri noktalarının doğrusal ayrıldığı durumda aranan $\alpha_i \geq 0$ koşulu, bu durumda $C \geq \alpha_i \geq 0$ koşuluna dönüşür. Vektörün normunu minimize edebilmek adına Langrange çarpımı yeni değişkenler ile tekrar oluşturulur. Çarpıma μ_i değişkeni

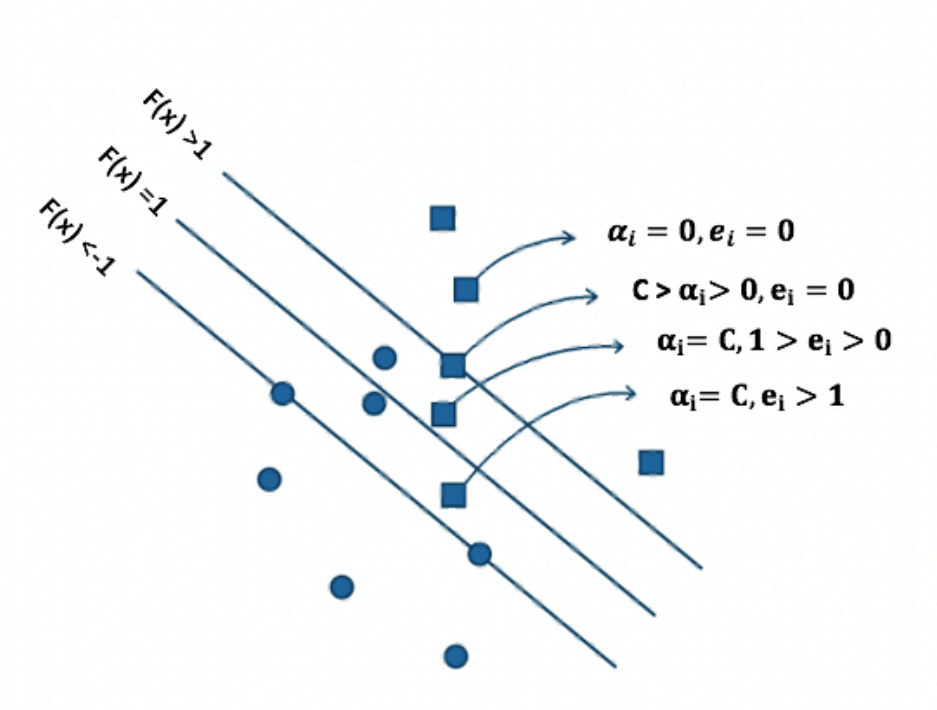
eklenir. μ_i, Langrage değişkeni $e_i \geq 0$ veya -$e_i \leq 0$ eşdeğeri olan -$e_i \leq 0 \geq e_i$ kısıtlama ile ilişkili Lagrange değişkenidir. Diğer bir deyişle, μ_i fonksiyonun negatif değer vermesini önleyen çarpandır (Yu ve Kim, 2012).

$$\mu_i e_i = 0, i = 1,2, \dots n \quad (4.\ 30)$$

$$\alpha_i + \mu_i = C$$

$$e_i = 0\,, \quad \alpha_i < C \text{ olduğu durumda}$$

$$e_i \geq 0\,, \quad \alpha_i = C \text{ olduğu durumda}$$



Şekil 4. 4. C Maliyet Parametresi Gösterimi (Yu ve Kim, 2012)

Şekil 4.4'te gösterildiği üzere marjin dışındaki veri noktaları α = 0 ve e_i= 0 değerlerine sahip olacaktır. Marjin üstündeki noktalar ise C> α > 0 olacak ancak yine de e_i= 0 olacaktır. Marjin içindeki veri noktaları için ise α = C olacaktır. Bunlar arasında, doğru sınıflandırılanların 1 >e> 0, yanlış sınıflandırılan puanları ise e> 1 olacaktır (Yu ve Kim, 2012). Bu koşullar altında optimal parametre değerleri seçilir.

Değişkenler Langrage çarpımında yerine konulduğunda:

$$L = \frac{1}{2}w'w + C\sum_{i=1}^{n} e_i + \sum_{i=1}^{n} \alpha_i \left(1 - e_i - y_i(w'x_\text{i} + b)\right) - \sum_{i=1}^{n} \mu_i e_i \quad (4.\ 31)$$

Benzer şekilde, α_i de, $y_i(w'x_\text{i} + b) \geq 1 - ei$ ya da $1{-}e_i - y_i(w'x_\text{i} + b) \leq 0$ sabitleri ile Langrange değişkenidir.

Bu durumda ağırlık vektörü

$$w = \sum_{\text{i}=1}^{n} \alpha_i y_i x_i \quad (4.\ 32)$$

L'yi b ile ayırarak ve 0'a eşitleyerek,

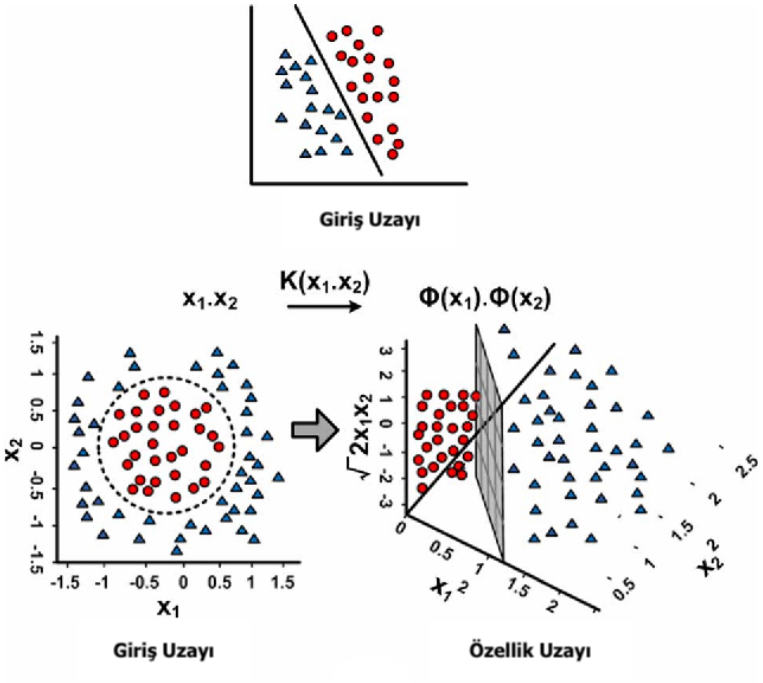
$$\sum_{\text{i}=1}^{n} \alpha_i y_i = 0 \quad (4.\ 33)$$

Soft marjinlerin çıkış noktasında olduğu gibi, tamamen doğrusal dağılan veriyi gerçek hayat problemlerinde elde etmek oldukça zordur. Bu amaçla bir hata terimi eklenerek hiperdüzlemi oluştururken hesaplamanın hata yapmasına izin verilir. Ancak bazı durumlarda soft marjin yöntemi de yetersiz kalabilmektedir. Buradan doğan ihtiyaç ile, doğrusal ayrılamayan verilerin girdi noktalarını özellik noktalarına eşleyerek yeni ve çok boyutlu uzay yaratılabilir ve doğrusal olmayan yüzeyleri ayırma çalışması yapılabilir (Cortes ve Vapnik, 1995).

Bu noktada eğitim verisindeki noktaların uzayına referans eden "giriş uzayı" ve verilerin dönüşümden sonraki çok boyutlu uzaya referans eden "özellik uzayı (φ(Xi)" terimleri kullanılır. Haritalama işlemini çalışmanın başında kestirmek zorunlu değildir; giriş uzayında x noktasını sonsuz boyutlu özellik uzayına haritalamak mümkün olabilir.

4.7.6. Kernel çekirdek fonksiyonları

Doğrusal olmayan DVM'lerin temel fikri, eğitim verilerini haritalama φ(x) üzerinden daha yüksek boyutlu bir özellik alanına eşleştirmek ve orada maksimum marjine sahip bir ayırıcı hiperdüzlem oluşturmaktır. Böylece, yeni uzayda doğrusal olmayan bir karar sınırı oluşturulur. Şekil 4.5'te de gösterildiği üzere, bir çekirdek fonksiyonu olan K(x,z) = ⟨φ(x),φ(z)⟩ kullanarak, ayıran hiperdüzlemi açık bir şekilde girdi uzayından özellik uzayına eşleyen hesaplamayı gerçekleştirmek mümkündür (Scholkopf, 2000).



Şekil 4. 5. Kernel Fonksiyonları ile Özellik Uzayına Haritalandırma (Bin Altaf ve Yoo, 2016)

Özellik alanında bir hiperdüzlem oluşturmak için öncelikle n boyutlu giriş vektörü x'i N boyutlu bir özellik vektörüne dönüştürmek için N boyutlu vektör fonksiyonu ϕ seçimi yapılır (Cortes ve Vapnik, 1995).

$$\phi = \Re^n \longrightarrow \Re^N$$

Destek vektör makineleri için özellik uzayı yaratma işleminde Hilbert uzayında nokta-ürün eşleşmesi fikrinden yararlanılır (Anderson ve Bahadur, 1966).

$$\phi(u).\phi(v) \equiv K(u,v) \tag{4. 34}$$

Hilbert-Schmidt Teorisi'ne göre K(u,v) ∈ L_2 ile herhangi bir K(u, v) simetrik bir fonksiyonu,

$$K(u,v) = \sum_{i=1}^{\infty} \lambda_i \phi_i\,(u).\,\phi_i(v) \tag{4. 35}$$

K (u, v) tarafından tanımlanan integral işleminin $\lambda_i \in \Re$ ve ϕ_i 'nin özdeğer ve öz fonksiyonları şeklinde genişletilebilir.

$$\int K(u,v)\,\phi_i(u)du = \lambda_i \phi_i(v) \tag{4. 36}$$

Bir özellik uzayında nokta-ürün tanımını oluşturabilmek için yeterli koşul, genişletme işleminde tüm öz değerlerinin pozitif olmasıdır. Bu katsayıların pozitif olduğunu garanti etmek için, gerekli ve yeterli durum Mercer teoremi ile sağlanır.

$$\int K(u,v)\,g(u)g(v)dudv > 0 \tag{4. 37}$$

Tüm g değerleri için

$$\int g^2\,(u)du < \infty \tag{4. 38}$$

Böylece Mercer Teoremini karşılayan nokta-ürün fonksiyonu oluşturulur.

$$K(u,v) = \exp\left(-\frac{|u-v|}{\sigma}\right) \tag{4. 39}$$

K çekirdek işlevi için çeşitli çekirdek fonksiyonları kullanılabilmektedir. Radyal tabanlı kernel, polinomial kernel, sigmoid kernel en çok kullanılan çekirdek

fonksiyonlarıdır (Amami vd., 2013). Girdi uzayından özellik uzayına olan haritalandırma işlemini yapabilecek çekirdek fonksiyonları aşağıda sıralanmıştır.

- **Radyal Tabanlı Kernel Fonksiyonu**

Gaussian radyal temelli fonksiyon (RBF) sınıflandırıcısıdır. Burada çekirdek fonksiyonunun sonucu x_i'in x_j'den Öklid uzaklığına bağlıdır. Destek vektörü σ, RBF'nin merkezi olacak ve bu destek vektörünün veri alanı üzerindeki etki alanını belirleyecek kullanıcı tanımlı bir parametredir. Bu değer büyüdükçe daha yumuşak bir karar yüzeyi ve daha düzenli karar sınırı ortaya çıkar. Bunun nedeni, yüksek bir RBF'nin bir destek vektörünün daha geniş bir alan üzerinde güçlü bir etkiye sahip olmasını sağlamasıdır. Daha büyük bir değer de destek vektörlerinin sayısını aşağı yönlü oynatacaktır (Girma, 2009).

$$K(x_{i,}x_j) = \exp\left(-\frac{\|x_{i,}-x_j\|^2}{2\sigma^2}\right) \quad (4.\ 40)$$

- **Polinominal Kernel Fonksiyonu**

Polinominal çekirdek fonksiyonu, durağan olmayan bir fonksiyondur. D parametresi polinomun derecesini veren parametre iken r sabit terimdir (Amami vd., 2013). Polinom çekirdeği fonksiyonu yönlüdür, yani çıkış düşük boyutlu uzaydaki iki vektörün yönüne bağlıdır. Bunun nedeni çekirdekteki nokta ürünüdür. Aynı yöndeki tüm vektörler çekirdekten yüksek bir çıkışa sahip olacaktır. Çıktının büyüklüğü de xj vektörünün büyüklüğüne bağlıdır (Girma, 2009).

$$K(x_i, x_j) = \left(\sigma x_i^T x_j + r\right)^d, \sigma > 0 \quad (4.\ 41)$$

- **Sigmoid Kernel Fonksiyonu**

Sinir ağları kökenli olan Sigmoid çekirdek fonksiyonunda κ ve θ kullanıcı tanımlı parametrelerdir.

$$K(x, y) = tanh(\kappa X^T y + \theta) \quad (4.\ 42)$$

Seçilen çekirdek hilesi ile girdi uzayından özellik uzayına vektör yaratılır. Bu dönüşüm ile birlikte:

- Giriş uzayında doğrusal olmayan bir sınır, özellik uzayında doğrusal bir karar sınırı kullanılarak yakalanabilir.
- Dönüştürme uygunsa, giriş uzayında sağlanamayan doğrusallık sorunu, özellik alanında elde edilen basit doğrusal sınıflayıcı ile çözülebilir
- Genellikle özellik uzayı giriş uzayı göre daha yüksek boyuta sahiptir.

Destek Vektör Makinelerinin çekirdek fonksiyonları kullanılarak sınıflandırılmasında öncelikli olarak N boyutlu doğrusal ayırıcı w ve b sapma ile sonra dönüştürülmüş vektörler kümesi oluşturulur.

$$\Phi(x_i) = \phi_1(x_i), \phi_2(x_i), \dots, \phi_N(x_i), \quad i = 1, \dots, l \quad (4.\ 43)$$

Bilinmeyen vektör x'in sınıflandırılması, önce vektörü ayırma alanına ($x \mapsto \phi(x)$) dönüştürerek ve daha sonra fonksiyonun işaretini alarak yapılır.

$$f(x) = w.\phi(x) + b \quad (4.\ 44)$$

Yumuşak marjin sınıflandırma yönteminin özelliklerine göre, özellik alanında w vektörü destek vektörlerinin doğrusal bir kombinasyonu olarak yazılabilir.

$$w = \sum_{i=1}^{l} y_i \alpha_i \phi(x_i) \quad (4.\ 45)$$

W ağırlık vektörü 4.45'teki fonksiyonda yerine konulduğunda sınıflandırma fonksiyonu yalnızca nokta uzayına bağlı olarak şekillenir.

$$f(\mathrm{x}) = \phi(\mathrm{x}).w + b = \sum_{\mathrm{i}=1}^{\mathrm{l}} \mathrm{y_i}\alpha_\mathrm{i}\phi(\mathrm{x}).\phi(x_\mathrm{i}) + b \quad (4.\ 46)$$

Buradan hareketle b:

$$b = y_i - W^t\phi(X_i) = y_p - \sum_{X_i \in S} \alpha_\mathrm{i} y_i \phi\left((X_i)\right)^t \phi(X_i) \quad (4.\ 47)$$

Girdi uzayından özellik uzayına haritalanan veriler ile maksimum marjini oluşturacak sınıflayıcı belirlenerek Lagrangian probleminde yeniden formüle edilir (Hofmann, 2006).

$$L_D\left(\underset{\alpha}{\rightarrow}\right) = \sum_{i=1}^{t} \alpha_i - \sum_{i=1}^{t}\sum_{j=1}^{t} \alpha_i\alpha_j y_i y_j \phi\left(\underset{x_i}{\rightarrow}\right)^T \phi\left(\underset{x_j}{\rightarrow}\right) \quad (4.\ 48)$$

Optimal ağırlık vektörü

$$\underset{w_o}{\rightarrow} = \sum_{j=1}^{t} \alpha_i\alpha_j y_i y_j \phi\left(\underset{x_i}{\rightarrow}\right) \quad (4.\ 49)$$

Optimal hiperdüzlem

$$\left(\underset{w_o}{\rightarrow}\right)^T \underset{x}{\rightarrow} + b_0 = \sum_{i=1}^{l} \alpha_i, y_{0i} \phi\left(\underset{x_i}{\rightarrow}\right)^T \phi\left(\underset{x}{\rightarrow}\right) + b_0 = 0 \quad (4.\ 50)$$

Optimal karar fonksiyonu,

$$g\left(\underset{x}{\rightarrow}\right) = sgn\left(\left(\underset{w_o}{\rightarrow}\right)^T \underset{x}{\rightarrow} + b_0\right) = sgn \sum_{i=1}^{l} \alpha_i, y_{0i} \phi\left(\underset{x_i}{\rightarrow}\right)^T \phi\left(\underset{x}{\rightarrow}\right) + b_0 \quad (4.\ 51)$$

Her yeni test verisi için destek vektörü çekirdek işlevinin yeniden hesaplanması gerekir.

DVM algoritmasının avantajları:

- Hatalı veriyi ayıklamada güçlü bir yöntemdir.

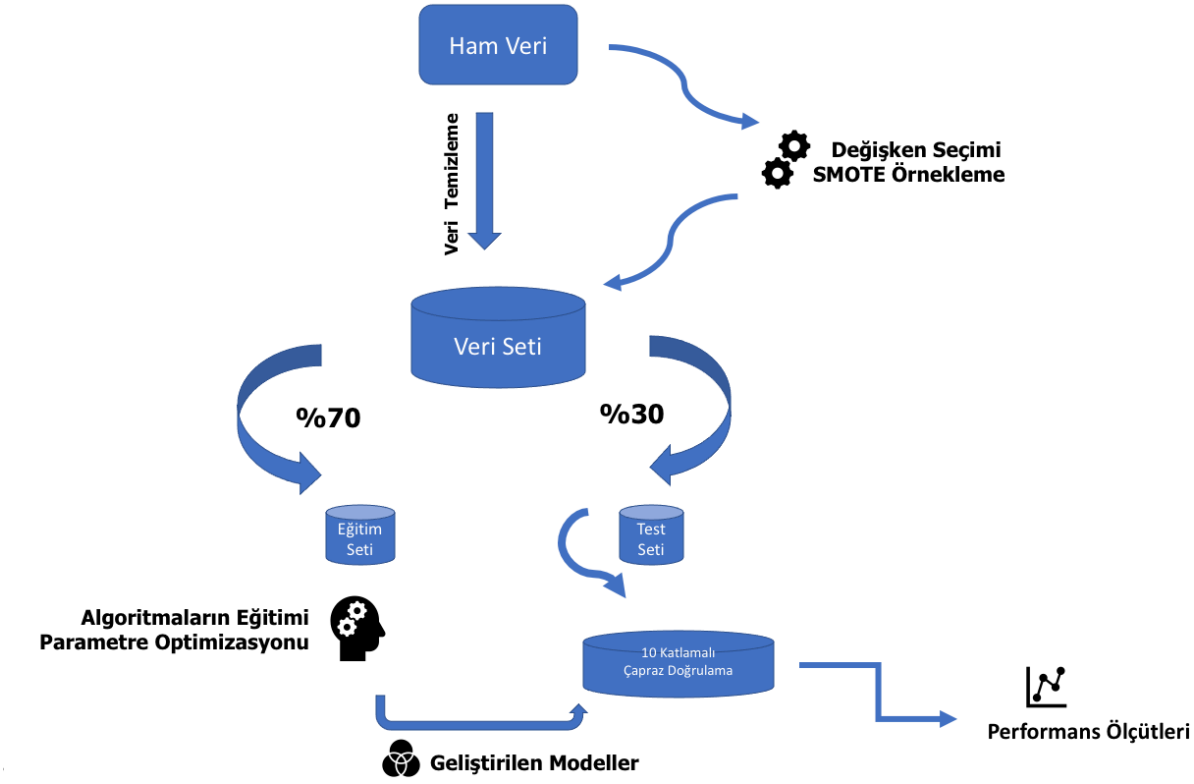
- Özellikle çok yüksek boyutlu verilerde etkin sonuçlar vermektedir.

DVM algoritmasının dezavantajları:

- Sınıflandırma süresi uzundur, bu nedenle zaman maliyeti fazladır (Archana ve Elangovan, 2014).

5. UYGULAMA

Çalışmanın bu aşamasında, tanıtılan makine öğrenmesi algoritmalarının iflas tahmin problemine uygulanması ile ilgili adımlara yer verilmiştir. Şekil 5.1.'de gösterildiği üzere, bu bölümde ilk olarak veri seti tanıtılmış, değişken seçimi adımları anlatılmıştır. Eğitim verisindeki hedef değişken oranından ötürü örnekleme çalışması yapılmış, bu adım sonrasında veri seti eğitim ve test verisi olarak ayrılmıştır. Konu edilen çalışmaları takiben Naive Bayes, k En Yakın Komşu ve Destek Vektör Makineleri algoritmaları ile sınıflandırma işlemi gerçekleştirilmiştir. Çalışma öncesinde tanıtılan performans ölçütleri ile algoritmalar karşılaştırılmıştır.



Şekil 5. 1. Uygulama Adımları Şeması

5.1. Araştırmada Kullanılan Veri Seti

Araştırmada kullanılan veriler, Kaliforniya Üniversitesi tarafından paylaşılan, makine öğrenmesi çalışmalarına yönelik açık kaynaklı veri setlerinin bulunduğu bir platformdan (UCI) alınmıştır. Veri toplanırken, imalat sektöründe faaliyet gösteren Polonyalı şirketler dikkate alınmıştır. 2004 yılından itibaren Polonya'da faaliyet gösteren şirketlerin finansal bilgilerine Piyasalar Bilgi Servisi (EMIS) veritabanı aracılığı ile ulaşılmıştır.

EMIS (www.emis.com) veritabanındaki verilerin kullanılabilirliğine göre, 2007-2013 yılları arasından iflas eden şirketler ve 2000-2012 arasında faaliyetlerine devam edebilen şirketler analize dahil edilmiştir. Böylece, iflas etmiş ve hali hazırda faaliyet gösteren şirket verileri birleştirilerek veri seti oluşturulmuştur. 2007-2013 döneminde yaklaşık 700 iflas eden firma için yaklaşık 2400 bilanço; 2000-2012 döneminde halen faaliyette olan 10.000'den fazla şirket için eğer bu firmalar arasında iflas beyan etmiş şirket var ise çalışmanın dışında tutularak 65 binden fazla bilanço dikkate alınmıştır. Çalışma sonucunda veri akışı sağlanmış Tablo 5.1'de verilen 64 finansal gösterge seçilerek veri seti oluşturulmuştur.

Tablo 5. 1. Değişken Listesi

Değişkenin Kodu	Değişkenin Tanımı	Değişkenin Kodu	Değişkenin Tanımı
X1	net kar / toplam varlıklar	X33	işletme giderleri / kısa vadeli yükümlülükler
X2	toplam yükümlülükler / toplam varlıklar	X34	işletme giderleri / toplam yükümlülükler
X3	çalışma sermayesi / toplam varlıklar	X35	satışlardaki kar / toplam varlıklar
X4	dönen varlıklar / kısa vadeli borçlar	X36	toplam satışlar / toplam varlıklar
X5	[(nakit + kısa vadeli menkul kıymetler + alacaklar - kısa vadeli borçlar) / (işletme giderleri - amortisman)] * 365	X37	(cari varlıklar - stoklar) / uzun vadeli yükümlülükler
X6	elde tutulan kazançlar / toplam varlıklar	X38	sabit sermaye / toplam varlıklar
X7	EBIT / toplam varlıklar	X39	satış / satış kar
X8	özkaynak / toplam yükümlülüklerin defter değeri	X40	(cari varlıklar - stok - alacaklar) / kısa vadeli yükümlülükler

X9	satış / toplam varlıklar	X41	toplam yükümlülükler / ((faaliyet karı + amortisman) * (12/365))
X10	özkaynak / toplam varlıklar	X42	faaliyet faaliyetleri / satış kar
X11	(brüt kar + olağanüstü kalemler + finansal giderler) / toplam varlıklar	X43	rotasyon alacakları + gün içinde stok devir hızı
X12	brüt kar / kısa vadeli borçlar	X44	(alacaklar * 365) / satış
X13	(brüt kar + amortisman) / satış	X45	net kar / stok
X14	(brüt kar + faiz) / toplam varlıklar	X46	(cari varlıklar - stok) / kısa vadeli yükümlülükler
X15	(toplam yükümlülükler * 365) / (brüt kar + amortisman)	X47	(stok * 365) / satılan ürünlerin maliyeti
X16	(brüt kar + amortisman) / toplam yükümlülükler	X48	FAVÖK (Faiz Amortisman ve Vergi Öncesi Kar) / toplam varlıklar
X17	toplam varlıklar / toplam yükümlülükler	X49	FAVÖK (faaliyet faaliyetlerinden elde eden kar - amortisman) / satışlar
X18	brüt kar / toplam varlıklar	X50	dönen varlıklar / toplam yükümlülükler
X19	brüt kar / satış	X51	kısa vadeli borçlar / toplam varlıklar
X20	(stok * 365) / satış	X52	(kısa vadeli yükümlülükler * 365) / satılan ürünlerin maliyeti)
X21	satış (n) / satış (n-1)	X53	özkaynak / sabit kıymetler
X22	faaliyetlerinden elde edilen kar / toplam varlıklar	X54	sabit sermaye / sabit kıymetler
X23	net kar / satış	X55	çalışma sermayesi
X24	brüt kar (3 yıl içinde) / toplam varlıklar	X56	(satış - satılan ürünlerin maliyeti) / satış
X25	(özkaynak - sermaye sermayesi) / toplam varlıklar	X57	(cari varlıklar - stok - kısa vadeli borçlar) / (satışlar - brüt kar - amortisman)
X26	(net kar + amortisman) / toplam yükümlülükler	X58	toplam maliyetler /toplam satışlar
X27	faaliyet faaliyetlerinden kar / finansal giderler	X59	uzun vadeli yükümlülükler / özkaynaklar
X28	çalışma sermayesi / sabit kıymetler	X60	satış / envanter
X29	toplam varlıkların logaritması	X61	satış / alacaklar
X30	(toplam yükümlülükler - nakit) / satış	X62	(kısa vadeli borçlar *365) / satış
X31	(brüt kar + faiz) / satış	X63	satış / kısa vadeli yükümlülükler
X32	(cari borçlar * 365) / satılan ürünün maliyeti	X64	satış / sabit kıymetler

Toplanan verilere dayanarak tahmin dönemi 5 farklı yıla ayrılmıştır.

- 5 Yıl Sonra İflas Eden Firmalar: Tahmin döneminin 1.yılından itibaren finansal oranları ve 5 yıl sonraki iflas durumunu gösteren ilgili sınıf etiketini içerir. 7027 veriden, 271'i iflas etmiş şirketleri, 6756'sı iflas etmemiş şirketleri temsil etmektedir.

- 4 Yıl Sonra İflas Eden Firmalar: Tahmin döneminin 2. Yılından itibaren finansal oranları ve 4 yıl sonraki iflas durumunu gösteren ilgili sınıf etiketini içerir. 10173 veriden, 400'ü iflas etmiş şirketleri, 9773'ü iflas etmemiş şirketleri temsil etmektedir.

- 3 Yıl Sonra İflas Eden Firmalar: Tahmin döneminin 3. Yılından itibaren finansal oranları ve 3 yıl sonraki iflas durumunu gösteren ilgili sınıf etiketini içerir. 10503 veriden, 495'i iflas etmiş şirketleri, 10008'i iflas etmemiş şirketleri temsil etmektedir.

- 2 Yıl Sonra İflas Eden Firmalar: Tahmin döneminin 4. Yılından itibaren finansal oranları ve 2 yıl sonraki iflas durumunu gösteren ilgili sınıf etiketini içerir. 9792 veriden, 515'i iflas etmiş şirketleri, 9277'si iflas etmemiş şirketleri temsil etmektedir.

- 1 Yıl Sonra İflas Eden Firmalar: Tahmin döneminin 5. Yılından itibaren finansal oranları ve 1 yıl sonraki iflas durumunu gösteren ilgili sınıf etiketini içerir. 5910 veriden, 410'u iflas etmiş şirketleri, 5500'ü iflas etmemiş şirketleri temsil etmektedir.

5.2. Araştırmanın Metodolojisi

Bu bölümde, iflas tahmin probleminin çözümlenmesinde kullanılacak analizlere yer verilmiştir. Belirlenen veri seti üzerinde tahmine yardımcı olacak optimal değişkenlerin seçimi ve örnekleme yöntemleri anlatılmıştır. Ardından algoritmaların karşılaştırılmasında kullanılacak ölçütler tanıtılmıştır.

5.2.1. Değişken seçimi

Teknolojik gelişmeler veri toplama, veriyi veritabanı oluşturarak saklama faaliyetlerinde büyük geliştirme göstermiş ve geliştirmeye de devam

etmektedir. Ancak analizler yapılırken her durumda, her türlü veriye ulaşmak mümkün olmamaktadır. Öte yandan, farklı kaynaklardan veri toplanıyor ise periyodik olarak tutulmuş veriye erişim konusunda sıkıntılar yaşanabilmektedir. Bu nedenle analizler oluştururken ortaya çıkacak maliyetler düşünülerek, problemi optimum çözüme kavuşturacak (maksimum doğruluk sağlayacak) minimum değişken seçimi yapmak önemlidir.

Toplanan 64 değişken için hem ileriki çalışmalarda ilgili 64 değişkene ulaşmanın maliyeti, hem de eğer değişkenin anlamlı bir etkisi yoksa algoritmayı yorma (sınıflama süresinin uzaması, yanlı sonuçlar doğurması vb.) durumu göz önünde bulundurularak değişkenlerde azaltma yoluna gidilmiştir.

Çalışmada değişken seçimi aşamasında Kanıt Ağırlığı (Weight of Evidence (WOE)) ve Bilgi Değeri (Information Value (IV)) ölçütleri kullanılmıştır. Kanıt Ağırlığı ve Bilgi Değeri değişken seçimi için kullanılan basit ve güçlü tekniklerdendir. Lojistik regresyon tabanlı bu seçim yöntemi ağırlıklı olarak kredi skorlama problemlerinde iyi ve kötü müşterileri ayırmak için kullanılmaktadır (Krishnan, 2018).

1940'ların sonlarında ortaya çıkan ve başlangıçta skorlama için geliştirilen bilgi kuramına dayanan Kanıt Ağırlıkları (WOE) ve Bilgi Değeri (IV) son yıllarda segmentasyon ve değişken azaltma gibi kullanımlar için giderek artan bir ilgi görmektedir. WOE ve IV hesaplanması, ilgili durumun olması ve oluşmaması arasındaki zıtlığı gerektirdiğinden ikili sonuçların analizi ve tahmini ile sınırlandırılmıştır (Lin ve Hsieh, 2014).

WOE, bağımlı değişkenin istenilen değerini tahmin etmede gruplanmış özniteliğin gücünü ölçer. WOE hesaplanırken veriler genelde parçalara (gruplara) bölünür. Her gruptaki ilgili olayın olma durumu ve olmama durumu sayısı hesaplanır ve yüzde değeri alınır. Bu iki değer birbirine oranlanarak ln tabanında yazılır. WOE terminolojisinde bir olayın gerçekleşmesi (bu çalışma için firmanın iflas etme olasılığı) "Bad"(b) olarak tanımlanır. Olayın

gerçekleşmeme durumu (firmanın iflas etmemesi) ise "Good"(g) olarak tanımlanır (Zeng, 2014).

$$WOE = ln\left(\frac{\frac{b_1}{b_1 + b_2 \ldots + b_{k+1}}}{\frac{g_1}{g_1 + g_2 \ldots + g_{k+1}}}\right) \quad (5.\ 1)$$

Böylece her grup için tablo 5.2'de gösterildiği gibi WOE hesaplanır.

Tablo 5. 2. WOE Hesaplanması (Zeng, 2014)

Grup	x	Good Sayısı	Bad Sayısı	WOE
1	x_1 x_2 … x_{n_1}	g_1	b_1	$\ln\frac{b_1/(b_1+b_2\ldots+b_{k+1})}{g_1/(g_1+g_2\ldots+g_{k+1})}$
2	x_{n_1+1} x_{n_1+2} … x_{n_2}	g_2	b_2	$\ln\frac{b_2/(b_1+b_2\ldots+b_{k+1})}{g_2/(g_1+g_2\ldots+g_{k+1})}$
…	…	…	…	…
k	$x_{n_{k-1}+1}$ $x_{n_{k-1}+2}$ … x_{n_k}	g_k	b_k	$\ln\frac{b_k/(b_1+b_2\ldots+b_{k+1})}{g_k/(g_1+g_2\ldots+g_{k+1})}$
$k+1$	x_{n_k+1} x_{n_k+2} … $x_{n_{k+1}}$	g_{k+1}	b_{k+1}	$\ln\frac{b_{k+1}/(b_1+b_2\ldots+b_{k+1})}{g_{k+1}/(g_1+g_2\ldots+g_{k+1})}$

WOE yardımı ile, IV değişkenin genel tahmin gücünü değerlendirir. Böylece, rakip değişkenler arasında tahmin gücünü değerlendirmek için kullanılan ana kriter IV haline gelir. Bilgi Değeri aşağıdaki gibi hesaplanır (Lin ve Hsieh, 2014). Bir değişkenin özniteliklerinin WOE değerinin ağırlıklı toplamıdır; bu da olan olayların oranı ile olmayan olayların oranı arasındaki farktır. IV aşağıdaki gibi hesaplanır (Wu vd., 2013).

$$IV = \sum_i \left(\frac{olayın\ gerçekleşme\ sayısı}{Tüm\ durum\ içinde\ olayın\ gerçekleşme\ sayısı} - \frac{Olayın\ gerçekleşmeme\ sayısı}{Tüm\ durum\ içinde\ olayın\ gerçekleşmeme\ sayısı} \right) . WOE_i \quad (5.\ 2)$$

Hesaplanan IV değeri 0,02'den küçük ise tahmin gücünün olmadığı kanısına varılır. Buna karşılık IV değerinin 0,5 değerinden büyük olması değişkenin şüpheli tahmin edici kategorisinde yer almasına neden olur.

Tablo 5. 3. IV Değer Karşılıkları (Krishnan, 2018)

IV	Tanım
< 0.02	Tahmin Gücü Yok
0.1	Zayıf Tahminci
0.1 - 0.3	Orta Güçlü Tahminci
0.3 - 0.5	Güçlü Tahminci
> 0.5	Şüpheli Tahminci

Belirlenen kriterler çerçevesinde, Python 3.7 üzerinde değişkenlerin WOE ve IV değerleri hesaplanmıştır. Tablo 5.4'te ** ile işaretlenen değişkenler algoritmanın tahmin gücüne istenen yönde şekil veren, analize dahil edilen değişkenlerdir.

Tablo 5. 4. Değişkenler için hesaplanan IV Değerleri

Değişken	5 Yıl Sonra İflas Eden Firmalar için IV Değeri	4 Yıl Sonra İflas Eden Firmalar için IV Değeri	3 Yıl Sonra İflas Eden Firmalar için IV Değeri	2 Yıl Sonra İflas Eden Firmalar için IV Değeri	1 Yıl Sonra İflas Eden Firmalar için IV Değeri
X1	0,27**	0,26**	0,19**	0,19**	0,13**
X2	0,3**	0,43**	0,42**	0,25**	0,17**
X3	0,54	0,64	0,58	0,49**	0,09
X4	0,39**	0,22**	0,18**	0,16**	0,22**
X5	0,25**	0,31**	0,33**	0,25**	0,06
X6	0,92	0,93	0,79	0,63	0,04
X7	0,29**	0,4**	0,4**	0,2**	0,12**
X8	0,39**	0,22**	0,18**	0,16**	0,17**
X9	0,71	0,93	0,77	0,62	0,42**
X10	0,39**	0,22**	0,18**	0,16**	0,18**
X11	0,26**	0,4**	0,29**	0,32**	0,09

X12	0,31**	0,24**	0,21**	0,11**	0,09
X13	0,99	1,22	1	0,79	0,07
X14	0,28**	0,44**	0,36**	0,28**	0,12**
X15	0,34**	0,52	0,43**	0,35**	0,03
X16	0,27**	0,36**	0,32**	0,26**	0,07
X17	0,21**	0,24**	0,24**	0,14**	0,3**
X18	0,44**	0,52	0,56	0,48**	0,12**
X19	0,26**	0,17**	0,19**	0,2**	0,09
X20	0,35**	0,4**	0,35**	0,28**	0,2**
X21	0,18**	0,28**	0,23**	0,2**	0,54
X22	0,29**	0,32**	0,23**	0,15**	0,08
X23	0,49**	0,66	0,55	0,52	0,1**
X24	0,3**	0,41**	0,43**	0,32**	0,09
X25	0,16**	0,22**	0,26**	0,25**	0,09
X26	0,23**	0,29**	0,23**	0,26**	0,09
X27	0,2**	0,26**	0,25**	0,16**	0,07
X28	0,35**	0,48**	0,5**	0,36**	0,07
X29	0,4**	0,42**	0,32**	0,31**	0,47**
X30	0,36**	0,46**	0,42**	0,32**	0,14**
X31	0,42**	0,49**	0,43**	0,34**	0,07
X32	0,58	0,61	0,48**	0,39**	0,19**
X33	0,16**	0,28**	0,22**	0,2**	0,16**
X34	0,92	1,02	0,98	0,9	0,08
X35	0,29**	0,46**	0,33**	0,31**	0,1**
X36	0,35**	0,43**	0,39**	0,34**	0,24**
X37	0,33**	0,56	0,43**	0,36**	0,13**
X38	0,27**	0,37**	0,34**	0,38**	0,13**
X39	0,17**	0,28**	0,28**	0,24**	0,05
X40	0,08	0,19**	0,12**	0,13**	0,09
X41	0,13**	0,14**	0,09	0,09	0,1**
X42	0,28**	0,27**	0,24**	0,18**	0,45**
X43	0,21**	0,25**	0,23**	0,18**	0,29**
X44	0,23**	0,35**	0,24**	0,19**	0,14**
X45	0,24**	0,23**	0,24**	0,21**	0,09
X46	0,29**	0,25**	0,21**	0,17**	0,12**
X47	0,3**	0,28**	0,26**	0,14**	0,12**
X48	1,26	0,86	0,89	0,78	0,1**
X49	0,29**	0,25**	0,25**	0,18**	0,09
X50	0,19**	0,2**	0,19**	0,09	0,17**
X51	0,28**	0,26**	0,36**	0,18**	0,16**
X52	0,21**	0,31**	0,22**	0,19**	0,16**
X53	0,31**	0,28**	0,24**	0,16**	0,1**
X54	0,19**	0,24**	0,2**	0,17**	0,18**
X55	0,23**	0,27**	0,23**	0,15**	0,04
X56	0,17**	0,26**	0,21**	0,21**	0,05
X57	0,27**	0,22**	0,23**	0,19**	0,1**

X58	0,14**	0,31**	0,18**	0,2**	0,43**
X59	0,2**	0,26**	0,15**	0,15**	0,06
X60	0,23**	0,23**	0,21**	0,2**	0,17**
X61	0,26**	0,37**	0,27**	0,19**	0,11**
X62	0,26**	0,23**	0,18**	0,13**	0,2**
X63	0,11**	0,12**	0,11**	0,11**	0,17**
X64	0,28**	0,43**	0,34**	0,32**	0,09
IV değeri 0,1 - 0,5 arasındaki değişkenler analize dahil edilmiştir.					

Tablo 5.4'ten hareketle,

5 yıl sonra iflas eden firmaların tahmini için 56 değişken,

4 yıl sonra iflas eden firmaların tahmini için 53 değişken,

3 yıl sonra iflas eden firmaların tahmini için 55 değişken,

2 yıl sonra iflas eden firmaların tahmini için 56 değişken,

1 yıl sonra iflas eden firmaların tahmini için 38 değişken ile sınıflama algoritmaları eğitilecektir.

5.2.2. Veri ön işleme, eğitim ve test seti ayrımı

Çalışmada uygulanacak makine öğrenmesi algoritmaları denetimli makine öğrenmesi altında yer almaktadır. Bu nedenle, denetimli öğrenme makineleri süreci gereği veri seti eğitim seti ve test seti olarak ayrılmalıdır. Eğitim seti ve test seti oranları çalışmalara göre farklılık gösterebilir, %66.6- %33.3, %70- %30, %80- %20 gibi farklı oranlarda ayrılabilmektedir. Bu çalışmada, veri seti %70'i eğitim seti, %30'u test seti olacak şekilde ayrılmasına karar verilmiştir.

Veri seti eğitim ve test seti olarak ayrılmadan önce, tahmin edilecek yıl bazında mevcut olan iflas eden ve iflas etmeyen firma sayıları analiz edilmiştir.

Tablo 5. 5. Hedef Değişkenin Sınıf Hacimleri

Tahmin Edilecek Yıl	İlgili Yılda İflas Eden Firma Sayısı	İlgili Yılda İflas Etmeyen Firma Sayısı	İlgili Yılda İflas Eden Firmaların Veri Seti İçerisindeki Oranı
1.yıl	30	3.171	1%
2.yıl	117	6.251	2%
3.yıl	107	4.823	2%
4.yıl	120	4.695	2%
5.yıl	191	3.038	6%

Tablo 5.5'te verildiği üzere, sınıflandırma algoritmasının kurgulanmasına yardımcı olacak eğitim seti içerisindeki iflas eden firma oranı 5 yılın sonunda iflas eden firmalar tahmin edilmek istendiğinde %1, 4, 3 ve 2 yılın sonunda iflas eden firmalar tahmin edilmek istediğinde %2 1 yılın sonunda iflas eden firmalar tahmin edilmek istendiğinde %6 olarak belirlenmiştir.

Veri setindeki bu dengesizlik, veri seti eğitim ve test seti olarak ayrıldıktan sonra da devam ederek sınıflandırma algoritmasının sürekli olarak iflas etmeyen firmaları tahmin etmesine neden olacağı öngörülebilmektedir. Algoritmanın iflas etmeyen firmaları yüksek oranda tahmin etmesi, doğruluk oranını istemsiz olarak yukarı çekecek, algoritmanın doğruluk gücünde yanlı sonuçlara neden olabilecektir.

Dengesiz veri setleri, sınıflandırma yapılacak hedef değişkeni içerisinde azınlık durumda (az sayıda gözlem bulunan) sınıf ya da sınıfların bulunmasından kaynaklanmaktadır. Çalışma kapsamında bu durumun yaratabileceği olumsuz durumları önlemek adına veri setinde örnekleme yapma yoluna gidilmiştir. Dengesiz veri setinin sonuçları saptırmasını önlemek adına veri seti üzerinde Sentetik Azınlık Aşırı Örnekleme Yöntemi (Synthetic Minority Oversampling Technique (SMOTE)) kullanılmıştır.

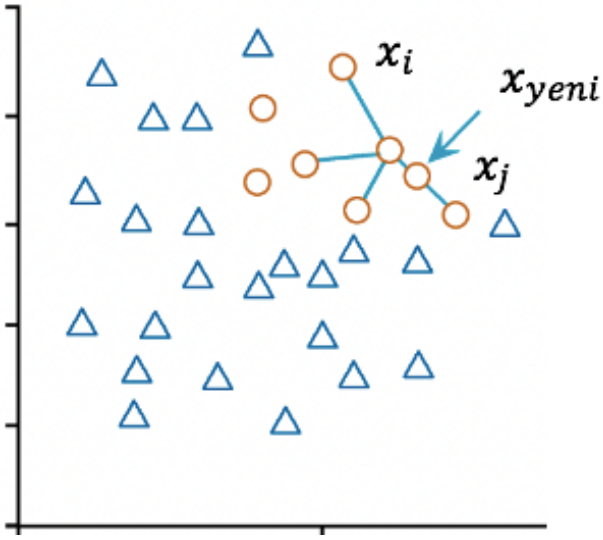
SMOTE örnekleme yöntemi, gerçek azınlık örnekleri arasındaki özellik alanı benzerliklerini temel alan sentetik veri noktaları oluşturur. Bu örnekleme yöntemi ile k tane en yakın komşuları dikkate alınarak her örnek için Euclidean uzaklık ölçütüne dayalı uzaklık hesaplanır (Le vd., 2018).

Azınlık sınıfı, her azınlık sınıfı örneğini alarak ve k azınlık sınıfına en yakın komşuların herhangi birini/tamamını birleştirerek çizgi segmentleri boyunca sentetik örnekler sunarak aşırı örneklenir. Gerekli aşırı örnekleme miktarına bağlı olarak, k en yakın komşulardan komşular rastgele seçilir (Chawla vd., 2002).

Şekil 5.2'de verildiği üzere örnekleme adımları:

- Azınlık sınıfına ait her gözlemin k yakın komşusu belirlenir, azınlık sınıfına ait gözlem ile k yakın komşusu (kNN) olan gözlem arasındaki fark alınır.
- (0,1) arasında rastgele bir sayı (α) seçilir, ve bulunan fark ile bu sayı çarpılır (Haklı, 2018).

$$x_{yeni} = x_i + (x_j - x_i).a \qquad (5.\ 3)$$



Şekil 5. 2. SMOTE Örneklemesi (Haklı, 2018)

Çalışmada veri setindeki dengesizliği gidermek amacıyla sırasıyla aşağıdaki adımlara başvurulmuştur.

Azınlık sınıfı olarak nitelendirilen iflas eden firma sınıfındaki veri adedi sabit tutularak, iflas etmeyen firmalar üzerinde rastgele örnekleme yapılmıştır. Örnekleme sonucunda iflas eden firmaların iflas etmeyen firmalara oranı 1/10 olacak şekilde örnekleme gerçekleştirilmiştir.

1/10 olarak belirlenen sınıf oranlarının ardından Python 3.7 üzerinde SMOTE örnekleme yöntemi çalıştırılmıştır. En yakın 5 komşu göz önünde

bulundurularak azınlık sınıfı olan iflas eden firmalar sınıfı yeniden örneklenmiştir. Örnekleme sonucunda iflas eden firmaların iflas etmeyen firmalara oranı 1/4 olacak şekilde örnekleme gerçekleştirilmiştir.

Örnekleme ile dengesiz veri setinin neden olabileceği sorunların önüne geçilmesi hedeflenmiştir. Çalışma sonucunda veri seti Tablo 5.6'da verilen hali almıştır. İlgili veri seti %70 eğitim seti, %30 test seti olacak şekilde parçalanmış ve algoritmalar eğitilmiştir.

Tablo 5. 6. Örnekleme Sonucu Sınıf Verileri

Tahmin Edilecek Yıl	**Örnekleme Öncesi İlgili Yılda İflas Eden Firma Sayısı**	**Örnekleme Öncesi İlgili Yılda İflas Etmeyen Firma Sayısı**	**SMOTE Örnekleme Sonrası İlgili Yılda İflas Eden Firma Sayısı**	**SMOTE Örnekleme Sonrası İlgili Yılda İflas Etmeyen Firma Sayısı**
1.yıl	30	3.171	75	300
2.yıl	117	6.251	292	1.170
3.yıl	107	4.823	267	1.070
4.yıl	120	4.695	300	1.200
5.yıl	191	3.038	477	1.910

5.2.3. Performans ölçütleri

Sınıflandırma problemlerinde, algoritmanın tahmin gücünü tespit etmek ve olası diğer algoritmalarla karşılaştırabilmek için çeşitli ölçütler mevcuttur. Denetimli öğrenme metodolojisi altında yer alan bu problem türünde, eğitim seti ve test seti bilindiğinden, algoritmaların performans ölçümlerinin birincil kaynağı bir karmaşıklık matrisidir. Tablo 5.7 iki sınıflı bir sınıflandırma problemi için bir karmaşıklık matrisini gösterir.

Tablo 5. 7. Karmaşıklık Matrisi

		Tahmini Sınıf	
		İflas Etti	İflas Etmedi
Gerçek Sınıf	İflas Etti	TP	FP
	İflas Etmedi	FN	TN

Üst soldan sağa çapraz boyunca sayılar doğru kararları temsil eder ve bu diyagonal dışındaki sayılar hataları temsil eder.

TP (True Positive): Gerçekte iflas eden ve algoritma test sonucunda da iflas etmiş olarak sınıflanan firmalardır.
TN (True Negative): Gerçekte iflas etmeyen ve algoritma test sonucunda da iflas etmemiş olarak sınıflanan firmalardır.
FP (False Positive): Gerçekte iflas etmiş ancak algoritma test sonucunda iflas etmemiş olarak sınıflanan firmalardır. Tip I hatası olarak da adlandırılır.
FN (False Negative): Gerçekte iflas etmeyen ancak algoritma test sonucunda iflas etmiş olarak sınıflanan firmalardır. Tip II hatası olarak da adlandırılır
Verilen tanımlardan hareketle algoritmaların performansını etkileyen ölçütler aşağıda verilmiştir.

Keskinlik, tüm pozitif tahminler içinde gerçek pozitif durumların tahmin edilme oranını göstermektedir.

$$Keskinlik = \frac{TP}{TP + FP}$$

Duyarlılık, tüm veri seti içinde pozitif durumların tahmin edilme oranını ifade eder.

$$Duyarlılık = \frac{TP}{TP + FN}$$

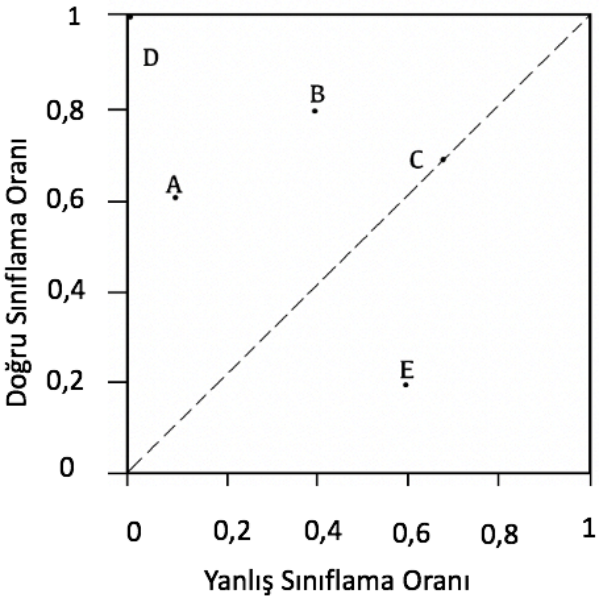
F_1 Puanı ise Keskinlik ve Duyarlılık değerlerinin harmonik ortalamasıdır (Bulut ve Osmani, 2017).

$$F\ Puanı = 2\frac{Keskinlik\ x\ Duyarlılık}{Keskinlik + Duyarlılık}$$

Doğruluk, doğru tahmin edilen sınıfların tüm veriye oranıdır.

$$Doğruluk = \frac{TN + TP}{TP + FP + TN + FN}$$

Doğruluk ölçütü, tüm veri seti içinde doğru tahminlerin oranını hesaplamaktadır. Bu ölçüt, uzun yıllar boyunca algoritmaların performanslarını karşılaştırmada ana gösterge olmuştur. Ancak sadece doğru sınıflanan verinin oranını bilmek, algoritmanın yanlış sınıfladığı verilerin oranını göz ardı etmek algoritmanın maliyetini göz ardı etmek anlamına gelir. Bu nedenle, zaman içerisinde Alıcı Çalışma Özellikleri Eğrisi (ROC) adı verilen bir performans ölçüm tekniği geliştirilmiştir. Tutarlılık ve ayrıştırıcılık bakımından ROC eğrisinin doğruluk oranından daha iyi bir ölçüt olduğunu kanıtlanmıştır (Ling vd., 2003). ROC eğrisi sınıflama algoritmalarının davranışını analiz etme aynı zamanda da görselleştirme amacıyla kullanılmaktadır.



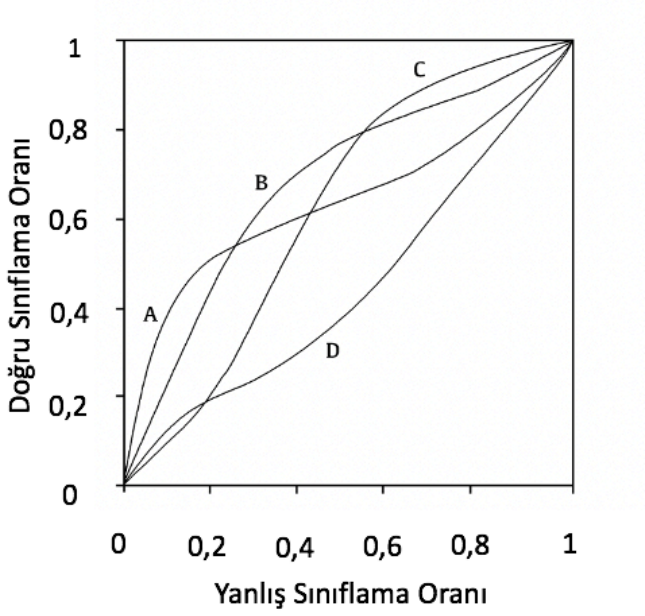
Şekil 5. 3. ROC Eğrisi (Fawcett, 2006)

Şekil 5.3'te gösterildiği üzere, ROC grafikleri, Y ekseninde doğru sınıflama (TP) oranının çizildiği ve X ekseninde yanlış sınıflama (FP) oranın çizildiği iki boyutlu grafiklerdir. ROC eğrisi faydalar (gerçek pozitifler) ve maliyetler (yanlış pozitifler) arasındaki göreceli dengeyi betimletir. Gerçek pozitifler oranı hesaplanan Duyarlılık oranına eşdeğerdir. Maliyetler olacak düşünebilecek yanlış pozitifler oranı ise özgüllük değeri ile hesaplanabilir.

$$\text{Özgüllük} = \frac{TN}{TN + FP}$$

Sınıflandırma işlemi sırasında her bir veri için bir fayda maliyet çifti üretilir ve bu çiftler şekilde verilen grafiğe entegre edilir. (0; 1) noktası ROC eğrisinde en iyi sınıflandırmayı temsil eder, D noktası sınıflandırma performansının çok iyi olduğunu gösterir. C noktasının üzerinde bulunduğu doğru ise, bu modelin sınıf ayırma kapasitesi olmadığı anlamına gelir (Olson ve Delen, 2008).

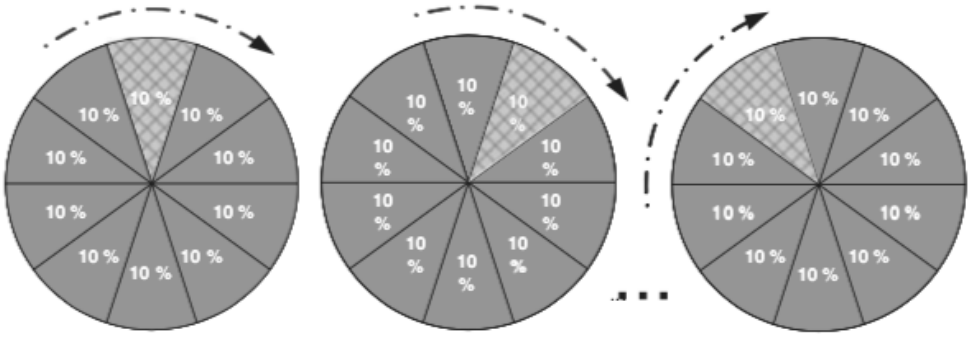
ROC eğrisi, sınıflandırıcı performansını hem fayda hem de maliyeti aynı anda göz önünde bulundurarak görselleştirmek amacıyla kullanılan bir grafiktir. ROC eğrisinin sınıflandırma performansı skaler bir büyüklüğe dönüştürülmek istendiğinde, grafik üzerinde ROC eğrisinin altında kalan alan (AUC) hesaplanır.



Şekil 5. 4. Eğri Altında Kalan Alan (Huang & Ling, 2005)

Şekil 5.4'te gösterildiği üzere, AUC değeri matematiksel hesaplamalar gereği 0 ile 1 arasında değer alan bir ölçüttür. AUC değerinin 1'e yakın olması, sınıflandırma algoritmasının performansının yüksek olduğunu gösterir. Paralel olarak, kötü bir sınıflandırma algoritması, 0'a yakın bir AUC değerine sahiptir. AUC 0,5 ise, algoritmanın sınıf ayırma kapasitesi olmadığı, ikili sınıflama için rastgele tahmin yapmak ile aynı tahmin kabiliyetine sahip olduğu anlamına gelir (Sarang, 2018).

Denetimli öğrenme algoritmalarında, veri seti eğitim ve test seti olarak ayrılır. Algoritmalar, verilen eğitim seti ile birlikte her girdiye karşılık gelen çıktıyı öğrenerek, yeni gelecek verinin nasıl bir davranış sergileyeceğini önceden tahmin etmeye çalışır. Bu tip algoritmalarda, yeni gelecek verinin davranışını doğru tahmin edilip edilemediği test seti üzerinden analiz edilir. Bu nedenle, veri seti hacmi küçük olduğunda, verilerin dağılımı düzgün olmadığında algoritma sonuçları taraflı ya da aşırı öğrenme sonucu oluşsa bile test veri seti üzerinden bu sorunlar saptanamayabilmektedir.



Şekil 5. 5. Çapraz Doğrulama Gösterimi (Olson ve Delen, 2008)

Algoritmanın taraflı tahmin, aşırı öğrenme sorunlarını minimuma indirmek adına, test veri setine Çapraz Doğrulama (Cross Validation (CV)) yöntemi uygulanır. Şekil 5.5'te verildiği gibi, ilgili veri seti eşit boyutlarda k adet rastgele parçalara, katlamalara (fold) bölünür. Bu parçaların k-1 adedi algoritmayı eğitmek amacıyla eğitim seti olarak, 1 adedi de test seti olarak kullanılır. Her tekrarda elde edilen doğruluk, algoritmanın final doğruluğunu elde etmek için ortalama olarak alınır.

5.3. Analiz ve Bulgular

Çalışmanın bu bölümünde, sınıflandırma algoritmalarının performanslarına ilişkin sonuçlara yer verilmiştir. Ham veri seti üzerinde yapılan WOE ve IV değer hesaplamalarının ardından algoritmayı olumlu yönde besleyecek potansiyel değişkenler, 5 farklı tahmin dönemi için ayrı ayrı belirlenmiştir. Ardından, veri setindeki dengesizlik durumunu gidermek amacıyla yüksek veri sayısına sahip olan sınıf üzerinde rastgele örnekleme yapılarak ilgili sınıfın verisi azaltılmıştır. Azınlık durumunda bulunan sınıf verileri için SMOTE örnekleme yöntemi yardımıyla yeniden örnekleme yapılmış, azınlık sınıfı, diğer sınıfa oranı ¼ olacak şekilde örnekleme yöntemiyle çoğaltılmıştır.

Veri seti tüm sınıflandırma algoritmaları için %70 eğitim seti ve %30 test seti olacak şekilde parçalanmıştır. Algoritmalar eğitim seti ile verilerin davranışını öğrendikten sonra test olarak ayrılan veri seti üzerinden doğru sınıflama metrikleri hesaplanmıştır. Olası yanlı tahmin problemini ortadan kaldırmak adına test veri setine Çapraz Doğrulama yöntemi uygulanmıştır. k=10 seçilerek, 10-katmanlı Çapraz Doğrulama ile türetilen yeni veri seti üzerinden tüm algoritmaların performansları ölçümlenmiştir. Tüm sınıflandırma çalışmaları Python 3.7 üzerinde yapılmıştır.

5.3.1. Naive bayes algoritması ile sınıflandırma

Bu bölümde, ön işleme süreci tamamlanmış, test ve eğitim seti olarak ayrılmış veri setine Naive Bayes algoritmasıyla yapılan sınıflandırma sonuçlarına yer verilmiştir.

Naive Bayes algoritmasının performansı, test seti olarak ayrılan verilerin 10 katlamalı Çapraz Doğrulama yöntemi ile farklılaştırılması yardımıyla yeni oluşturulan veri seti üzerinde sınanmıştır. Performans ölçütü olarak Keskinlik, Duyarlılık, F puanı ve Doğruluk değerleri hesaplanmıştır.

Tablo 5. 8. Naive Bayes ile Sınıflama Sonuçları

		Keskinlik	Duyarlılık	F_1 Puanı	Doğruluk
5 yıl Sonraki İflas Tahmini	**İflas Etmedi**	0,96	0,28	0,43	
	İflas Etti	0,22	0,95	0,36	
	Ortalama	0,59	0,61	0,4	0,4
4 yıl Sonraki İflas Tahmini	**İflas Etmedi**	0,9	0,19	0,31	
	İflas Etti	0,21	0,92	0,33	
	Ortalama	0,56	0,55	0,33	0,33
3 yıl Sonraki İflas Tahmini	**İflas Etmedi**	0,92	0,29	0,44	
	İflas Etti	0,25	0,9	0,39	
	Ortalama	0,58	0,59	0,41	0,413
2 yıl Sonraki İflas Tahmini	**İflas Etmedi**	0,83	0,16	0,27	
	İflas Etti	0,2	0,87	0,33	
	Ortalama	0,52	0,51	0,3	0,299
1 yıl Sonraki İflas Tahmini	**İflas Etmedi**	0,85	0,16	0,27	
	İflas Etti	0,2	0,88	0,32	
	Ortalama	0,52	0,52	0,29	0,295
5 Dönem Ortalama Tahmin	**İflas Etmedi**	0,89	0,22	0,34	
	İflas Etti	0,22	0,9	0,35	
	Ortalama	0,55	0,56	0,35	0,346

Tablo 5.8'de Naive Bayes algoritması ile 5 farklı döneme ait iflas tahmin sonuçları verilmiştir. Algoritmaların genel doğruluk oranları dikkate alındığında, 0,50 değerinin altında kaldığından, algoritma performansının düşük olduğu söylenebilir.

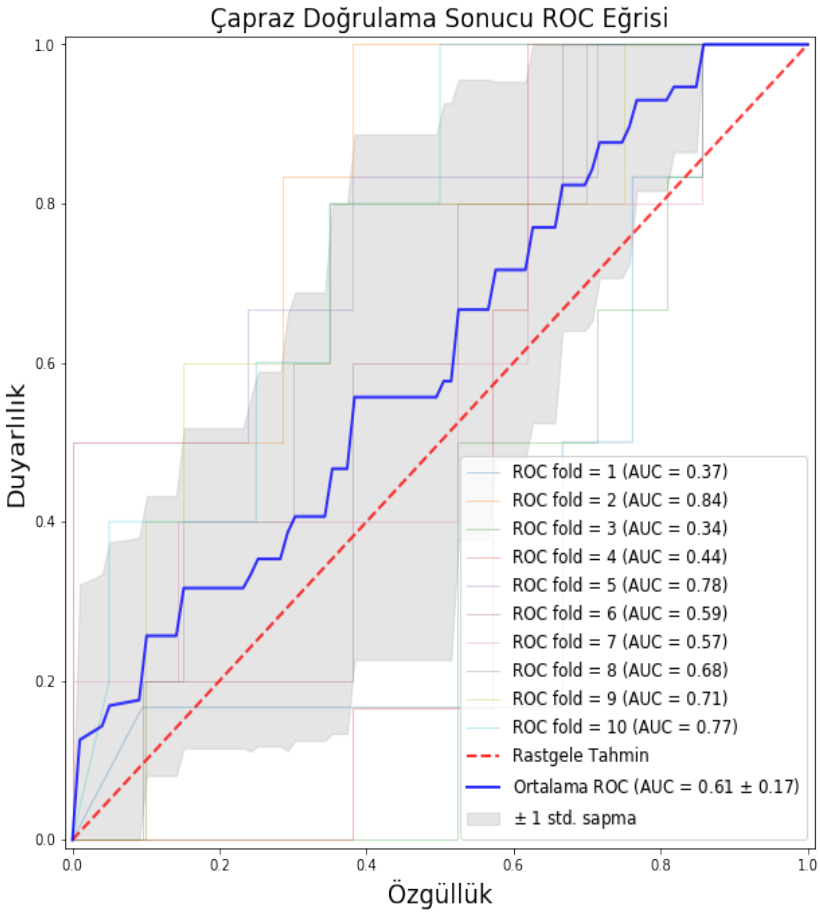
Naive Bayes algoritması ile sınıflandırmada en düşük doğruluk değerine sahip dönem 0,295 doğruluk oranı ile 1 yıl sonraki iflas tahmini olmuştur. İlgili dönemde gerçekte iflas eden müşteriler arasından, sınıflama sonucu iflas edeceği tahmin edilen firmaların %20'si, iflas etmeyen firmalar arasından %85'i doğru tahmin edilmiştir.

5 yıl sonraki iflas verileri göz önünde tutulduğunda, iflas etmedi olarak sınıflandırılan firmalar içinde gerçekten iflas etmemiş firmaların oranı %96; iflas etti olarak sınıflandırılan firmalar içinde gerçekten iflas etmiş firmaların oranı %22 olarak belirlenmiştir. Tüm sınıflandırma seti düşünüldüğünde iflas etmeyen firmaların doğru sınıflama oranı %28, iflas eden firmaların doğru tahmin oranı %95 olarak hesaplanmıştır. Algoritmanın tüm sınıflar bazında genel doğruluk oranı 0,40'tır.

4 yıl sonraki iflas tahmini probleminde, gerçekte iflas eden firmalar arasından, sınıflama sonucu iflas edeceği belirlenen firmaların %21'i, iflas etmeyen firmalar arasından %90'ı doğru tahmin edilmiştir. Tüm sınıflar göz önünde tutulduğunda algoritmanın doğru sınıflama oranı 0,325 kabul edilir.

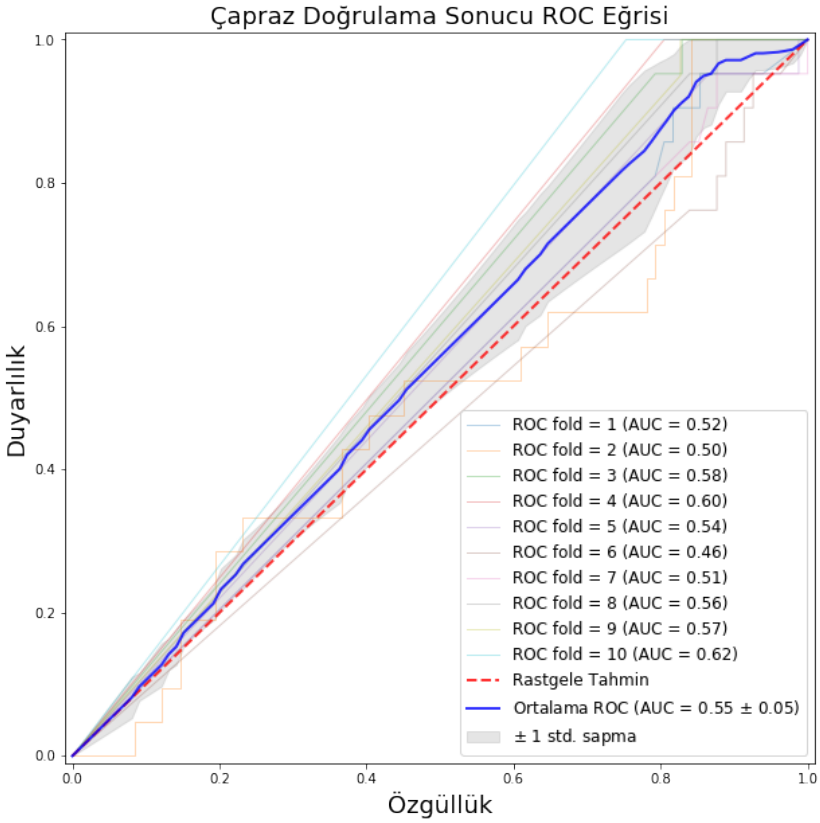
3 yıl sonraki iflasın tahmin edilmesi probleminde, tüm test setinde yer alan firmalardan, iflas edeceği tahmin edilen firmaların %90'ı gerçekten iflas etmiştir. Sınıflandırma işlemi sonucu, gerçekte iflas eden tüm firmalar arasından, %25'i doğru olarak tespit edilmiştir. Algoritmanın genel sınıflandırma doğruluğu ise 0,41'dir.

2 yıl sonraki iflas tahmini probleminde, gerçekte iflas eden firmalar arasından, sınıflama sonucu iflas edeceği belirlenen firmaların %87'si, tüm veride gerçekte iflas eden firmalar arasından %20'si doğru tahmin edilmiştir. Algoritmanın geneli ele alındığında doğru sınıflandırma oranı 0,299 kabul edilir.



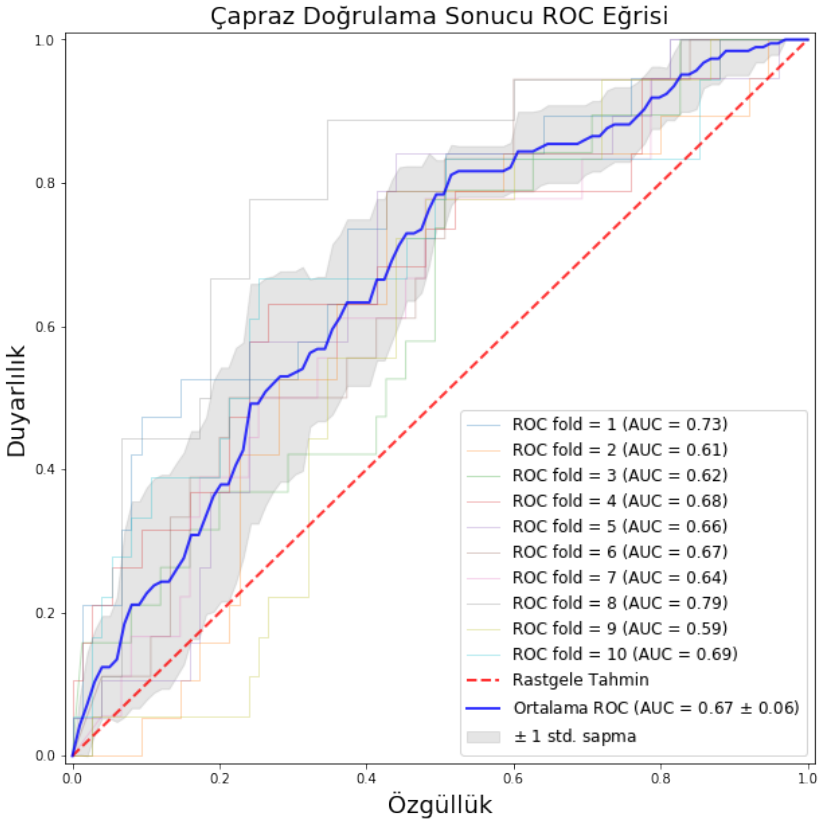
Şekil 5. 6. NB ile 5 yıl Sonraki İflas Tahmini

Şekil 5.6'da 5 yıl sonraki iflas tahmini için hesaplanan ROC eğrisi verilmiştir. Doğru sınıflandırma ve yanlış sınıflandırmanın birer eksende yer aldığı ROC eğrisinde 10 katlamalı çapraz doğrulama işleminin her katı için eğri altındaki alan hesaplanmıştır. 1. katlama durumunda AUC değeri 0,37'dir. En düşük AUC değeri katlama = 3 olduğu durum ile 0,34 değeridir. En yüksek AUC değerine ise 0,84 ile katlama=2 durumunda ulaşılmıştır. Veri setinden kaynaklanabilecek sapma ve yanlı tahmin problemini engellemek için belirlenen 10 katlamanın ortalaması alınarak analize dahil edilmiştir. Ortalama 0,61 AUC değeri ile algoritmanın sınıflama probleminde rastgele tahminden daha yüksek bir ayrıştırıcılığı olduğu söylenebilir.



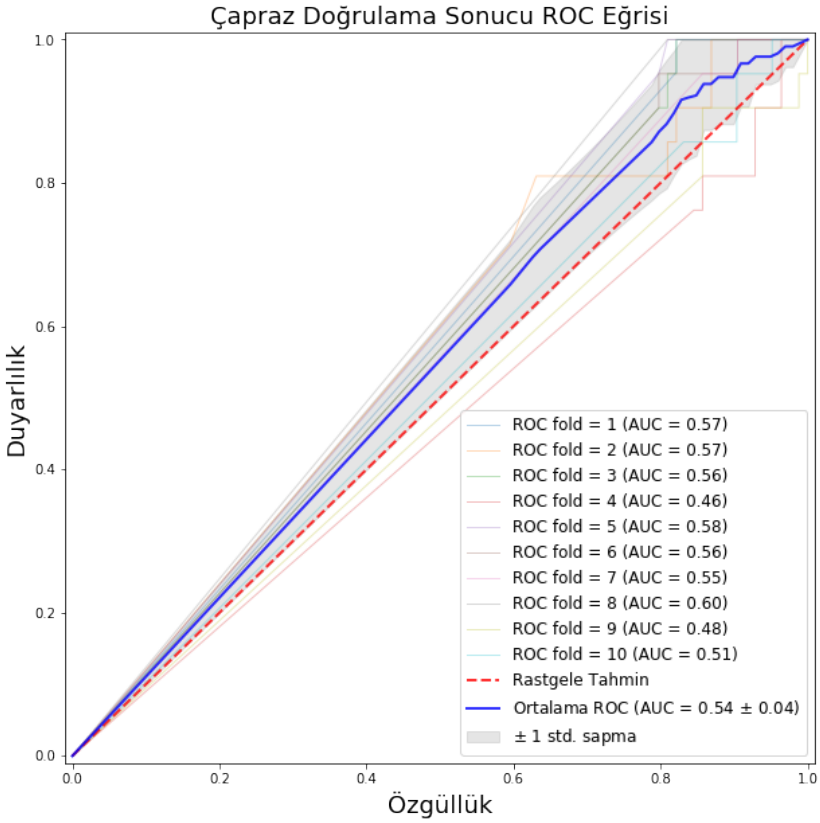
Şekil 5. 7. NB ile 4 yıl Sonraki İflas Tahmini

Şekil 5.7'de 4 yıl sonraki iflas tahmini için hesaplanan ROC eğrisi verilmiştir. En düşük AUC değeri katlama=6 olduğu durum ile 0,46 değeridir. En yüksek AUC değerine ise 0,62 ile katlama=10 durumunda ulaşılmıştır. Veri setinden kaynaklanabilecek sapma ve yanlı tahmin problemini engellemek için belirlenen 10 katlamanın ortalama AUC değeri 0,55 AUC hesaplanmıştır, algoritmanın sınıflama probleminde rastgele tahminden küçük bir fark ile daha yüksek bir ayrıştırıcılığı olduğu söylenebilir.



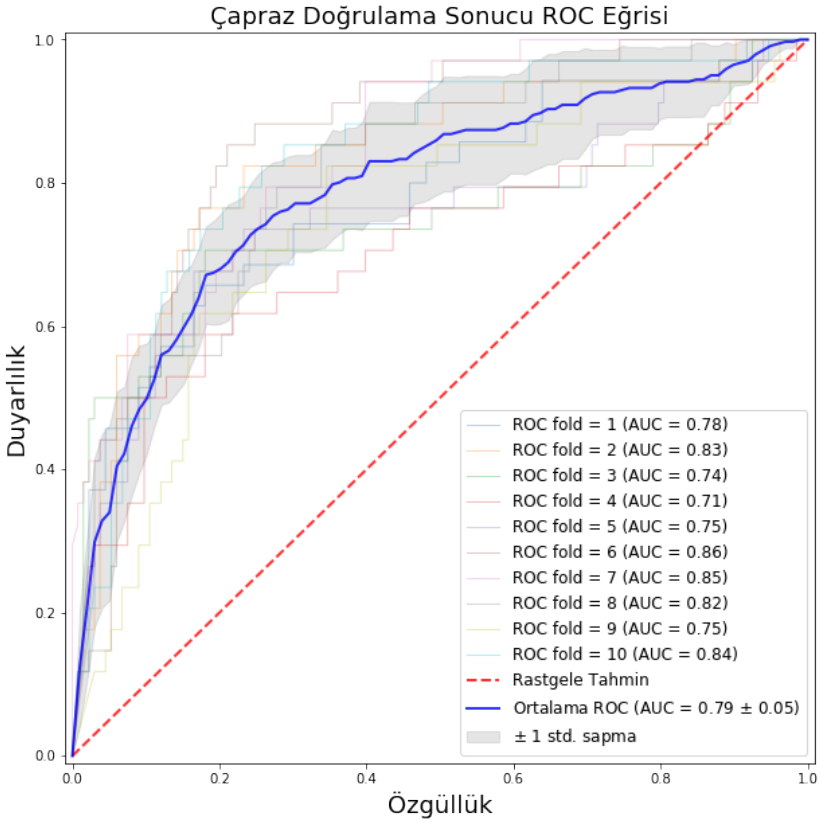
Şekil 5. 8. NB ile 3 yıl Sonraki İflas Tahmini

Şekil 5.8'de 3 yıl sonraki iflas tahminin ROC eğrisi verilmiştir. En düşük AUC değeri katlama=9 olduğu durum ile 0,59 değeridir. En yüksek AUC değerine ise 0,79 ile katlama=8 durumunda ulaşılmıştır. Ortalama 0,67 AUC değeri ile algoritmanın sınıflama probleminde rastgele tahminden yüksek bir ayrıştırıcılığı olduğu söylenebilir.



Şekil 5. 9. NB ile 2 yıl Sonraki İflas Tahmini

2 yıl sonraki iflas tahmini için ROC eğrisi Şekil 5.9'da verilmiştir. En düşük AUC değeri katlama=4 olduğu durum ile 0,46 değeridir. En yüksek AUC değerine ise 0,60 ile katlama=8 durumunda ulaşılmıştır. Ortalama 0,54 AUC değeri ile algoritmanın sınıflama probleminde rastgele tahmine yakın bir ayrıştırıcılığı olduğu söylenebilir.



Şekil 5. 10. NB ile 1 yıl Sonraki İflas Tahmini

Şekil 5.10'da 2 yıl sonraki iflas tahmini için hesaplanan ROC eğrisi verilmiştir. En düşük AUC değeri katlama=4 olduğu durum ile 0,71 değeridir. En yüksek AUC değerine ise 0,86 ile katlama=6 durumunda ulaşılmıştır. Ortalama 0,79 AUC değeri ile algoritmanın sınıflama probleminde rastgele tahmine kıyasla yüksek bir ayrıştırıcılığı olduğu söylenebilir.

5.3.2. k en yakın komşuluk algoritması ile sınıflandırma

K En Yakın Komşu algoritmasının performansı, test seti olarak ayrılan verilerin 10 katlamalı Çapraz Doğrulama yöntemi ile farklılaştırılması sonucu yeni oluşturulan veri seti üzerinde sınanmıştır. Python üzerinde kurulan döngü sonucunda sınıflama hatasını en aza indiren k değeri 1 olarak bulunmuştur. Böylece en yakın 1 komşu alınarak algoritma eğitilmiştir.

Tablo 5. 9. k En Yakın Komşuluk ile Sınıflama Sonuçları

		Keskinlik	Duyarlılık	F_1 Puanı	Doğruluk
5 yıl Sonraki İflas Tahmini	**İflas Etmedi**	0,9	0,8	0,85	
	İflas Etti	0,39	0,6	0,47	
	Ortalama	0,64	0,7	0,66	0,76
4 yıl Sonraki İflas Tahmini	**İflas Etmedi**	0,88	0,85	0,87	
	İflas Etti	0,45	0,52	0,49	
	Ortalama	0,67	0,69	0,68	0,787
3 yıl Sonraki İflas Tahmini	**İflas Etmedi**	0,89	0,89	0,89	
	İflas Etti	0,57	0,57	0,57	
	Ortalama	0,73	0,73	0,73	0,825
2 yıl Sonraki İflas Tahmini	**İflas Etmedi**	0,88	0,9	0,89	
	İflas Etti	0,55	0,51	0,53	
	Ortalama	0,71	0,7	0,71	0,819
1 yıl Sonraki İflas Tahmini	**İflas Etmedi**	0,94	0,91	0,92	
	İflas Etti	0,66	0,75	0,7	
	Ortalama	0,8	0,83	0,81	0,88
5 Dönem Ortalama Tahmin	**İflas Etmedi**	0,9	0,87	0,88	
	İflas Etti	0,52	0,59	0,55	
	Ortalama	0,71	0,73	0,72	0,814

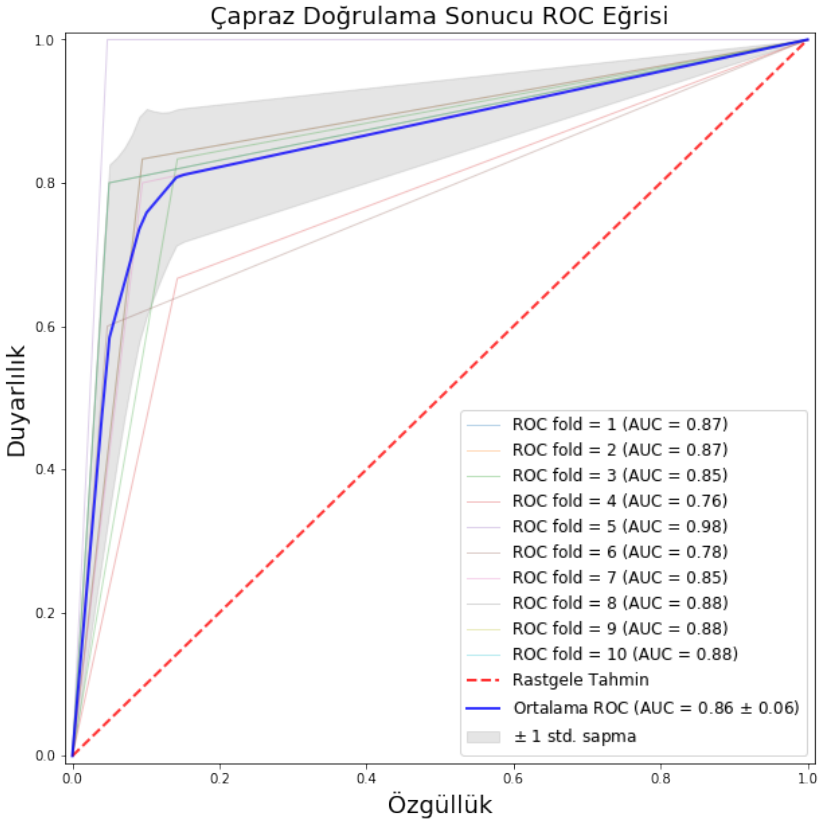
Tablo 5.9' da k En Yakın Komşu algoritması ile 5 farklı döneme ait iflas tahmin sonuçları verilmiştir. Algoritmaların genel doğruluk oranları dikkate alındığında algoritma performansının görece yüksek olduğu söylenebilir. k En Yakın Komşu algoritması ile sınıflandırmada en düşün doğruluk değerine sahip dönem 0,760 doğruluk oranı ile 5 yıl sonraki iflas tahmini olmuştur.

4 yıl sonraki iflas tahmini probleminde, gerçekte iflas eden firmalar arasından, sınıflama sonucu iflas edeceği belirlenen firmaların %45'i, iflas etmeyen firmalar arasından %88'i doğru tahmin edilmiştir. Algoritmanın geneli ele alındığında doğru sınıflandırma oranı 0,787 kabul edilir.

3 yıl sonraki iflas tahmini probleminde tüm test setinde yer alan firmalardan iflas edeceği tahmin edilen firmaların %57'si gerçekten iflas etmiştir. Sınıflandırma işlemi sonucu, gerçekte iflas eden tüm firmalar arasından, yine %57'si doğru olarak tespit edilmiştir. Algoritmanın genel sınıflandırma doğruluğu ise 0,825 oranındadır.

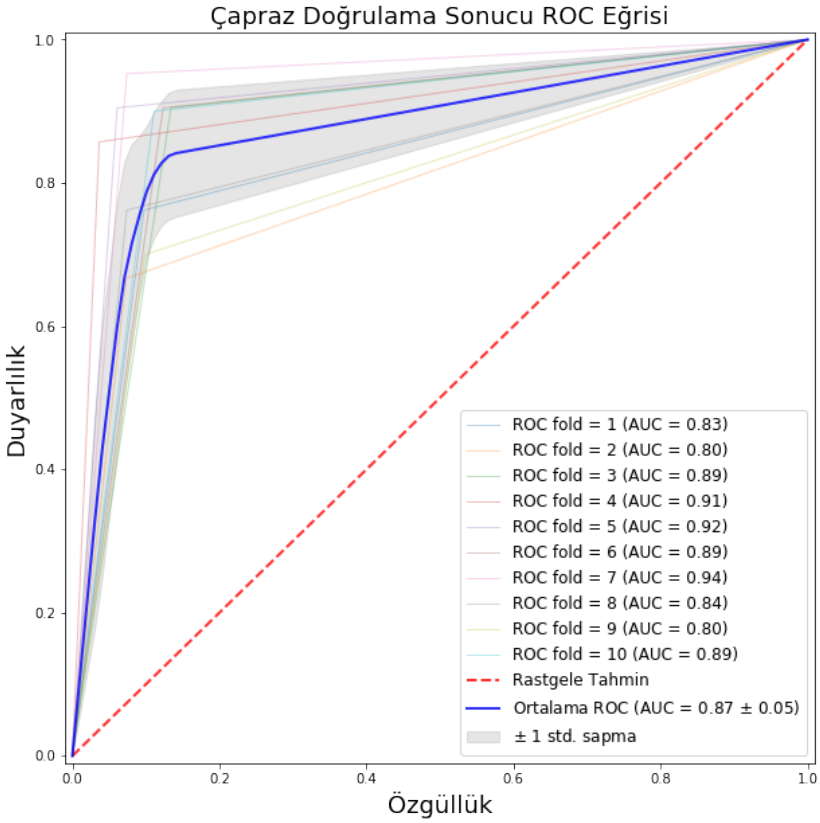
2 yıl sonraki iflas tahmini probleminde, gerçekte iflas eden firmalar arasından, sınıflama sonucu iflas edeceği belirlenen firmaların %51'i, iflas etmeyen firmalar arasından %55'i doğru tahmin edilmiştir. Algoritmanın geneli ele alındığında doğru sınıflandırma oranı 0,819 kabul edilir.

1 yıl sonraki iflas tahmini probleminde, gerçekte iflas eden firmalar arasından, sınıflama sonucu iflas edeceği belirlenen firmaların %75'i, iflas etmeyen firmalar arasından %66'sı doğru tahmin edilmiştir. Algoritmanın geneli ele alındığında doğru sınıflandırma oranı 0,88 kabul edilir.



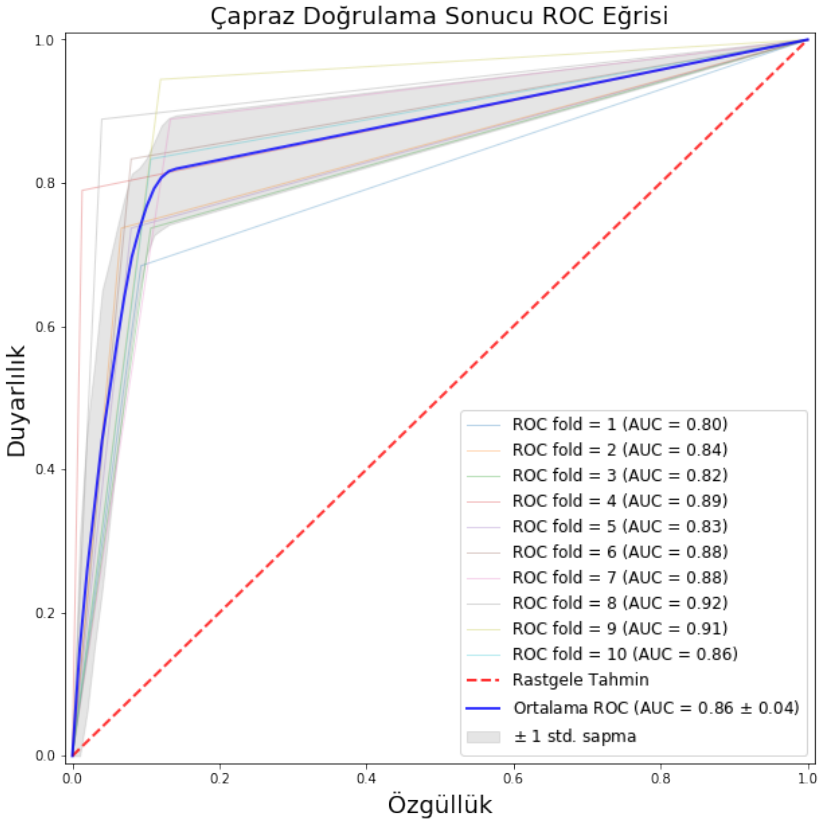
Şekil 5. 11. kNN ile 5 yıl Sonraki İflas Tahmini

Şekil 5.11'de 5 yıl sonraki iflas tahmini için ROC eğrisinde 10 katlamalı çapraz doğrulama işleminin her katlama için eğri altındaki alanı hesaplanmıştır. Katlama=1 durumunda AUC değeri 0,87'dir. En düşük AUC değeri katlama=4 olduğu durum ile 0,76 değeridir. En yüksek AUC değerine ise 0,98 ile katlama=5 durumunda ulaşılmıştır. Veri setinden kaynaklanabilecek sapma ve yanlı tahmin problemini engellemek için belirlenen 10 katlamanın ortalaması alınarak analize dahil edilmiştir. Ortalama 0,86 AUC değeri ile algoritmanın sınıflama probleminde yüksek bir ayrıştırıcılığı olduğu söylenebilir.



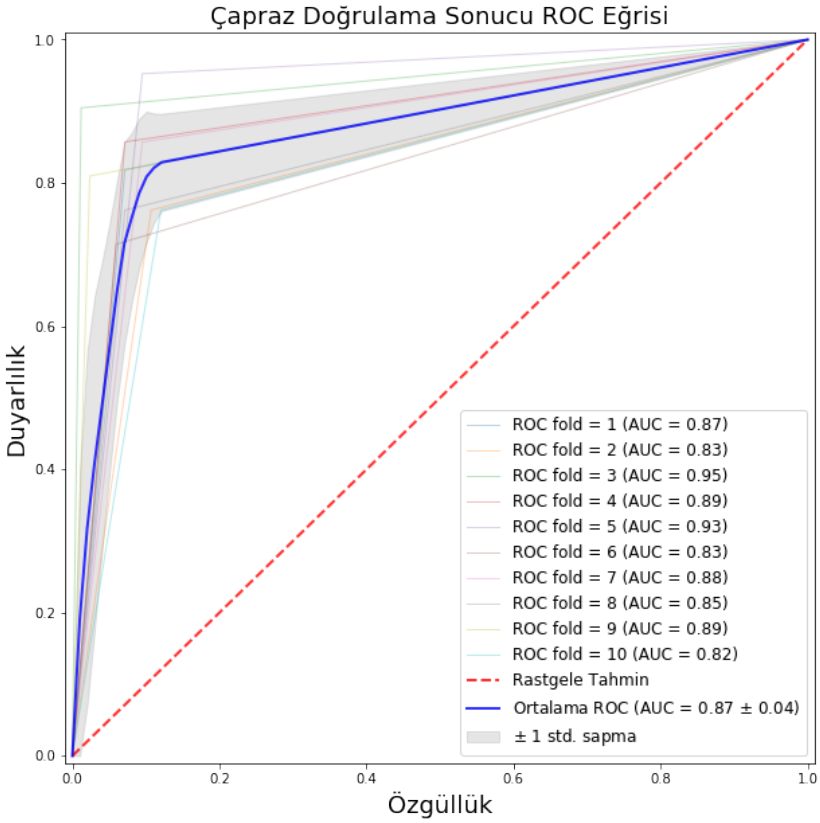
Şekil 5. 12. kNN ile 4 yıl Sonraki İflas Tahmini

4 yıl sonraki iflas tahmini performansı Şekil 5.12'de hesaplanan ROC eğrisi ile verilmiştir. En düşük AUC değeri katlama=2 ve 9 olduğu durum ile 0,80 değeridir. En yüksek AUC değerine ise 0,94 ile katlama=7 durumunda ulaşılmıştır. 10 katlamanın ortalaması 0,87 AUC değeri ile algoritmanın sınıflama probleminde yüksek bir ayrıştırıcılığı olduğu söylenebilir.



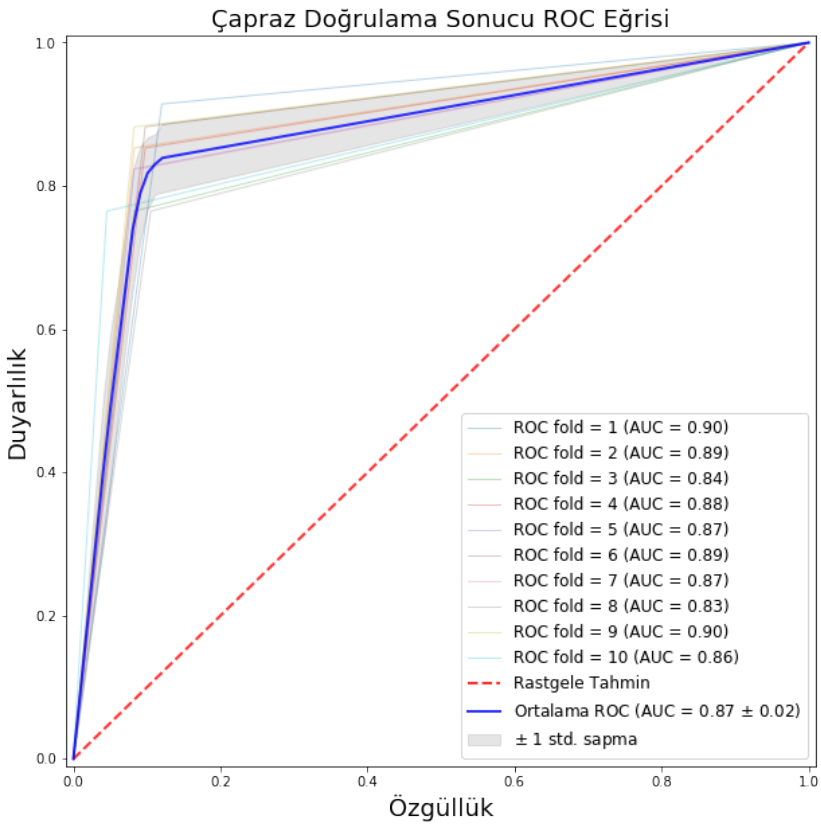
Şekil 5. 13. kNN ile 3 yıl Sonraki İflas Tahmini

Şekil 5.13'te 3 yıl sonraki iflas tahmini için hesaplanan ROC eğrisi verilmiştir. En düşük AUC değeri katlama=1 olduğu durum ile 0,80 değeridir. En yüksek AUC değerine ise 0,92 ile katlama=8 durumunda ulaşılmıştır. Ortalama 0,86 AUC değeri ile algoritmanın sınıflama probleminde yüksek bir ayrıştırıcılığı olduğu söylenebilir.



Şekil 5. 14. kNN ile 2 yıl Sonraki İflas Tahmini

Şekil 5.14'te 2 yıl sonraki iflas tahmini için hesaplanan ROC eğrisi verilmiştir. En düşük AUC değeri katlama =10 olduğu durum ile 0,82 değeridir. En yüksek AUC değerine ise 0,95 ile katlama=3 durumunda ulaşılmıştır. Sapma ve yanlı tahmin problemini engellemek için belirlenen 10 katlamanın ortalaması alınarak, ortalama 0,87 AUC değeri ile algoritmanın sınıflama probleminde yüksek bir ayrıştırıcılığı olduğu söylenebilir.



Şekil 5. 15. kNN ile 1 yıl Sonraki İflas Tahmini

Şekil 5.15'te 1 yıl sonraki iflas tahmini için hesaplanan ROC eğrisi verilmiştir. En düşük AUC değeri katlama=8 olduğu durum ile 0,83 değeridir. En yüksek AUC değerine ise 0,90 ile katlama=1 ve 9 durumunda ulaşılmıştır. Ortalama 0,87 AUC değeri ile algoritmanın sınıflama probleminde yüksek bir ayrıştırıcılığı olduğu söylenebilir.

5.3.3. Destek vektör makinesi algoritması ile sınıflandırma

Destek Vektör Makineleri algoritmasının performansı, test seti olarak ayrılan verilerin 10 katlamalı Çapraz Doğrulama yöntemi ile farklılaştırılması sonucu yeni oluşturulan veri seti üzerinde sınanmıştır. Veri setindeki noktaları çok boyutlu uzaya haritalamak amacıyla Radyal Tabanlı Kernel fonksiyonu kullanılmıştır. Kernel fonksiyonunun optimal parametrelerini bulmak amacıyla bir dizi C ve gamma parametresi döngü oluşturacak şekilde kullanılarak Python üzerinde kodlanmıştır. En yüksek doğruluğu sağlayan C ve gamma parametresi optimum olarak belirlenmiştir.

Performans ölçütü olarak Keskinlik, Duyarlılık, F puanı ve Doğruluk değerleri hesaplanmıştır.

Tablo 5. 10. Destek Vektör Makinesi ile Sınıflama Sonuçları

			Keskinlik	Duyarlılık	F_1 Puanı	Doğruluk
5 yıl Sonraki İflas Tahmini	**İflas Etmedi**	**C = 1,**	0,93	1	0,96	
	İflas Etti	**gamma = 0.0001**	1	0,65	0,79	
	Ortalama		0,64	0,7	0,66	0,884
4 yıl Sonraki İflas Tahmini	**İflas Etmedi**	**C = 1,**	0,88	1	0,94	
	İflas Etti	**gamma = 0.1**	1	0,44	0,61	
	Ortalama		0,94	0,72	0,77	0,864
3 yıl Sonraki İflas Tahmini	**İflas Etmedi**	**C = 1,**	0,87	1	0,93	
	İflas Etti	**gamma = 0.001**	1	0,41	0,59	
	Ortalama		0,93	0,71	0,76	0,88
2 yıl Sonraki İflas Tahmini	**İflas Etmedi**	**C = 1,**	0,86	1	0,92	
	İflas Etti	**gamma = 1**	1	0,31	0,48	
	Ortalama		0,93	0,66	0,7	0,893
1 yıl Sonraki İflas Tahmini	**İflas Etmedi**	**C = 1,**	0,88	1	0,93	
	İflas Etti	**gamma = 0,5**	1	0,39	0,56	
	Ortalama		0,94	0,69	0,75	0,937
5 Dönem Ortalama Tahmin	**İflas Etmedi**		0,88	1	0,94	
	İflas Etti		1	0,44	0,61	
	Ortalama		0,88	0,7	0,73	0,891

Tablo 5.10'da Destek Vektör Makinesi algoritması ile 5 farklı döneme ait iflas tahmin sonuçları verilmiştir. Algoritmaların genel doğruluk oranları dikkate alındığında algoritma performansının yüksek olduğu söylenebilir. Destek

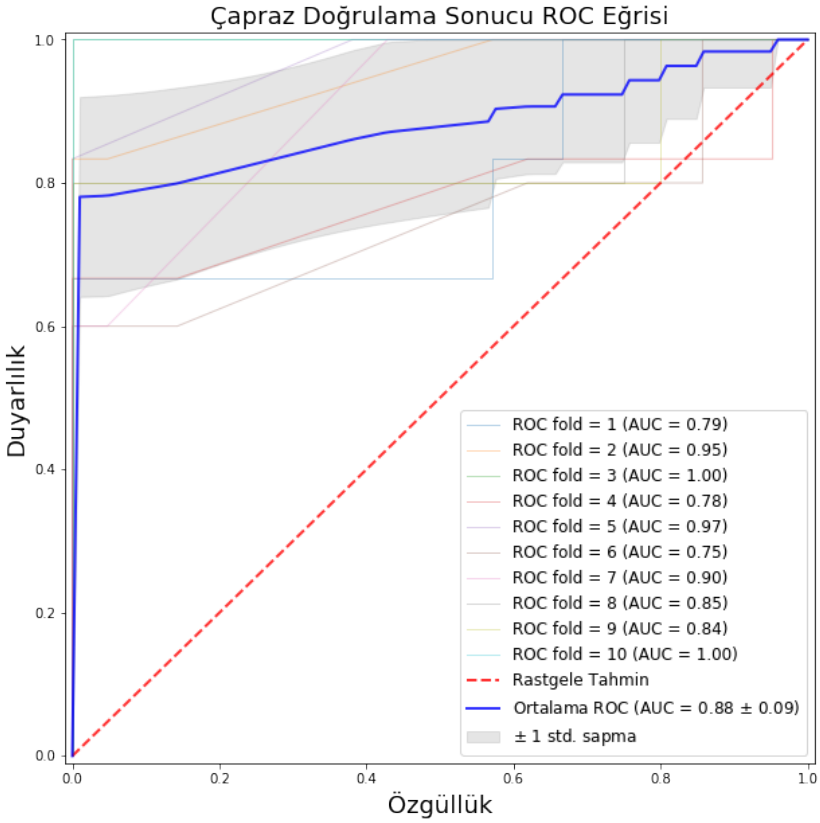
Vektör Makineleri algoritması ile sınıflandırmada en düşük doğruluk değerine sahip dönem 0,864 doğruluk oranı ile 4 yıl sonraki iflas tahmini olmuştur.

5 yıl sonraki iflas tahmini probleminde, gerçekte iflas eden firmalar arasından, sınıflama sonucu iflas edeceği belirlenen firmaların tamamını, iflas etmeyen firmalar arasından %88'i doğru tahmin edilmiştir. Algoritmanın geneli ele alındığında doğru sınıflandırma oranı 0,893 kabul edilir.

3 yıl sonraki iflas tahmini probleminde tüm test setinde yer alan firmalardan iflas edeceği tahmin edilen firmaların %100'ü gerçekten iflas etmiştir. Sınıflandırma işlemi sonucu, gerçekte iflas eden tüm firmalar arasından %41'i doğru olarak tespit edilmiştir. Algoritmanın genel sınıflandırma doğruluğu ise 0,88 oranındadır.

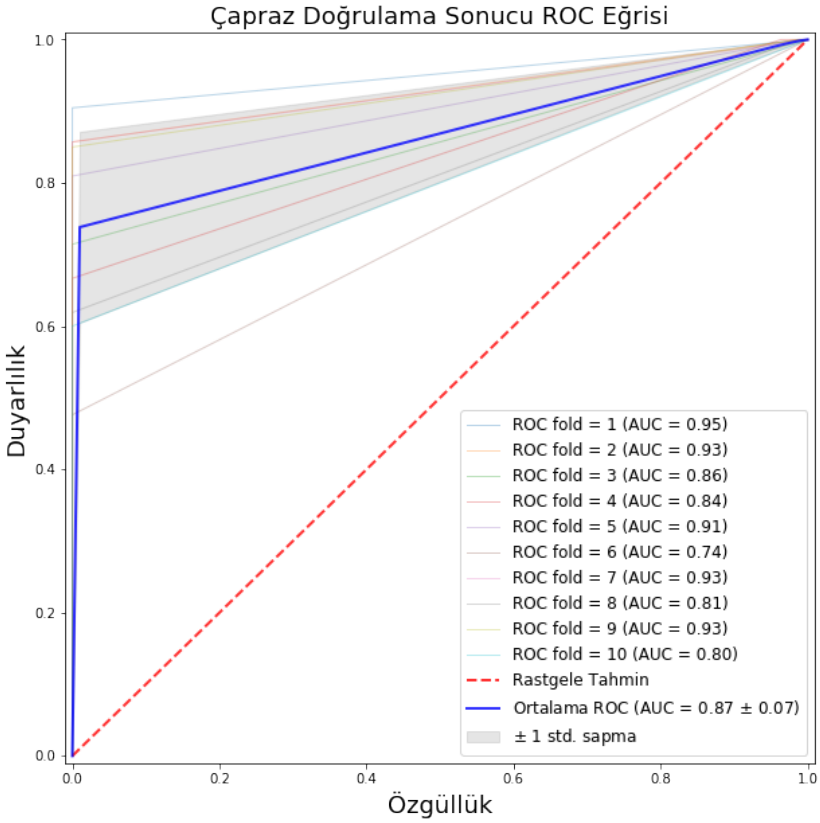
2 yıl sonraki iflas tahmini probleminde, gerçekte iflas eden firmalar arasından, sınıflama sonucu iflas edeceği belirlenen firmaların %31'i, iflas etmeyen firmaların %100'ü doğru tahmin edilmiştir. Algoritmanın geneli ele alındığında doğru sınıflandırma oranı 0,864 kabul edilir.

1 yıl sonraki iflas tahmini probleminde, gerçekte iflas eden firmalardan, sınıflama sonucu iflas edeceği tahmin edilenlerin %39'u, iflas etmeyen firmalar arasından %100'ü doğru tahmin edilmiştir. Algoritmanın genel doğru sınıflandırma oranı 0,884 kabul edilir.



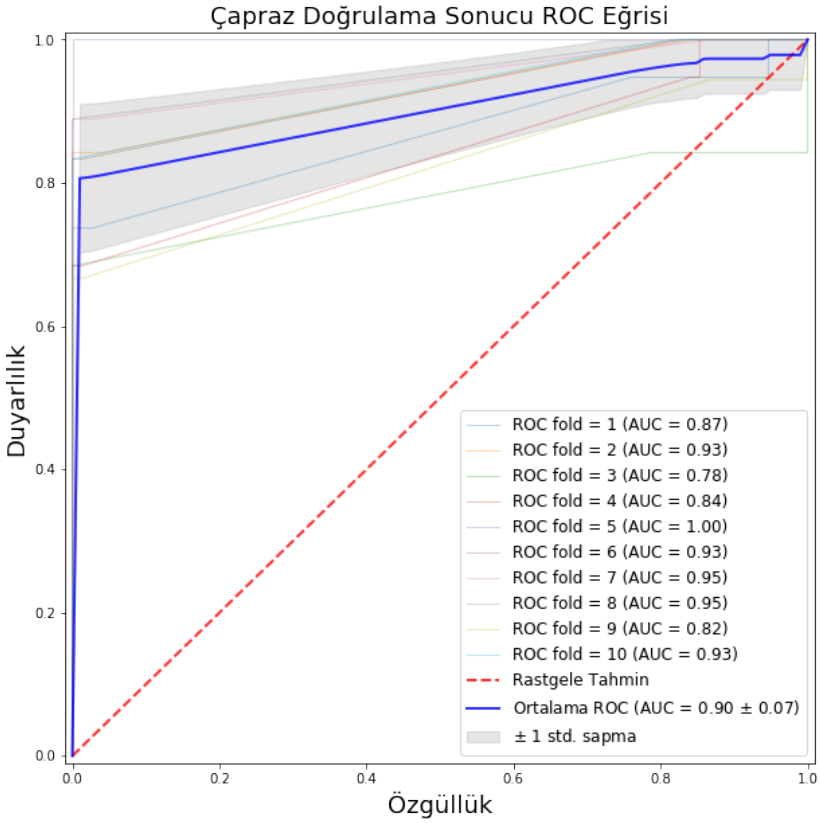
Şekil 5. 16. DVM ile 5 yıl Sonraki İflas Tahmini

Şeki 5.16'da 5 yıl sonraki iflas tahmini için ROC eğrisinde 10-katlamalı çapraz doğrulama işleminin her katmanı için eğri altındaki alan hesaplanmıştır. Katman=1 durumunda AUC değeri 0,79'dur. En düşük AUC değeri katman=6 olduğu durum ile 0,75 değeridir. En yüksek AUC değerine ise 1,00 ile katman=3 ve 10 durumunda ulaşılmıştır. 10 katmanın ortalaması alınarak, ortalama 0,88 AUC değeri ile algoritmanın sınıflama probleminde yüksek bir ayrıştırıcılığı olduğu söylenebilir.



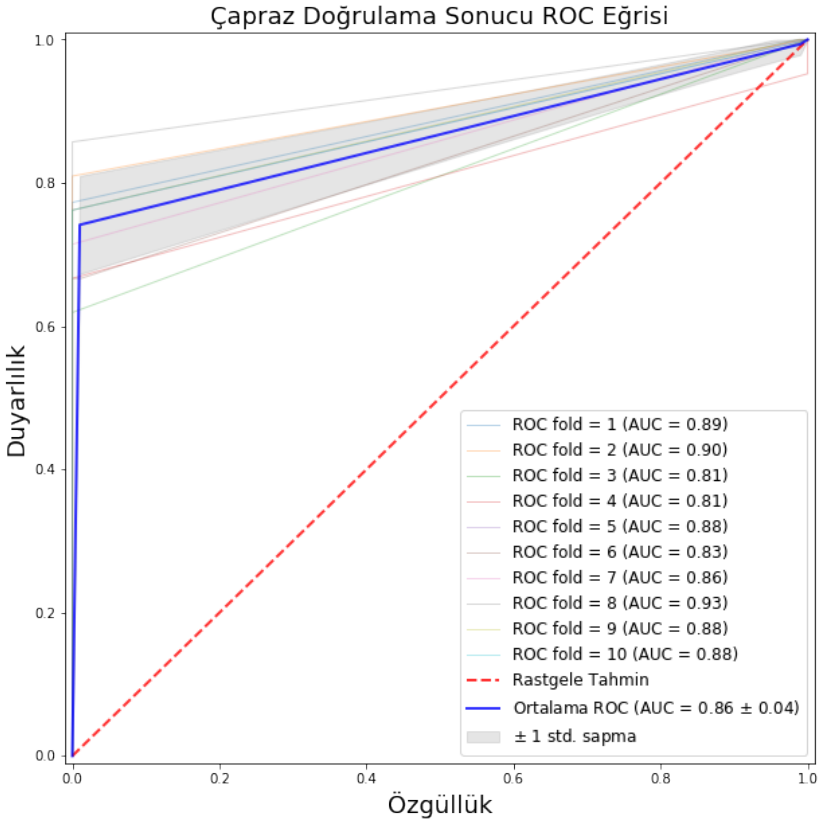
Şekil 5. 17. DVM ile 4 yıl Sonraki İflas Tahmini

4 yıl sonraki iflas tahmini için Şekil 5.17'de verilen ROC eğrisi hesaplanmıştır. En düşük AUC değeri katlama=10 olduğu durum ile 0,80 değeridir. En yüksek AUC değerine ise 0,95 ile katlama=1 durumunda ulaşılmıştır. Ortalama 0,87 AUC değeri ile algoritmanın sınıflama probleminde yüksek bir ayrıştırıcılığı olduğu söylenebilir.



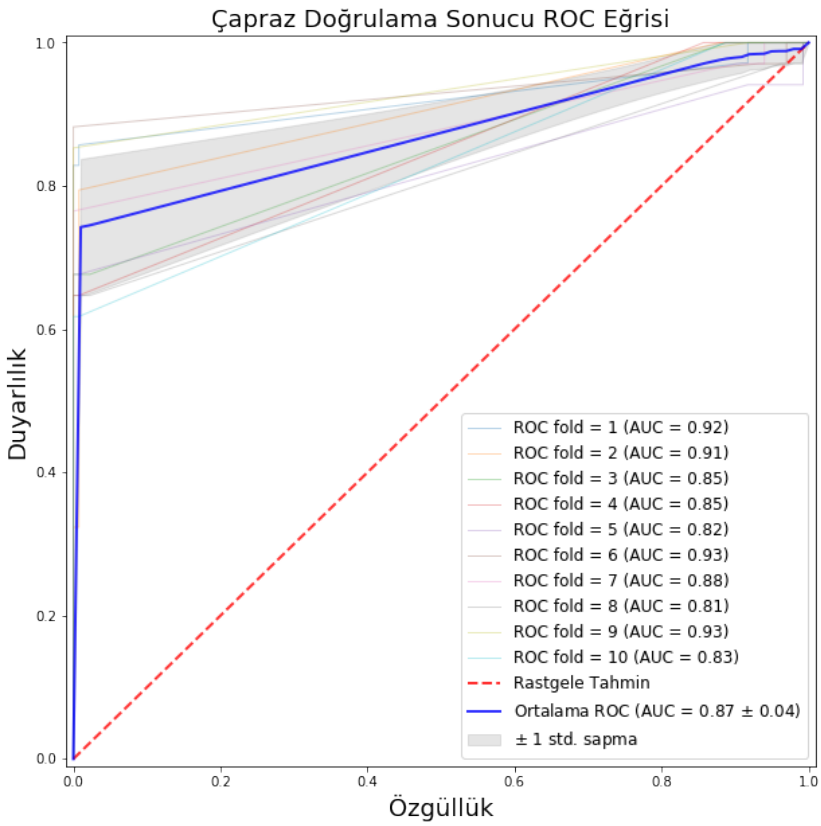
Şekil 5. 18. DVM ile 3 yıl Sonraki İflas Tahmini

Şekil 5.18'de 3 yıl sonraki iflas tahmini ROC eğrisine göre, en düşük AUC değeri katlama=9 olduğu durum ile 0,82 değeridir. En yüksek AUC değerine ise 0,95 ile katlama=7,8 durumunda ulaşılmıştır. 10 katlamanın ortalaması 0,90 AUC değeri ile algoritmanın sınıflama probleminde yüksek bir ayrıştırıcılığı olduğu söylenebilir.



Şekil 5. 19. DVM ile 2 yıl Sonraki İflas Tahmini

Şekil 5.19'da 2 yıl sonraki iflas tahmini için ROC eğrisi verilmiştir. En düşük AUC değeri katlama=3 olduğu durum ile 0,81 değeridir. En yüksek AUC değerine ise 0,93 ile katlama=8 durumunda ulaşılmıştır. 0,86 ortalama AUC değeri ile algoritmanın sınıflama probleminde yüksek bir ayrıştırıcılığı olduğu söylenebilir.



Şekil 5. 20. DVM ile 1 yıl Sonraki İflas Tahmini

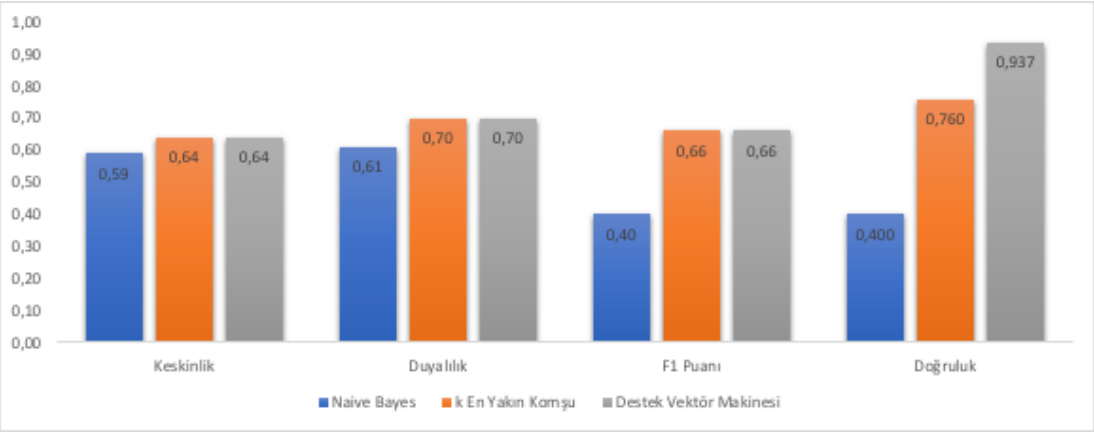
Şekil 5.20'de 1 yıl sonraki iflas tahmini için hesaplanan ROC eğrisi verilmiştir. En düşük AUC değeri katlama=8 olduğu durum ile 0,81 değeridir. En yüksek AUC değerine ise 0,93 ile katlama=6,9 durumunda ulaşılmıştır. Çapraz doğrulama katlamalarının ortalama 0,87 AUC değeri alması ile algoritmanın sınıflama probleminde yüksek bir ayrıştırıcılığı olduğu söylenebilir.

5.3.4. Sınıflama algoritmalarının karşılaştırılması

İflas tahmininde kullanılan Naive Bayes, k Yakın Komşu, Destek vektör Makinesi algoritmaları arasında hangi algoritmanın en iyi tahminlemeyi yaptığını tespit etmek amacıyla hesaplanan performans metriklerini karşılaştırma yoluna gidilmiştir.

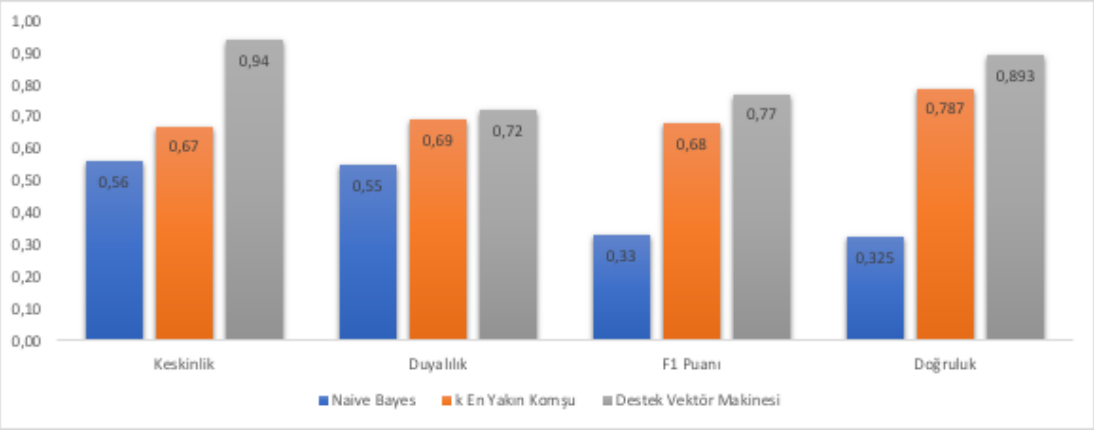
Veri setindeki sınıf verilerinin oranları algoritmaların performansını etkilemektedir. Bu nedenle, algoritmaların performans sonuçları, öncelikle iflas

tahmini yapılan yıllara göre ayrı ayrı karşılaştırılmıştır daha sonra 5 dönemin ortalaması üzerinden ayrıca karşılaştırma yapılmıştır.



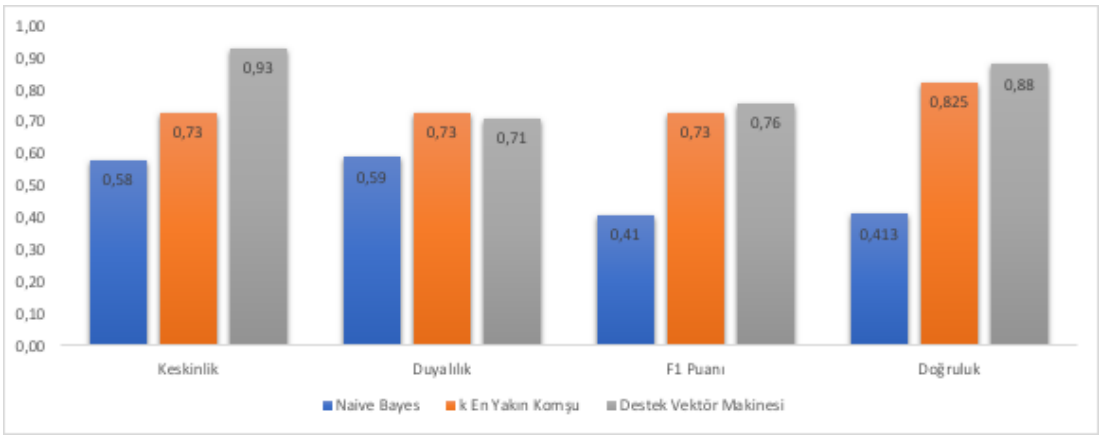
Şekil 5. 21. 5 Yıl Sonraki İflas Tahmini Performans Ölçütleri Karşılaştırması

Şekil 5.21'de 5 yıl sonraki iflas tahminine ait performans karşılaştırması verilmiştir. Keskinlik ve Duyarlılık değerlerinde kNN ve DVM algoritmaları eşit sonuç vermesine karşın en yüksek doğruluk değerine sahip algoritma DVM olmuştur.



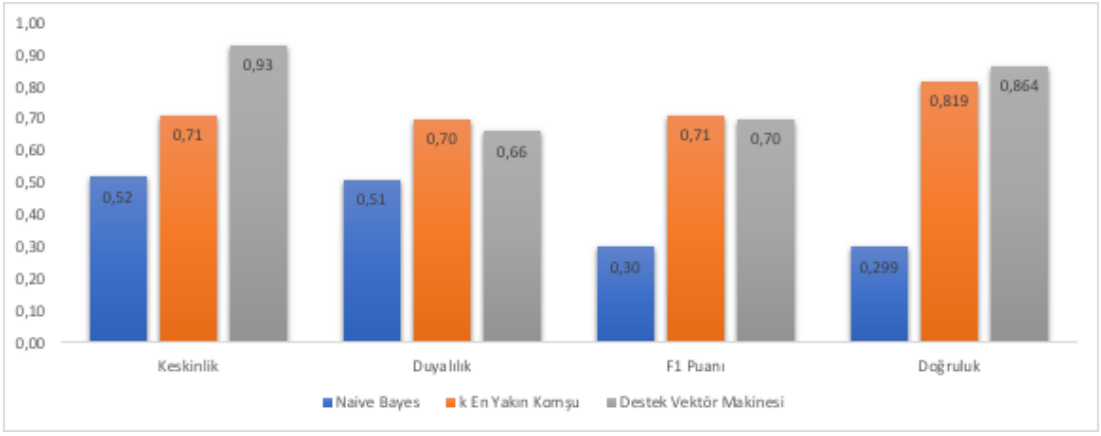
Şekil 5. 22. 4 Yıl Sonraki İflas Tahmini Performans Ölçütleri Karşılaştırması

4 yıl sonraki iflas tahminine ait performans karşılaştırması Şekil 5.22'de verilmiştir. 4 yıl sonraki iflas tahmini verileriyle eğitilen algoritmada tüm ölçütler için DVM algoritması en yüksek değere sahip olmuştur. En düşük performanslı algoritma ise kNN'dir.



Şekil 5. 23. 3 Yıl Sonraki İflas Tahmini Performans Ölçütleri Karşılaştırması

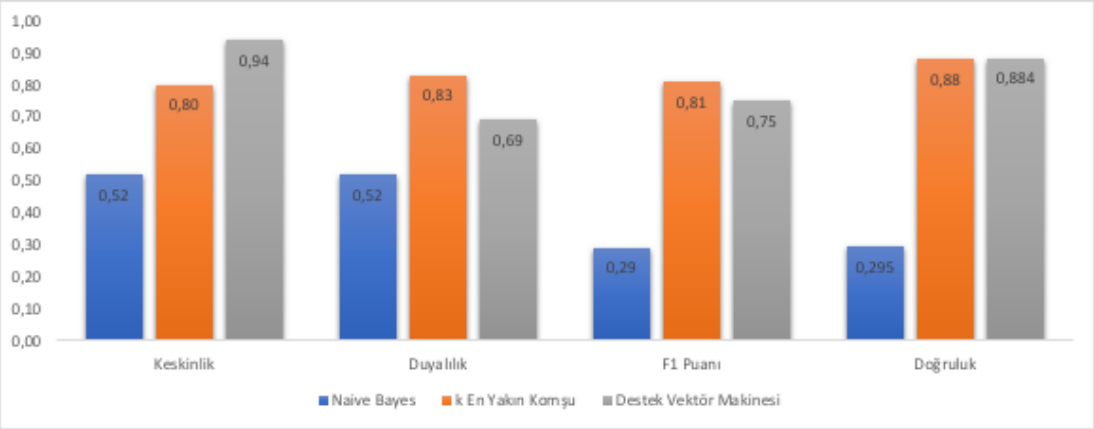
3 yıl sonraki iflas tahminine ait performans karşılaştırması Şekil 5.23'te verilmiştir. Duyarlılık ölçütünde kNN 0,02 oranlık fark ile DVM algoritmasından üstün gelmiştir. Doğruluk değerleri karşılaştırıldığında ise DVM algoritması en yüksek değere sahip olmuştur. En düşük performanslı algoritma ise yine kNN'dir.



Şekil 5. 24. 2 Yıl Sonraki İflas Tahmini Performans Ölçütleri Karşılaştırması

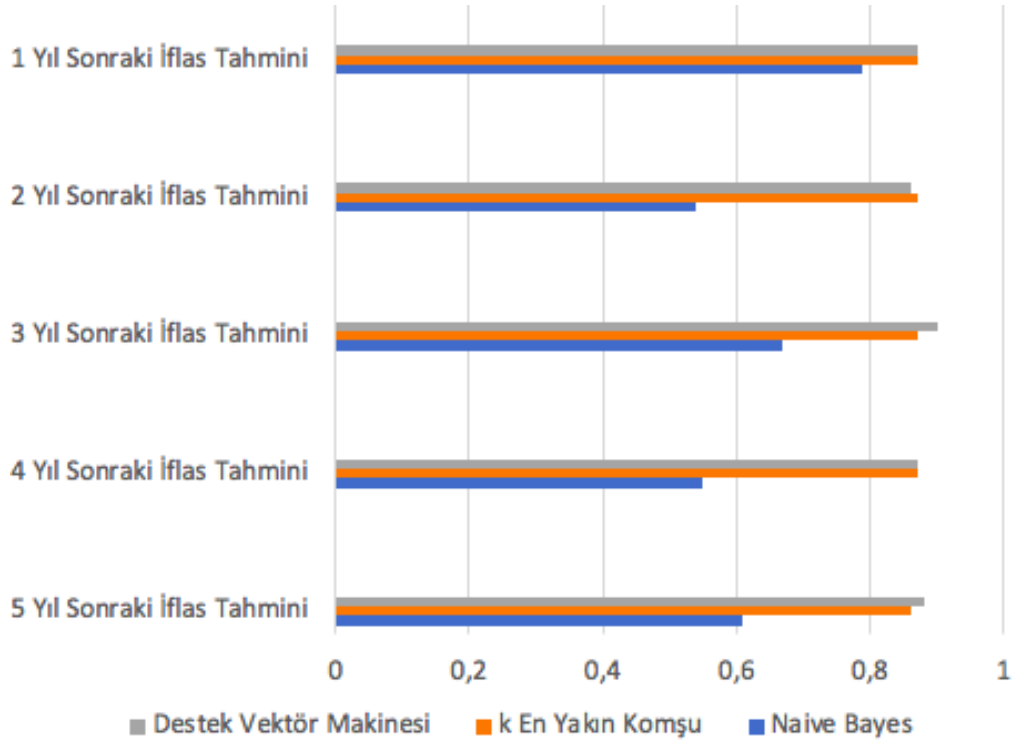
Şekil 5.24'te 2 yıl sonraki iflas tahminine ait performans karşılaştırması verilmiştir. Duyarlılık ölçütü ve F1 puanı özelinde kNN algoritması 0,04 ve 0,01oranlık fark ile DVM algoritmasından üstün gelmiştir. 2 yıl sonraki iflas

tahmini verileriyle eğitilen algoritmada tüm ölçütler için en yüksek performanslı DVM, en düşük performanslı algoritma ise kNN olmuştur.



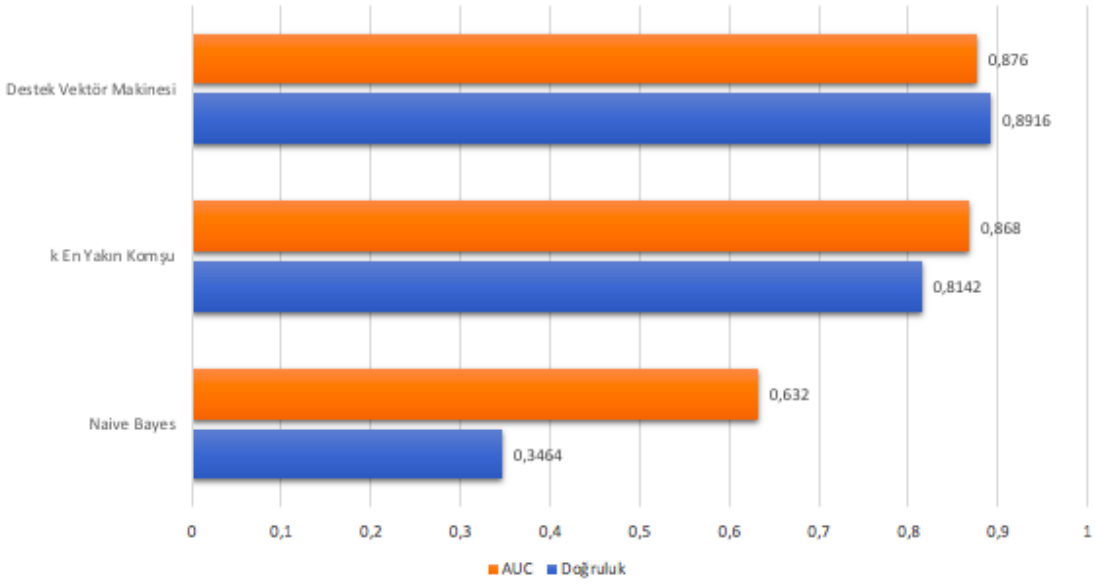
Şekil 5. 25. 1 Yıl Sonraki İflas Tahmini Performans Ölçütleri Karşılaştırması

1 yıl sonraki iflas tahminine ait performans incelendiğinde, Duyarlılık ölçütü ve F1 puanı özelinde kNN algoritması 0,14 ve 0,06 oranlık fark ile DVM algoritmasından üstün gelmiştir. İlgili dönem için doğruluk değerleri ise kNN ve DVM için hemen hemen aynı oranda olup, kNN algoritması bu dönem için de düşük performans sergilemiştir.



Şekil 5. 26. Tüm Dönemlerin AUC Değer Karşılaştırması

AUC değeri karşılaştırmalarına Şekil 5.26'da yer verilmiştir. Dönemlere göre AUC değerleri karşılaştırıldığında, yalnızca 2 yıl sonraki iflas tahmin edilmek istendiğinde k En Yakın Komşu algoritmasının en yüksek performansa sahip olduğu söylenebilir. Diğer durumlarda en yüksek performansa sahip algoritma Destek Vektör Makinesi olmuştur.



Şekil 5. 27. Ortalama AUC ve Doğruluk Karşılaştırması

5 dönemin ortalama doğruluk ve AUC değerleri Şekil 5.27'de verilmiştir. Ortalama doğruluk ve AUC değerleri göz önünde bulundurulduğunda da, üç algoritma arasından en düşük performansa sahip algoritma 0,34 doğruluk ve 0,63 AUC değeri ile Naive Bayes algoritması olmuştur.

K En Yakın Komşu algoritması performans karşılaştırmasında ikinci sırada gelmektedir. kNN algoritması, 0,81 doğruluk oranı ve 0,86 AUC değeriyle yüksek performans göstermiş olmasına karşı Destek Vektör makinelerinden görece düşük performans göstermiştir. Destek Vektör Makineleri ise 0,89 doğruluk ve 0,87 AUC değeriyle belirlenen üç algoritma arasından en yüksek performansa sahip model seçilmiştir.

6. SONUÇ ve ÖNERİLER

Makine öğrenmesi, istatistik ve bilgisayar bilimlerinin ortak amaçlarından doğmuş bir disiplindir. Günümüzde, verilerin saklanabilmesi, işlenebilmesi için başvurulan yöntemlerin çoğalması ile birlikte bu verileri anlamlı sonuçlara dönüştürme isteği artmış ve bu amaçla, birçok alanda, makine öğrenmesine gösterilen ilgi artmıştır. Makine öğrenmesi çözümlerini kullanan alanlardan biri de iflas tahmini olarak karşımıza çıkmaktadır. Makine öğrenmesi disiplini firmanın iflas edip etmeyeceğini ikili bir sınıflandırma problemine uyarlayarak bu konuyu tahminleme yapmaya açık hale getirmektedir.

İş dünyasında yaygın bir olgu olan iflas, şirketlerin finansal açıdan işlevini yerine getirememesi; şirketin borçlarını belirlenen süre içinde alacaklılara ödeyememesi ile ortaya çıkan bir süreçtir. Bu durumda şirkete sağlanan nakit ile şirketin altında bulunduğu borç miktarı dengeli değildir; firmanın elde ettiği kar, operasyonu devam ettirecek yeterliğe ulaşamamaktadır.

İflas, sonucu ağır kayıplara neden olabilen bir durumdur ancak, iflas zamana yaygın olarak gelişir, firmalar, hukuk dışı bir durum gerçekleşmiyorsa bir gecede iflas etmemektedir. Sürece yaygın olduğundan, firmanın gidişatı detaylı olarak incelendiğinde iflasın sinyallerini önceden fark etmek mümkün olabilmekte ve ağır kayıpların önüne geçilebilmektedir. Firmaların maddi konumlarını düzenli olarak kontrol ederek iflas olasılığını tespit etmeleri, ilerleyen dönemlerde olası bir riski önceden görmeleri ve tedbir almalarına yardımcı olur. Bununla birlikte, iflas riskini önceden belirlemek sadece iflasın önlemek için değil, aynı zamanda firmaların durumunu iyileştirmek amacıyla stratejik çözümler bulmak için de bir kaynak olabileceği düşünülmelidir. Bu amaçla literatürde birçok tahmin yöntemi kullanılmış, farklı alanlardaki çözümler iflas tahmini problemlerine de uygulanmıştır.

Firmaları iflasa götüren temel eylem borçlarını ödeyememe durumu olduğundan, iflasın temel nedeni kaynak sıkıntısı olarak düşünülebilir. Buradan

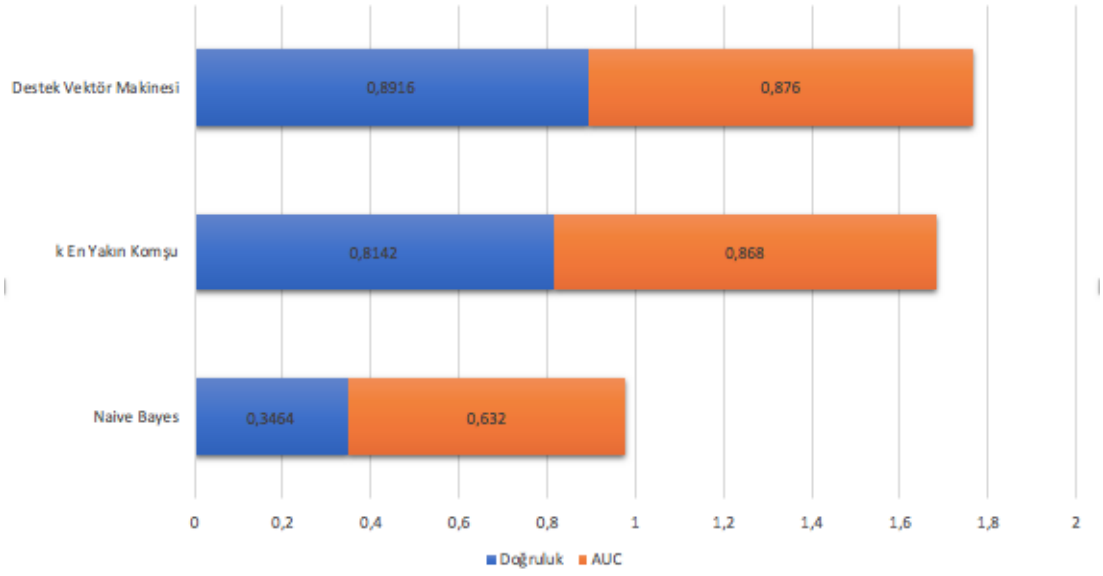
hareketle, iflas tahmini problemi genellikle firmanın finansal kalemlerinin değerlendirmesi temelli çalışmalar oluşturmaktadır. 1930'lar günümüze uzanan iflas tahmini çalışmalarında, başlangıçta, teorik yöntemler olarak adlandırılan finansal oranların birbirleri ile karşılaştırmasına dayanan oran analizlerine yer verilmektedir. Yıllar içerisinde literatüre yeni tahmin yöntemlerinin eklenmesi ile, 1967'de Beaver ve 1968'de Altman istatistiki tahmin yöntemlerine başvurulmuştur. Bu çalışmalar ile birlikte tek değişkenli analiz ve çoklu diskriminant analizi iflas tahmininde popüler hale gelmiştir. 1980'lerde ise lojit ve probit modeller ile sağlanan iflas tahmin çalışmalarına rastlanmaktadır. Ancak, istatistiksel yöntemlerin çeşitli varsayımları bulunduğundan ve bu varsayımları sağlamak çoğunlukla zor olduğundan modellerde açık noktalar kalmıştır. 1990'dan sonra ise yapay sinir sağları metodolojisinin iflas tahminine uygulanmasıyla birlikte bu çalışma alanında makine öğrenmesi teknikleri kullanılmaya başlanmıştır. Zaman içerisinde birçok makine öğrenmesi algoritması iflas tahmin problemine uyarlanmış ve uyarlanmaya devam etmektedir. Son zamanlarda ise farklı veri setleri ile algoritmaları karşılaştırma yoluna gidilmiş ve en iyi tahmini yapan algoritmalar seçilmiştir.

Bu çalışmada iflas tahmini motivasyonuyla, makine öğrenmesi sınıflanma problemlerinde kullanılan denetimli öğrenme algoritmalarından olasılık tabanlı Naive Bayes, tembel öğrenici K En Yakın Komşu ve klasik istatistiğin kuramlarının dışında, yapısal risk minimizasyonu temeliyle hareket eden Destek Vektör Makineleri kullanılmıştır.

EMIS veri tabanından toplanan verilerin kullanıldığı çalışmada 5 farklı dönemin iflas tahmini için 64 adet değişken elde edilmiştir. Çalışma öncesinde, her bir dönem için WOE ve IV değerleri analiz edilerek, 64 değişken arasından modele katkı sağlayacağı öngörülen değişkenler seçilmiştir. Analiz sonucunda, 5 yıl sonra iflas eden firmaların tahmini için 56 değişken; 4 yıl sonra iflas eden firmaların tahmini için 53 değişken; 3 yıl sonra iflas eden firmaların tahmini için

55 değişken; 2 yıl sonra iflas eden firmaların tahmini için 56 değişken ve 1 yıl sonra iflas eden firmaların tahmini için 38 değişken seçilmiştir.

Algoritmaların eğitim sırasında kullanılacak veri, tüm veri setinin %70'i olarak belirlenmiştir. Geriye kalan %30'luk veri seti ise test seti olarak ayrılmıştır. Algoritmaların performansları karşılaştırılırken bu test setinden yararlanılmıştır. Test setine çapraz doğrulama yöntemi uygulanılarak test veri seti 10 katlama ile farklılaştırılmıştır. Performans ölçütü olarak Duyarlılık, Keskinlik, F1 puan ve Doğruluk değerleri, bunların yanında duyarlılık ve özgüllük değerlerinin bir grafiğe işlenmesinden ortaya çıkan ROC eğrisi ve bu eğrinin altında kalan AUC değeri hesaplanmıştır. Modellerin karılaştırmasında temel gösterge olarak AUC değeri ve doğruluk değeri dikkate alınmıştır. Belirlenen üç algoritmaya ilişkin hesaplanan performans ölçütlerine tabloda yer verilmiştir.



Şekil 6. 1. AUC ve Doğruluk Sonuç Karşılaştırması

Tabloda paylaşıldığı üzere kNN ve DVM algoritmaları için her sınıflama problemi özelinde optimum parametreler belirlenmiştir. Sonuçlar karşılaştırıldığında, tüm algoritmaların iflas tahmini konusunda gözle görülebilir bir tahmin yeteneği olduğu söylenebilir. Karşılaştırılan üç algoritma arasından en düşük tahmin yeteneğine sahip algoritma 0,632 AUC değeri ile Naive Bayes olmuştur. kNN

algoritması ise 0,868 AUC değeri ile oldukça yüksek tahmin oranıyla sıralamada Naive Bayes algoritmasını takip etmektedir. Ancak DVM algoritması, ortalama 0,876 AUC değeri ile kNN'e görece üstünlük sağlayarak en iyi tahminlemeyi yapan algoritma olmuştur. Böylece, firmaların iflas edip etmeyeceğine dair öngörüde bulunulmak istendiğinde, Destek Vektör Makine algoritması tercih edilerek yapılacak tahminlerde, güçlü bir tahmin modeli ortaya çıkmış olacaktır.

0,867 tahmin oranı ile DVM algoritması oldukça yüksek bir oran ile iflas tahmini probleminde başarılı olmuştur. Ancak iflas, sonucu büyük kayıplara neden olan bir sorun olduğundan her zaman geliştirmeye ihtiyaç duyulan bir problem olarak düşünülmelidir. Bu nedenle iflas tahmini üzerine gelecekte yapılacak çalışmalar için öneriler aşağıda sunulmuştur.

Modelin geliştirmesine yönelik öneriler arasında ilk olarak modeli oluşturan metriklerde değişiklik yapmak düşünülebilir. Çalışmada kNN algoritması Euclidean uzaklık ölçütü kullanılarak oluşturulmuştur. İflas, kNN algoritması özelinde farklı uzaklık ölçütleri ile tahmin edilebilir. Farklı çalışmalar incelendiğinde Chebyshev, Manhattan, Minkowski gibi farklı uzaklık hesaplama ölçütleri kullanılarak farklı k değerleri hesaplandığı saptanmıştır.

Destek Vektör Makineleri algoritması özelinde, veriler özellikle uzayına haritalanırken kullanılabilecek çeşitli kernel fonksiyonları mevcuttur. Çalışmada Radyal Tabanlı fonksiyonlara her verilmiştir. Ancak Sigmoid, Polinomial fonksiyonlar kullanılarak algoritmalar tekrar oluşturulabilir. Farklı kernel fonksiyonlarının algoritma üzerindeki etkisi ölçümlenebilir.

Veri setindeki gözlem sayısı da modelin tahmin sonuçlarında oldukça etkili olmaktadır. Özellikle firma iflas gibi istenmeyen, yaşanmaması için çabalanan durumlarda bu durumun yaşandığı örnekler bulmakta zorlanılmaktadır, iflas eden firma azınlık sınıfı konumuna düşebilmektedir. Çalışmada bu durum için belli oranda SMOTE aşırı örnekleme tekniği kullanılmıştır. İleriki çalışmalarda

farklı oranlarda SMOTE tekniği ya da farklı aşırı örnekleme teknikleri kullanılabilir.

Veri seti oluşturulmasına yönelik öneriler arasında ise ilk olarak iflas tanımı gelmektedir. Literatürde iflas tanımına ilişkin oldukça geniş çaplı terimler, durumlar mevcuttur. İflas tanımının geniş oluşu, araştırmalarda veri seti oluşturulurken birtakım sapmaların meydana gelmesine neden olabilmektedir. Bu nedenle, iflas tanımı daraltılarak, bazı kriterlere indirgenerek incelenebilir. Örnek olarak, her ülkenin hukuk yapısının, iflas tanımının farklı olabileceği öngörüldüğünde iflas tanımının ülke bazında incelenmesi, tahminlerin tek bir ülke verisi üzerinden yapıldığında daha yüksek oranlarda tahminleme yapılabileceği düşünülmektedir.

Modeller kurulurken genellikle firmaların bilançolarından elde edilen finansal değişkenler kullanılmaktadır. Bu bağlamda bilanço rakamlarının doğruluğu modelde sapmalara neden olmamak amacıyla büyük önem taşımaktadır. İflas tahmininde veri seti oluşturulurken denetlenmiş bilançolara sahip firmaların finansal kalemlerinin toplanmasına dikkat edilmelidir.

İflasın nedenleri arasında firmaların ekonomik durumundan sonra yönetimle ilgili sıkıntılar gelmektedir. Modellerde firma yönetimine ilişkin değişkenler bulundurmak modelin daha geniş bir perspektiften değerlendirme yapmasına yardımcı olabilir. Firma ortakları sayısı, ortakların firmadaki hisse büyüklüğü, ortakların yaşı, firma üst yönetimi ile kan bağı gibi nitelikler değişken olarak modellere eklenebilir.

Çevresel faktörler de firmaların iflas etmesine neden olan önemli noktalardan biri olarak kabul edilmektedir. Özellikle ülke bazında iflas tahmin çalışması yapılmak istendiğinde, modele gayri safi yurtiçi büyüme, enflasyon, kredi risk primi gibi makro ekonomik değişkenlerin eklenmesinin faydalı olacağı öngörülmektedir. Çevresel faktörler içinde sektör riski de göz ardı edilmemelidir. Farklı sektörlerden firmaların bulunduğu veri setleri ile model

oluşturulmak istendiğinde firmanın içinde bulunduğu sektörün riskini aktaran değişkenlere yer vermek modelin iflas öngörüsünü yükselteceği tahmin edilmektedir.

KAYNAKLAR

Alaka, H. A., Oyedele, L. O., Owolabi, H. A., Kumar, V., Ajayi, S. O., Akinade, O. O., & Bilal, M. (2018). Systematic review of bankruptcy prediction models: Towards a framework for tool selection. Expert Systems with Applications, 94, 164-184. https://doi.org/10.1016/j.eswa.2017.10.040

Alexandropoulos, S.-A. N., Aridas, C. K., Kotsiantis, S. B., & Vrahatis, M. N. (2019). A Deep Dense Neural Network for Bankruptcy Prediction. Içinde J. Macintyre, L. Iliadis, I. Maglogiannis, & C. Jayne (Ed.), Engineering Applications of Neural Networks (C. 1000, ss. 435-444). Springer International Publishing. https://doi.org/10.1007/978-3-030-20257-6_37

Altman, E. I. (1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. 22.

Altman, E. I., & Hotchkiss, E. (2006). Corporate Financial Distress and Bankruptcy (Third Edition).

Altman, E. I., & Narayanan, P. (1997). An International Survey of Business Failure Classification Models. Financial Markets, Institutions and Instruments, 6(2), 1-57. https://doi.org/10.1111/1468-0416.00010

Amami, R., Ayed, D. B., & Ellouze, N. (2013). Practical Selection of SVM Supervised Parameters with Different Feature Representations for Vowel Recognition. 7.

Archana, S., & Elangovan, D. K. (2014). Survey of Classification Techniques in Data Mining. 2, 7.

Argenti, J. (1976). Corporate planning and Corporate Collapse. Long Range Planning, 9(6), 12-17. https://doi.org/10.1016/0024-6301(76)90006-6

Arieshanti, I., Purwananto, Y., Ramadhani, A., Nuha, M. U., & Ulinnuha, N. (2013). Comparative Study of Bankruptcy Prediction Models. TELKOMNIKA (Telecommunication Computing Electronics and Control), 11(3), 591. https://doi.org/10.12928/telkomnika.v11i3.1143

Balcaen, S., & Ooghe, H. (2006). 35 years of studies on business failure: An overview of the classic statistical methodologies and their related problems. The British Accounting Review, 38(1), 63-93. https://doi.org/10.1016/j.bar.2005.09.001

Bartlett, J. (2019). What's the difference between statistics and machine learning? https://thestatsgeek.com/2019/08/08/whats-the-difference-between-statistics-and-machine-learning/

Bin Altaf, M. A., & Yoo, J. (2016). A 1.83 J/Classification, 8-Channel, Patient-Specific Epileptic Seizure Classification SoC Using a Non-Linear Support Vector Machine. IEEE Transactions on Biomedical Circuits and Systems, 10(1), 49-60. https://doi.org/10.1109/TBCAS.2014.2386891

Bousquet, O., Luxburg, U. von, & Rätsch, G. (Ed.). (2004). Advanced lectures on machine learning: ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003 [and] Tübingen, Germany, August 4-16, 2003: revised lectures. Springer.

Burksaitiene, D., & Mazintiene, A. (2011). THE ROLE OF BANKRUPTCY FORECASTING IN THE COMPANY MANAGEMENT. ECONOMICS AND MANAGEMENT, 7.

Chen, H.-L., Liu, D.-Y., Yang, B., Liu, J., Wang, G., & Wang, S.-J. (2011). An Adaptive Fuzzy k-Nearest Neighbor Method Based on Parallel Particle Swarm Optimization for Bankruptcy Prediction. J. Z. Huang, L. Cao, & J. Srivastava (Ed.), Advances in Knowledge Discovery and Data Mining (C. 6634, ss. 249-264). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-20841-6_21

Cortes, C., & Vapnik, V. (1995). Support-vector networks. 25.

Çomak, E. (2008). Destek Vektör Makinelerinin Etkin Eğitimi İçin Yeni Yaklaşimlar.

Erdal, H. (2015). Contribution of Machine Learning Methods to the Construction Industry: Prediction of Compressive Strength. Pamukkale University Journal of Engineering Sciences, 21(3), 109-114. https://doi.org/10.5505/pajes.2014.26121

Ertel, W. (2011). Introduction to Artificial Intelligence.

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861-874. https://doi.org/10.1016/j.patrec.2005.10.010

Fernández-Gámez, M. A., Diéguez-Soto, J., & Santos, J. A. C. (2019). Bankruptcy Prediction of Family Firms Using Combined Classifiers. 78, 5.

Gepp, A., & Kumar, K. (2015). Predicting Financial Distress: A Comparison of Survival Analysis and Decision Tree Techniques. Procedia Computer Science, 54, 396-404. https://doi.org/10.1016/j.procs.2015.06.046

Ghory, I. (2004). Reinforcement learning in board games. 57.

Girma, H. (2009). A Tutorial on Support Vector Machine. 18.

Haklı, D. A. (2018). Sinif Dengesizliği Sorununu Çözmek İçin Kullanilan Algoritmalarin Farkli Siniflandirma Yöntemlerinde Performanslarinin Karşilaştirilmasi. 102.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). Overview of Supervised Learning. In: The Elements of Statistical Learning. İçinde Overview of Supervised Learning. https://link.springer.com/chapter/10.1007/978-0-387-84858-7_2

Hofmann, M. (2006). Support Vector Machines—Kernels and the Kernel Trick. 16.

Horak, J., Vrbka, J., & Suler, P. (2020). Support Vector Machine Methods and Artificial Neural Networks Used for the Development of Bankruptcy Prediction Models and their Comparison. Journal of Risk and Financial Management, 13(3), 60. https://doi.org/10.3390/jrfm13030060

Huang, J., & Ling, C. X. (2005). Using AUC and accuracy in evaluating learning algorithms. IEEE Transactions on Knowledge and Data Engineering, 17(3), 299-310. https://doi.org/10.1109/TKDE.2005.50

Jakkula, V. (2006). Tutorial on Support Vector Machine (SVM). 13.

Keats, B. W., & Bracker, J. S. (1988). Toward a Theory of Small Firm Performance: A Conceptual Model. American Journal of Small Business, 12(4), 41-58. https://doi.org/10.1177/104225878801200403

Klepáč, V., & Hampel, D. (2016). Prediction of Bankruptcy with SVM Classifiers Among Retail Business Companies in EU. Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis, 64(2), 627-634. https://doi.org/10.11118/actaun201664020627

Koh, H. C., & Tan, G. (2005). Data Mining Applications in Healthcare. 9.

Korol, T. (2019). Dynamic Bankruptcy Prediction Models for European Enterprises. Journal of Risk and Financial Management, 12(4), 185. https://doi.org/10.3390/jrfm12040185

Krishnan, S. (2018). Weight of evidence and Information Value using Python. Weight of evidence and Information Value using Python. https://sundarstyles89.medium.com/weight-of-evidence-and-information-value-using-python-6f05072e83eb

Lachenbruch, P. A., & Goldstein, M. (1975). Discriminant Analysis. 18.

Le, T., Lee, M., Park, J., & Baik, S. (2018). Oversampling Techniques for Bankruptcy Prediction: Novel Features from a Transaction Dataset. Symmetry, 10(4), 79. https://doi.org/10.3390/sym10040079

Leskovec, J., Rajaraman, A., & Ullman, J. D. (2011). Mining of Massive Datasets. http://infolab.stanford.edu/~ullman/mmds/book0n.pdf

Lim, T. C., Lim, J. X. Y., Siwei, G., & Jiang, H. (2012). Bankruptcy Prediction: Theoretical Framework Proposal. International Journal of Management Sciences and Business Research, 1(9), 69-74.

Lin, A. Z., & Hsieh, T.-Y. (2014). Expanding the Use of Weight of Evidence and Information Value to Continuous Dependent Variables for Variable Reduction and Scorecard Development. 23.

Ling, C. X., Huang, J., & Zhang, H. (2003). AUC: a Statistically Consistent and more Discriminating Measure than Accuracy. 6.

Lu, Y., Zeng, N., Liu, X., & Yi, S. (2015). A New Hybrid Algorithm for Bankruptcy Prediction Using Switching Particle Swarm Optimization and Support Vector Machines. Discrete Dynamics in Nature and Society, 2015, 1-7. https://doi.org/10.1155/2015/294930

Luxburg, U. von, & Schoelkopf, B. (2008). Statistical Learning Theory: Models, Concepts, and Results. ArXiv:0810.4752 [Math, Stat]. http://arxiv.org/abs/0810.4752

Mai, F., Tian, S., Lee, C., & Ma, L. (2019). Deep learning models for bankruptcy prediction using textual disclosures. European Journal of Operational Research, 274(2), 743-758. https://doi.org/10.1016/j.ejor.2018.10.024

Murty, M. N., & Raghava, R. (2016). Kernel-Based SVM. M. N. Murty & R. Raghava, Support Vector Machines and Perceptrons (ss. 57-67). Springer International Publishing. https://doi.org/10.1007/978-3-319-41063-0_5

Nwogugu, M. (2005). TITLE: DECISION-MAKING, RISK AND CORPORATE GOVERNANCE: A CRITIQUE OF BANKRUPTCY/RECOVERY PREDICTION MODELS – PART ONE. 20.

Olson, D. L., & Delen, D. (2008). Advanced data mining techniques. Springer.

Olson, D. L., Delen, D., & Meng, Y. (2012). Comparative analysis of data mining methods for bankruptcy prediction. Decision Support Systems, 52(2), 464-473. https://doi.org/10.1016/j.dss.2011.10.007

Onan, A. (2015). Şirket İflaslarının Tahminlenmesinde Karar Ağacı Algoritmalarının Karşılaştırmalı Başarım Analizi. Bilişim Teknolojileri Dergisi, 8(1). https://doi.org/10.17671/btd.36087

Ooghe, H., & De Prijcker, S. (2008). Failure processes and causes of company bankruptcy: A typology. Management Decision, 46(2), 223-242. https://doi.org/10.1108/00251740810854131

Park, H.-A. (2013). An Introduction to Logistic Regression: From Basic Concepts to Interpretation with Particular Attention to Nursing Domain. Journal of Korean Academy of Nursing, 43(2), 154. https://doi.org/10.4040/jkan.2013.43.2.154

Ross, S. A., Westerfield, R. W., & Jordan, B. D. (2010). FUNDAMENTALS OF CORPORATE FINANCE.

Rossi, F., & Villa, N. (2006). Support vector machine for functional data classification. 13.

Sarang, N. (2018). Understanding AUC - ROC Curve. https://www.48hours.ai/files/AUC.pdf

Saritas, M. M. (2019). Performance Analysis of ANN and Naive Bayes Classification Algorithm for Data Classification. International Journal of Intelligent Systems and Applications in Engineering, 7(2), 88-91. https://doi.org/10.18201/ijisae.2019252786

Scott, J. (1981). The probability of bankruptcy. Journal of Banking & Finance, 5(3), 317-344. https://doi.org/10.1016/0378-4266(81)90029-7

Shariq, M. (2016). Bankruptcy Prediction by Using the Altman Z-score Model in Oman: A Case Study of Raysut Cement Company SAOG and its subsidiaries. Australasian Accounting, Business and Finance Journal, 10(4). https://doi.org/10.14453/aabfj.v10i4.6

Shi, Y., & Li, X. (2019). An overview of bankruptcy prediction models for corporate firms: A Systematic literature review. Intangible Capital, 15(2), 114. https://doi.org/10.3926/ic.1354

Staňková, M., & Hampel, D. (2018). Bankruptcy Prediction of Engineering Companies in the EU Using Classification Methods. Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis, 66(5), 1347-1356. https://doi.org/10.11118/actaun201866051347

Starbuck, W. H., & Hedberg, B. (2001). How Organizations Learn from Success and Failure. 26.

Thornhill, S., & Amit, R. (2003). Learning About Failure: Bankruptcy, Firm Age, and the Resource-Based View. Organization Science, 14(5), 497-509. https://doi.org/10.1287/orsc.14.5.497.16761

Tuba, E., Mrkela, L., & Tuba, M. (2016). Support vector machine parameter tuning using firefly algorithm. 2016 26th International Conference Radioelektronika (RADIOELEKTRONIKA), 413-418. https://doi.org/10.1109/RADIOELEK.2016.7477388

Vapnik, V. (1992). Principles of Risk Minimization for Learning Theory. 8.

Yu, H., & Kim, S. (2012). SVM Tutorial: Classification, Regression, and Ranking. 32.

Zeitun, R., Tian, G., & Keen, K. (2007). Default probability for the Jordanian companies: A test of cash flow theory. 21.

Zeng, G. (2014). A necessary condition for a good binning algorithm in credit scoring. Applied Mathematical Sciences, 8, 3229-3242. https://doi.org/10.12988/ams.2014.44300

Zhou, L., Lai, K. K., & Yen, J. (2014). Bankruptcy prediction using SVM models with a new approach to combine features selection and parameter optimisation. International Journal of Systems Science, 45(3), 241-253. https://doi.org/10.1080/00207721.2012.720293

Zięba, M., Tomczak, S. K., & Tomczak, J. M. (2016). Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. Expert Systems with Applications, 58, 93-101. https://doi.org/10.1016/j.eswa.2016.04.001

Zopounidis, C., & Doumpos, M. (1999). A Multicriteria Decision Aid Methodology for Sorting Decision Problems: The Case of Financial Distress. 22.

EKLER

Ek.1 WOE ve IV Hesaplama Kodları

```
def calculate_woe_iv(dataset, feature, target):
    lst = []
    for i in range(dataset[feature].nunique()):
        val = list(dataset[feature].unique())[i]
        lst.append({
            'Value': val,
            'All': dataset[dataset[feature] == val].count()[feature],
            'Good': dataset[(dataset[feature] == val) & (dataset[target] == 0)].count()[feature],
            'Bad': dataset[(dataset[feature] == val) & (dataset[target] == 1)].count()[feature]
        })

    dset = pd.DataFrame(lst)
    dset['Distr_Good'] = dset['Good'] / dset['Good'].sum()
    dset['Distr_Bad'] = dset['Bad'] / dset['Bad'].sum()
    dset['WoE'] = np.log(dset['Distr_Good'] / dset['Distr_Bad'])
    dset = dset.replace({'WoE': {np.inf: 0, -np.inf: 0}})
    dset['IV'] = (dset['Distr_Good'] - dset['Distr_Bad']) * dset['WoE']
    iv = dset['IV'].sum()

    dset = dset.sort_values(by='WoE')

    return dset, iv
```

```
for col in dataset.columns:
    if col == 'class': continue
    else:
        print('WoE and IV for column: {}'.format(col))
        df, iv = calculate_woe_iv(dataset, col, 'class')
        print(df)
        print('IV score: {:.2f}'.format(iv))
        print('\n')
```

Ek.2 SMOTE Örnekleme ve Test Train Ayrımı Kodları

```
import pandas as pd
import numpy as np
import imblearn
from sklearn.model_selection import train_test_split
from collections import Counter
from sklearn.datasets import make_classification
from imblearn.over_sampling import RandomOverSampler
from imblearn.under_sampling import RandomUnderSampler
from imblearn.under_sampling import RandomUnderSampler
from sklearn.datasets import make_classification
from imblearn.over_sampling import SMOTE
df = pd.read_excel('1yearafterIV.xlsx')
df = df.dropna()
data = df.values
X = data[:,0:56]
y = data[:,56]
print(Counter(y))
undersample = RandomUnderSampler(sampling_strategy=0.1, random_state= 42)
# fit and apply the transform
X_over, y_over = undersample.fit_resample(X, y)
X= X_over
y=y_over
print(Counter(y))
# transform the dataset
sm = SMOTE(ratio = 'minority', random_state=42)
X_sm, y_sm = sm.fit_sample(X,y)
print('smote resample dataset shape %s' %Counter(y_sm))
#X_sm, y_sm = oversample.fit_resample(X, y)
over = RandomOverSampler(sampling_strategy=0.25)
X_sm, y_sm = over.fit_resample(X, y)
print('resample dataset shape %s' %Counter(y_sm))
X = X_sm
y = y_sm
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.30, random_state=42)# transform the dataset
print('train  dataset shape %s' %Counter(y_train))
```

Ek.3 NB ile Sınıflandırma Çalışması Kodları

```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb_model = nb.fit(X_train, y_train)
nb_model
```

```
GaussianNB(priors=None, var_smoothing=1e-09)
```

```
from sklearn.datasets import make_classification
from sklearn.model_selection import cross_val_predict
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report

# generate some artificial data with 11 classes
#X, y = make_classification(n_samples=2000, n_features=20, n_informative=10, n_classes=11, random_state=0)

# your classifier, assume GaussianNB here for non-integer data X
estimator = GaussianNB()
# generate your cross-validation prediction with 10 fold Stratified sampling
y_pred = cross_val_predict(estimator, X_test, y_test, cv=10)
y_pred.shape

#Out[91]: (2000,)

# generate report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

         0.0       1.00      0.27      0.42        93
         1.0       0.23      1.00      0.37        20

   micro avg       0.40      0.40      0.40       113
   macro avg       0.61      0.63      0.40       113
weighted avg       0.86      0.40      0.41       113
```

```
cv = StratifiedKFold(n_splits=10)
classifier = GaussianNB()

tprs = []
aucs = []
mean_fpr = np.linspace(0, 1, 100)
plt.figure(figsize=(10,10))
i = 1
for train, test in cv.split(X_train, y_train):
    probas_ = classifier.fit(X_train[train], y_train[train]).predict_proba(X_train[test])
    # Compute ROC curve and area the curve
    fpr, tpr, thresholds = roc_curve(y_train[test], probas_[:, 1])
    tprs.append(interp(mean_fpr, fpr, tpr))
    tprs[-1][0] = 0.0
    roc_auc = auc(fpr, tpr)
    aucs.append(roc_auc)
    plt.plot(fpr, tpr, lw=1, alpha=0.3,
             label='ROC fold = %d (AUC = %0.2f)' % (i, roc_auc))

    i += 1
plt.plot([0, 1], [0, 1], linestyle='--', lw=2, color='r',
         label='Rastgele Tahmin', alpha=.8)

mean_tpr = np.mean(tprs, axis=0)
mean_tpr[-1] = 1.0
mean_auc = auc(mean_fpr, mean_tpr)
std_auc = np.std(aucs)
plt.plot(mean_fpr, mean_tpr, color='b',
         label=r'Ortalama ROC (AUC = %0.2f $\pm$ %0.2f)' % (mean_auc, std_auc),
         lw=2, alpha=.8)

std_tpr = np.std(tprs, axis=0)
tprs_upper = np.minimum(mean_tpr + std_tpr, 1)
tprs_lower = np.maximum(mean_tpr - std_tpr, 0)
plt.fill_between(mean_fpr, tprs_lower, tprs_upper, color='grey', alpha=.2,
                 label=r'$\pm$ 1 std. sapma')

plt.xlim([-0.01, 1.01])
plt.ylim([-0.01, 1.01])
plt.xlabel('Özgüllük',fontsize=18)
plt.ylabel('Duyarlılık',fontsize=18)
plt.title('Çapraz Doğrulama Sonucu ROC Eğrisi',fontsize=18)
plt.legend(loc="lower right", prop={'size': 12})
plt.show()
```

Ek.4 kNN ile Sınıflandırma Çalışması Kodları

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=3, metric='euclidean')
classifier.fit(X_train, y_train)
```

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='euclidean',
           metric_params=None, n_jobs=None, n_neighbors=3, p=2,
           weights='uniform')
```

```
y_pred = classifier.predict(X_test)
```

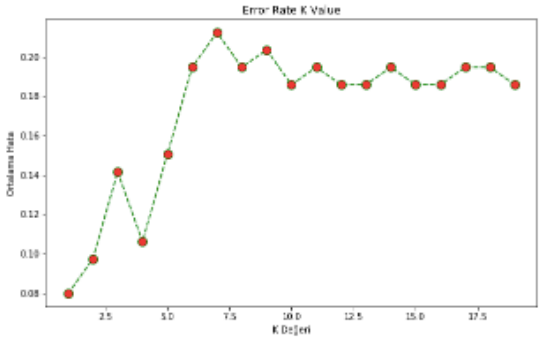
```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[84  9]
 [ 7 13]]
              precision    recall  f1-score   support

         0.0       0.92      0.90      0.91        93
         1.0       0.59      0.65      0.62        20

   micro avg       0.86      0.86      0.86       113
   macro avg       0.76      0.78      0.77       113
weighted avg       0.86      0.86      0.86       113
```

```
error = []
for i in range(1, 20):
    knn = KNeighborsClassifier(n_neighbors=i, metric='euclidean')
    knn.fit(X_train, y_train)
    pred_i = knn.predict(X_test)
    error.append(np.mean(pred_i != y_test))
```

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
plt.plot(range(1, 20), error, color='green', linestyle='dashed', marker='o',
         markerfacecolor='red', markersize=10)
plt.title('Error Rate K Value')
plt.xlabel('K Değeri')
plt.ylabel('Ortalama Hata')
```

```
Text(0, 0.5, 'Ortalama Hata')
```




```
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
from sklearn.model_selection import RepeatedKFold
from sklearn.model_selection import cross_val_score
# 10-fold cross-validation with the best KNN model
knn = KNeighborsClassifier(n_neighbors=1, metric='euclidean')
#cv = RepeatedKFold(n_splits=10, n_repeats=10, random_state=1)

# Instead of saving 10 scores in object named score and calculating mean
# We're just calculating the mean directly on the results
print(cross_val_score(knn, X_test, y_test, cv=10, scoring='accuracy').mean())
```

```
0.7606060606060607
```

```
from sklearn.datasets import make_classification
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import classification_report

# generate some artificial data with 11 classes
#X, y = make_classification(n_samples=2000, n_features=20, n_informative=10, n_classes=11, random_state=0)

# your classifier, assume GaussianNB here for non-integer data X
estimator = knn = KNeighborsClassifier(n_neighbors=1, metric='euclidean')
# generate your cross-validation prediction with 10 fold Stratified sampling
y_pred = cross_val_predict(estimator, X_test, y_test, cv=10)
y_pred.shape

#Out[91]: (2000,)

# generate report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

         0.0       0.90      0.80      0.85        93
         1.0       0.39      0.60      0.47        20

   micro avg       0.76      0.76      0.76       113
   macro avg       0.64      0.70      0.66       113
weighted avg       0.81      0.76      0.78       113
```

```
cv = StratifiedKFold(n_splits=10, random_state=42)
classifier = KNeighborsClassifier(n_neighbors=1, metric='euclidean')

tprs = []
aucs = []
mean_fpr = np.linspace(0, 1, 100)
plt.figure(figsize=(10,10))
i = 1
for train, test in cv.split(X_train, y_train):
    probas_ = classifier.fit(X_train[train], y_train[train]).predict_proba(X_train[test])
    # Compute ROC curve and area the curve
    fpr, tpr, thresholds = roc_curve(y_train[test], probas_[:, 1])
    tprs.append(interp(mean_fpr, fpr, tpr))
    tprs[-1][0] = 0.0
    roc_auc = auc(fpr, tpr)
    aucs.append(roc_auc)
    plt.plot(fpr, tpr, lw=1, alpha=0.3,
             label='ROC fold = %d (AUC = %0.2f)' % (i, roc_auc))

    i += 1
plt.plot([0, 1], [0, 1], linestyle='--', lw=2, color='r',
         label='Rastgele Tahmin', alpha=.8)

mean_tpr = np.mean(tprs, axis=0)
mean_tpr[-1] = 1.0
mean_auc = auc(mean_fpr, mean_tpr)
std_auc = np.std(aucs)
plt.plot(mean_fpr, mean_tpr, color='b',
         label=r'Ortalama ROC (AUC = %0.2f $\pm$ %0.2f)' % (mean_auc, std_auc),
         lw=2, alpha=.8)

std_tpr = np.std(tprs, axis=0)
tprs_upper = np.minimum(mean_tpr + std_tpr, 1)
tprs_lower = np.maximum(mean_tpr - std_tpr, 0)
plt.fill_between(mean_fpr, tprs_lower, tprs_upper, color='grey', alpha=.2,
                 label=r'$\pm$ 1 std. sapma')

plt.xlim([-0.01, 1.01])
plt.ylim([-0.01, 1.01])
plt.xlabel('Özgüllük',fontsize=18)
plt.ylabel('Duyarlılık',fontsize=18)
plt.title('Çapraz Doğrulama Sonucu ROC Eğrisi',fontsize=18)
plt.legend(loc="lower right", prop={'size': 12})
plt.show()
```

Ek.5 DVM ile Sınıflandırma Çalışması Kodları

```
from sklearn.svm import SVC
svc_model = SVC(kernel = "rbf").fit(X_train, y_train)
svc_model
```

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='rbf', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)
```

```
y_pred = svc_model.predict(X_test)
accuracy_score(y_test, y_pred)
```

```
0.9823008849557522
```

```
svc_params = {"C": [0.0001, 0.001, 0.1, 1, 5, 10 ,50 ,100],
              "gamma": [0.0001, 0.001, 0.1, 1, 5, 10 ,50 ,100]}
```

```
svc = SVC()
svc_cv_model = GridSearchCV(svc, svc_params,
                            cv = 10,
                            n_jobs = -1,
                            verbose = 2)

svc_cv_model.fit(X_train, y_train)
```

```
Fitting 10 folds for each of 64 candidates, totalling 640 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done  39 tasks      | elapsed:    2.4s
[Parallel(n_jobs=-1)]: Done 640 out of 640 | elapsed:    4.5s finished
GridSearchCV(cv=10, error_score='raise-deprecating',
       estimator=SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='rbf', max_iter=-1, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False),
       fit_params=None, iid='warn', n_jobs=-1,
       param_grid={'C': [0.0001, 0.001, 0.1, 1, 5, 10, 50, 100], 'gamma': [0.0001, 0.001, 0.1, 1, 5, 10, 50, 100]},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
       scoring=None, verbose=2)
```

```
print("En iyi parametreler: " + str(svc_cv_model.best_params_))
```

```
En iyi parametreler: {'C': 1, 'gamma': 0.0001}
```

```
svc_tuned = SVC(kernel = "rbf", C = 1, gamma = 0.0001).fit(X_train, y_train)
```

```
from sklearn.datasets import make_classification
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import classification_report

# generate some artificial data with 11 classes
#X, y = make_classification(n_samples=2000, n_features=20, n_informative=10, n_classes=11, random_state=0)

# your classifier, assume GaussianNB here for non-integer data X
estimator = SVC(kernel = "rbf", C = 1, gamma = 0.0001)
# generate your cross-validation prediction with 10 fold Stratified sampling
y_pred = cross_val_predict(estimator, X_test, y_test, cv=10)
y_pred.shape

#Out[91]: (2000,)

# generate report
print(classification_report(y_test, y_pred))
```

```
cv = StratifiedKFold(n_splits=10, random_state=42)

classifier = SVC(kernel = "rbf", C = 1, gamma = 0.0001, probability=True).fit(X_train, y_train)

import sklearn.metrics as metrics
# calculate the fpr and tpr for all thresholds of the classification
probs = svc_tuned.predict_proba(X_test)

tprs = []
aucs = []
mean_fpr = np.linspace(0, 1, 100)
plt.figure(figsize=(10,10))
i = 1
for train, test in cv.split(X_train, y_train):
    probas_ = classifier.fit(X_train[train], y_train[train]).predict_proba(X_train[test])
    # Compute ROC curve and area the curve
    fpr, tpr, thresholds = roc_curve(y_train[test], probas_[:, 1])
    tprs.append(interp(mean_fpr, fpr, tpr))
    tprs[-1][0] = 0.0
    roc_auc = auc(fpr, tpr)
    aucs.append(roc_auc)
    plt.plot(fpr, tpr, lw=1, alpha=0.3,
             label='ROC fold = %d (AUC = %0.2f)' % (i, roc_auc))
```

```
    i += 1
plt.plot([0, 1], [0, 1], linestyle='--', lw=2, color='r',
         label='Rastgele Tahmin', alpha=.8)

mean_tpr = np.mean(tprs, axis=0)
mean_tpr[-1] = 1.0
mean_auc = auc(mean_fpr, mean_tpr)
std_auc = np.std(aucs)
plt.plot(mean_fpr, mean_tpr, color='b',
         label=r'Ortalama ROC (AUC = %0.2f $\pm$ %0.2f)' % (mean_auc, std_auc),
         lw=2, alpha=.8)

std_tpr = np.std(tprs, axis=0)
tprs_upper = np.minimum(mean_tpr + std_tpr, 1)
tprs_lower = np.maximum(mean_tpr - std_tpr, 0)
plt.fill_between(mean_fpr, tprs_lower, tprs_upper, color='grey', alpha=.2,
                 label=r'$\pm$ 1 std. sapma')

plt.xlim([-0.01, 1.01])
plt.ylim([-0.01, 1.01])
plt.xlabel('Özgüllük',fontsize=18)
plt.ylabel('Duyarlılık',fontsize=18)
plt.title('Çapraz Doğrulama Sonucu ROC Eğrisi',fontsize=18)
plt.legend(loc="lower right", prop={'size': 12})
plt.show()
```

ÖZGEÇMİŞ

Adı Soyadı : Gizem DİLKİ

Doğum Yeri ve Yılı : İstanbul, 15/06/1994

Medeni Hali : Bekar

Yabancı Dili : İngilizce

E-posta : gizemdilki2@gmail.com

Eğitim Durumu

Lise : Sabancı Lisesi, 2012

Lisans : İstanbul Ticaret Üniversitesi, İnsan ve Topum bilimleri Fakültesi, İstatistik Bölümü

Mesleki Deneyim

Crif Enformasyon Derecelendirme ve Danışmanlık AŞ 2018-2020

Alternatif Bank 2020-…(devam ediyor)

Yayınları

Dilki, G., Deniz Başar, Ö. (2020). İşletmelerin İflas Tahmininde K- En Yakın Komşu Algoritması Üzerinden Uzaklık Ölçütlerinin Karşılaştırılması, İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, 19 (38), 224-233.

Dilki, G. (2020). Makine Öğrenmesi Algoritmalarının Sınıflama Problemleri Üzerinden Karşılaştırılması: Satış Tahmini, Pressacademia Procedia, 12 (1) , 82-83.