ÇANKIRI KARATEKİN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

DİYABET TESPİTİNDE MAKİNE ÖĞRENMESİ ALGORİTMALARI YAKLAŞIMINI KULLANARAK YAPILMIŞ BİR ÇALIŞMA

Ayşe DOĞRU

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ÇANKIRI
2021

TEZ ONAYI

Ayşe DOĞRU tarafından hazırlanan **"Diyabet Tespitinde Makine Öğrenmesi Algoritmaları Yaklaşımını Kullanarak Yapılmış Bir Çalışma"** adlı tez çalışması 26/01/2022 tarihinde aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Çankırı Karatekin Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalında **Yüksek Lisans Tezi** olarak kabul edilmiştir.

Danışman : Prof. Dr. Murat ARI

Eş Danışman : Dr. Öğr. Üyesi Selim BUYRUKOĞLU

Jüri Üyeleri :

Başkan : Prof. Dr. Murat ARI
Telekomünikasyon Anabilim Dalı
Çankırı Karatekin Üniversitesi

Üye : Dr. Öğr. Üyesi Serkan SAVAŞ
Bilgisayar Bilimleri Anabilim Dalı
Çankırı Karatekin Üniversitesi

Üye : Dr. Öğr. Üyesi Yıldıran YILMAZ
Bilgisayar Donanımı Anabilim Dalı
Recep Tayyip Erdoğan Üniversitesi

Yukarıdaki sonucu onaylarım

Prof. Dr. İbrahim ÇİFTÇİ

Enstitü Müdürü

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Çankırı Karatekin Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğine göre hazırlamış olduğum "**Diyabet Tespitinde Makine Öğrenmesi Algoritmaları Yaklaşımını Kullanarak Yapılmış Bir Çalışma**" konulu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı, tezin içerdiği yenilik ve sonuçları başka bir yerden almadığımı, tezde kullandığım eserleri usulüne göre kaynak olarak gösterdiğimi, tezin Çankırı Karatekin Üniversitesi Fen Bilimleri Enstitüsü'nden başka bir bilim kuruluna akademik amaç ve unvan almak amacıyla vermediğimi ve bu çalışmanın Çankırı Karatekin Üniversitesi tarafından kullanılan "Bilimsel İntihal Tespit Programı"yla tarandığını, "intihal içermediğini" beyan ederim. Çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması halinde ortaya çıkacak tüm ahlaki ve hukuki sonuçlara razı olduğumu bildiririm. Çankırı Karatekin Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca gereğinin yapılmasını arz ederim (26/01/2022).

Ayşe DOĞRU

ÖZET

Yüksek Lisans Tezi

DİYABET TESPİTİNDE MAKİNE ÖĞRENMESİ ALGORİTMALARI YAKLAŞIMINI KULLANARAK YAPILMIŞ BİR ÇALIŞMA

Ayşe DOĞRU

Çankırı Karatekin Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Murat ARI
Eş Danışman: Dr. Öğr. Üyesi Selim BUYRUKOĞLU

Klasik yöntemlerle diyabet hastalığının erken tanı konulmasında ve teşhis edilmesinde kullanılan belirti ve bulgular her zaman yeterli olmayabilir. Bu tez çalışmasında, makine öğrenmesi algoritmalarını kullanarak diyabet hastalığına tanı ve teşhis konulmasında klasik yöntemler dışında farklı bir bakış açısı geliştirilmeye çalışılmıştır. Çalışmada UCI Machine Learning Repository'den temin edilen Early-stage diabetes risk prediction veri kümesi kullanılmıştır. Veri seti 320'si diyabet hastası, 200'ü diyabet hastası olmayan 520 hasta kaydından oluşmaktadır. Hasta profil bilgileri 16 öznitelik ve diyabetli ve diyabetli olmadığını gösteren bir sınıf bilgisinden oluşmaktadır. Veri seti eğitim (%70) ve test (%30) veri seti olmak üzere iki kısma ayrılmıştır. Veri setinden daha iyi performans alabilmek için veri ön işleme adımları uygulanmıştır. Öznitelik seçim tekniklerinden ki-kare yöntemi kullanılarak 9 özellik (Polydipsia, Polyuria, Sudden weight loss, Partial paresis, Gender, Irritability, Polyphagia, Alopecia, Age) seçilmiştir. Doğruluk performansını arttırmak için 10 kat çapraz doğrulama yapılmıştır. Modelimizin sınıflandırma performans kalitesini arttırmak için doğruluk, kesinlik, duyarlılık, F1-skor ve ROC eğrisi doğrulama metrikleri kullanılmıştır. Tez çalışmasındaki amacımız, diyabet hastalığını tahmin edebilmek için tek tabanlı makine öğrenmesi algoritmaları (Lojistik Regresyon, Destek Vektör Makineleri, K-En Yakın Komşu, Naif Bayes ve Yapay Sinir Ağları) ve topluluk makine öğrenmesi (Karar Ağacı, Rastgele Orman, AdaBoost, Gradyan Arttırma, XGBoost, Oylama (Voting), Super Learner) algoritmaları ile doğruluk performanslarını karşılaştırmaktır. En yüksek doğruluk tahmini %99,4 değer ile Super Learner topluluk öğrenme algoritmasıyla elde edilmiştir. Bu çalışmada, Early-stage diabetes risk prediction veri setini kullanarak diyabet hastalığının tahmininde topluluk öğrenmesi algoritmalarının tek tabanlı makine öğrenmesi algoritmalarından daha iyi kabul edilebileceği anlamına gelmektedir. Bu yöntemle hekime yardımcı olması amacıyla diyabet hastalığının erken teşhis edilmesi hedeflenmiştir.

2022, 116 sayfa

ANAHTAR KELİMELER: Diyabet, Tahmin, Makine öğrenmesi, Topluluk öğrenmesi, Rastgele orman, Super learner, Torbalama, Arttırma, Yığınlama

ABSTRACT

Master of Science Thesis

A STUDY USING THE APPROACH OF MACHINE LEARNING ALGORITHM IN DETECTION OF DIABETES

Ayşe DOĞRU

Çankırı Karatekin University
Graduate School of Natural and Applied Sciences
Department of Electrical-Electronics Engineering

Advisor: Prof. Dr. Murat ARI
Co-Advisor: Asst. Prof. Dr. Selim BUYRUKOĞLU

The signs and symptoms used in the early detection and diagnosis of diabetes with classical methods may not always be sufficient. In this thesis, it has been tried to develop a different perspective other than classical methods in the detection and diagnosis of diabetes by using machine learning algorithms. The Early-stage diabetes risk prediction dataset obtained from UCI Machine Learning Repository was used in the study. The dataset consists of 520 patient records, 320 of whom are diabetic and 200 are not. The patient profile information consists of 16 attributes and class information indicating diabetic and non-diabetic. The data set is divided into two parts as training (70%) and test (30%) dataset. Data preprocessing steps were applied to get better performance from the dataset. Nine features (Polydipsia, Polyuria, Sudden weight loss, Partial paresis, Gender, Irritability, Polyphagia, Alopecia, Age) were selected using the chi-square method, one of the feature selection techniques. To improve the accuracy performance, 10-fold cross-validation was performed. Accuracy, precision, sensitivity, F1-score, and ROC curve validation metrics were used to improve the classification performance quality of our model. Our aim in the thesis study is to compare the accuracy performances of single-based machine learning algorithms (Logistic Regression, Support Vector Machines, K-Nearest Neighbor, Naive Bayes, and Artificial Neural Networks) and ensemble machine learning algorithms (Decision Trees, Random Forest, AdaBoost, Gradient Boosting, XGBoost Voting, Super Learner) which are used to predict diabetes. The highest accuracy estimate was obtained with the Super Learner ensemble learning algorithm, with a value of 99,4%. This study means that ensemble machine learning algorithms can be considered better than single-based machine learning algorithms in predicting diabetes by using the Early-stage diabetes risk prediction dataset. This method is aimed at early detection of diabetes to assist the clinics.

2022, 116 pages

Keywords: Diabetes Mellitus, Prediction, Machine learning, Ensemble learning, Random forest, Super Learning, Bagging, Boosting, Stacking

ÖNSÖZ VE TEŞEKKÜR

"Diyabet Tespitinde Makine Öğrenmesi Algoritmaları Yaklaşımını Kullanarak Yapılmış Bir Çalışma" isimli bu tez çalışması 2019-2022 yılları arasında hazırlanarak Çankırı Karatekin Üniversitesi Fen Bilimleri Enstitüsü'ne "Yüksek Lisans" tezi olarak sunulmuştur.

Yüksek Lisans tez çalışmam ve araştırmalarım sırasında bana sağladığı destek, göstermiş olduğu sabır ve sahip olduğu tecrübelerle beni aydınlatan, çalışma alanımda beni yönlendiren tez danışmanım Prof. Dr. Murat ARI'ya teşekkürlerimi sunarım.

Yüksek Lisans tez çalışmam sırasında farklı fikir ve bakış açılarıyla beni yönlendiren, sahip olduğu bilgi ve tecrübelerle karşılaştığım zorlukların üstesinden gelmeme yardımcı olan ve sürekli desteğini benden esirgemeyen eş tez danışmanım Dr. Öğr. Üyesi Selim BUYRUKOĞLU'na teşekkürlerimi sunarım.

Tez çalışmam süresince gösterdikleri sevgi, destek, anlayış ve fedakarlıklarından dolayı eşim Sadettin ve çocuklarım Yusuf Kadir ve Muhammed Aras'a çok özel teşekkürlerimi sunarım. Ayrıca her daim sevgi ve desteklerini esirgemeyen annem Şengül SÜRÜN, babam Halit SÜRÜN'e ve kardeşlerime en içten sevgi, saygı ve teşekkürlerimi sunarım.

Ayşe DOĞRU
Çankırı, Ocak 2022

İÇİNDEKİLER

KISALTMALAR DİZİNİ

AB	AdaBoost
ANN	Yapay Sinir Ağları (Artificial Neural Network)
BC	Harmanlama Sınıflandırıcı (Blending Classifier)
CV	Çapraz Doğrulama (Cross Validation)
DM	Diyabet Hastalığı (Diabetes Mellitus)
DS	Karar Kütüğü (Decision Stump)
DT	Karar Ağaçları (Decision Tree)
FN	Yanlış Negatif (False Negative)
FP	Yanlış Pozitif (False Positive)
GADA	Glutamik Asit Dekarboksilaz Antikor
GB	Gradyan Arttırma (Gradient Boosting)
GDM	Gestasyonel Diyabet Hastalığı (Gestasyonel Diabetes Mellitus)
IAA	İnsülin Otoantikoru (Insulin Auto Antibody)
ICA	Adacık Hücre Antikoru (Islet Cell Antibodies)
IDF	Uluslararası Diyabet Federasyonu (International Diabetes Federation)
KNN	K-En Yakın Komşu (K-Near Neighbours)
LADA	Erişkin Latent Otoimmün Diyabet
LR	Lojistik Regresyon (Logistic Regression)
ML	Makine Öğrenmesi (Machine Learning)
NB	Naif Bayes (Naive Bayes)
PIDD	Pima Hint Diyabet Veri seti (Pima Indian Diabetes Dataset)
SC	Yığınlama Sınıflandırıcı (Stacking Classifier)
SL	Süper Öğrenci (Super Learner)
SVM	Destek Vektör Makineleri (Support Vector Machines)
TN	Doğru Negatif (True Negative)
TP	Doğru Pozitif (True Positive)
UCI	Univercity of California, Irvine
XGBoost	Aşırı Gradyan Arttırma (eXtreme Gradient Boosting)
WHO	Dünya Sağlık Örgütü (World Health Organization)

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

1. GİRİŞ

Diabetes Mellitus küresel çapta hızla ilerleyen insanın yaşam kalitesini olumsuz yönde etkileyen ciddi bir metabolik hastalıktır. Uluslararası Diyabet Federasyon'u (IDF) ve Dünya Sağlık Örgütü'nün (WHO) paylaşmış olduğu bilgilere baktığımızda son on yılda hastalığın gelişim sürecinin hızlandığı görülmektedir. IDF'nin 2021 yılında açıklamış olduğu verilerde şu anda teşhis koyulan 537 milyon diyabetli kişi olduğu bilinmektedir. Bu rakamın 2030 yılında 643 milyona yükseleceği ve 2045 yılına kadar da 784 milyon ulaşacağı varsayılmaktadır (Diabetes Atlas 2021). Diğer bir deyişle her 10 kişiden 1'ine diyabet teşhisi koyulmaktadır. Tanı koyulan hastaların yaş aralığı, orta yaşlı ve yaşlı hasta iken şuan yaş aralığı (20-79) gençlere doğru gittikçe düşmeye başlamıştır. 2021 yılında alınan istatiksel raporlara göre 6,7 milyon kişinin ölüm nedeni diyabet olduğu bilinmektedir (Diabetes Atlas 2021).

Diabetes Mellitus, günlük hayatta şeker hastalığı olarak tabir ettiğimiz, vücuttaki pankreas salgı bezi tarafından üretilen insülin hormonunun görevini yerine getirememesi, yeterli miktarda salgılanmaması veya eksikliğinden kaynaklanan kronik hastalıklardan biridir. İnsülin hormonu kanda bulunan glikozu (şeker) hücre, doku ve organlara taşıyarak vücudun ihtiyacı olan enerji ihtiyacını karşılamış olur. Fakat insülin hormonu bu görevi yerine getiremediği zaman kandaki glikoz seviyesi normal düzeyinin üstüne çıkmaya başlamasıyla birlikte hiperglisemi ortaya çıkmış olur (KI 2007).

Hipergliseminin başlıca belirtileri ise vücuttan sık sık idrar atma (poliüri), aşırı susuzluk (polidipsi), halsizlik, aşırı yemek tüketmek (polifaji), ani kilo kaybı, görsel bulanık, genital pamukçuk ve kaşıntı olarak sıralanabilir (Gavin *et al.* 2003). Uzun vadede ise böbrek yetmezliği (nefropati), periferik nöropati, kardiyovasküler, gastrointestinal, ayak ülseri, görme yetisini kaybetme (retinopati) gibi ciddi hastalıkların ortaya çıkmasına sebep olmaktadır (Horton and Barrett 2021).

Bu bölümün geri kalan kısmında Diyabetin Etyolojik Sınıflandırması (1.1), Diyabette tanı kriterleri (1.2), Diyabet tanısının önemi (1.3), Amaç ve Hedefler (1.4), Araştırma

Soruları (1.5), Önerilen Modelin Özeti (1.6), Literatüre Katkısı (1.7), Tez Çalışmasının Şeması (1.8) hakkında bilgi verilecektir.

1.1 Diyabet'in Etyolojik Sınıflandırması

Diyabet'in sınıflandırılmasında 4 klinik tip bulunmaktadır. Primer (tip-1, tip-2 ve gestasyonel diyabet) ve sekonder diyabet formları olmak üzere diyabet'in etiyolojik sınıflandırılması Çizelge 1.1'de verilmiştir (Türkiye Diyabet Vakfı 2021).

Çizelge 1.1 Diabetes mellitus'un etyolojik sınıflanması (Türkiye Diyabet Vakfı 2021)

1.Tip-1 Diyabet Mellitus A. İmmün aracılı B. İdiyopatik
2.Tip-2 Diyabet Mellitus
3.Gestasyonel Diyabet Mellitus
4.Diğer Spesifik Tipler A. Beta hücre fonksiyonunun genetik defektleri B. İnsülin etkisinde genetik defektler C. Ekzokrin pankreas hastalıklar D. Endokrinopatiler E. İlaç ve kimyasal maddelerle oluşan diyabet F. İnfeksiyonlar G. İmmün ilişkili diyabetin sık olmayan formları H. Diyabetle birlikte görülebilen diğer genetik sendromlar

1.1.1 Tip 1 diyabet

Diyabetli hasta sayısının %5'i ile %10'unu tip 1 diyabet hastaları oluşturmaktadır (Diabetes Atlas 2019). Genellikle 30 yaş altı kişilerde ve çocuklarda görülme olasılığı fazla olması rağmen her yaş grubundan insanın yaşamını etkileyebilir. Pankreas salgı bezi tarafından salgılanan insülin hormonunun vücuttaki ihtiyacı karşılayacak düzeyde

olmaması ya da hiç üretilmediği otoimmün bir hastalık türüdür. Bağışıklık sisteminde vücudun insülin üretimini yapan beta hücrelerini tahrip etmesiyle ortaya çıkar. Tanı koyulduktan sonra insülin kullanımı gerekmekte ve tedavinin devam ettirilmesi önem arz etmektedir (Katsarou *et al.* 2017).

Tip-1A (immün aracılı) diyabet türü genetik olarak diyabete yatkın olan bireylerde çevresel faktörlerin de etkisiyle vücutta beta hücre yıkımı başlar. Bunun etkisiyle diyabet belirtileri görülmeye başlar. İmmün aracılı diyabet olduğunda kanda adacık hücre antikorları (ICA), ICA-512, İnsülin Otoantikoru (IAA) olarak bilenen proteine, Glutamik Asit Dekarboksilaz Antikora (GADA) rastlanır (Anteby *et al.* 2021). Yetişken bireylerde Latent Autoimmune Diabetes of Adults (LADA) olarak bilenen beta hücrelerinin yavaş ilerleyen yıkımı şeklinde bir süreç gözlemlenebilir. Çoğu zaman Tip-2 diyabet türü ile karıştırılıp, yanlış teşhis konulmaktadır (Sugihara *et al.* 2021).

Tip-1B (İdiyopatik) diyabet türünde ise kanda herhangi bir beta hücre yıkımı olduğuna dair adacık antikorları bulunmaz. Otoimmün dışında kalan insülin eksikliğinde ortaya çıkan bir diyabet türüdür (Balasubramanian *et al.* 2003) .

1.1.2 Tip 2 diyabet

Diyabetli hasta sayısının %90'ının oluşturduğu çok yaygın bir diyabet türüdür. Vücutta salgılanan insülin hormonunun düzgün kullanılmadığında karşılaşılan durum Tip-2 diyabet türüdür. İnsüline karşı vücut bir direnç oluşturur. Buna karşın vücutta beta hücrelerinde hasar meydana gelir. Bunun sonucu olarak ilk önce tokluk kan şekeri ilerleyen zamanlarda da açlık kan şekerinde glikoz seviyesi değişime uğrar (Bailey *et al.* 2018, Evert *et al.* 2019). Tip-2 diyabet hastalarında tedavi yöntemi olarak fiziksel aktivelerinin arttırılması, sağlıklı besinler tüketerek diyet yapılması ve ilaç alımı ile hastalık kontrol altına alınmalıdır. Ancak değişen yaşam tarzlarıyla birlikte düzensiz beslenme, fazla kalorili besinlerin tüketiminin artması, hareketsizlik ve stresle beraber tip-2 diyabet hasta sayısında artış meydana gelmektedir. Yaş grubu olarak orta ve yaşlı gruplarda daha sıklıkla görülmesine karşın gençler ve çocuklarda da bu hastalık görülmeye başlamıştır (Henry and Hague 2015).

1.1.3 Gestasyonel diyabet

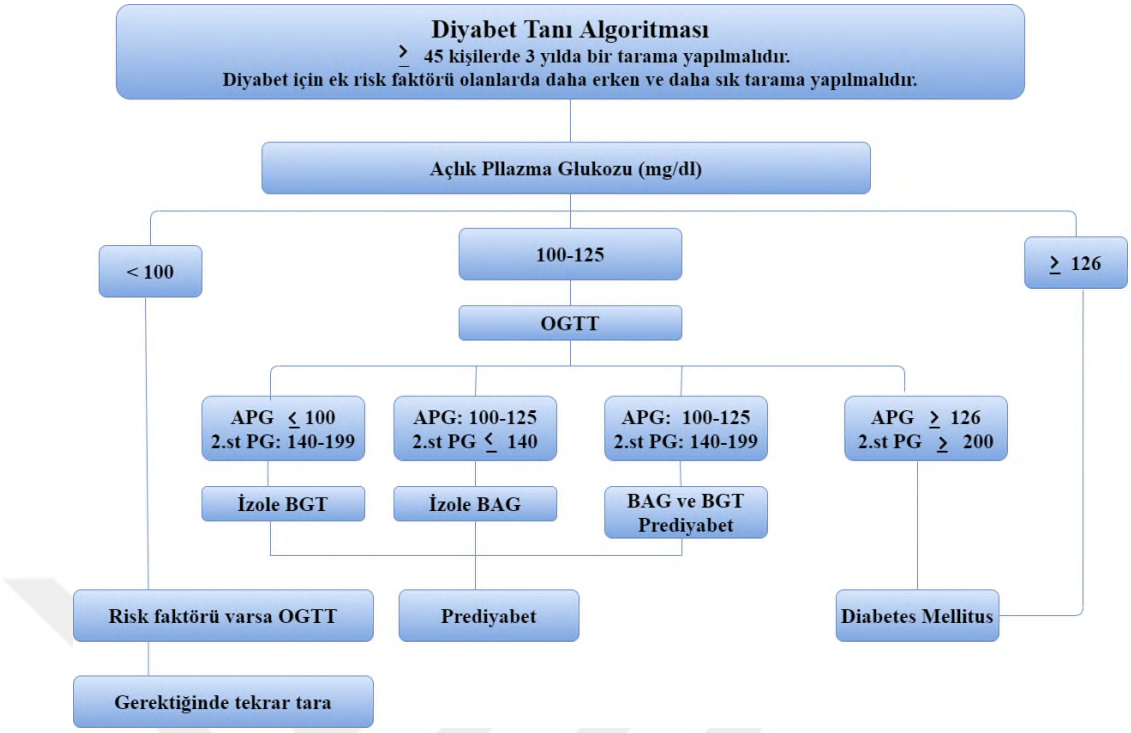
Hamilelik döneminde insülin hormonunun yeterli düzeyde salgılanamamasından dolayı ortaya çıkan bir durumdur. Hem anne ve hem de bebek için dikkat edilmesi gereken bir dönemdir. Ailede diyabet hastalığı geçmişi varsa ya da fazla kilo alımı olduysa riskli bir hamilelik dönemini geçirirler. Hamilelik sona erdikten sonra çoğunlukla diyabet ortadan kaybolur fakat anne ve bebek bazen etkilenebilir ve ilerleyen zamanlarda tip-2 diyabet olarak karşılarına çıkabilir (Diabetes Atlas 2019).

1.2 Diyabette Tanı Kriterleri

Klasik klinik yöntemlerle hastadan alınan kan örneğinden glikoz toleransına, açlık kan şekerine veya tokluk kan şekerine bakılarak diyabet tanısı koyulabilir (Association 2010). Diyabet tanısı koyarken Türkiye Diyabet Vakfi'nın hazırlamış olduğu Diyabet Tanı ve Tedavi Rehberi 2021 verilerine göre aşağıdaki adımlar uygulanır:

- Açlık plazma glukozu (APG) ölçebilmek için en az sekiz saat önceden gıda alımı yapılmamalıdır. APG değerinin 126 mg/dl yüksek bir değer olması,
- Rastlantısal plazma glukozu ile ölçüm yapabilmek için gıda alınıp alınmadığına bakılmadan günün herhangi bir saat diliminde yapılabilir. Bu değerin de 200 mg/dl yüksek olması,
- Oral glukoz tolerans testi (OGTT) için de 75 gr glukoz alımı yapılır. İki saat sonra tekrar 75 gr glukoz alarak test yapılır. Bu değerin de 200mg/dl yüksek olması,
- HbA1c değerinin de %6,5'ten yüksek olması durumunda diyabet tanısı konulabilir (Türkiye Diyabet Vakfı 2021).

APG'nin 100 mg/dl ile 126 mg/dl aralığında, OGTT'nin ise ikinci saat sonunda 140 mg/dl ile 199 mg/dl aralığında olması ise gizli şeker hastalığı (pre-diyabet) olduğunu sinyalini verir, takip edilmesi gereken bir dönemdir. Diyabet tanı koyma algoritması Şekil 1.1'de sunulmaktadır.



Şekil 1.1 Diyabet tanı algoritması (Türkiye Diyabet Vakfı 2021)

1.3 Diyabet Tanısının Önemi

Diyabet tanısı koyulan hastalar ömürleri boyunca bu hastalıkla ile yaşarlar. Hastalığın tanısı erken dönemlerde koyulamadığı için hastalık çok sinsi bir şekilde ilerlemeye devam eder. Diyabet teşhisi yapılmayan birçok diyabetli hasta bulunmaktadır. Çünkü hastalığın belirtilerinin ortaya çıkması uzun zaman alır.

Diyabet evreleri olan bir hastalıktır. Bireyin aile geçmişinde genetik yatkınlığı varsa ya da düzensiz beslenme, hareketsizlik gibi bir yaşam tarzı varsa ilk önce riskli grupta yer alır. İlerleyen zamanlarda gizli diyabet olarak karşımıza çıkar. Daha sonra ise diyabet teşhisi yapılır. Diyabetli hasta tedaviye uymalı, kontrollerini yaptırmalı ve günlük hayatta buna uygun yaşamalıdır. Yoksa ilerleyen zamanlarda diyabet diğer organlara da hasar vermeye başlar ve çeşitli hastalıkların ortaya çıkmasına neden olur. Bu nedenlerden dolayı diyabette erken tanı konulması önem kazanmaktadır (Aronson 2008, Cole and Florez 2020).

Diyabet hastalığının erken evrelerde teşhis edilmesi hastalığın ilerlemesini yavaşlatır ve bireylerin yaşam kalitelerini arttırmış olur. Hastaların rutin kontrollerinde ve fiziki muayenelerinde temin edilen veriler, makine öğrenmesi algoritmaları kullanılarak diyabetin erken teşhis edilmesine yardımcı olur. Bu yöntem klasik yöntemlere göre daha hızlı yanıt verir ve maliyeti düşürdüğü için tercih sebebi olmaya başlamıştır. Yöntem hekimlere bireyler hakkında ön referans bilgi sağlamış olur (Alghamdi *et al.* 2017, Kavakiotis *et al.* 2017).

1.4 Amaç ve Hedefler

Bu çalışmanın amacı, klinik verilerinden elde edilen veri setiyle makine öğrenmesi algoritmalarını kullanarak, diyabet teşhisinin konulması ve sınıflandırmasında en iyi doğruluk tahmin performansı gösteren topluluk öğrenme modellerini belirlemektir. Bu amaç doğrultusundaki hedefler aşağıda belirtilmiştir:

- Veri setinin elde edilmesi,
- Veri setinde veri ön işleme adımlarının uygulanması,
- Veri setinde özellik seçim yöntemlerinin uygulanması,
- Modelin doğruluğunu arttırmak için k-kat çapraz doğrulama tekniğinin uygulanması,
- Tek (single) tabanlı makine öğrenmesi algoritmaları ve topluluk (ensemble) öğrenmesi modellerinin belirlenmesi,
- En hassas doğruluk performansı sonucu alabileceğimiz optimum modelin belirlenmesi,
- Modellerin sonuç performansının elde edilmesi ve karşılaştırılma yapılmasıdır.

1.5 Araştırma Soruları

Bu çalışmanın temel amaç ve hedefleri doğrultusunda 5 farklı araştırma sorusu hazırlanmıştır;

1. Diyabet hastalığının tespit edilmesinde torbalama topluluk (bagging ensemble) makine öğrenmesi modelleri, tek tabanlı (single base) makine öğrenmesi algoritmalarından daha iyi performans sağlayabilir mi?
2. Diyabet hastalığının tespit edilmesinde arttırma topluluk (boosting ensemble) makine öğrenmesi modelleri, tek tabanlı (single base) makine öğrenmesi algoritmalarından daha iyi performans sağlayabilir mi?
3. Diyabet hastalığının tespit edilmesinde yığınlama topluluk (stacking ensemble) makine öğrenmesi modelleri, tek tabanlı (single base) makine öğrenmesi algoritmalarından daha iyi performans sağlayabilir mi?
4. Diyabet hastalığının tespit edilmesinde yığınlama topluluk (stacking ensemble) makine öğrenmesi modelleri, torbalama topluluk (bagging ensemble) makine öğrenmesi modelleri daha iyi performans sağlayabilir mi?
5. Diyabet hastalığının tespit edilmesinde yığınlama topluluk (stacking ensemble) makine öğrenmesi modelleri, arttırma topluluk (boosting ensemble) makine öğrenmesi modelleri daha iyi performans sağlayabilir mi?

1.6 Önerilen Modelin Özeti

Çalışmada kullandığımız diyabet veri seti UCI Machine Learning Reporsitory'den ücretsiz olarak temin edilmiştir. Elde ettiğimiz veri setine veri ön işleme adımları uygulanmıştır. Daha sonra veri setimiz eğitim (%70) ve test veri (%30) seti olarak iki kısma ayrılmıştır. Modelimize (5 ve 10) kat çapraz doğrulama yöntemleri uygulanmıştır. Eğitim veri setine hem tek tabanlı (single base) hem de topluluk (ensemble) tabanlı makine öğrenmesi modelleri eğitime tabi tutulmuştur. Modelin performans değerlendirmesini yapmak için test veri setinde doğruluk, kesinlik, duyarlılık ve F1-Skor'a göre tahmin performanslarını analiz edilip, değerlendirip sonuçlar karşılaştırılmıştır. Super Learner topluluk öğrenmesi modeliyle %99,4 değerinde en iyi doğruluk tahmini elde edilmiştir. Özetle, topluluk öğrenmesi modellerinin, tek tabanlı makine öğrenmesi algoritmalarından daha iyi tahmin performansı gösterdiği ortaya çıkmıştır.

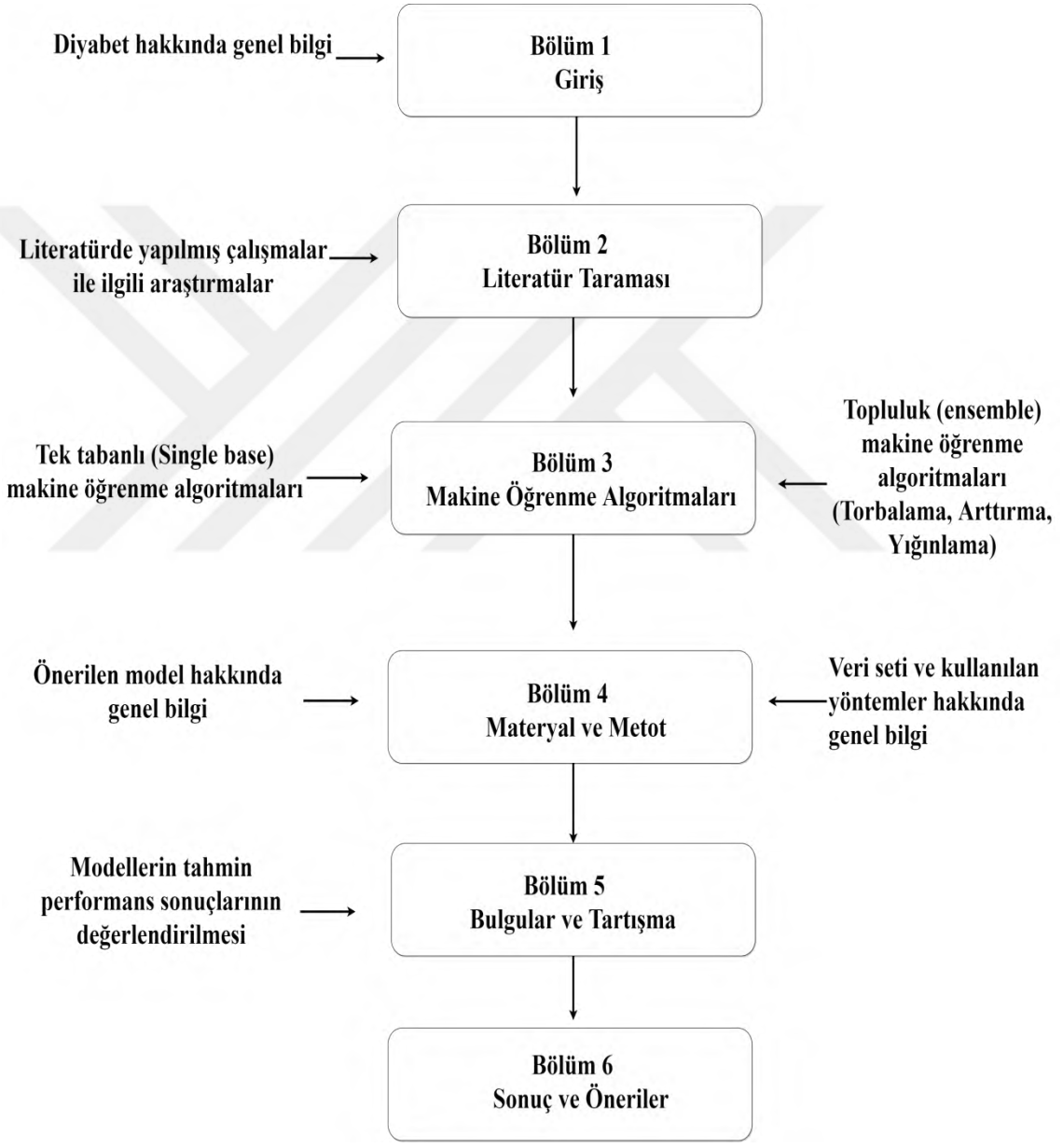
1.7 Literatüre Katkısı

Bu tez çalışmasının literatüre katkısı; Super Learner topluluk makine öğrenmesi modelini kullanarak, diyabet hastalığına tanı ve teşhis konulmasında kolaylık sağlamasıdır. Tahmin performansındaki doğruluğu arttırmak için Seviye-0'da farklı makine öğrenmesi algoritmalarının birden fazla kombinasyonları kullanılmıştır. Optimum modeli belirleyebilmek için hem 5-kat çapraz doğrulama ve hem de 10-kat çapraz doğrulama yapılarak sonuçlar gözlemlenmiştir. Seviye-1'de meta learner olarak genellikle Lojistik Regresyon makine öğrenmesi algoritması kullanılmıştır. Fakat bu çalışmada 6 farklı (Lojistik Regresyon, Rastgele Orman, Karar Ağacı, K-En Yakın Komşuluk, Ekstra Karar Ağaçları, Destek Vektör Makineleri) makine öğrenmesi algoritması kullanarak modelin performans sonuçlarını değerlendirilmiştir. Meta learner olarak Destek Vektör Makineleri algoritmasını seçtiğimiz de daha iyi tahmin performans sonucu elde edilmiştir. Bu çalışmalar sonucunda optimum makine öğrenmesi modelini bulmak için Super Learner topluluk öğrenmesi modelinin diğer makine öğrenmesi algoritmalarından daha avantajlı olduğu söylenebilir. Çünkü hangi makine öğrenmesi algoritmasının hangi problemlerde daha iyi sonuç vereceğini daha önceden belirlemek zor bir süreçtir. Super Learner topluluk öğrenmesi modelinde ise bu seçimi model kendisi öğrendiği için bu adım daha kolay bir hale gelmiş olur. Sonuç olarak diyabet hastalığı tespiti için farklı Super Learner yapıları oluşturarak optimum Super Learner modelinin diyabet hastalığı teşhisinde kullanılması bu tez çalışmasının literatüre olan katkısıdır.

1.8 Tez Çalışması Akış Şeması

Tez çalışmamızın akış şeması Şekil 1.2'de sunulmuştur. Bölüm 1'de diyabet hakkında genel bilgiler verilmiştir. Bölüm 2'de literatürde makine öğrenmesi algoritmaları ile diyabet hastalığını teşhis etmek için yapılmış çalışmalar araştırılmıştır. Kullanılan teknikler incelenmiştir. Bölüm 3'te çalışmamızda kullanılan makine öğrenmesi algoritmaları ve öznitelik seçme teknikleri açıklanmıştır. Bölüm 4'te önerilen yöntem hakkında bilgi verilmiştir. Veri seti ve veri setine uygulanan veri ön işleme adımları ve doğrulama ve değerlendirme metrikleri anlatılmıştır. Çalışmada kullandığımız makine

öğrenmesi algoritmalarının çalışma prensibi anlatılmıştır. Bölüm 5'te önerilen modelde kullandığımız makine öğrenmesi algoritmalarının tahmin performanslarının karşılaştırılması yapılmıştır. Bölüm 6'da çalışmamızın sınırlarını, katkılarını ve bu çalışmadan çıkardığımız sonuçlar anlatılmaktadır. Son olarak gelecekte yapılacak çalışmalara yön gösterebilmek adına tavsiyeler içermektedir.



Şekil 1.2 Tez çalışması akış şeması

2. LİTERATÜR TARAMASI

Bu bölümde diyabet hastalığının teşhisi ile alakalı literatürde yapılmış çalışmalar incelenmiş olup, araştırmalar hakkında detaylı bilgi verilmiştir.

2.1 Makine Öğrenmesi Algoritmalarını Uygulayan Tahmin Modelleri

Makine öğrenmesi algoritmaları medikal alanda başta kronik hastalıklar (diyabet, tiroid, kalp vb.) olmak üzere kanser gibi birçok hastalığın erken teşhis edilmesi ve tanı konulmasında kullanılmaktadır (Nayeem *et al.* 2021, Qin e*t al.* 2020). Herhangi bir hastalığın tanı konulmasında geleneksel yöntemlerle yapılması mümkün olmayan veya sürecin uzamasına neden olan durumlarda makine öğrenmesi algoritmaları performanslarıyla dikkat çekmiştir.

Bu bölümün geri kalan kısmında diyabet hastalığı tespitinde kullanılmış makine öğrenmesi algoritmaları ve öznitelik seçim yönteminin tahmin performansına olan etkisi üzerine yapılan çalışmalar hakkında bilgi verilmiştir. Makine öğrenmesi algoritmaları tek ve topluluk tabanlı makine öğrenmesi modelleri olmak üzere iki alt bölümden oluşmaktadır.

2.1.1 Tek (single) tabanlı makine öğrenmesi modelleri

Literatürde makine öğrenmesi algoritmalarıyla diyabet gibi kronik hastalıkların erken teşhis edilmesinde kullanılan çeşitli uygulamalar bulunmaktadır (Kaur and Kumari 2019). Farklı yaklaşımlar uygulayarak hem maliyet açısından hesaplı hem de daha hızlı sonuç verebilen, performans düzeyi yüksek olan makine öğrenmesi algoritmaları kullanılmıştır. Diyabet hastalığı tespitinde önerilmiş olan tek tabanlı makine öğrenmesi algoritmalarını şu şekildedir; Lojistik Regresyon (LR), Naif Bayes (NB), Yapay Sinir Ağları (ANN), K-En Yakın Komşu (KNN), Destek Vektör Makineleri (SVM) gibi algoritmaları sayabiliriz. Makine öğrenmesi algoritmalarını kullanarak sağlık alanında yapılan çalışmalar şu şekildedir;

Gajendra Sharma ve Umesh Hengaju (2020) çalışmalarında Pima Indian Diabetes Dataset (PIDD) veri seti ile KNN, NB, Rastgele Orman (RF), SVM, Karar Ağacı (DT) makine öğrenmesi algoritmaları ile diyabet hastalığını teşhis edilmesini önermişlerdir. 10 kat çapraz doğrulama yapılmıştır. Kazanç Oranı (GR), Information Gain (IG), Gini Index (GI) öznitelik seçim yöntemleri uygulanmıştır. Doğruluk, kesinlik, F1-Puanı, ROC ölçüm metrikleri kullanılmıştır. Deney sonuçlarına göre doğruluk oranları NB (%76,30), DT (%73,82), KNN (%71,65), RF (%68,7), SVM (%65,10) alınmıştır. En iyi doğruluk değeri Naif Bayes makine öğrenmesi algoritmasıyla elde edilmiştir.

Ram ve Vishwakarma (2021) tarafından yapılan çalışmada ise PIDD veri seti ile diyabet hastalık tahmini için KNN, SVM, LR, Gaus Naif Bayes (GNB) ve RF makine öğrenmesi algoritmalarını kullanmışlardır. 10 kat çapraz doğrulama yapılmıştır. Özyinelemeli öznitelik eleme yaklaşımı (RFE) uygulanarak (BMI, Age, Glucose, Pregnancies, Diabetes Pedigree Function) 5 öznitelik en iyi olarak seçilmiştir. Bu çalışmadaki en iyi doğruluk değeri %85 ile Lojistik Regresyon makine öğrenmesi algoritmasıyla elde edilmiştir.

Sisodia ve Sisodia (2018) yaptıkları çalışmada PIDD veri seti ile diyabet tahmini için yapılan sınıflandırma işleminde GNB, SVM ve DT makine öğrenmesi algoritmalarını kullanılmışlardır. Bu üç algoritmanın da performans, doğruluk, kesinlik ve geri çağırma metrikleri ile sonuçları değerlendirilmiştir. Bu çalışmada en iyi tahmin performansı %76.30 değeriyle Naif Bayes makine öğrenmesi algoritmasıyla elde edilmiştir. Bu çalışmanın dezavantajı veri setinde seçilen öznitelik değerleri için hangi öznitelik seçim yönteminin uygulandığı hakkında bir bilgi bulunmamaktadır. Veri setinde veri ön işleme adımları yapılmamıştır. Tahmin kalitesinin performansını arttırmak için bu adımlar yapılsaydı daha iyi olabilirdi.

Alcalá-Rmz ve diğerleri (2021) çalışmalarında Medical Research Unit in Biochemistry tarafından tahsis edilen diyabet veri setinde lipit profil seviyesine göre makine öğrenmesi algoritmaları ile diyabet teşhisi konulmasını önermişlerdir. Diyabetli bir hastada lipit profili kolesterol, yüksek kolesterol (HDL), düşük kolesterol (LDL), trigliseritlerden oluşmaktadır (Ergin *et al.* 2013). LR, KNN, Yapay Sinir Ağları (ANN),

RF, DT makine öğrenmesi algoritmaları kullanılmıştır. Bu algoritmaların performansını ölçmek için ROC, doğruluk, özgüllük, duyarlılık metrikleri kullanılmıştır. Z-score yöntemi kullanılarak veriler üzerinde standart sapması 1 ve ortalaması 0 olan bir dağılım yaparak verileri normalize edilmesini sağlamışlardır. Veri setinde seçilen özellikler 6 adet giriş parametresi (Age, Gender, HDL, LDL, Cholesterol, Triglycerides) ve 1 çıkış parametresinden (Treatment (TX)) oluşmaktadır. Deney sonuçlarına göre en iyi doğruluk %72,9, duyarlılık %85,2 ve AUC %79,5 oranları ile LR makine öğrenmesi algoritmasından alınmıştır. En kötü performans ise %66,9 doğruluk oranıyla KNN makine öğrenmesi algoritmasından alınmıştır. Bu çalışmanın dezavantajı ise KNN algoritmasında komşu sayısı olarak k=16 olarak ayarlanmıştır. K değerinde herhangi bir artırıp, azaltma yapılmadan sonuç alınmıştır. Optimal olarak k'nın değeri (3,10) arasında ve tek sayı olarak seçilmesi sınıflandırma yapılırken eşitliği bozduğu için daha yararlıdır (Hall *et al.* 2008) .

Mujawar ve diğerleri (2021) çalışmalarında Swad Diabetic Care Centre'den diyabetli hastaların verileri toplanarak oluşturulmuş bir veri seti kullanmışlardır. Veri setinde 461 hasta kaydı bulunmaktadır. Veri setinde veri ön işleme adımları olarak eksik değerler için verilerin ortalaması alınarak doldurulmuştur, veri normalleştirilmesi veya veri ölçeklendirmesi için öznitelik değerleri 0 ve 1 aralığına alınmıştır. Veri seti eğitim (%85) ve test (%15) veri seti olarak ikiye ayrılmıştır. Yapay sinir ağları kullanılarak online bir sistemle diyabet teşhisi koymayı önermişlerdir. Yapay sinir ağı modelinde giriş katmanı ve gizli katmanda ReLU aktivasyon fonksiyonu, çıkış katmanında ise Sigmoid aktivasyon fonksiyonu kullanılmıştır. Parametre ayarları epoch:50, batch_size:40, optimizer: Adam, learn_rate: 0,1 momentum: 0,4 olarak ayarlanmıştır. Giriş katmanındaki girdi sayısı:5, gizli katmandaki nöron sayısı: 8, çıkış katmanında bir nöron sayısı bulunmaktadır. Yapılan deney sonuçlarına göre eğitim veri setindeki doğruluk oranı %95,52 iken test veri setindeki doğruluk oranı %94,20'dir. Bu çalışmanın dezavantajı model performans değerlendirmesinde hold-out tekniği kullanılmıştır fakat k-kat çapraz doğrulama yöntemi de kullanılsaydı tahmin performansında artış sağlanabilirdi. Veri setindeki hasta sayısının az olmasından dolayı daha büyük ve farklı veri setlerinde tahmin sonuçları gözlemlenmemiştir. Ayrıca başka makine öğrenmesi algoritmalarıyla performans karşılaştırılması yapılmamıştır.

Kowsher ve diğerleri (2019) çalışmalarında Tip-2 diyabet hastalığını önceden tespit edebilmek için (LR, Lineer Diskriminant Analizi (LDA), KNN, SVM, NB, RF ve ANN) 7 farklı makine öğrenmesi algoritmalarını kullanmışlardır. Veri seti 9483 diyabetli hasta kaydından oluşmaktadır. Veri seti eğitim (%80) ve test (%20) veri seti olarak ikiye ayrılmıştır. Veri setinde veri ön işleme yapılmıştır. Eksik değer kontrolü için ortalama yöntemi uygulanmıştır. Veri etiketleme işlemleri için de one hot encoding ile yapılmıştır. Özellik seçim yöntemi olarak korelasyon katsayısı kullanılmıştır. Fasting, 2 Hours after Glucose Load, Duration, BMI, High Cholesterols, Heart Diseases, Kidney Diseases öznitelikleri seçilmiştir. Boyut azaltmak için PCA yöntemi uygulanmıştır. Yapay sinir ağında 6 gizli katman ve çıkış katmanı olarak tek çıkışlı olan bir yapıdan oluşmaktadır. Gizli katman ve çıktı katmanı arasında softmax aktivasyon fonksiyonu kullanılmıştır. Aktivasyon fonksiyonu olarak ReLU kullanılmıştır. Epoch: 25 olarak verilmiştir. 10 kat çapraz doğrulama yapılmıştır. Yapılan deney sonuçlarına göre doğruluk oranı %94,7 ile Yapay Sinir Ağları en iyi sonucu vermiştir. Model değerlendirme yönteminde seçilen k-kat çapraz doğrulama sayısını artırıp, azaltarak performans değerlendirmesi yapılmaması bu çalışmanın dezavantajı olarak düşünebilir. Farklı ve daha büyük veri setlerinde uygulama yapılarak doğruluk tahmin sonuçlarına tekrar gözlemlenebilirdi.

Mareeswari ve diğerleri (2017) PIDD veri setini kullanarak yapılan bu çalışmada ise diyabet hastalığının olup olmadığı hakkında öneride bulunmuşlardır. Java ve Weka yazılımlarından faydalanarak KNN makine öğrenmesi algoritmasını kullanılmışlardır. 10 kat çapraz doğrulama uygulanmıştır. Parametre ayarlarında K=3 ve K=7 olarak ayarlanıp deney yapılmıştır. Yapılan deney sonuçlarına göre K=3 iken doğruluk oranı %72,65, K=7 iken doğruluk oranı %74,73 olarak alınmıştır. K değerini arttırdıkça doğru tahmin oranı da artmaya başlamıştır. Bu çalışmanın dezavantajı komşu sayısı k'nın (k=3, k=7) parametre olarak tanımlanırken dikkatli olunması gerekir. Çünkü daha büyük bir veri seti ile çalışıldığında tanımlanmış olan parametre değerleri algoritmanın çalışma performansını etkilemiş olacaktır. Daha büyük veri seti daha büyük bir alanı kapladığı için alan içinde tanımlı nokta sayısı değişikliğe uğrayacaktır (Taşcı and Onan 2017). Veri setinde özellik seçim yaklaşımlarının uygulanamaması bir dezavantaj olarak düşünebilir. Eğer yapılsaydı modelin performansında artış sağlanabilirdi. Farklı makine

öğrenmesi algoritmaları kullanılıp performans karşılaştırılması ile veri setine en uygun algoritma belirlenebilirdi.

Tek tabanlı makine öğrenmesi algoritmaları ile yapılmış çalışmalar Çizelge 2.1'de sunulmuştur. En iyi doğruluk tahmin performansını Kowsher ve diğerlerinin Yapay Sinir Ağlarını kullanarak %94,7 doğruluk oranıyla, en düşük performans ise Mareeswari ve diğerlerinin K-En Yakın Komşu makine öğrenmesi algoritmasını kullanarak %74,73 doğruluk oranıyla almışlardır.

Çizelge 2.1 Tek tabanlı makine öğrenmesi algoritmaları ile yapılmış çalışmalar

Yazarlar	**Yıl**	**Veri Seti**	**Algoritma**	**ACC**
Mareeswari ve diğerleri	2017	PIDD	KNN	0,7473
Sisodia ve Sisodia	2018	PIDD	NB	0,7630
Kowsher ve diğerleri	**2019**	**9483 veri seti**	**ANN**	**0,947**
Gajendra Sharma ve Umesh Hengaju	2020	PIDD	NB	0,7630
Ram ve Vishwakarma	2021	PIDD	LR	0.85
Alcalá-Rmz	2021	Medical Research Unit in Biochemistry	LR	0,795
Mujawar ve diğerleri	2021	Swad Diabetic Care Centre	ANN	0,942

ACC (Accuracy) : Tahmin performansındaki doğruluk değeri

2.1.2 Literatür taraması genel değerlendirme

Tek tabanlı makine öğrenmesi algoritması kullanarak modelden yüksek performans beklemek her zaman mümkün olmayabilir. Yapılmış çalışmalar bize gösteriyor ki bazen

veri seti değişkliği bazen parametre ayarlarında yapılan farklılıklar doğru tahmin performansını etkileyebilmektedir (Mareeswari *et al.* 2017). Sadece çapraz doğrulama yapmak bile model performansını değiştirebilmektedir (Kowsher *et al.* 2019, Mareeswari *et al.* 2017, Ram and Vishwakarma 2021). Bireysel olarak yapılan sınıflandırma işlemlerine kıyasla birden fazla makine öğrenmesi kullanarak elde edeceğimiz doğruluk tahmin oranlarını arttırabiliriz. Bu durumda topluluk öğrenme modelleri (ensemble learning models) kullanılabilir çünkü performansı artırmak için birden fazla makine öğrenmesi algoritmasının tahminlerini birleştirip ona göre bir tahminde bulunur. Farklı makine öğrenmesi algoritmalarının kullanılması çeşitlilik sağlayarak algoritmaların eksik yönlerinden kaynaklanan hata oranlarını düşürmemize olanak sağlamaktadır (Polikar 2012). Bölüm 2.1.3'te diyabet hastalığı teşhisi için önerilmiş olan topluluk makine öğrenmesi modelleri hakkında bilgi verilmiştir.

2.1.3 Topluluk (ensemble) tabanlı makinesi öğrenmesi modelleri

Hibrit veya topluluk öğrenmesi modellerinin tek bir modele kıyasla daha iyi sonuç vereceği birçok araştırmacı tarafından belirtilmiştir (Joshi and Alehegn 2017). Hem modeldeki varyansı azaltmak hem de modele olan bağlılığı azaltmak için alternatif başka bir yaklaşım ise topluluk (ensemble) öğrenmesi algoritmalarını kullanmaktır. Topluluk öğrenmesi modelleri ile birden fazla ve birbirinden bağımsız bir şekilde eğitilen modelleri birleştirerek elde edeceğimiz tahminlerimizin doğruluğunu artırabiliriz. Çoklu öğrenme algoritmalarını kullanan topluluk yaklaşımlarının, sınıflandırma doğruluğunu iyileştirmenin etkili bir yol olduğu görülmüştür (Polikar 2012).

Literatürde üç tane aktif olarak kullanılan topluluk öğrenmesi modeli bulunmaktadır (Bauer and Kohavi 1999). Bu topluluk öğrenmesi sınıf tekniklerini aşağıdaki gibi sıralayabiliriz.

- **Bagging (Torbalama):** Birkaç model birbirinden bağımsız olarak eğitilip, tahmin sonuçlarının ortalaması alınır. Bagging algoritması karar ağaçlarından oluşan bir

önyükleme topluluk öğrenmesi algoritmasıdır. Veri seti ilk önce alt kümelere ayrılır. Her alt model birbirinden bağımsız bir şekilde eğitilir. Daha sonra modellerden gelen tahminler birleştirilerek değerlendirmeye alınır (Breiman 1996). Bagging algoritmasında modeller homejen bir yaklaşımla (homogeneous approach) seçilir. En çok kullanılan Bagging topluluk öğrenmesi algoritmaları arasında Rastgele Orman, Torbalanmış Karar Ağacı, Karar Kütüğü (Decision Stump) makine öğrenmesi algoritmalarını sayabiliriz (Ho 1998). Modeller hakkında daha detaylı bilgi Bölüm 4.6.2'de verilmiştir.

- **Boosting (Yükseltme / Artırma):** Modeller sıralı bir şekilde eğitilip, tahmin sonuçlarındaki varyans azaltılmaya çalışılır. Boosting algoritması çok sayıda zayıf öğrenen bir araya getirerek daha güçlü bir öğrenen elde etmeyi amaçlamaktadır (Freund and Schapire 1997). En çok kullanılan Boosting topluluk öğrenmesi algoritmaları AdaBoost (AB), Ektra Gradyan Arttırma (XGBoost), Gradient Boosting (GB), LightGBM, CatBoost algoritmalarından oluşmaktadır. Modeller hakkında daha detaylı bilgi Bölüm 4.6.2'de verilmiştir.

- **Stacking (Yığma):** Her model ayrı ayrı tahminde bulunur. Stacking algoritmasında ise heterojen bir yaklaşım (heterogeneous approach) izlenir. Bu tahminler daha sonra bir meta learner sınıflandırıcısına girdi olarak alınır. Daha sonra tahminde bulunulur. Modeller hakkında daha detaylı bilgi Bölüm 4.6.3'te verilmiştir.

2.1.3.1 Torbalama (Bagging) ve Arttırma (Boosting) topluluk öğrenmesi algoritması ile yapılan çalışmalar

Literatürde diyabet hastalığı tespitinde kullanılan Torbalama ve Arttırma topluluk öğrenmesi modelleri bu bölümde sunulmuştur.

Vijayan ve Anjali (2016) çalışmalarında eğitim veri seti olarak UCI makine öğrenmesi deposundan elde edilen küresel veri kümesini kullanmışlardır. Doğrulama işlemi için test veri seti ise Kerala'da local olarak elde edilen veriler kullanılarak diyabet hastalığını

tahmin eden bir sistem önermişlerdir. AdaBoost (AB) ile birlikte DT, SVM, NB ve Karar Kütüğü (DS) olmak üzere 4 farklı makine öğrenmesi algoritmalarını uygulamışlardır. Deney sonuçlarına göre tek tabanlı makine öğrenmesi algoritmalarının performansına bakıldığında en iyi doğruluk tahmin performans oranı %79,68 ile SVM makine öğrenmesi algoritmasından sonuç alınmıştır. Fakat AdaBoost algoritmasıyla birlikte performanslarına bakacak olursak %80,729 oranında AB + DS modeli en iyi doğruluk oranını sağlamıştır. SVM algoritmasında ise bir değişiklik görülmemiştir. Bu çalışmanın dezavantajı KNN, ANN gibi diğer makine öğrenmesi algoritmalarının uygulanmaması ve farklı veri setlerinde test işlemlerinin uygulanmaması olabilir. Modelin doğruluğunun artırmak için çapraz doğrulama teknikleri kullanılabilir. Veri seti özellik seçim yaklaşımları uygulanabilir.

Perveen ve diğerleri (2016) tarafından CPCSSN diyabet veri seti ile AdaBosst (J48 Karar ağacı), Torbalama (J48 Karar ağacı) ve J48 karar ağacı algoritmalarını kullanarak diyabet hastalığını tahmin etmeyi önermişlerdir. Farklı yaş grupları ile diyabet ilişkisini araştırmak için diyabet hastalarını (18-35), (36-55), 55 yaş ve üzeri olmak üzere üç yaş grubuna ayırmışlardır. Veri seti eğitim veri seti (%60) ve test veri seti (%40) olarak iki kısma ayrılmıştır. Öznitelik seçim tekniği olarak Ki-Kare yöntemi kullanılmıştır. Deneylerde Weka programından yararlanılmıştır. ROC eğrisi performans sonuçlarına baktığımızda en iyi doğruluk performans %98 değerle Torbalama (J48 Karar ağacı) model ile elde edilmiştir. Genel olarak AdaBoost topluluk öğrenmesi modeli her üç yaş grubunda da iyi bir performans göstermiştir. Bu çalışmanın dezavantajı model doğrulama yöntemlerinde sadece hold-out tekniği kullanılıp, k-kat çaprazlama gibi doğrulamam teknikleri uygulanmamıştır.

Birjais ve diğerleri (2019) çalışmalarında UCI makine öğrenmesi deposundan PIDD diyabet veri seti ile diyabet hastalığının teşhis edilmesi için LR, NB ve GB makine öğrenmesi algoritmalarını kullanmışlardır. Üçlü çapraz doğrulama yapılmıştır. Deney sonuçlarına göre doğruluk oranları sırasıyla %86 GB, %77 NB ve %79 LR elde edilmiştir. Gradyan Arttırma topluluk öğrenmesi modeli diğer algoritmalara göre daha iyi doğruluk oranına sahip olmuştur.

Zou ve diğerleri (2018) çalışmalarında PIDD ve Luzhou diyabet veri seti ile Rastgele Orman, Karar Ağacı ve Yapay Sinir Ağları makine öğrenmesi algoritmalarını kullanarak diyabet hastalığının tahmin edilmesini amaçlamışlardır. Yapay Sinir Ağı iki katmanlı bir geri besleme ağı bulunan ve sigmoid gizli ve softmax çıkış nöronlarına sahiptir. Gizlilik katman sayısı 10 olarak ayarlanmıştır. Bu çalışmada modelleri incelemek için 5'li çapraz doğrulama ve boyutsallığı azaltmak için temel bileşen analizi (PCA) kullanılmıştır. Birbirinden bağımsız 5 test yapılmış ve sonuç bu deneylerin ortalaması olmuştur. Bu çalışmada doğruluk, duyarlılık, özgüllük ve korelasyon katsayısı ölçütleri kullanılmıştır. Deneylerin sonucunda Rastgele Orman makine öğrenmesi algoritması Luzhou veri setinde %80,84, PIDD veri setinde ise %77,21 ile en iyi doğruluk oranını ulaşmıştır.

Lai ve diğerleri (2019) tarafından diyabet hastalığının teşhisi için veri seti olarak CPCSSN'den (www.cpcssn.ca) elde edilen verileri kullanmışlardır. Veri seti eğitim (%80) ve test (%20) veri seti olarak ikiye bölünmüştür. LR, GB ve RF makine öğrenmesi algoritmaları kullanılarak 10 kat çapraz doğrulama yapılmıştır. Sonuçları karşılaştırmak için PIDD veri seti üzerinde çalışma yapmışlardır. CPCSSN'deki veri setinde yapılan deney sonuçlarına göre doğruluk oranları sırasıyla GB (%84,7), LR (%84,4), RF (%83,4) elde edilmiştir. PIDD veri setinde yapılan deney sonuçlarına göre doğruluk oranları sırasıyla GB (%85,1), RF (%85,5), LR (%84,6) elde edilmiştir.

Daghistani ve Alshammari (2020) yapmış oldukları çalışmada diyabet hastalığının tanı konulmasında Suudi Arabistan'da Ulusal Muhafiz Hastane İşleri Başkanlığı (MNGHA) veri setinden alınan örneklerle RF ve LR makine öğrenmesi algoritmaları ile çalışma yapmışlardır. F-1 puanı, kesinlik, geri çağırma, gerçek pozitif oran, yanlış pozitif oran, AUC ölçütlerini kullanmışlardır. Modeli doğrulamak amacıyla 10 kat çapraz doğrulama yöntemini kullanmışlardır. Bu çalışma sonucunda en iyi doğruluk performansı %94,4 değeriyle RF makine öğrenmesi algoritmasıyla elde etmiştir. LR makine öğrenmesi algoritmasından elde edilen sonuç ise %70,8'dir.

Buyrukoğlu (2021) çalışmasında ADNI veri setini kullanarak Alzheimer hastalığının erken teşhis edilmesinde homojen topluluk (Kazanç Oranı, ReliefF, Ki-Kare, FCBC) ve heterojen topluluk (Kazanç Oranı, ReliefF, Ki-Kare, FCBC) olmak üzere 2 ayrı özellik seçim yöntemini uygulamıştır. Eğitim verileri 4 bölüme ayrılmıştır. Bu eğitim verilerine eşik değerleri (%10 , %25 , %50) olarak uygulanmıştır. En iyi öznitelik seçimi için eşik değeri %25 ile oluşturulan özellik veri setinden sonuca ulaşılmıştır. Homojen topluluk için Ki-Kare özniteliği en iyi doğruluk oranı vermiştir. Heterojen topluluk için Bilgi Kazanımı, Ki-Kare, ReliefF öznitelikleri seçilmiştir. RF, LR, SVM, NB ve Multilayer Perception makine öğrenmesi algoritmalarını kullanmıştır. En iyi doğruluk oranı RF ve Homojen topluluk (Ki-Kare) özellik seçimi ile %88, RF ve Heterojen topluluk (Kazanç Oranı, Ki-Kare, ReliefF, FCBC) özellik seçimi ile %91 oranında sonuca ulaşmıştır.

Emon ve diğerleri (2021) tarafından diyabet hastalık risk tahmini için Sylhet Diyabet Hastanesi'nden alınan anket sonuçlarına göre oluşturulan veri setini kullanmışlardır. 11 farklı makine öğrenmesi algoritması ile birlikte bir çalışma yapılmıştır. Doğruluk, kesinlik, F-1 puanı, AUC ve geri çağırma ölçütleri uygulanmıştır. Bu çalışma sonucunda en iyi doğruluk performansı %98 değer ile Rastgele Orman makine öğrenmesi algoritmasıyla elde edilmiştir. Çalışmada kullanılan diğer makine öğrenmesi algoritmalarının sonuçları ise şu şekildedir; SVM (%96), AdaBoost (%94), Karar Ağacı (%93), Lojistik Regresyon (%92), MLP (%92), KNN %91, Gauss Süreci (%88), Naif Bayes (%84), Boosting (%9)'dir.

Yang ve diğerleri (2021) çalışmalarında diyabet hastalığının tahmini için Luzhou Belediye Sağlık Komisyonundan elde edilen verilerle veri setini oluşturmuşlardır. Üç farklı fizik muayene verisi birleştirilmiştir. Bu veri setine XGBoost, LR, RF makine öğrenmesi algoritmaları kullanılmıştır. Karşılıklı Bilgi (MI), Varyans Analizi (ANOVA), Gini Safsızlığı (GI) olmak üzere üç öznitelik seçme tekniğini uygulamışlardır. Bu çalışma sonucunda en iyi tahmin performansı makine öğrenmesi algoritması olarak XGBoost ve öznitelik seçimi olarak GI seçildiğinde %87,68 doğruluk oranı elde edilmiştir.

Shamreen Ahamed ve Sumeet Arya (2021) çalışmalarında PIDD diyabet veri setini kullanarak diyabet hastalığını tahmin edebilmek için makine öğrenmesi algoritmalarıyla bir model geliştirmişlerdir. Lojistik Regresyon, Karar Ağacı, Rastgele Orman, Gradyan Arttırma, XGBoost, Extra Karar Ağaçları, LGBM gibi makine öğrenmesi algoritmalarını kullanmışlardır. Deney sonuçlarına göre en iyi doğruluk tahmin performansını %95,2 değer ile LGBM algoritmasıyla elde edilmiştir.

Çizelge 2.2'de en iyi performans gösteren çalışmalar %98 doğruluk oranı ile iki çalışma bulunmaktadır. Birinci Perveen ve diğerlerinin (2016) Torbalama (J48 Karar ağacı) algoritmasını kullanarak yaptıkları çalışma, diğeri ise Emon ve diğerlerinin (2021) Rastgele Orman makine öğrenmesi algoritmasını uygulayarak elde etmişlerdir. En düşük performans oranı ise %77,2 ile Zou ve diğerleri (2018) Rastgele Orman makine öğrenmesi algoritmasını kullanarak elde etmişlerdir. Bir diğer düşük performans gösteren çalışma ise Vijayan ve Anjali (2016) %80,7 doğruluk oranı ile AdaBoost algoritmalarını kullanarak elde etmişlerdir. Bu çalışmalarda farklı ağaç sayıları ve diğer hiper parametre ayarlarında değişiklik yapılıp modelin performansı arttırılabilirdi.

Çizelge 2.2 Torbalama(bagging) ve arttırma (boosting) topluluk öğrenmesi ile diyabet hastalığı üzerinde yapılmış uygulamalar

Yazarlar	**Yıl**	**Veri Seti**	**Algoritma**	**ACC**
Vijayan ve Anjali	2016	Küresel ve local diyabet veri setleri	AdaBoost, Karar Kütüğü	0,807
Perveen ve diğerleri	**2016**	**CPCSSN**	**Torbalama**	**0,98**
Birjais ve diğerleri	2019	PIDD	Gradyan Arttırma	0,86
Zou ve diğerleri	2018	V1: PIDD, V2: Luzhou	Rastgele Orman	V1: 0,772 V2: 0,808
Lai ve diğerleri	2019	V1:PIMA, V2:CPCSSN	Rastgele Orman, Gradyan Arttırma	V1: 0,855 V2: 0,847
Daghistani ve Alshammari	2020	MNGHA	Rastgele Orman	0,944
Buyrukoğlu	2021	ADNI	Rastgele Orman	0,91
Emon ve diğerleri	2021	Sylhet Diyabet Hastanesi	Rastgele Orman	0,98
Yang ve diğerleri	2021	Luzhou	XGBoost	0,8768
Shamreen Ahamed ve Sumeet Arya	2021	PIMA	LGBM	0,952

2.1.3.2 Literatür taraması genel değerlendirme

Torbalama topluluk öğrenmesi modeli varyansı düşürmeye odaklanır, Arttırma topluluk öğrenmesi modelinde ise düşük önyargılı modeller elde etmeye çalışır. Yığınlama (Stacking) topluluk öğrenmesi modelleri, torbalama(bagging) ve arttırma (boosting) topluluk öğrenmesi modellerinden farklı olarak hem varyansı azaltmayı hem de önyargıyı azaltmayı hedefler. Bir diğer farkı ise heterojen bir yaklaşım izleyerek

birbirinden bağımsız ve farklı makine öğrenmesi algoritmalarını kullanarak birleştirme işlemi yapar (Kuhn and Johnson 2013) .

2.1.3.3 Yığınlama (Stacking) topluluk öğrenmesi algoritması ile yapılan çalışmalar

Yığınlama topluluk öğrenmesi modellerinin mimarisi, temel modeller ve meta modeli kullanarak farklı bir yaklaşım ile topluluk öğrenmesi modeli oluşturmuştur. İki seviye kullanılarak sınıflandırma işlemi yapılır. Seviye-0'da temel makine öğrenmesi modelleri bir tahminde bulunur. Seviye-1'de ise bu tahminler meta learner'a girdi parametresi olarak alınır ve tahminde bulunur (Brownlee 2021). Yığınlama topluluk öğrenmesi modelini kullanan algoritmalar Oylama (Voting), Ağırlıklı Ortalama (Weighted Avarage), Harmanlama (Blending), Yığınlama (Stacking), Süper Öğrenci (Super Learner) gibi makine öğrenmesi algoritmalarını sayabiliriz (Wolpert 1992). Detaylı bilgi Bölüm 4.2.3 verilmiştir. Yığınlama topluluk öğrenmesi modellerinin tekniğini kullanarak yapılan çalışmalardan aşağıda bahsedilmiştir.

Kumari ve diğerleri (2021) çalışmalarında PIDD diyabet ve meme kanseri veri setlerinde diyabet hastalığını erken bir aşamada tahmin etmeyi önermişlerdir. Önerilen modelde Yumuşak Oylama (Soft Voting) Sınıflandırıcısı ile Rastgele Orman, Lojistik Regresyon ve Naif Bayes makine öğrenmesi algoritmalarından oluşmaktadır. Modelin doğruluğunu değerlendirmek için Destek Vektör Makineleri, AdaBoost, Lojistik Regresyon, CatBoost, Gradyan Arttırma ve Torbalama (Bagging) gibi farklı makine öğrenmesi algoritmalarını kullanmışlardır. Doğruluk, kesinlik, AUC, F1-Puanı ve geri çağırma ölçütlerini uygulamışlardır. Deney sonuçlarına göre PIDD veri setinde en yüksek tahmin performansını Yumuşak Oylama Sınıflandırıcı ile %79,04 doğruluk, %73,48 kesinlik, %71,56 F1-puanı ve %80,6 AUC oranları alınmıştır. Meme kanseri veri setinde ise en iyi doğruluk performans oranı %97,02 Yumuşak Oylama sınıflandırıcı ile elde etmişlerdir.

PIDD veri seti ile yapılan bir başka çalışmada Taz ve diğerleri (2021) diyabet hastalığının tahmininde bulunmak için AdaBoost, XGBoost, LightGBM, Yumuşak

Oylama ve Rastgele Orman topluluk öğrenmesi modellerini kullanmışlardır. Veri seti eğitim (%70) ve test (%30) veri seti olarak ikiye bölünmüştür. PIDD diyabet veri setinde hem veri ön işleme yaparak hem de veri ön işleme yapılmadan performans değerlendirilmesini ölçmek istemişlerdir. Veri setinde eksik değerlerin tanımlanması için her bir özelliğin ortalama değerini vermişlerdir. Ayrıca normalizasyon ve veri temizleme gibi veri ön işleme adımlarını uygulamışlardır. Modelin doğruluğunu arttırmak için 10 kat çapraz doğrulama yapmışlardır. Doğruluk, kesinlik, F-1 puanı ve geri çağırma ölçütlerini uygulamışlardır. Yapılan deney sonuçlarına göre veri ön işlemeden sonra en iyi doğruluk tahmin performans oranı %94 ile LightGBM makine öğrenmesi algoritması tarafından elde etmişlerdir.

Alghamdi ve diğerleri (2017) çalışmalarında Henry Ford Sağlık Sistemlerindeki doktorlar tarafından egzersiz koşu bandı stres testi uygulanan ve 5 yıllık takip sonucunda diyabet tanısı koyulan hastaların kayıtlarını kullanmışlardır. Veri setinde 62 öznitelik bulunmaktadır. Çoklu Doğrusal Regresyon (MLR) ve Bilgi Kazanımı (IG) öznitelik seçimleri kullanılarak 13 öznitelik (Age, Resting Heart Rate, Metabolic Equivalent, Resting Systolic Blood Pressure, Resting Diastolic Blood Pressure, Sedentary Lifestyle, Black, Obesity, Hypertension, HR Achieved, Hyperlipidemia, Asprin, Family History of Premature Coronary Artery Disease) seçilmiştir. Veri setinde veri ön işleme olarak veriler “Evet”, ”Hayır” olarak etiketlemişlerdir. Modeldeki veri dengesizliğini önlemek amacıyla SMOTE (Sentetik Azınlık Aşırı Örnekleme Tekniği) kullanmışlardır. SMOTE tekniği ile azınlıkta olan sınıfa ait verilerin oluşturduğu küme sınır çizgisindeki örneklerin sayısı arttırılarak veri dengesizliği giderilmeye çalışılmaktadır. Model geçerliliği için 10-kat çaprazlama tekniğini kullanmışlardır. Random Forest, Logistic Regresyon, Naive Bayes üç makine öğrenmesi algoritmaları kullanarak Oylama yöntemi ile topluluk öğrenmesi yaklaşımını önermişlerdir. Deneyler sonucunda %92,2 doğruluk oranı, %91,3 F1-Puanı, %92 ROC, %76,8 Kappa, %74,7 Özgünlük oranları alınmıştır. Bu çalışmada daha farklı makine öğrenmesi algoritmaları kullanarak modelin tahminde bulunma kalitesinde artış sağlanabilirdi.

Yadav ve Pal (2021) UCI makine öğrenmesi deposundan alınan veri seti ile diyabet hastalığının özelliklerinin çeşitliliği üzerinde yaptıkları bu çalışmada kural tabanlı

makine öğrenmesi algoritmalarıyla (Karar Ağacı, Jrip, OneR) sınıflandırma yapmışlardır. OneR algoritmasında her sınıf için bir kural oluşturup, daha az hata ile en iyi kuralı seçmeye çalışmaktadır. Jrip algoritmasında ise sınıfın büyüklüğünü hesaplar ve veri setindeki hataları azaltmaya hedeflemektedir. Veri setinde veri ön işleme olarak eksik değerlerli veriler kaldırmışlardır. Öznitelik seçiminde Ki-Kare kullanılarak 9 tane öznitelik (serum_insulin, plasma_glucose, class, age, body_mass_index, times_pregnant, tricep_skin_fold_thickness, diastolic_blood_pressure, diabetes_pedigree_function) çıkarılmıştır. Her özelliğin ısı haritası ve korelasyon matrisi oluşturulmuştur. Doğruluk, kesinlik, F-1 puanı ve geri çağırma ölçütleri uygulanmıştır. Arttırmalı topluluk öğrenmesi modeli (Karar Ağacı, Jrip, OneR) ile torbalama topluluk öğrenmesi modeli (Karar Ağacı, Jrip, OneR) uygulanıp oylama (voting) ile sonuçlandırılmıştır. Deney sonucuna göre en iyi doğruluk performans %98 değeri Torbalama topluluk öğrenmesi modeliyle elde etmişlerdir. Bu çalışmada modelin performans kalitesini arttırmak için torbalama ve arttırma topluluk öğrenmesi modellerini oluştururken farklı makine öğrenmesi modelleri kullanarak çalışma yapılabilirdi.

Kabir ve Ludwig (2019) Diyabetik Retina veri seti (Messidor), Wiscoin Meme Kanseri veri seti (WBC), PIDD, Karaciger Hasta veri seti (Indian Liver Patient Dataset) olmak üzere 4 farklı sağlık veri seti üzerinde yapılan bu çalışmada Super Learning (SL) ve Yığınlama topluluk öğrenmesi modellerini uygulamışlardır. GB, RF, Derin Sinir Ağları (DNN) makine öğrenmesi algoritmalarını kullanmışladır. GB‘de kullanılan ağaç sayısı: 80, öğrenme hızı: [0.01,0.03,0.05,0.1], maksimum derinlik: [6,8,10]. RF’de kullanılan ağaç sayısı: 100, maksimum derinlik: [6,8,10]. DNN’de aktivasyon fonksiyonu: tanh, rectifier, maxout. Gizli katman sayısı: 50, eğitim tur sayısı: 20 olarak hiper parametreleri ayarlamışlardır. Meta learner olarak GLM (Generalized Linear Model) algoritmasını kullanmışlardır. Temel makine öğrenmesi algoritmaları, Yığınlama topluluk öğrenmesi modeli, SL2 (GBM-RF) iki temel makine öğrenmesi algoritması, SL3(GBM-RF-DNN) üç temel makine öğrenmesi algoritmasıyla yapılan ayrı ayrı deney sonuçlar şu şekildedir: Doğruluk oranlarına göre Messidor veri setinde en iyi performans SL3 ve DNN (%77,92) , WBC veri setinde en iyi performans SL3 ve DNN (%99,29), PIDD veri setinde en iyi performans SL3 ve RF (%81,17), ILPD veri setinde

en iyi performans SL3 (%71,80) ve DNN (%70,16) makine öğrenmesi algoritmasıyla elde etmişlerdir. AUC oranlarına göre Messidor veri setinde en iyi performans SL3 (%84,7) ve DNN (%83,8), WBC veri setinde en iyi en iyi performans SL3 ve SL2 (%99,8), PIDD veri setinde en iyi performans SL3 (%88,6) ve RF (%88,2), ILPD veri setinde en iyi performans SL3 ve DNN (%73,4), makine öğrenmesi algoritmasıyla elde etmişlerdir.

Akula ve diğerleri (2019) çalışmalarında PIDD diyabet veri seti ile Tip-2 diyabet hastalığını teşhis etmek için topluluk makine öğrenmesi modelini önermişlerdir. Veri setinde veri ön işleme olarak veri temizliği yapılmıştır. Doğruluk, kesinlik, duyarlılık, negatif tahmin değeri ve F-1 Puanı ölçüm metriklerini kullanmışlardır. Bu çalışmada diyabet hastası olmayan olarak etiketlenen hastanın testine bakmamışlardır. Diğer bir deyişle yanlış negatiften daha çok yanlış pozitif alınması konusunda daha hassasiyet göstermişlerdir. Önerdikleri modelde Seviye-0'da (Karar Ağacı, Rastgele Orman, Yapay Sinir Ağı, Destek Vektör Makineleri, Gradyan Arttırma, K-En Yakın Komşu ve Naif Bayes) 7 farklı makine öğrenmesi algoritmaları kullanmışlardır. Seviye-0'da tanımlanan algoritmaların sonucunu alıp, algoritmaya atanan ağırlıkla çarpıp sonucu belirlemek için de bir topluluk modeli oluşturmuşlardır. Bu topluluk modelindeki Seviye-1'de (SVM, RF ve NB) 3 farklı makine öğrenmesi algoritmalarından oluşturmuşlardır. Verileri test etmek içinde Practice Fusion veri setini kullanmışlardır. Deney sonucuna göre Ağırlıklı Ortalama Topluluk öğrenmesi modelinde (Weighted Average Ensemble) en iyi doğruluk performans değerleri Practice Fusion veri setinde %85 ve PIDD veri setinde %89 olarak elde etmişlerdir.

Deberneh ve Kim (2021) tarafından makine öğrenmesi algoritmalarını kullanarak Tip-2 diyabet tahmini yapmayı önermişlerdir. LR, SVM, RF, XGBoost, Yumuşak Oylama (SV) ve Yığınlama (ST) makine öğrenmesi algoritmalarını kullanmışlardır. Veri seti olarak Güney Kore Seul'deki Hanaro Tıp vakfından altı yıllık (2013-2018) bir tıbbı kayıt listesinden oluşturmuşlardır. Özellik seçim yöntemlerinden RFE, ANOVA testi, Ki-Kare tekniklerini kullanmışlardır. Ortaya çıkan öznitelikler (FBG, BMI, Age, Sex, Physical Activity, HbA1c, Gamma-GTP, Drinking, Triglycerides, Smoking, Uric Acid, Family History) 12 öznitelik elde etmişlerdir. 10 kat çapraz doğrulama yapmışlardır.

F1- Puanı, doğruluk, hassasiyet, geri çağırma değerlendirme ölçüm metriklerini kullanmışlardır. Çapraz doğrulama yaptıktan sonra topluluk öğrenme modelleri (Yumuşak Oylama ve Yığınlama), tekli modellerden daha iyi bir performans göstermişlerdir. Topluluk öğrenme modellerinin doğruluk tahmin performans sonuçları yaklaşık olarak %80 oranındadır.

Çizelge 2.3'te en iyi performans gösteren çalışma Kabir ve Ludwig'in (2019) WBC veri setinde %99,8 değer ile Super Learner topluluk öğrenmesi modelini kullanarak elde etmişlerdir. Bu doğruluk oranına en yakın bir diğer çalışma ise %98 değer ile Yadav ve Pal'ın (2021) Torbalama (Decision Tree, Jrip, OneR) topluluk öğrenmesi modeliyle yapmış oldukları çalışmadır. En düşük performans ise %79 oranı ile Kumari ve diğerleri'nin (2021) Yumuşak Oylama (Soft Voting) ile yaptığı çalışmadır. Yığınlama topluluk öğrenmesi modelinde oylama (voting) tekniği her zaman yüksek performans gösteremeyebilir. Oylama topluluk öğrenmesi modeli meta learner olarak seçilen makine öğrenmesi algoritmasının doğrusal olmakla sınırlı iken Yığınlama topluluk (stacking ensemble) öğrenmesi modelinde böyle bir sınırlama yoktur. Süper Öğrenci topluluk öğrenmesi modelindeki mimaride ise temel olarak k-katlı çapraz doğrulama üzerine inşa edilmiştir. Aşırı uyumu önlemek amacıyla çapraz doğrulama tekniğini kullanır (Van Der Laan *et al.* 2007).

Çizelge 2.3 Yığınlama topluluk (stacking ensemble) öğrenmesi ile diyabet hastalığı üzerinde yapılmış uygulamalar

Yazarlar	**Yıl**	**Veri Seti**	**Stacking Yöntemi**	**Algoritmalar**		**ACC**
				Seviye-0	**Seviye-1**	
Alghamdi ve diğerleri	2017	Henry Ford Sağlık Sis.	Oylama (Voting)	J48, NB, RF, LR	RF, LR ve NB	0,922
Akula ve diğerleri	2019	PIDD	Ağırlıklı Ortalama (Weigthed Average)	KNN, SVM, DT, RF, GB, MLP, NB	SVM, RF ve GB	0,89
Kabir ve Ludwig	**2019**	**V1:Messidor V2:WBC V3:PIDD V4:ILP**	**Süper Öğrenci (Super Learner)**	**SL(GBM-RF-DNN), S(GBM-RF-DNN)**	**GLM**	V1: 0,84 **V2: 0,99** V3: 0,88 V4: 0,73
Yadav ve Pal	**2021**	**UCI makine öğrenmesi deposundan**	**Oylama (Voting)**	**DT, OneR, JRip**	**Torbalama (DT, OneR, JRip)**	**0,98**
Kumari ve diğerleri	2021	PIDD	Oylama (Voting)	LR, NB, RF	Yumuşak Oylama (LR, NB, RF)	0,790
Taz ve diğerleri	2021	PIDD	Oylama (Voting)	LightGBM, XGBoost, AdaBoost, RF	Yumuşak Oylama (Soft Voting)	0,95
Deberneh ve Kim	2021	Hanaro Tıp vakfı	Oylama (Voting)	SVM, XGBoost, RF	Yumuşak Oylama , Yığınlama	~0,80

2.1.3.4 Literatür taraması genel değerlendirme

Yukarıda bahsedilen çalışmalarda da görüldüğü üzere topluluk öğrenmesi modelleri tek tabanlı makine öğrenmesi modellerine kıyasla genellikle daha iyi tahmin performansı göstermişlerdir (Odegua 2020). Topluluk öğrenmesi modellerinde de heterojen yaklaşım izleyen Yığınlama (stacking ensemble) topluluk öğrenmesi modeli, homojen yaklaşım izleyen torbalama ve arttırma topluluk öğrenmesi modellerine kıyasla daha iyi tahmin performansı göstermiştir (Tsai *et al.* 2014). Homojen yaklaşımlı yapıda model

oluşturulurken aynı türde temel makine öğrenmesi algoritmaları kullanılırken heterojen yaklaşımlı yapıda farklı türde temel makine öğrenmesi algoritmaları kullanarak model oluşturulmaktadır (Elish *et al.* 2013).

Super Learner topluluk öğrenmesi modelinde çapraz doğrulama kullanarak model performansını iyileştirip daha sonra birbirinden farklı makine öğrenmesi algoritmaları kullanarak model birleştirme işlemi yapar. Super Learner topluluk öğrenmesi modeli diğer topluluk öğrenmesi modellerine kıyasla farklı problemlerde ve veri setlerinde daha esnek ve daha uygulanabilir olduğunu göstermiştir (Li *et al.* 2021). Bizim yapacağımız çalışmada da diyabet hastalığının erken tanı ve teşhisini koymak için Super Learner topluluk öğrenmesi modelini yeni bir diyabet veri setinde uygulayarak doğru tahmin performansı elde etmektir. Bu bölümün geri kalan kısmında özellik seçim yaklaşımının tahmin modellerindeki etkisi hakkında bilgi verilecektir.

2.2 Özellik Seçim Yaklaşımının Tahmin Modellerindeki Etkisi

Özellik seçim yöntemi, veri setinde tanımlanmış özelliklerden hedef değişkenini saptayabilmek için en alakalı özellikleri bir alt özellik kümesi olarak tanımladığımız seçme tekniğidir (Kuhn and Johnson 2013). Modelin hesaplama maliyetini ve performansı artırmak için özellik seçim yöntemleri uygulanır. Bölüm 2.2.1'de Bilgi Kazanımı (Informatin Gain), Kazanç Oranı (Gain Ratio), Pearson Korelasyon, Ki-Kare (Chi-Square), Varyans Analizi (ANOVA), Özyinelemeli Öznitelik Eleme (RFE) çeşitli öznitelik seçme teknikleri hakkında bilgi ve Bölüm 2.2.2'de özellik seçim yaklaşımları ile ilgili çalışmalar hakkında bilgi verilmiştir.

2.2.1 Özellik seçim modelleri

2.2.1.1 Bilgi Kazanımı (Informatin Gain)

Veri setindeki rastgele seçilen bir özelliğin veri sınıflandırmadaki sonucunu ölçer. Bilgi kazanımını anlayabilmek için entropi kavramını bilmemiz gerekir. Entropi bir olayın

olması için gereken bilgi değeridir (Tom M. Mitchell 1999). Bir olayın düşük olasılıklı olması entropisinin yüksek, bir olayın yüksek olasılıklı olması ise entoropi değerinin düşük olduğunu ifade eder. Bilgi kazanımını ile entropi arasında ters bir ilişki vardır. Bilgi kazanımının yüksek olması için entropi'nin düşük olması gerekir. Genel olarak karar ağaçlarında kullanılmaktadır.

2.2.1.2 Kazanç Oranı (Gain Ratio)

Bilgi kazanımının birden fazla farklı değere sahip olan özelliklerde seçim yapmasının önüne geçebilmek için alternatif olarak kullanılan özellik seçim tekniklerinden biridir (Quilan 1988). 0 - 1 arası değer alır. Kazanç oranı 1 değerini aldığında veri seti hakkında daha fazla bilgiye sahip olduğunu, 0 değeri aldığında ise herhangi bir bilgiye ulaşamadığını gösterir.

2.2.1.3 Pearson Korelasyon

İki ya da daha fazla değişkenin arasındaki ilişkiyi tanımlamak için kullanılır. Korelasyon değeri -1 ile +1 arasında değer alır. Korelasyon değeri pozitif ise değişkenlerin arasında doğru orantılı bir ilişki olduğunu gösterirken, korelasyon değeri negatif ise değişkenler arasında ters orantılı bir ilişkinin olduğunu söyleyebiliriz (Cohen 2013).

2.2.1.4 Ki-Kare (Chi-Square)

Kategorik tanımlı değişkenlerde kullanılan yöntemdir. İki ya da daha çok girdi değişkeninin ilişkisini ölçerek en yüksek orana sahip olan özellikleri listeler (Freedman *et al.* 1998). Hedef değişken ile seçilen özelliklerin arasında yüksek oranda bir ilişki vardır ama değişkenlerin kendi aralarında bir ilişki yoktur.

2.2.1.5 Varyans Analizi (ANOVA)

Bir değişken değerini en az 3 veya daha fazla grup ile karşılaştırmak için kullanılan bir yöntemdir. Bu yöntemde grubun içinden en az birinin diğerlerinden farklı olup olmadığını ortaya koymak için tercih edilen bir yöntemdir.

2.2.1.6 Özyinelemeli Öznitelik Eleme (RFE)

Veri setindeki özelliklere hedef değişkenle arasındaki ilişkiye göre bir katsayı verilir. Daha sonra önemi az olan özellik kaldırılır. Geriye kalan özellikler arasında tekrar aynı şekilde bir katsayı verilerek önem sırasına göre listelenip, önemi az olan özellik kaldırılır. Modele uyan en iyi özniteliği bulana kadar bu işlem devam eder (Kuhn and Johnson 2013).

2.2.2 Özellik seçim yaklaşımını uygulayan tahmin modelleri

Literatür taramasında özellik seçimi ve sınıflandırmaya dayalı makine öğrenmesi algoritmalarını kullanan pek çok çalışma mevcut bulunmaktadır. Özellik seçim tekniğini kullanarak modelden elde edilen sonuçların doğruluk oranlarının artmasıyla birlikte bu tekniklerin kullanımı da yaygınlaşmıştır. Çizelge 2.4'te yapılan uygulamalar hakkında detaylı bilgi verilmiştir.

Mishra ve diğerleri (2016) tarafından Diabetes Healthcare veri setını kullanarak diyabet hastalığının tahmini için öznitelik seçim yöntemlerini uygulamışlardır. RBF, IBK, JRip üç tane sınıflandırma algoritmaları kullanmışlardır. Doğruluk, AUC, F-1 puan ölçüm metriklerini uygulamışlardır. Ki-Kare yöntemi %84,3, Kappa İstatistik ile Korelasyon Yöntemi %66,7 sonuçlarını almışlardır. En uygun doğruluk oranı ise %98,78 ile Bilgi Kazanımı elde etmişlerdir.

Yuvaraj ve SriPreethaa (2019) çalışmalarında National Institute of Diabetes tarafından 75.664 diyabet hasta kaydıyla oluşturulan diyabet veri seti ile diyabet hastalığının teşhis

edilmesini önermişlerdir. Rastgele Orman, Karar Ağacı, Naif Bayes olmak üzere 3 farklı makine öğrenmesi algoritmalarını kullanmışlardır. Hadoop tabanlı yazılama ekleyerek yeni bir uygulama geliştirmişlerdir. Hadoop açık kaynak yazılımı destekleyen, çok büyük veri kümeleri için tahmine dayalı makine öğrenmesi algoritmalarını çalıştırabilmek ve uygulayabilmek için kullanılan ücretsiz bir platformdur. Doğruluk, F-1 puan, kesinlik ve geri çağırma ölçüm metriklerini kullanmışlardır. Özellik seçim yöntemlerinden Bilgi Kazanımı özellik yöntemini uygulamışlardır. Veri setindeki 13 öznitelik arasından 7 özniteliği (Plasma glucose concentration, Diastolic blood pressure, Triceps skin fold thickness, 2-hour serum insulin, Body mass index, Diabetes pedigree function ve Age) seçmişlerdir. Yapılan deney sonuçlarına göre en iyi doğruluk performansı %94 oranıyla Rastgele Orman makine öğrenmesi algoritması ile elde etmişlerdir.

Çizelge 2.4'te özellik seçim yöntemlerinden Ki-Kare tekniğini kullanarak %98 değerinde en iyi performansı gösteren Perveen ve diğerleri (2016) ve Yadav ve Pal (2021) olmak üzere iki çalışma vardır. Ki-Kare tekniği kullanarak %80 değerinde en düşük doğruluk performansını gösteren iki çalışma ise %80 değer ile Putri ve diğerleri (2019) ve Deberneh ve Kim (2021)'dir.

Çizelge 2.4 Özellik seçim yöntemleri uygulayan çalışmalar

Yazar Adı	**Yıl**	**Makine Öğrenmesi Algoritması**	**Özellik Seçimi**	**ACC**
Perveen ve diğerleri	2016	Torbalama	Ki-Kare	0,98
Mishra ve diğerleri	2016	RBF, IBK, JRip	Bilgi Kazanımı	0,843
Alghamdi ve diğerleri	2017	Oylama (RF, LR, NB)	Bilgi Kazanımı	0,922
Putri ve diğerleri	2019	LVQ	Ki-Kare	0,80
Yuvaraj ve SriPreethaa	2019	RF	Bilgi Kazanımı	0,94
Kowsher ve diğerleri	2019	ANN	Korelasyon Katsayısı	0,947
Ram ve Vishwakarma	2021	LR	RFE	0,85
Buyrukoğlu	2021	RF	G1:Ki-Kare G2: Kazanç Oranı, Ki-Kare, RelifF, FCBC	G1:0,88 G2:0,91
Yang ve diğerleri	2021	XGBoost	Karşılıklı Bilgi (MI), Varyans Analizi (ANOVA), Gini Index	0,8768
Yadav ve Pal	2021	Torbalama	Ki-Kare	0,98
Deberneh ve Kim	2021	Yumuşak Oylama, Yığınlama	RFE, ANOVA testi, Ki-Kare	~ 0,80

Bu tez çalışmasında kullanılan veri seti kategorik bir veri seti olduğu için özellik seçim yöntemlerinden Ki-Kare tekniğini kullanılmıştır. Veri seti ile alakalı detaylı bilgi Bölüm 4.1'de mevcut bulunmaktadır.

3. MAKİNE ÖĞRENMESİ ALGORİTMALARI

Bu bölümde tek tabanlı makine öğrenmesi algoritmaları, topluluk tabanlı makine öğrenmesi modelleri ve öznitelik seçim yöntemlerinden bahsedilmiştir.

3.1 Tek (Single) Tabanlı Makine Öğrenmesi Modelleri

Bu bölümde tez çalışmamızda kullandığımız tek tabanlı makine öğrenmesi algoritmalarından Lojistik Regresyon (Logistic Regression), Destek Vektör Makineleri (Support Vector Machines), K-En Yakın Komşu (KNN), Naif Bayes (Naive Bayes) ve Yapay Sinir Ağları (Artificial Neural Networks) hakkında detaylı bilgi verilecektir.

3.1.1 Lojistik Regresyon (Logistic Regression)

Lojistik Regresyon modeli literatürde 1944 yılında Joseph Berkson tarafından ilk kez tanıtıldı ve daha sonra yaygın olarak kullanılmaya başlandı (Cramer 2005). Veri setindeki veri tiplerinin kategorik olarak tanımlandığı, çıktı parametresi olarak bağımlı değişkenin olduğu durumlarda kullanılan bir makine öğrenmesi algoritmasıdır. Örneğin hastanın diyabet olup (1) olmadığına (0) bakar.

Lojistik Regresyon modelinin temelinde Maksimum Olabilirlik Tahmin Yöntemi (Maximum Likelihood Estimation) vardır (Gourieroux and Monfort 1981). Doğruluk tahmini yapabilmek için bu yöntemi kullanır. Lojistik Regresyon modelinde kullanılan lojistik fonksiyonu aynı zamanda sigmoid fonksiyonu olarak da bilinmektedir. Sigmoid fonksiyonu “0” ve “1” arasında değer alır ve “S” şeklinde bir eğriden oluşmaktadır. Maliyet fonksiyonu hesaplandıktan sonra bu oranı en aza indirgemek için Gradient Descent algoritması kullanılır. Gradient Descent algoritması verilen değerin minimum bir değere indirgemeyi amaçlamaktadır (Lemaréchal 2012).

Lojistik Regresyon modeli ayrıca ikili bir sınıflandırma yapabilmek için kullanılan yaygın yöntemlerden biridir. Modelde sınıflandırma yapılırken bir sınır değeri belirlenir

ve bu sınır değerinin üstünde kalan değerler bir sınıfa tanımlanır, sınır değerinin altında kalan değerler ise başka bir sınıf olarak etiketlenmiş olur. Lojistik Regresyon makine öğrenmesinin avantaj ve dezavantajları Çizelge 3.1'de verilmiştir.

Çizelge 3.1 Lojistik regresyon makine öğrenmesinin avantajları ve dezavantajları

Avantajlar	Dezavantajlar
• Basit ve kolay olması sebebiyle çok yaygın olarak kullanılır. • Karmaşık bir yapısı olmadığı için uygulama hızı yüksektir.	• Karmaşık ve büyük veri setlerine sahip modellerde çoğunlukla çok iyi performans gösteremez.

Lojistik Regresyonun matematiksel olarak gösterimi Denklem (3.1)'de verilmektedir:

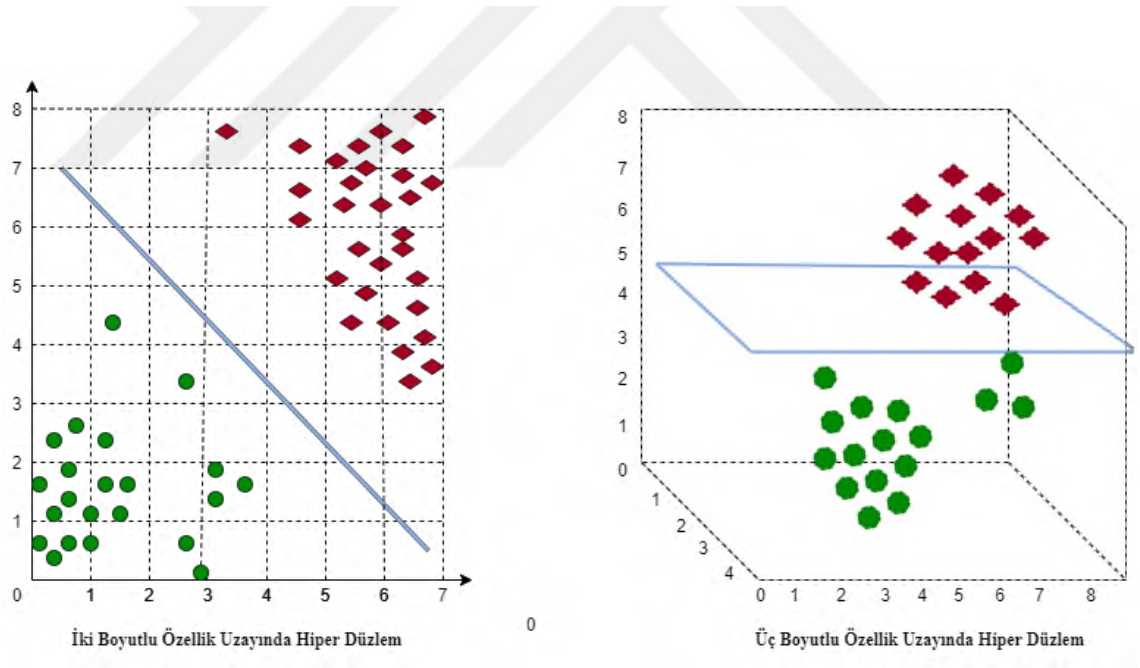
$$p = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + \cdots + + b_p x_p)}} \tag{3.1}$$

3.1.2 Destek Vektör Makineleri (Support Vector Machines)

Destek Vektör Makineleri, hem regresyon hem de sınıflandırma problemlerinde veri analizi yapabilmek için kullanılan denetimli bir makine öğrenmesi algoritmasıdır. Literatüre girmesi ve tarihsel gelişimine bakacak olursak ilk zamanlarda regresyon problemleri için kullanılacağı söylense de genellikle sınıflandırma problemleri için kullanılmıştır (Dr. Soman K. P.and Loganathan 2009). Temeli 1936 yılında Fisher'ın Doğrusal Diskriminant Yöntemi ile başlamıştır. 1957 yılında Frank Rosenblatt'ın Perceptron isimli basit bir doğrusal model fikrini ortaya atmasıyla devam eder. 1974 yılında Vapnik ve Chervonenkis tarafından "Maksimal Marj Sınıflandırıcı" isimli yeni bir model önerilmiştir. Bu modelin temeli istatiksel öğrenme metoduna dayanmaktadır. 1992 yılında da Vapnik ve arkadaşları tarafından Çekirdek Trick (Kernel Trick) isimli

doğrusal olmayan veriler için sınıflandırma yapmak için kullanılmaya başlanmıştır. En sonunda ise 1995 yılında Cortes ve Vapnik tarafından "Esnek Marjinal Sınıflandırıcı" isimli SVM algoritmasında bazı yanlış sınıflandırmaları kabul etmemizi sağlayan makaleyi tanıtmışlardır (Cortes and Vapnik 1995). Destek Vektör Makinelerinde sıklıkla kullanılan terimler aşağıdaki gibidir:

Hiper Düzlem (Hyper Plane): En az iki özelliğe sahip bir sınıf için, veri kümelerini doğrusal bir şekilde birbirinden ayıran ve sınıflandırma yapmamızı sağlayan bir karar sınırı olarak ifade edebiliriz. Sınıfımızın özellik sayısına göre bu hiper düzlemin boyutu değişir. Eğer sınıfımızın özellik sayısı 2 ise hiper düzlemimizin bir boyutlu bir doğrudur. Eğer sınıfımızın özellik sayısı 3 ise hiper düzlemimiz iki boyutlu bir düzlem olur. Bu şekilde, n boyutlu bir Öklid Uzayı için, onu iki parçaya ayıran n-1 boyutlu bir alt düzleme sahip oluruz. Aşağıdaki Şekil 3.1'de İki boyutlu ve üç boyutlu özellik uzayında hiper düzlem gösterilmiştir.



Şekil 3.1 İki boyutlu ve üç boyutlu özellik uzayında hiper düzlem (R gibi ikinci dereceden denklemde hiper düzlem bir çizgidir. R gibi üçüncü derece denklemde hiper düzlem bir yüzey şeklindedir.) (Morales 2017)

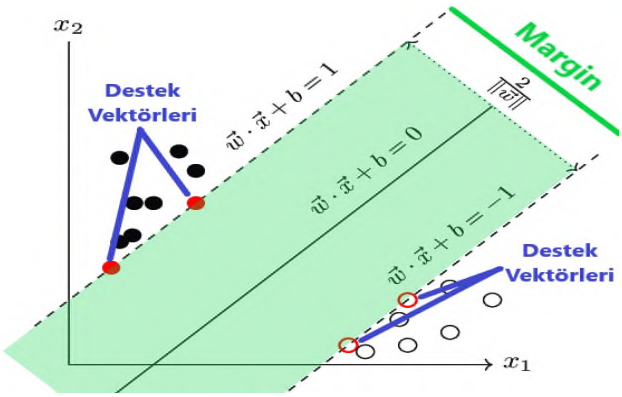
Destek Vektörleri (Support Vectors): Bir sınıftaki hiper düzleme en yakın olan farklı veri noktaları olarak tanımlayabiliriz. Bu veri noktalarını en doğru bir şekilde bölen

sınır çizgileri çizer. Destek vektörler üzerinde yapacağımız değişiklikler hiper düzlemde konum ve yön değişikliğine neden olacaktır (Burges 1998).

Alt Düzlem: Farklı sınıflara ait bir dizi nesnenin birbirinden bölünmüş bir şekilde olduğu karar düzlemi olarak tanımlayabiliriz.

Marjin (Margin): Bir hiper düzlem ile farklı sınıfa ait veri noktaları arasındaki mesafeyi maksimize edebilmemiz için en doğru hiper düzlemi seçmemize yardımcı olacaktır. Yüksek marjlı bir hiper düzlem ile en iyi sınıflandırma yapma olasılığını yakalamış oluruz.

Destek vektör makinelerinde sınıflandırma problemlerinde iki sınıfın birbirinden ayrılmasını sağlamak için bir doğru çizilir. Bu doğrunun ±1'i arasında kalan yeşil bölgeye Marjin olarak tanımlanır. Marjinin geniş olması sınıfların birbirinden ayrılmasına kolaylık sağlamaktadır. Şekil 3.2'de Destek Vektör Makinelerindeki hiper düzlem üzerinde bulunan iki sınıfın sınıflandırılması sunulmaktadır.



Şekil 3.2 Destek vektör - hiper düzlem (Medium Akca M.F. 2020)

3.1.2.1 Destek Vektör Makinelerin Çalışma Şekli

Bir SVM modelinin temeli veri noktalarını net bir biçimde sınıflandırma yapabilmek için N boyutlu uzayda (N: özellik sayısı) bir hiper düzlemde göstermesidir. Verilerimiz

iki boyutta ise en iyi doğruyu, ikiden fazla boyutta ise en iyi hiper düzlemi seçebilmektir. Hatayı en aza indirebilmek için SVM yinemeli olarak hiper düzlem oluşturur. Hiper düzlem maksimum marj üzerinden hesaplanır. Maksimum marjinal hiper düzlem (MMH) elde edebilmek için veri setimizdeki sınıfları bölme işlemi gerçekleştirilir. Daha sonra bunların arasından en doğru sınıflandırmayı yapan, diğer bir deyişle her iki sınıfın veri noktaları arasındaki maksimum mesafeye sahip olan bir hiper düzlemin seçimini yapacaktır. Sınıflandırma işlemi yaparken veri kümelerimiz;

- Bir boyutta ise bir nokta ile,
- İki boyutta ise çizgi (doğru) ile,
- Üç boyutta ise bir düzlem ile,
- Çok boyutta ise onu bir hiper düzlem kullanarak sınıflandırabiliriz.

Destek Vektör Makineleri girdi verilerine göre ikiye ayrılır:

1. **Doğrusal Destek Vektör Makineleri:** Girdi verileri doğrusal bir çizgiyle ayrılarak marjı en büyük olan doğruyu seçme işlemidir.

2. **Doğrusal Olmayan Destek Vektör Makineleri:** Doğrusal olmayan girdi verilerini sınıflandırabilmemiz için doğrusal hiper düzlemi kullanarak çözüme kavuşamayız. Ama bu verileri daha yüksek boyutta doğrusal bir şekilde ayırabileceğimiz verilere dönüştürebiliriz. Bunun için çekirdek numarası adı verilen yöntem kullanılır.

Destek vektör makinelerinin avantajları ve dezavantajları Çizelge 3.2'de verilmiştir.

Çizelge 3.2 Destek vektör makinelerinin avantanjları ve dezavantajları

Avantajlar	Dezavantajlar
• Çok boyutlu veri setleri için kullanıma uygundur. • Veri setinde hem etiketlendirilmiş verilerle hem de etiketlenmemiş veriler için kullanıma uygundur. • Doğrusal ve doğrusal olmayan veri setleri için kullanıma uygundur.	• Çekirdek türü seçiminde dikkatli olunmalıdır. Birden çok çekirdek türü olduğu için seçim yapmayı zorlaştırabilir.

3.1.3 K-En Yakın Komşu (K- Nearest Neighbors)

KNN algoritması, kullanımı basit ve anlaşılır olması sebebiyle hem sınıflandırma hem de regresyon problemlerinde kullanılan denetimli bir makine öğrenmesi algoritmasıdır. Genel olarak sınıflandırma problemlerinde tercih edilmektedir. 1967 yılında T.Cover ve P.Hart tarafından ele alınan "En yakın komşu desen sınıflandırılması" makalesiyle KNN algoritmasının temelleri atılmıştır (Cover and Hart 1967). Tembel ve parametrik olmayan bir öğrenme modelidir. Bu iki terim aşağıda tanımlanmıştır:

- **Tembel Öğrenme Modeli:** Bu modelde eğitim aşama kısmı olmadan sınıflandırma işlemi yapan ve test aşaması sırasında bütün verileri kullanan tembel bir öğrenme modelidir. Bu yöntem eğitimi hızlandırır ancak test aşaması sırasında maliyet artırır ve yavaşlar. Bunun için de bellek ve zamana olan ihtiyaç artmaktadır.

- **Parametrik Olmayan Öğrenme Modeli:** Veriler hakkında herhangi bir bilgisi olmadığı için varsayımda bulunmaz.

KNN algoritmasının çalışma modeli şu şekildedir:

1.) KNN algoritmamızı uygulayabilmek için ilk önce veri setini yüklememiz gerekir.
2.) En yakın veri noktalarını seçebilmemiz için K'ya bir değer atanır.
3.) Verilerdeki her nokta için aşağıdaki adımlar tekrarlanır.
    a) Test veri seti ile her eğitim veri seti arasındaki mesafe hesaplanır. Manhattan, Minkowski, Hamming ve Euclidean yöntemlerinden biriyle hesaplama işlemi yapılır. En çok tercih edilen yöntemlerden biri ise Euclidean'dir.
    b) Hesaplama değerine göre küçükten büyüye doğru artan bir şekilde sıralama işlemi yapılır.
    c) Sıralanan listeden en üstteki K satırı seçilir.
    d) Eğer regresyon ise K etiketlerinin ortalaması döndürülür.
    e) Eğer sınıflandırma işlemi ise, K etiketlerinin modu döndürülür (Hierons 1999).

Sürekli değişkenlerde Euclidean, Manhattan, Minkowski fonksiyonları kullanılabilir. Bu fonksiyonların mesafe ölçümü için matematiksel gösterimi Euclidean Denklem (3.2)'de, Manhattan Denklem (3.3)'te, Minkowski Denklem (3.4)'te verilmiştir. Kategorik değişkenlerde ise Hamming fonksiyonu kullanılır. Hamming fonksiyonun matematiksel gösterimi Denklem (3.5)'te verilmiştir. Eğer veri setinde sayısal değişkenler ile kategorik değişkenler birlikte bulunuyorsa, 0 ile 1 aralığında sayısal değer alır. Bu hesaplama ölçümünün matematiksel gösterimi Denklem (3.6)'da verilmiştir.

$$Euclidean = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{3.2}$$

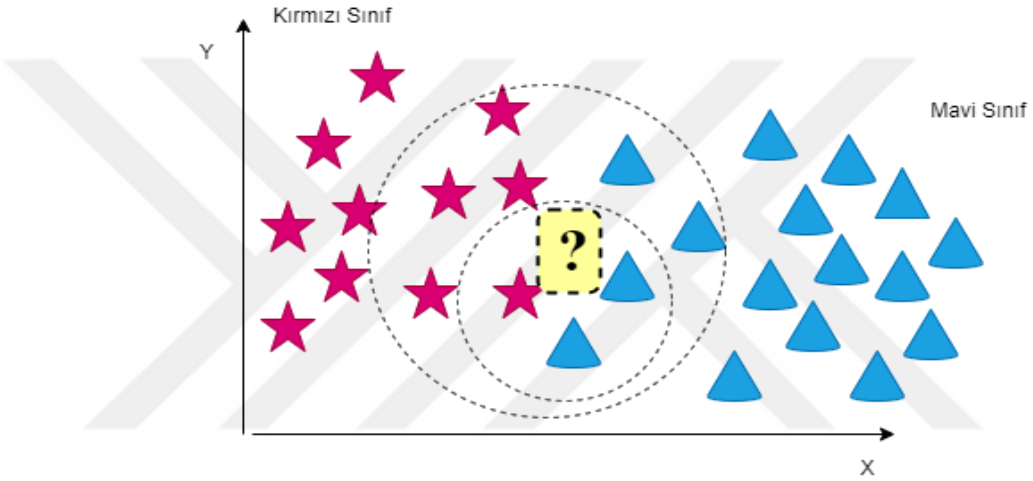
$$Manhattan = \sum_{i=1}^{k}|x_i - y_i| \tag{3.3}$$

$$Minkowski = \left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{\frac{1}{q}} \quad (3.4)$$

$$Hamming \quad = D_H = \sum_{i=1}^{k}|x_i - y_i| \quad (3.5)$$

$$x = y \Rightarrow D_H = 0, \quad x \neq y \Rightarrow D_H = 1 \quad (3.6)$$

Şekil 3.3'te k değerinin 3 ve 9 olarak seçildiğinde sınıflandırma örneği sunulmuştur.



Şekil 3.3 K=3 ve K=9 için K en yakın komşu sınıflandırma örneği

K parametresine optimum değeri verebilmek için verileri çok iyi bilmemiz gerekmektedir. K değeri büyük seçilirse gürültüyü azaltmış oluruz. Daha doğru tahminlerde bulunma olasılığımız artar. K değerini düşürdükçe tam tersi durum olur. Tahminlerimizin gerçekleşme olasılığı düşer. Genel olarak veri seti için en uygun K değeri 3-10 aralığında seçilmesi daha uygun olmaktadır (Hall *et al.* 2008).

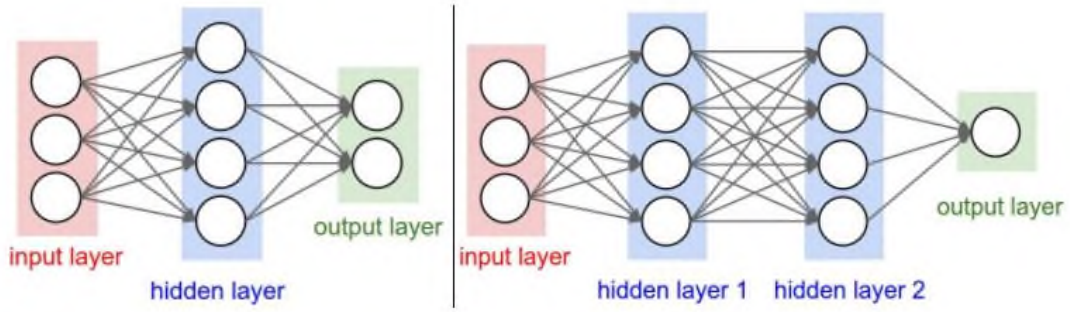
KNN makine öğrenmesinin avantajları ve dezavantajları Çizelge 3.3'te verilmiştir.

Çizelge 3.3 K en yakın komşu makine öğrenmesinin avantajları ve dezavantajları

Avantajlar	Dezavantajlar
• KNN algoritması basit ve uygulanması kolay olduğu için çok tercih edilmektedir. • Bir model oluşturmaya, birkaç parametreyi ayarlamaya veya ek varsayımlar yapmaya gerek yoktur. • Algoritma çok yönlüdür. Sınıflandırma, regresyon ve arama için kullanılabilir. • Algoritma yeni veri eklemeye elverişlidir.	• Örneklerin ve bağımsız değişkenlerin sayısı arttıkça algoritma önemli ölçüde yavaşlar. • En yakın komşuları tahmin etmek maliyet açısından biraz yüksektir.

3.1.4 Yapay Sinir Ağları (Artificial Neural Networks)

Yapay sinir ağlarını daha iyi anlayabilmek için beynimizdeki sinir ağlarının çalışma yapısını anlamamız gerekmektdir. Beyin, çevremizdeki her şeyi anlayıp yorumlamaya çalışmaktadır. Gördüğümüz nesneleri tanımlar, sınıflandırır, duyduğumuz sesleri ayırt eder, konuşmaları tanır ve bunları nöronlar sayesinde yapmaktadır. Nöronlar bu girdileri alır ve çıktı olarak bize sunar. Bir nöronun çıktısı diğerinin girişi olabilir. Bu şekilde çok katmanlı bir sinir ağı oluşmaktadır. Yapay sinir ağları da beynimizdeki bu sinir ağlarını taklit etmektedir. Tıpkı insan beyni gibi eğitilip, öğrenebilen ve tahminde bulunan makinaların tasarlanması amaçlanmıştır (Alpaydin 2021). Şekil 3.4‘teki gibi tek gizli katmanlı ve çok katmanlı sinir ağı yapısı gösterilmiştir.



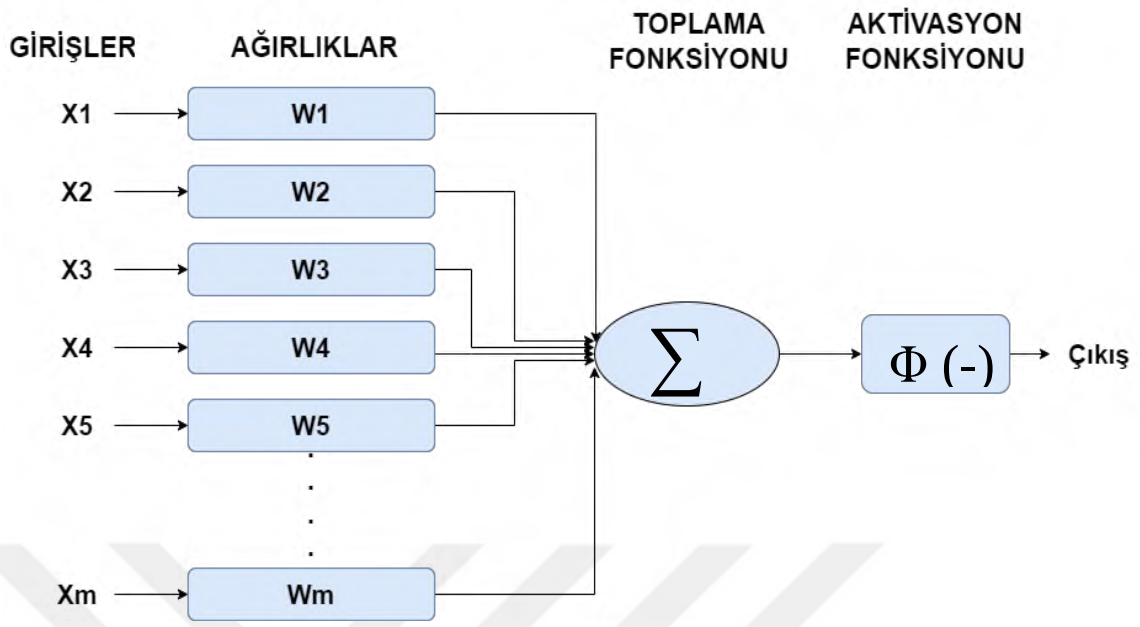
Şekil 3.4 Tek gizli katmanlı ve çok katmanlı sinir ağı yapısı (CS231n 2020)

Nöronları üç farklı katman tipinde ele almak mümkündür:

Giriş Katmanı (Input Layer): Giriş verilerimizi almaktadır. Yapay bir sinir ağında her zaman bir giriş katmanı olması gerekmektedir. Buradaki nöron sayımız, veri setimize bağlıdır.

Gizli Katmanlar (Hidden Layer): Giriş ve çıkış katmanı arasında kalan ve dış sistemler tarafından görülmedikleri için gizli katman olarak adlandırılmaktadır. Giriş katmanındaki nöronların çıktıları bu gizli katmanların giriş verisi olarak almaktadır. Bu katman hesaplamaların yapıldığı kısımdır. Daha basit problemlerde katman sayısı ve nöron sayısı az tutulurken, karmaşık problemlerin çözümünde bu sayı artabilmektedir. Katman sayısındaki artış hesaplamalarda karmaşıklığa ve hesaplama süresinin uzamasına neden olmaktadır. Buradaki gizli katman sayısının ne olduğuna karar verebilmek için sinir ağının yapısını iyi analiz etmek gerekmektedir.

Çıkış Katmanı (Output Layer): Çıkış verilerinin bulunduğu katmandır. Yapay sinir ağımızda her zaman bir çıkış katmanımızın olması gerekmektedir. Nihai sonuç ile ilgili bir tahminde bulunmaktadır. Şekil 3.5'te yapay bir sinir ağının yapısı sunulmuştur.



Şekil 3.5 Yapay sinir ağının yapısı

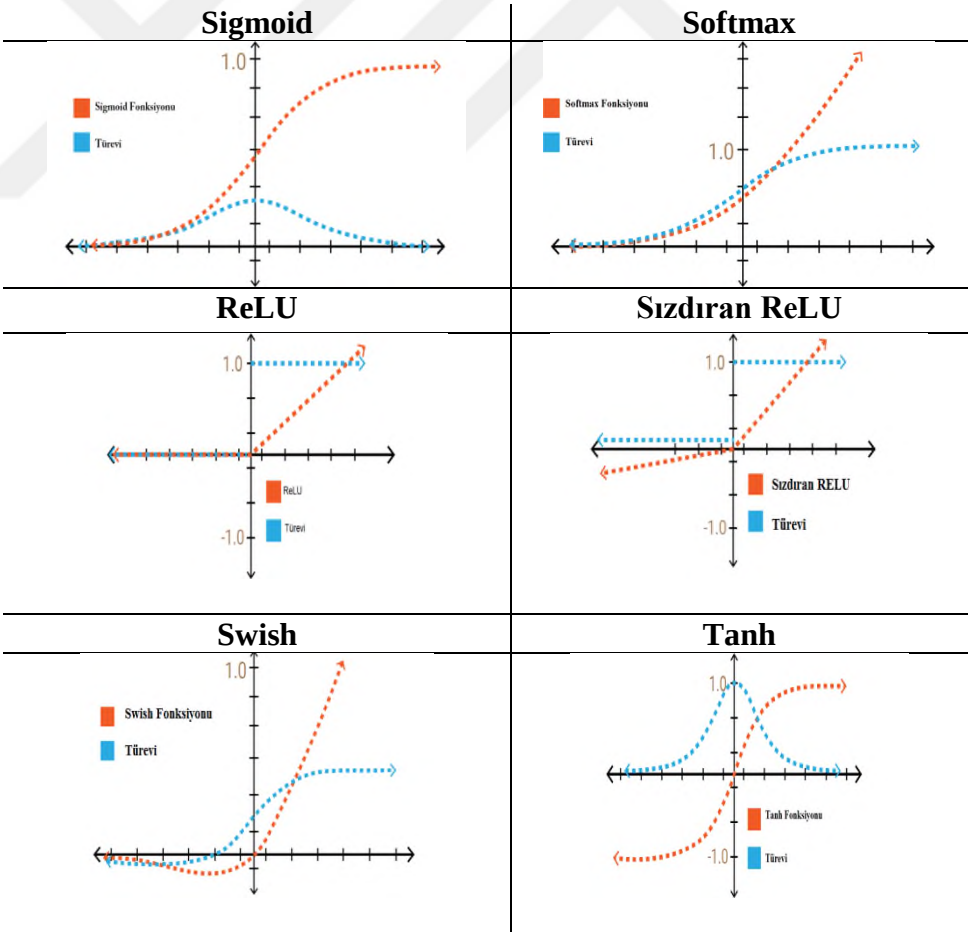
Ağırlıklar (Weights): İnsan beynindeki sinir ağları arasındaki bağlantı aksonlar tarafından sağlanmaktadır. Yapay sinir ağlarındaki bu bağlantı ise sayısal bir değer olan ağırlıklar tarafından gerçekleştirilmektedir. Başlangıç değerleri rastgele olarak atanabilir veya daha önce eğitilmiş bir modelin ağırlıkları kullanılabilir. Sabit ya da sıfıra yakın pozitif veya negatif değerlerde verilebilir. Ağırlığın sıfır olması nöronun çıktısını etkilemez ve böylece öğrenme gerçekleşmemiş olur. Buradaki en önemli konu ise eğitim sürecinde her bir iterasyonda hatayı yayarak olması gereken optimum ağırlık değerlerinin tekrar hesaplanıp, güncellenmesidir. Eğitim süresince bu şekilde güncelleme devam eder. Ağırlıklar uygun seviyede ayarlandıktan sonra eğitim aşamasına geçilebilir.

Aktivasyon Fonksiyonu: Yapay sinir hücresinde gelen girdi değerine x ve ağırlık değerine de w olarak tanımlarız. Aktivasyon fonksiyonu hesaplama sonucu olarak bu ağın çıkış fonksiyonu olarak üretilen f(x) değerinin bir sonraki katmana iletip iletilmeyeceğine karar vermektedir. Aktivasyon fonksiyonu bizim sinir ağımızı eğitmemiz için gereken bir fonksiyondur. Çünkü dış dünyadan gelecek olan karmaşık verilerin (ses, video, metin gibi) analiz edilmesinde doğrusal olmayan fonksiyonlara ihtiyaç duyarız. Bu şekilde çok katmanlı bir sinir ağına öğrenme yetisi kazandırabiliriz.

Ağın performans ve başarı kriterini artırmak için aşağıdaki hususlara dikkat etmemiz gerekecektir;

- Aktivasyon fonksiyonun doğrusal olmaması (Non-Lineer olması)
- Seçilen aktivasyon fonksiyonun türev alma işleminin kolay olması gerekmektedir. Çünkü her katmanda hatanın geri yayılmasında türev işlemi uygulanır. Eğitim süresinin kısa olması için türevi kolay alınabilen fonksiyonları tercih etmek gerekir.

En çok tercih edilen aktivasyon fonksiyonları ise Sigmoid, Softmax, ReLU, Sızdıran ReLU, SWISH, TanH'dır. Şekil 3.6'da aktivasyon fonksiyonlarının diyagramı, Çizelge 3.4'de aktivasyon fonksiyonlarının matematiksel formülü sunulmuştur.



Şekil 3.6 Aktivasyon fonksiyonlarının diyagramı (Ayyüce Kızrak 2019)

1. Sigmoid Fonksiyonu: En yaygın kullanılan aktivasyon fonksiyonlarının başında gelmektedir. İkili sınıflandırma problemlerinde daha çok tercih edilmektedir. Şekil 3.6'daki Sigmoid grafiğin X değer aralığını -2 ile +2 arasında inceleyecek olursak, y değerlerinin hızla değiştiğini görmekteyiz. Bu da bizim sınıflandırma problemimizi çözmemizde kullanabileceğimizi göstermektedir. Grafiğin uç noktalarına baktığımızda ise x değerlerindeki değişikliğe karşı y değerlerinde değişiklik çok fazla değildir. Bu bölgede uygulanan türev değeri ise çok düşüktür ve sıfıra yaklaşmaktadır. Bu de demek oluyor ki öğrenme burada çok azdır. Sıfırda ise öğrenme yoktur. Sınıflandırma problemimizde maksimum doğruluk değerini yakalamamız zorlaşabilmektedir. Bu nedenle bilgi kaybı gerçekleşeceği için bir dezavantajdır.

2. Softmax Fonksiyonu: Sigmoid aktivasyon fonksiyonun genelleştirilmiş halidir. Sigmoid aktivasyon fonksiyonu ikili sınıflandırmalarda kullanılırken softmax aktivasyon fonksiyonu ise çok sınıflı sınıflandırmalarda kullanılmaktadır. Genelde yapay sinir ağında son katmanda kullanılmaktadır. Şekil 3.6'daki Softmax grafiğin 0 ile 1 arasında olasılık değeri alan ve olasılıkların toplamı 1'e eşit olan bir fonksiyon olduğu gösterilmektedir. Her sınıf için olasılıkların hesaplandığı, verilen girdi için hedef sınıfın tahmin edilmesinde en yüksek olasılığa sahip olan sınıf etiketlenmektedir (Eli Bendersky 2003).

3. ReLU Fonksiyonu (Doğrusallaştırılmış Doğrusal Ünite): Derin öğrenme veya evrişimli sinir ağlarında en çok kullanılan aktivasyon fonksiyonudur. Şekil 3.6'daki ReLU grafiğe baktığımızda sıfır ile sonsuz arasında değer alan bir fonksiyondur. ReLU pozitif olduğunda girdi ile aynı çıktıyı vermektedir. Negatif olduğunda ise çıktı sıfıra eşittir. Buradaki dezavantaj ise ReLU aktivasyon fonksiyonuna verilen tüm negatif değerleri direk sıfıra dönüştürmesidir. Böylece negatif değerler hep aynı sonucu verdiği için öğrenme olmayacaktır.

4. Sızdıran ReLU: ReLU aktivasyon fonksiyonunun geliştirilmiş halidir. ReLU'ya negatif değerlerde çözüm olarak kullanılan fonksiyondur. Şekil 3.6'daki grafiğinde Sızdıran ReLU fonksiyonun aralığı (–) sonsuz ile (+) sonsuz aralığındadır. Türevi negatif değerlerde 0.01 değerini alır, pozitif değerleri için 1 değerini alır. Grafiğin sol

tarafındaki bu küçük değişiklik negatif değerler için sıfırdan farklı bir değer alır ve bu bölgede öğrenme gerçekleşebilecektir.

5. Swish Fonksiyonu: Google'daki araştırmacılar tarafından keşfedilen, ReLU aktivasyon fonksiyonuna çok benzeyen aktivasyon fonksiyonudur. Farkı ise değer aralığının (–) sonsuz ile (+) sonsuz arasında olmasıdır. Sızdıran ReLU aktivasyon fonksiyonunda farkı ise giriş değerleri artsa bile fonksiyonun çıkış değerinin düşebilmesidir. Monoton olmamasıdır. Bu özellik swish'e özgüdür (Ramachandran *et al.* 2018).

6.Tanh Fonksiyonu: Sigmoid aktivasyon fonksiyonuna çok benzemektedir. Fark ise orjin etrafında simetrik olmasıdır. Tanh fonksiyonun değer aralığı ise -1 ile +1 arasındadır.

Çizelge 3.4 Aktivasyon fonksiyonlarının matematiksel gösterimi

Fonksiyonun Adı	**Matematiksel Gösterimi**	**Türev Fonksiyonun Matematiksel Gösterimi**
Sigmoid	$f(x) = 1 / (1 + e^{-x})$	f '(x) = 1-sigmoid (x)
Softmax	$\phi = \frac{e^{i}}{\sum_{j=0}^{k} e^{j}} \quad i = 0,1,2, \ldots \ldots k$	
ReLU	$f(x) = \begin{cases} 0, x < 0 \\ x, x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0, x < 0 \\ 1, x \geq 0 \end{cases}$
Sızdıran ReLU	$f(x) = \begin{cases} 0, x < 0 \\ x, x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0{,}01, x < 0 \\ 1, x \geq 0 \end{cases}$
Swish Fonksiyonu	f (x) = x * sigmoid (x) $f(x) = x / (1 - e^{-x})$	
Tanh	f(x) = tanh(x)	$f'(x)=1-f(x)^2$

3.1.5 Naif Bayes (Naive Bayes)

Naif Bayes makine öğrenmesi algoritması (1701-1761) yıllarında Thomas Bayes tarafından geliştirilen Bayes Teoreminin daha basit bir örneğidir (Barnard 1958). Belirli bir değişkenin ön plana çıkmasında diğer değişkenlerin bir ilgisi olmadığını, bağımsız olduğunu varsayım yapmaktadır (Harry Zhang and Sheng 2004). Bayes Teoreminin matematiksel olarak gösterimi Denklem (3.7)'de verilmiştir:

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \tag{3.7}$$

P(X): X olayının gerçekleşme olasılığı (ön olasılık),

P(Y):Y olayının gerçekleşme olasılığı (marjinal olasılık),

P(Y|X): X olayı gerçekleştiğinde Y olayının gerçekleşme olasılığı,

P(X|Y): Y olayı gerçekleştiğinde X olayının gerçekleşme olasılığını ifade etmektedir. Bu olasılığa arka olasılık olarak tanımlanmaktadır (Shatovskaya *et al.* 2006).

Naif Bayes sınıflandırıcısının matematiksel olarak gösterimi Denklem (3.8) ve Denklem (3.9)'da verilmiştir:

$$P(Y|X_1, X_2, X_3, X_4, X_5, \dots . X_n) = \frac{P(X_1, X_2, X_3, X_4, X_5, \dots . X_n|Y)P(Y)}{P(X_1, X_2, X_3, X_4, X_5, \dots . X_n)} \tag{3.8}$$

$$X = [X_1, X_2, X_3, X_4, X_5, \dots . X_n] \tag{3.9}$$

Bu formülde X birbirinden bağımsız değişkenler olduğu varsayımı yapılmaktadır. Y sınıfındaki olasılıklar için bunların hesaplaması yapılır ve max(P) olan değerimizi sınıf olarak etiketlenmektedir.

Naif Bayes makine öğrenmesi algoritmaları e-posta spam filtreleme, text sınıflandırması, öneri sistemleri ve duygu analizlerinde kolay kullanımı ve sınıflandırmadaki iyi performansından dolayı tercih edilmektedir (Haiyi Zhang and Li 2008). Naif Bayes makine öğrenmesinin avantajları ve dezavantajları Çizelge 3.3'te verilmiştir.

Çizelge 3.5 Naif Bayes makine öğrenmesi avantajları ve dezavantajları

Avantajlar	**Dezavantajlar**
• Model kurulumu basit ve çok hızlı olduğu için kullanımı çok yaygındır. • Küçük veri setlerinde eğitimi oldukça kolay olmaktadır. • Büyük ve karmaşık veri setlerinde de iyi performans gösterdiği için tercih edilmektedir.	• Bütün değişkenleri bağımsız olarak varsayım yapmasıdır. • Sıfır Frekans (Zero Frekans) sorunu vardır. Her değişken bir sınıf etiketiyle tanımlandığı için, test veri setimizdeki bir değişken eğitim veri setinde yoksa bu sınıf etiketi eksik olduğu için olasılık değeri sıfır olarak tanımlanmaktadır. Hesaplama yapılırken olasılıklar çarpıldığı için sonucumuz da sıfır olacaktır. Bu sorunu çözebilmek içinde Laplace tekniği kullanılmaktadır. Sıfır olasılıklı değere +1 eklenerek sorun çözülmektedir (Witten et al. 2016).

3.2 Topluluk (Ensemble) Tabanlı Makine Öğrenmesi Modelleri

Makine öğrenmesi algoritmalarının farklı kullanım alanlarındaki karmaşık ve zor problemlerin çözümünde elde ettiği yüksek performans yaygın bir şekilde kullanımını arttırmıştır. Makine öğrenmesi algoritmaların gittikçe artan kullanım alanları bazı sorumlulukları da beraberinde getirmiştir. Büyük ve karmaşık veri setlerinde bazı durumlarda tahminlerdeki performans düşüklüğü ve yüksek varyans farklı yaklaşım fikirlerinin ortaya çıkmasını sağlamıştır. Bu yaklaşımlardan biri de Topluluk Öğrenme (Ensemble Learning) modellerini oluşturmuştur. Topluluk öğrenmesi modelleri 1990'larda hızla gelişim gösterip yaygın olarak kullanılmaya başlanmıştır (Zhou 2012). 2000'li yıllarda Kaggle'daki yarışma ve Netflix ödülleri ile daha fazla benimsenmiştir (Töscher *et al.* 2009).

Topluluk öğrenmesi sonuca ulaşabilmek için birden çok model kullanarak doğru sonuçlar tahmin etmemize olanak sağlamaktadır. Topluluk öğrenmesi modelleri çoğunlukla tek bir sınıflandırıcı ile tahmin yapan modellerden daha iyi performans göstermiştir (Emon *et al.* 2021, Kabir and Ludwig 2019; Perveen et al. 2016, D. C. Yadav and Pal 2021). Çünkü makine öğrenmesi algoritmaları farklı şekillerde çalışmaktadır. Her modelin iyi ve kötü performans gösterdiği kısımlar bulunmaktadır. Ama birlikte kullanarak oluşturulan topluluk modellerinde birbirlerinin zayıflıklarını ortadan kaldırmış olurlar.

Bir modelde genellikle oluşan hataları anlamak için önyargı, varyans ve indirgenemez hatalara bakmak gerekmektedir. Önyargı, farklı bir eğitim veri setinde modelde ne kadar bir değişiklik olacağına bakmaktadır. Varyans ise modelin gerçek verilerle tahmin ettiği veriler arasındaki farkına bakmaktadır. Modelde yüksek varyans varsa aşırı öğrenme vardır. İndirgenemez hatada ise veri setimizde aykırı değerler, eksik değerler gibi durumlar vardır. Bunun için verinin temizlenmesi gerekmektedir. Genelde önyargıyı azaltmak için varyans arttırılmaktadır. Ya da tam tersi durum geçerli olmaktadır.

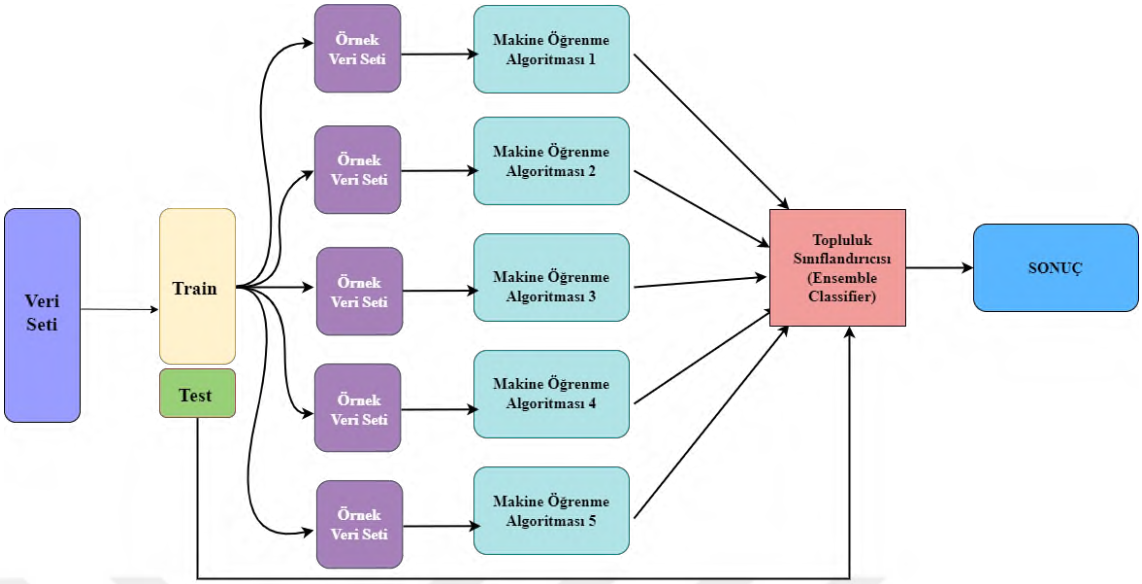
Topluluk Öğrenmesinde kullanılan yöntemlerin en yaygın olanları aşağıdaki gibidir:

- Torbalama (Bagging)
- Arttırma (Boosting)
- Yığınlama (Stacking)

3.2.1 Torbalama Topluluk Öğrenmesi Modeli (Bagging Ens. Learning Model)

Torbalama topluluk öğrenmesi (Boostrap AGGregating ensemble) modeli literatür de 1996 yılında Leo Breiman tarafından ele alınmıştır (Breiman 1996). Bu modelde ilk önce veri setimizi eğitim ve test veri seti olarak ayırırız. Ayırdığımız eğitim veri setinde rastgele seçim yaparak farklı eğitim setleri oluşturulur. Eğitim veri setleri birbirinden bağımsız bir şekilde paralel olarak eğitilirler. Farklı eğitim setleri farklı tahmin sonuçlarını doğurmuş olur. Alınan tahmin sonuçlarının ortalaması veya oylama işlemi gibi istatistik yöntemlerle birleştirilir. Bu yönteme önyükleme adı verilir (Varian 2005).

Torbalama topluluk öğrenmesi modelinde düşük sapmalı yüksek varyanslı modeller için daha uygundur. Amacımız varyansı azaltarak hatayı azaltmaktır. Torbalama algoritması olarak Karar Ağacı (Decision Tree), Karar Kütüğü (Decision Stump) ve Random Forest (Rastgele Orman) ve algoritmalarını söyleyebiliriz (Ho 1998). Torbalama topluluk öğrenme modelinin diyagramı Şekil 3.7‘de sunulmuştur.



Şekil 3.7 Torbalama topluluk öğrenme modeli diyagram

3.2.1.1 Karar Ağacı (Decision Tree)

Karar Ağacı hem sınıflandırıcı hem de regresyon problemlerinde tercih edilen denetimli bir makine öğrenmesi algoritmasıdır. Karar ağacının yaygın olarak kullanım alanları ise veri kümelerinin yinelemeli olarak kategorilere ayrılmasında bir sınıflandırıcı olarak kullanılmasıdır. Karar Ağacı modelimizi oluşturmak için ilk olarak bir özniteliğe göre yalnızca bir kök düğüme sahip olması gerekmektedir. Kök düğüm bizim problemi çözmemizdeki amacımızı belirlemektedir. Bundan sonra kök düğüm karar kurallarına göre dallara ve nihai karar olarak da yaprak düğümlere ayrılmaktadır. Yaprak düğüm bizim sınıf etiketimizi temsil eder. Modelimiz ağaç benzeri bir yapı almış olur (Rokach and Maimon 2009).

Gerçek hayatta karar ağacı yapısı insanın düşünme yeteneğini taklit etmeye dayanmaktadır. İnsanlar bir problemi çözüme kavuşturmak için çeşitli sorular sorarak uygun cevaplar bulmak ister. Karar Ağacı algoritması da aynı bu şekilde çalışmaktadır. Bir soru sorar ve aldığı cevaba göre (Evet / Hayır) Şekil 3.8'deki gibi ağaç yapısını oluşturmuş olur. Karar Ağacı algoritmasını kullanırken sıklıkla duyacağımız temel kavramlar aşağıda açıklanmıştır:

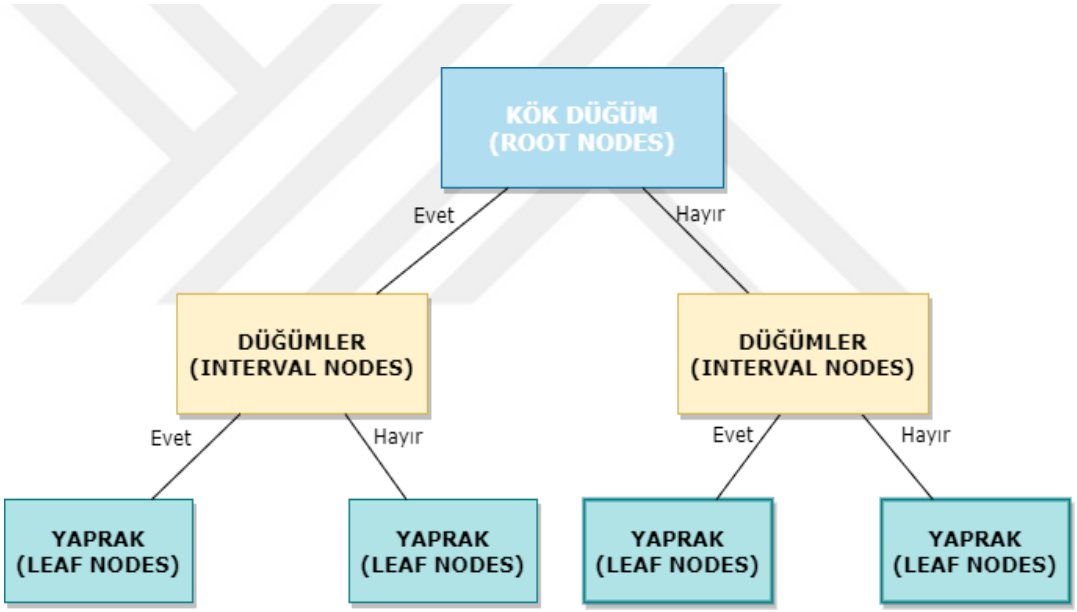
Kök Düğüm: Karar ağacının ilk başlama noktasıdır. Tüm veri kümemizi temsil etmektedir.

Bölme: Veri kümemizi verilen koşullara göre alt düğümlere bölme işlemidir.

Karar Düğümü: Düğümün dallara ayrılmasını sağlamaktadır.

Yaprak Düğümü: Düğümün nihai kararıdır. Bundan sonra herhangi bir bölünme işlemi olmaz.

Üst ve Alt Düğüm: Bir düğümün başka düğüme karşı pozisyonunu belirtir.



Şekil 3.8 Karar ağacı modeli (Medium Akca M.F. 2020)

Karar Ağacımızı oluştururken düğümlerimizin belirlenmesi çok önemlidir. Çünkü veri kümemiz düğümlere sorduğumuz sorulara göre dallanacak ve ona göre bir ağaç yapısı şekillenmiş olacaktır. Bu nedenle sorduğumuzun soruların veri kümemizi doğru bir şekilde ayırması gerekmektedir. Aksi takdirde dallanma sayımız ve ağacımızın derinliği artacaktır. Böylece karmaşık ve büyük boyutlu yapılar elde etmiş olacağız. Bu tür yapılar genellikle algoritmanın başarı seviyesini düşürmektedir. Bu sorunları aşabilmek için bazı yöntemler geliştirilmiştir. Bunlar Bilgi Kazancı (Information Gain), Gini

Indeksi, Ki-Kare gibi yöntemlerdir (Kothari *et al.* 2001). Özellik seçim yaklaşımları ile ilgili daha detaylı bilgi Bölüm 3.3'te verilmiştir. Karar ağacı makine öğrenmesinin avantaj ve dezavantajları Çizelge 3.6'da verilmiştir. Karar ağacı türleri, hedef değişkenimizin türüne bağlı olarak iki kısma ayrılmaktadır:

- **Sınıflandırma Karar Ağaçları:** Kategorik hedef değişkene sahip olan veri setlerinde sınıflandırma yapmamıza yardımcı olmaktadır. Örnek: Hedef değişkenin "Öğrenci futbol oynar veya oynamaz" olduğu gibi durumlarda öğrenci problemi senaryosunda, cevap "EVET" veya "HAYIR" gibidir.

- **Regresyon Karar Ağaçları:** Sürekli hedef değişkeni mevcut olan veri setlerinde kullanılmaktadır. Örneğin: Bir kişinin aylık geliri gibi vb.

Çizelge 3.6 Karar ağacı makine öğrenmesinin avantajları ve dezavantajları

Avantajlar	Dezavantajlar
• Karar ağaçlarının anlaşılması ve yorumlanabilmesi kolaydır. Yorumlanabilir olması nedeniyle karar ağaçlarını beyaz kutu modeli kategorisine koyabiliriz. Karar ağacımızı daha anlaşılır olması için görsel yapı kullanarak da ifade edebiliriz. • Model oluşturulurken diğer algoritmalara nispeten daha az veri normalizasyonu gerektirir. Kukla değişken oluşturmak veya boş verileri kaldırmakla uğraşmaz. Dağınık verilerle çalışmaya uygundur. Veri hazırlama kısmı çok az veya hiç yok diyebiliriz. Ama eksik değerleri destelemez. • Hem sürekli hem de kategorik verilerle çalışabilir. • Veri setine yeni özellikler ekleyebiliriz.	• Verilerin iyi bir şekilde açıklanmadığı karmaşık yapılı bir ağaç yapısı oluşabilir. Bu duruma aşırı uyum denir. Aşırı uyumu engellemek için geliştirilen yöntemler vardır. • Sınıflandırma karar ağaçları, sınıf dengesizliği olan veri kümelerindeki baskın sınıfı tahmin etme eğilimindedir. • Parametre ayarlarına dikkat etmek gerekir. Eğitim verisindeki en ufak bir değişiklik tamamen farklı bir ağaç oluşturabilir.

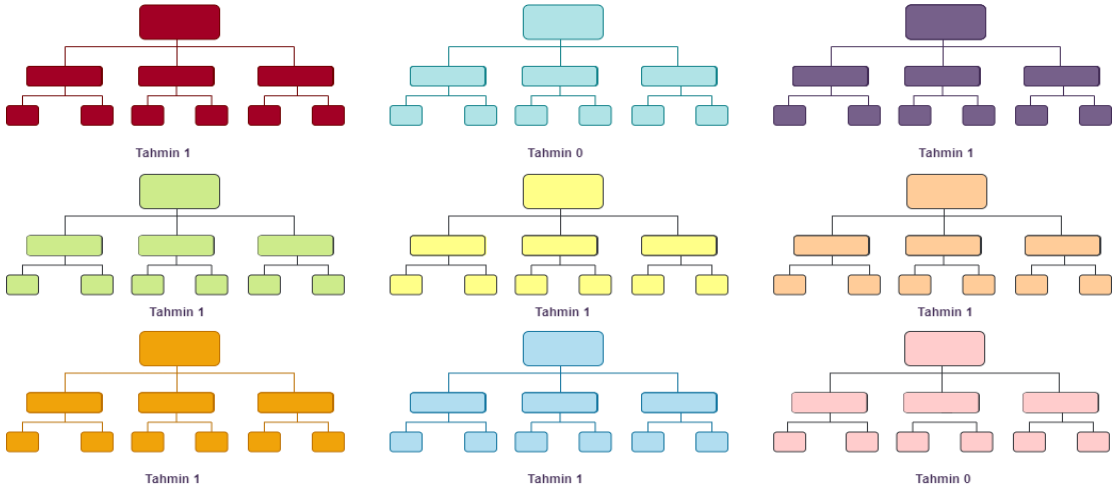
3.2.1.2 Rastgele Orman (Random Forest)

Rastgele ormanlar, çok sayıda karar ağacı kullanarak oluşturulan ve rastgele seçilen parametrelere göre dalları olan bir ağaç yapısıdır. L.Breiman (1996, 2000, 2001, 2004) tarafından hazırlanan birçok makale ve teknik raporda hem regresyon hem de sınıflandırma problemlerini çözmek amacıyla model toplama fikrine dayanan çok popüler bir algoritmadır.

Rastgele Orman torbalama topluluk öğrenmesi modeli uygulandığında her bir karar ağacı bir sınıf tahmininde bulunur ve sonuçlar oylama ile birleştirilerek en çok oyu olan sınıf, modelimizin tahminini belirler. Birbirinden ilişkisiz olan bu karar ağaçlarının sonuçları bireysel tahminlerin herhangi birinden daha doğru olan topluluk tahminleri üretebilir. Rastgele Orman modelinin avantajları ve dezavantajları Çizelge 3.7'de ve tahminde bulunan karar ağaçlarından oluşmuş görsel şeması ise Şekil 3.9'da ve sunulmuştur.

Çizelge 3.7 Rastgele orman makine öğrenmesinin avantajları ve dezavantajları

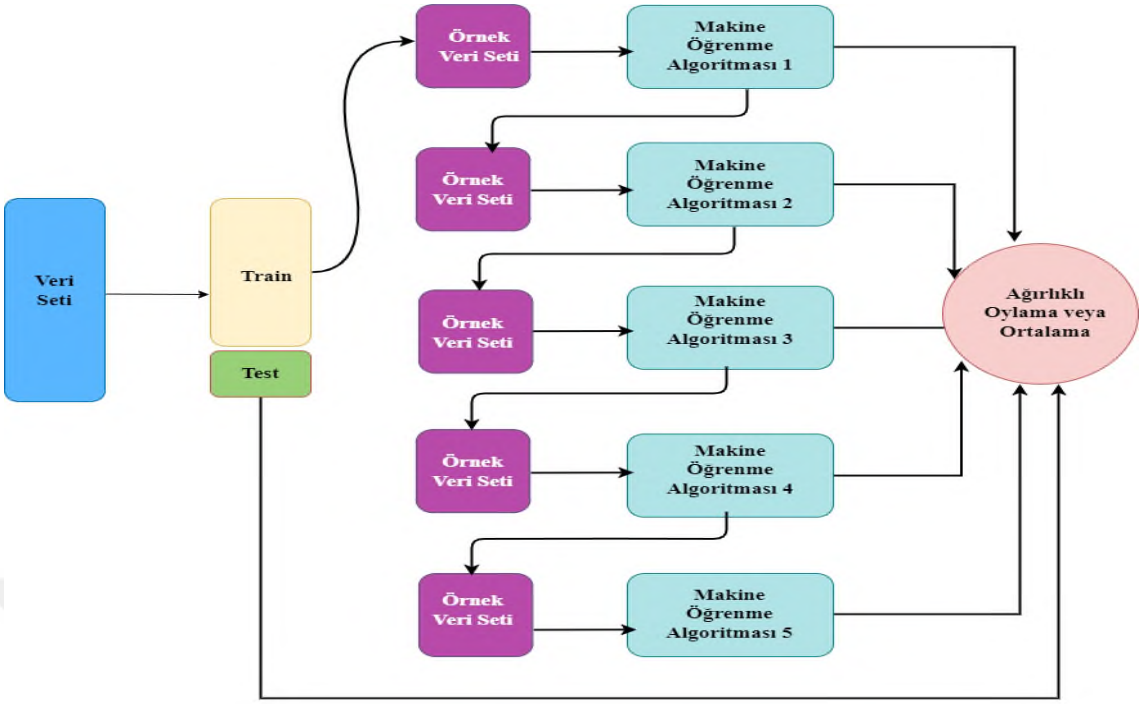
Avantajlar	Dezavantajlar
• Hem regresyon hem de sınıflandırma problemlerinde kullanılabilir olması, • Modelin aşırı uyum (overfitting) riskini ortadan kaldırmak, • Test verileri kullanarak eğitim süresini kısaltılmak, • Büyük veri tabanlarında çok verimli bir şekilde çalışır. Eksik olan veriler hakkında da tahminde bulunur.	• Model oluşturduktan sonra tahmin etme işleminin yavaşlamasıdır.



Şekil 3.9 Rastgele orman model yapısı (Towardsdatascience 2019)

3.2.2 Arttırma Topluluk Öğrenme Modeli (Boosting Ens. Learning Model)

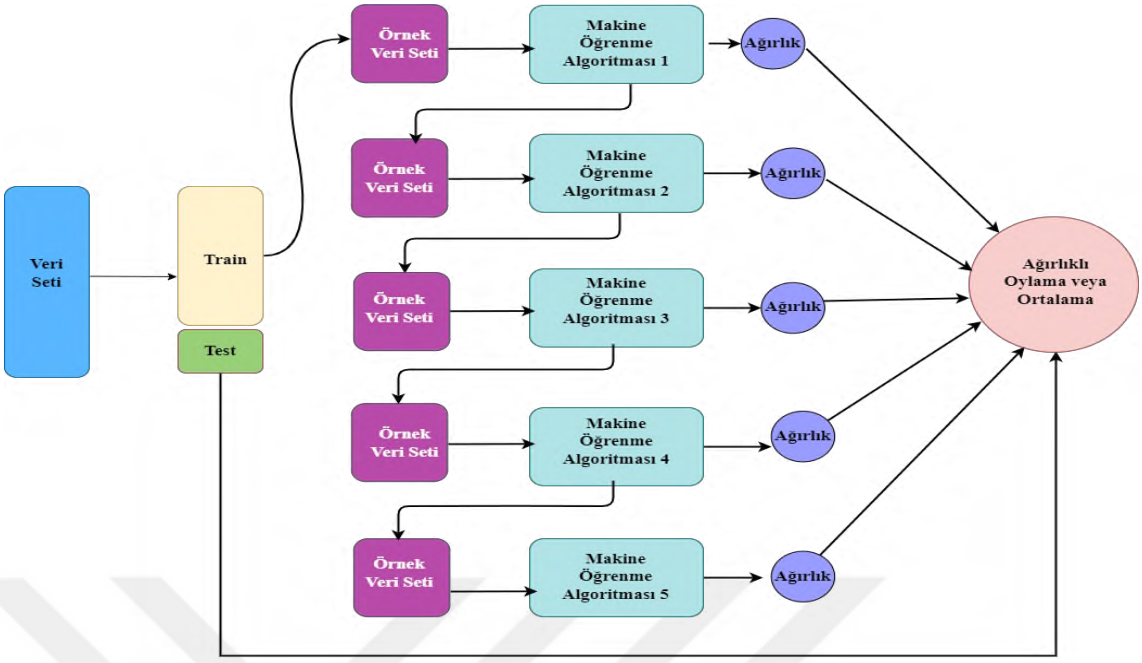
Attırma topluluk (Boosting ensemble) öğrenme modeli literatürde 1990 yılında Robert E. Schapire tarafından zayıf sınıflandırıcıları birleştirerek güçlendirme olarak tanımladığı, daha güçlü bir sınıflandırıcı yapılabileceği fikrini ortaya atmıştır (Schapire 1990). Torbalama topluluk öğrenmesi modelinden farklı olarak bir önceki modelin yaptığı hatayı öğrenerek eğitime devam eder. Yine eğitim veri kümesinden rastgele seçim yapılır. Alınan sonuçlardan yanlış olanlar tespit edilir. Daha sonraki eğitimde yanlış olanlara öncelik verilerek öğrenme süreci hızlandırılmış olunur. Böylece diğer yöntemlere göre hem hızlı çalışır hem de az bellek tüketir. Torbalama topluluk öğrenmesi modelinde zayıf öğrenenler rastgele seçim yapılarak eğitim alırken, Arttırma topluluk öğrenmesi modelinde ise zayıf öğrenenler sırayla eğitim almış olur. Bu yöntemde de temel amacımız varyansı azaltmaktır. Şekil 3.10'da Arttırma topluluk öğrenmesi modelinin yapısı sunulmuştur. Attırma topluluk öğrenmesi modelinin dezavantajı her adımda önceki adımın hatalarını düzeltmesi gerektiği için aykırı değerlere çok bağımlı olmasıdır. Attırma topluluk öğrenmesi modeli olarak AdaBoost (Adaptive Boosting), Gradyan Arttırma (Gradient Boost), XGBoost (eXtreme Gradient Boosting) algoritmalarını söyleyebiliriz.



Şekil 3.10 Arttırmalı topluluk öğrenme model yapısı

3.2.2.1 AdaBoost (Adaptive Boosting)

AdaBoost algoritması uyarlamalı güçlendirici bir topluluk öğrenmesi algoritmasıdır ve ilk başarılı Attırma (boosting) topluluk öğrenmesi modellerindendir (Freund 2001). Yoav Freund ve Robert Schapire yaptıkları çalışmadan dolayı 2003 yılında Gödel Ödülü'nü kazanmışlardır (Eatcs 2003). Genellikle Karar ağacı algoritmasıyla iyi bir şekilde çalışır. Tek seviyeli Karar Ağacı Kütüğü (Decision Tree Stump) olan zayıf modelleri de kullanır. Çalışma şekli ilk önce veri setinde eşit ağırlıklar verir. Veri setimiz N ile gösterildiğini varsayalım 1/N olarak tüm üyelere eşit ağırlıklar verilir. Bir Karar Ağacı ile eğitir. Daha sonra yanlış tahmin olan veri noktalarının ağırlıkları artırır. Doğru tahminde bulunan veri noktalarının ise ağırlıkları düşürür. Sonra tekrardan karar ağacının ağırlıkları hesaplanır. Tekrar eğitim süreci başlar. Bu işlem eklenen ağaç sayısına ulaşana kadar devam eder. Son olarak tahminde bulunulur (Schapire 2013). Şekil 3.11'de AdaBoost topluluk öğrenmesi modelinin yapısı sunulmuştur.



Şekil 3.11 AdaBoost topluluk öğrenme model yapısı

Rastgele Orman algoritmasından farkı burada paralel bir öğrenme süreci değil, sıralı bir öğrenme süreci vardır. Veri noktalarının ağırlıkları eşit değildir. AdaBoost algoritmasının dezavantajı aykırı değerler ve gürültü verilere karşı çok duyarlı olmasıdır (Schapire 2013).

3.2.2.2 Gradyan Arttırma (Gradient Boosting)

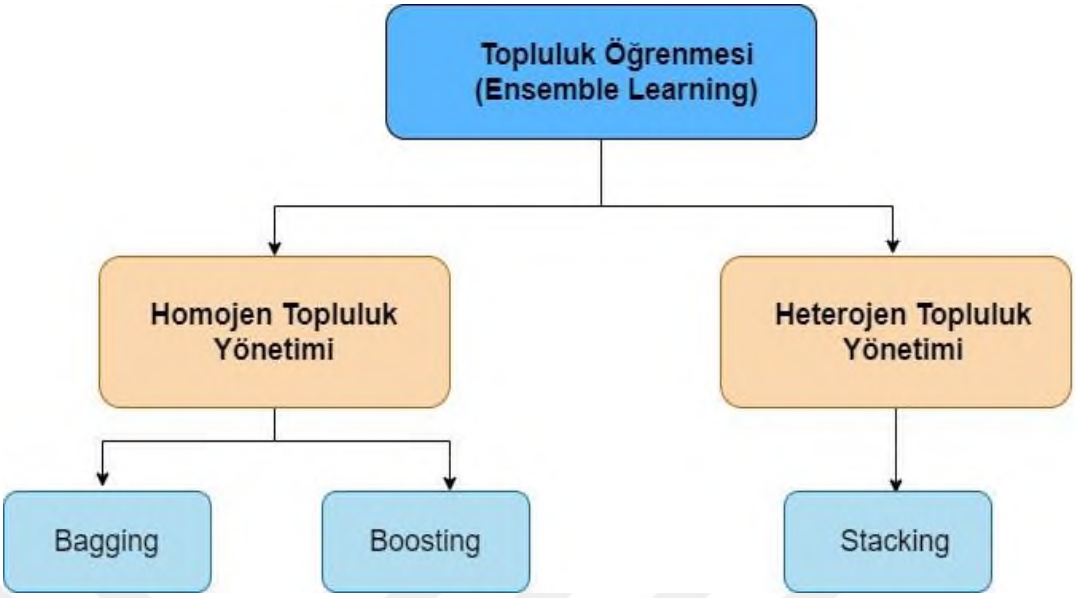
Gradyan Arttırma topluluk öğrenmesi algoritması Jerome H.Friedman tarafından 2001 yılında "Greedy Function Approximation: Gradient Boosting Machine" ve 2002 yılında "Stochastic Gradient Boosting" yazılan makaleler ile tanıtmışlardır (Friedman 2001, Friedman 2002). Jonathan, Frean, Mason, Baxter aynı zamanlarda fonksiyonel gradyan inişi anlattıkları "Boosting Algorithms as Gradient Descent in Function Space" makaleyi yayınlanmışlardır. Gradyan Arttırma diğer bir güçlendirme algoritmasıdır. Modele sırayla eklenen Karar Ağaçlarından oluşan bir topluluk öğrenmesi modelidir. AdaBoost'tan farklı olarak veri noktalara ağırlık vermez. Gradyan Arttırma algoritması kayıp fonksiyonunu en aza indirgemeyi amaçlayarak zayıf öğrenenleri eğitir (Mason *et al.* 2000). Sezgisel olarak yanlış tahminde bulunulan noktalara daha fazla odaklanarak zayıf öğrenenler eklenir.

3.2.2.3 XGBoost (eXtreme Gradient Boosting)

XGBoost algoritması 2016 yılında Washington Üniversitesi'nden Tianqi Chen ve Carlos Guestrin tarafından hazırlanan "XGBoost: A Scalable Tree Boosting System" adlı bir makale ile tanıtmışlardır (Chen and Guestrin 2016). XGBoost, gradyan arttırma yöntemini kullanan bir diğer algoritmadır. Karar ağaçlarından oluşan bir topluluk modelidir. Modelin hızını ve performansı artırmayı hedeflemektedir. Gradyan arttıma tekniklerini kullanan diğer algoritmalara göre hızı neredeyse 10 kat daha hızlıdır. Kullanımı kolaydır. Parametreleri değiştirmek kolaydır. Böylece modele yüksek esneklik kazandırılmış olunur. Eksik değerlerin işlenmesine olanak tanır.

3.2.3 Yığınlama Topluluk Öğrenmesi Modeli (Stacking Ens. Learning Model)

Yığınlama (Stacking) modeli iki ya da daha fazla makine öğrenmesi algoritması tarafından oluşturulmuş topluluk öğrenmesi modelidir (Witten *et al.* 2016). Torbalama ve arttırma topluluk öğrenmesi modelleri homojen bir yaklaşımla model oluştururken yığınlama topluluk öğrenmesi modeli ise heterojen bir yaklaşımla model oluşmaktadır. Homojen yaklaşımla model oluşturmak istendiğinde aynı tür makine öğrenmesi algoritmalarını tercih edilir fakat heterojen yaklaşımla model oluşturulmak istenildiğinde farklı tür makine öğrenmesi algoritmaları bir arada kullanılarak model oluşturulmaktadır (Elish *et al.* 2013). Topluluk öğrenmesi modelinde homojen ve heterojen yaklaşım yapısı Şekil 3.12'de sunulmuştur.



Şekil 3.12 Topluluk öğrenme model yapısı (Dataaspirant 2020)

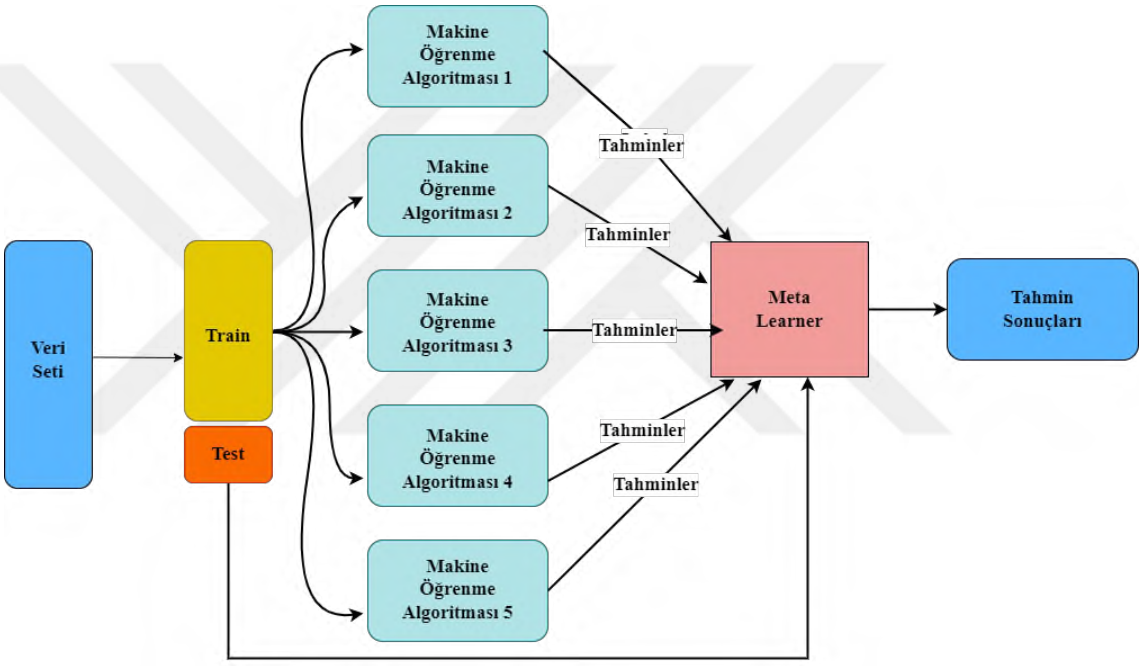
Yığınlama modelinde Oylama (Voting), Ağırlıklı Ortalama (Weighted Avarage), Harmanlama (Blending), Yığınlama (Stacking), Süper Öğrenci (Super Learner) gibi algoritmalar kullanılır. Bu makine öğrenmesi algoritmalarından aşağıda kısaca bahsedilmiştir.

3.2.3.1 Oylama (Voting)

Kullanımı en kolay topluluk öğrenmesi modellerinden biridir. Sınıflandırma işlemlerinde oylama yöntemi kullanılırken, regresyon işlemlerinde ortalama alınarak tahminde bulunur. Model içerisinde kurulan diğer modellere aynı düzeyde yetenekli olduğu varsayılır ve basit istatistikler kullanılarak birleştirme işlemi yapılır (Witten *et al.* 2016). Oylama (Voting) topluluk öğrenmesi modeli Sert Oylama (Hard Voting) ve Yumuşak Oylama (Soft Voting) olmak üzere iki ayrı kısımdan oluşmaktadır. Her bir modelden gelecek olan tahmin bir oydur. Yumuşak Oylama sınıflandırmada işleminde her sınıfa yapılan tahmin olasılık değerlerinin toplanması ve bunların içinde en fazla olasılığa sahip olan sınıfın etiketlenmesiyle yapılırken, Sert Oylamada en çok oya sahip olan sınıfın etiketlenmesiyle yapılmaktadır (Brownlee 2021).

3.2.3.2 Yığınlama (Stacking)

Ağırlıklı Ortalama (Weighted Avarage), Oylamanın (Voting) biraz daha geliştirilmiş halidir. Bu yöntemde Oylama'dan farklı olarak kurulan model üzerinde her alt modele eşit ağırlık verilmemektedir. Modele katkısı olan alt modellerin ağırlıkları farklı değerlendirilir. Bu yöntemle model üzerinde iyileştirilme yapılması hedeflenmiştir (Sollich and Krogh 1995). Şekil 3.13'te Yığınlama topluluk öğrenmesi modelinin yapısı sunulmuştur.



Şekil 3.13 Yığınlama topluluk öğrenme model yapısı

3.2.3.3 Harmanlama (Blending)

Harmanlama (Blending) model yığınlama topluluk öğrenmesi modelin farklı bir yaklaşımla oluşturduğu bir modeldir. Harmanlama, Netflix yarışmalarında kazanan ekipler tarafından kullanılan bir terimdir (Töscher *et al.* 2009). Yığınlanmış Genelleme (Stacked Generalization) yöntemine genel olarak çok benzeyen bir yöntemdir. Bazı uygulamalarda model olarak birbirlerinin yerine kullanılır (Sill *et al.* 2009). Harmanlama topluluk öğrenmesi modelin mimarisi iki seviyeden oluşur. Seviye-0 temel

modellerin eğitildiği kısımdır. Seviye-1 ise temel modellerin tahminlerinin meta modele giriş olarak verildiği kısımdır. Veri seti test ve doğrulama olmak üzere iki kısma ayrılır. Harmanlama topluluk öğrenmesi modelinde eğitim veri setinin bir kısmını bekleme alanı olarak ayırır ve burada eğitim yapar (Michailidis 2017). Meta model olarak genelde Lojistik Regresyon veya Lineer Regresyon gibi doğrusal modeller kullanılır.

3.2.3.4 Süper Öğrenci (Super Learner)

Super Learner topluluk öğrenmesi model de Harmanlama (Blending) topluluk öğrenmesi modeli gibi farklı bir yaklaşımla oluşturulmuş topluluk öğrenmesi modelidir. Super Learner topluluk öğrenmesi modeli 2007 yayınlanan "Super Learner" adlı makale ile tanıtmışlardır. Bu makaleyi California Üniversitesi Berkeley'den Mark Van der Laan, Alan Hubbard ve Eric Polley tarafından kaleme alınmıştır (Van Der Laan *et al.* 2007). Super Learner topluluk öğrenmesi model mimarisi aşağıdaki gibidir:

1.) Temel makine öğrenmesi algoritmaları bir listede oluşturulur. Bu listeyi L olarak tanımlayalım.

2.) Meta Learner olarak Lojistik Regresyon veya Lineer Regresyon gibi doğrusal modeller tercih edilir.

3.) Temel makine öğrenmesi modelleri üzerinde k-kat çapraz doğrulama gerçekleştirilir. Bunların tahminleri toplanır. Her bir modelden çapraz doğrulama yapılarak elde edilen tahmini değerler n x L şeklinde bir matris tutulur. Bu matrise "birinci düzey" veriler olarak tanımlanır.

4.) Son olarak meta learner'a bu tahminler verilir.

Super Learner aşırı uyumu önlemek ve modellerin tahmin doğruluğunu artırmak için kullanılan bir yöntemdir. Temel seviyede makine öğrenmesi algoritmalarının en uygun kombinasyonunu bulmayı hedeflemişlerdir. Biz bu çalışmamızda hem Seviye-0'da hem

de Seviye-1'de farklı makine öğrenmesi algoritmalarının kombinasyonlarını deneyerek performanslarını karşılaştırdık. Deney sonuçları hakkında detaylı bilgi Bölüm 5.2'de verilmiştir.

3.3 Özellik Seçim Yaklaşımları

Bir düğümü daha homojen bir yapı elde edebilmek için alt düğümlere böleriz. Bu bölme işlemi hedef değişkenimizin türüne bağlı olarak iki gruba ayrılır.

3.3.1 Kategorik Hedef Değişken

3.3.1.1 Bilgi Kazanımı

Bilgi kazanımı daha iyi anlayabilmemiz için öncelikle entropi kavramını incelememiz gerekmektedir.

Entropi (Entropy): Rastgele seçilen bir değişken için hesaplanan bilgi miktarının ölçüsüdür. Veri kümemizdeki safsızlık veya rastlantısallığı ölçmeye çalışır. Bir kutuda 100 tane mavi top olduğunu düşünelim. Bu kutudan bir top seçtiğimizde entropisinin sıfır olduğunu söyleyebiliriz. Kutudan 50 top alıp, 20 sarı ve 30 kırmızı top ile değiştirelim. Tekrar kutudan bir top seçtiğimizde bunun mavi olma olasılığı 1'den 0,5'e düşmüş olacaktır. Entropi artmış olur. Entropi ne kadar düşük olursa elde edilen bilginin ne olduğunu tahmin edebiliriz. Ama entropi artıkça bilginin ne olduğunu tahmin edebilmek gittikçe zorlaşır. Entropi değeri sıfır olması onun bir yaprak düğümü olduğunu gösterir. Eğer sıfırdan büyükse bölünmeye ihtiyacı olduğunu gösterir. Shannon'a göre Entropi'nin hesaplanması için Denklem (3.10)'daki matematiksel gösterimi verilmiştir (Shannon 1948).

$$H(X) = -\sum_{i=1}^{n} P(x_i)\, log_2\, P(x_i) \tag{3.10}$$

Denklem (3.10)'da $P(x_i)$ veri kümemizdeki (x_i) sınıfının tüm sınıf içindeki gelme olasılığıdır. Entropi için maksimum değer, sınıf sayısına bağlıdır.

- 2 sınıf için maksimum entropi değeri 1'dir.
- 4 sınıf için maksimum entropi değeri 2'dir.
- 8 sınıf için maksimum entropi değeri 3'tür.
- 16 sınıf için maksimum entropi değeri 4'tür.

Bilgi Kazanımı (Information Gain): Veri seti içindeki hangi özniteliğinin entropi kavramına göre maksimum bilgi sağladığını ölçmek için kullanılır (Srinivas 2013). Entropi kavramı bilgi kazanımının hesaplanmasında büyük rol oynamaktadır. Kök düğümden yaprak düğüme kadar entropi düzeyini düşürmeye çalışır. Bilgi kazanımını ölçmek için matematiksel gösterimi Denklem (3.11)'de verilmiştir.

$$Bilgi\ Kazanımı\ (Y,X) = Entropi\ (Y) - \sum P\ (Y|X) * Entropi(Y|X) \tag{3.11}$$

Veri setinde bulunan her özellik için bilgi kazanımı tek tek hesaplanır. Bunların arasından bilgi kazanımı en yüksek olan kök düğüm olarak belirlenir. Daha sonra aynı işlem diğer düğümler için tekrar edilir. Yaprak düğümlere varıldığında işlem sona ermiş olur.

3.3.1.2 Gini Safsızlık (Gini Index)

Veri setindeki rastgele seçilmiş bir değişkenin yanlış sınıflandırma olasılığının sıklığını ölçer. Her bir sınıfın olasılıklarının karelerini toplar. Ve daha sonra bunu 1'den çıkararak hesaplama yapar. Gini indeksinin hesaplanması için matematiksel gösterimi Denklem (3.12)'de verilmiştir.

$$Gini\ Index = 1 - \sum_{i=1}^{n} (P(x_i))^2 \tag{3.12}$$

Denklem (3.12)'de P(x_i) bir değişkenin farklı bir sınıf için sınıflandırma olasılığını gösterir. Gini indeks değeri sıfır ile bir arasında değer alır. Bu değer sıfıra ne kadar yakınsa o kadar iyi bir sınıf ayrımı yapmış olur. Bu yüzden bölme işlemi için gini indeksi düşük olan bir özellik seçilir. Gini indeksine hesaplama maliyeti yönünden bakacak olursak logaritma fonksiyonu kullanılmadığı için daha kolay ve daha hızlıdır. Bu nedenle bilgi kazanımının (information gain) yerine Gini indeksi tercih edilir.

3.3.1.3 Ki-Kare (Chi-Square)

Kök düğüm ile alt düğümler arasındaki farkın istatiksel önemi üzerinde çalışır. İki veya daha fazla bölme işlemi yapabilir. Hedef değişkenin gözlemlenen ve beklenen frekansları arasındaki standartlaştırılmış farkların karelerinin toplamı ile hesaplanır. Ki-Kare'nin hesaplanması için matematiksel gösterimi Denklem (3.13)'te verilmiştir.

$$\mathcal{X}^2 = \sum \frac{(O-E)^2}{E} \tag{3.13}$$

Denklem (3.13)'te E (Beklenen değer), kök düğümümüzdeki sınıfların dağılımına göre bir alt düğümdeki bir sınıf için beklenen değerdir. O (Gerçek değer) ise, bir alt düğümdeki bir sınıf için bulunan gerçek değerdir (Analytics Vidhya 2016).

3.3.1.4 Kazanç Oranı (Gain Ratio)

Kazanç oranının hesaplanması için matematiksel gösterimi Denklem (3.14) ve Denklem (3.15)'te verilmiştir.

$$Kazanç\ Oranı = \frac{Bilgi\ Kazanımı}{Bölünmüş\ Bilgi} \tag{3.14}$$

$$Kazanç\ Oranı = \frac{Entropi\ (Y) - \sum_{i=i}^{n} P\ (Y|X) * \ Entropi(Y|X)}{-\sum_{i=1}^{n} P(x_i)\ log_2\ P(x_i)} \quad (3.15)$$

3.3.2 Sürekli Hedef Değişkeni

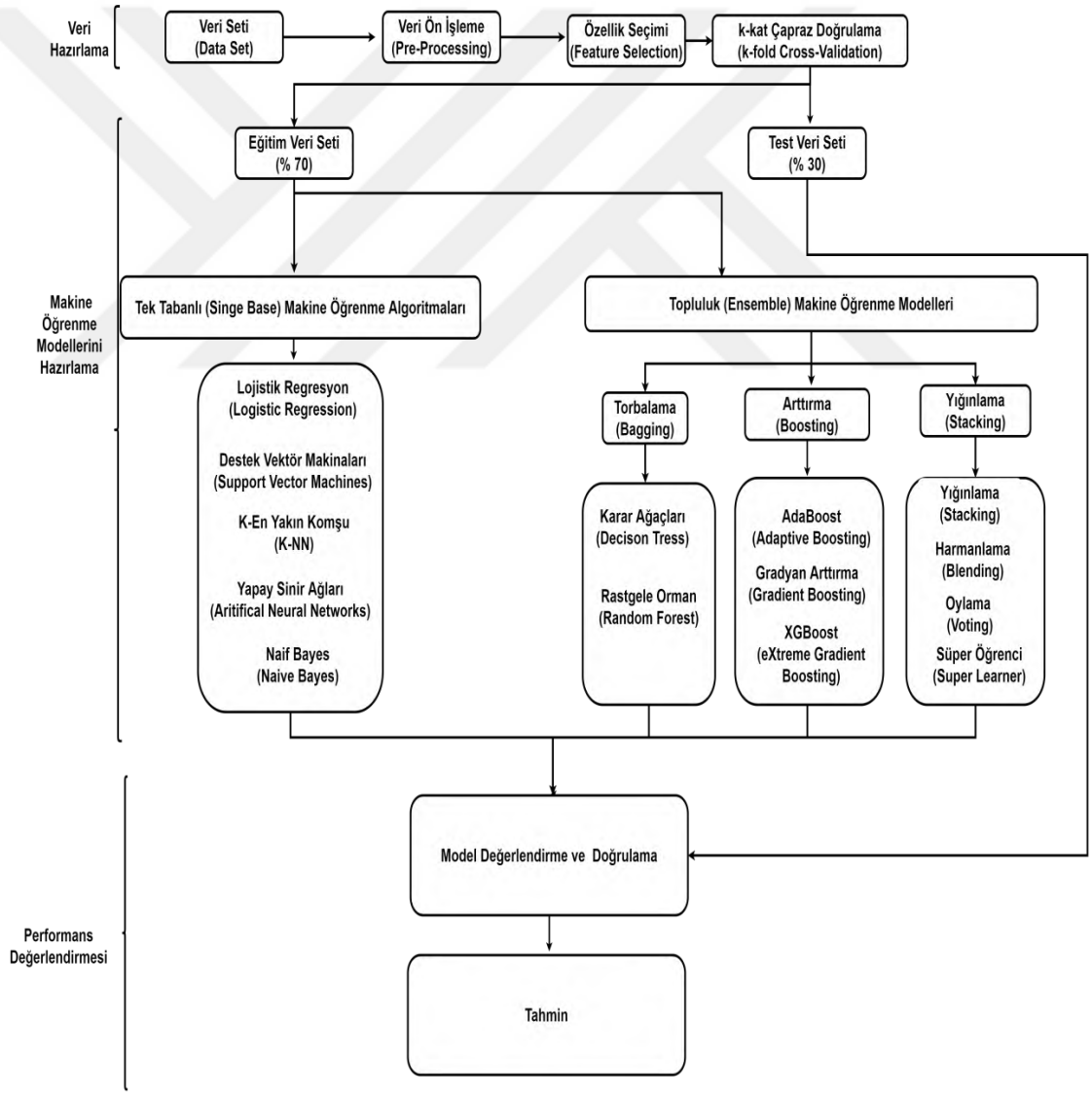
3.3.2.1 Varyans Azaltma

Bu yöntemde düğümleri alt düğümlere ayırırken karar vermemizi sağlayan bir ölçü olarak standart varyans formülü kullanılır. Regresyon problemlerinde, yani değişkenimizin sürekli olduğu durumlarda kullanılan bir yöntemdir. Varyans bir düğümdeki homojenliği ölçmek amacıyla kullanılır. Düğümdeki homojenlik ne kadar fazla ise varyansı da sıfıra o kadar yakındır. Varyans azaltmanın hesaplanması için matematiksel gösterimi Denklem (3.16)'da verilmiştir.

$$Varyans = \frac{\sum(X - \overline{X})^2}{n} \quad (3.16)$$

4. MATERYAL VE METOT

Bu bölümde çalışma yaptığımız veri seti üzerinde uyguladığımız aşamalar anlatılacaktır. Diyabet hastalığında teşhis koymak amacıyla hem tek tabanlı (Lojistik Regresyon, Destek Vektör Makineleri, K-En Yakın Komşu, Yapay Sinir Ağları ve Naif Bayes) makine öğrenmesi algoritmaları hem de topluluk (torbalama, arttırma, yığınlama) öğrenmesi modelleriyle birlikte çalışılmıştır. Önerilen modelde topluluk öğrenmesi modellerinden Super Learner modeliyle %99,4 oranında bir başarı elde edilmiştir. Şekil 4.1'de önerilen modelin yapısı sunulmaktadır.



Şekil 4.1 Önerilen model yapısı

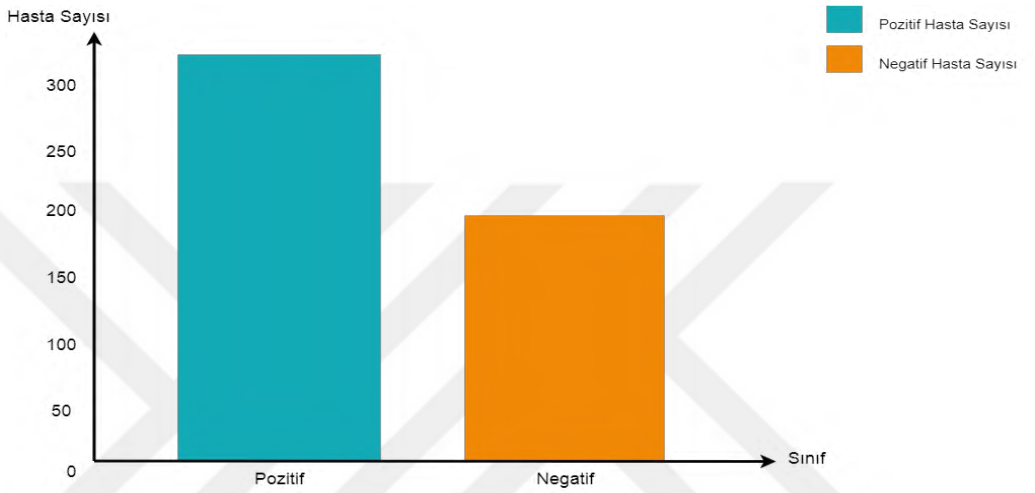
4.1 Veri Seti Açıklaması

Bu çalışmada kullanılan "Early stage diabetes risk prediction" veri seti Bangladeş, Sylhet'teki Sylhet Diyabet Hastanesindeki diyabet hastalarına yapılan doğrudan anket sonuçlarının doktor tarafından onay alınmasıyla hazırlanmıştır. Veriler 12.07.2020 tarihinde UCI'ye bağışlanmıştır. UCI Machine Learning Repository tarafından ücretsiz olarak bu veri setine ulaşılabilir (UCI Machine Learning Repository, 2020). Veri seti yeni diyabet hastası olan veya diyabet hastası olabilecek semptomları taşıyan verilerden oluşmaktadır. Toplam 520 hastadan alınan bilgiler doğrultusunda 16 öznitelik ve bir sınıf bilgisine sahip bir veri setidir. Veri setindeki özniteliklerle ilgili detaylı bilgi Çizelge 4.1'de gösterilmektedir.

Çizelge 4.1 Veri seti öznitelik bilgileri

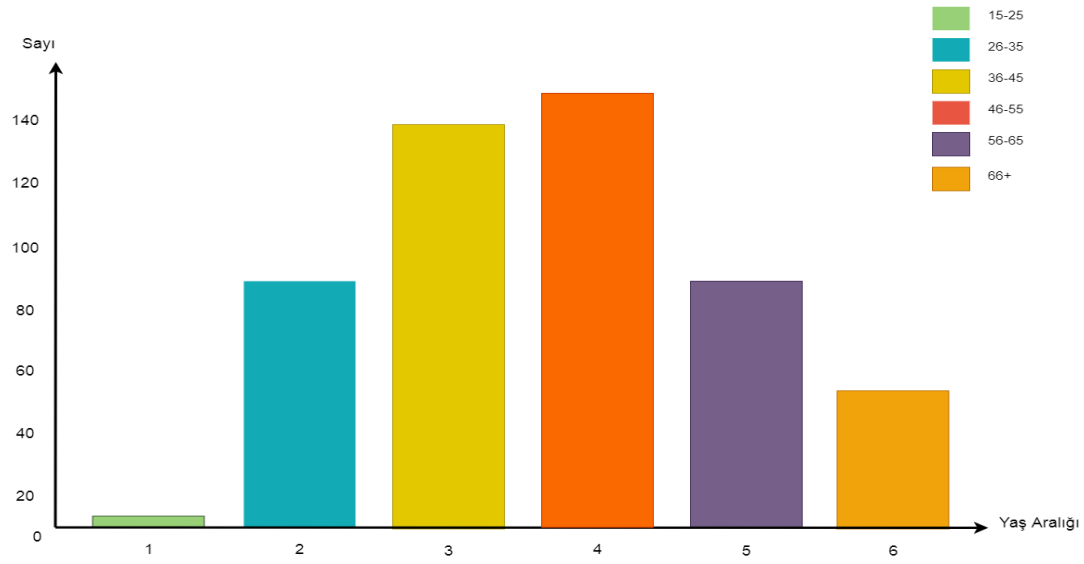
Sıra No	Öznitelik	Açıklama
1	Yaş (Age)	1. 20-35, 2. 36-45, 3. 46-55, 4. 56-65, 5. 65 üstü
2	Cinsiyet (Gender)	1. Erkek, 2. Kadın
3	Poliüri (Polyuria)	1. Evet, 2. Hayır
4	Polidipsi (Polydipsia)	1. Evet, 2. Hayır
5	Ani kilo kaybı (Sudden Weight Loss)	1. Evet, 2. Hayır
6	Halsizlik (Weakness)	1. Evet, 2. Hayır
7	Polifaji (Polyphagia)	1. Evet, 2. Hayır
8	Genital Pamukçuk (Genital Thrush)	1. Evet, 2. Hayır
9	Görsel Bulanıklık (Visual Blurring)	1. Evet, 2. Hayır
10	Kaşıntı (Itching)	1. Evet, 2. Hayır
11	Sinirlilik (Irritability)	1. Evet, 2. Hayır
12	Geç İyileşme (Delayed Healing)	1. Evet, 2. Hayır
13	Kısmi Parezi (Partial Paresis)	1. Evet, 2. Hayır
14	Kas Sertliği (Muscle Stiffness)	1. Evet, 2. Hayır
15	Alopesi (Alopecia)	1. Evet, 2. Hayır
16	Obezite (Obesity)	1. Evet, 2. Hayır
17	Sınıf (Class)	1. Pozitif, 2. Negatif

Sınıf özellik bilgisindeki pozitif değer hastanın diyabetli olduğunu negatif değer ise hastanın diyabetli olmadığını göstermektedir. Diyabetli olan pozitif değerli hasta sayımız 320 negatif değerli hasta sayımız ise 200'dür. Şekil 4.2'de veri setindeki diyabet dağılımı grafiksel olarak sunulmuştur. Veri setinde 328'si erkek ve 192'si kadın hasta olmak üzere toplam 520 hasta kayıt bilgisi bulunmaktadır. Diyabet hastası olan erkek hastalarının oranı %45 ve kadın hastaların oranı ise %90'dır.



Şekil 4.2 Veri setindeki diyabetli hasta ve diyabetli olmayan hasta dağılımı

Şekil 4.3'te grafiksel olarak yaş dağılımını gösterilmektedir. Yaş öznitelik bilgisi 5 gruba ayrılmıştır. Hastaların yaş aralığı 16 ile 90 arasında değişkenlik göstermektedir. Diğer öznitelik bilgileri ise 2 gruba ayrılmıştır. Bu özniteliklerden "Evet" seçeneği semptomun var olduğunu "Hayır" seçeneği ise semptomun olmadığı anlamına gelmektedir.



Şekil 4.3 Veri setindeki yaş dağılımı

4.2 Keşfedici Veri Çözümlemesi (Exploratory Data Analysis)

Üzerinde çalıştığımız veri setindeki veriler hakkında bilgi sahibi olacağımız veri analiz kısmıdır. Bu verileri makine öğrenmesi algoritmalarında girdi verisi olarak kullanabilmemiz için verileri hazır hale getirmemiz gerekmektedir (Tukey 1977).

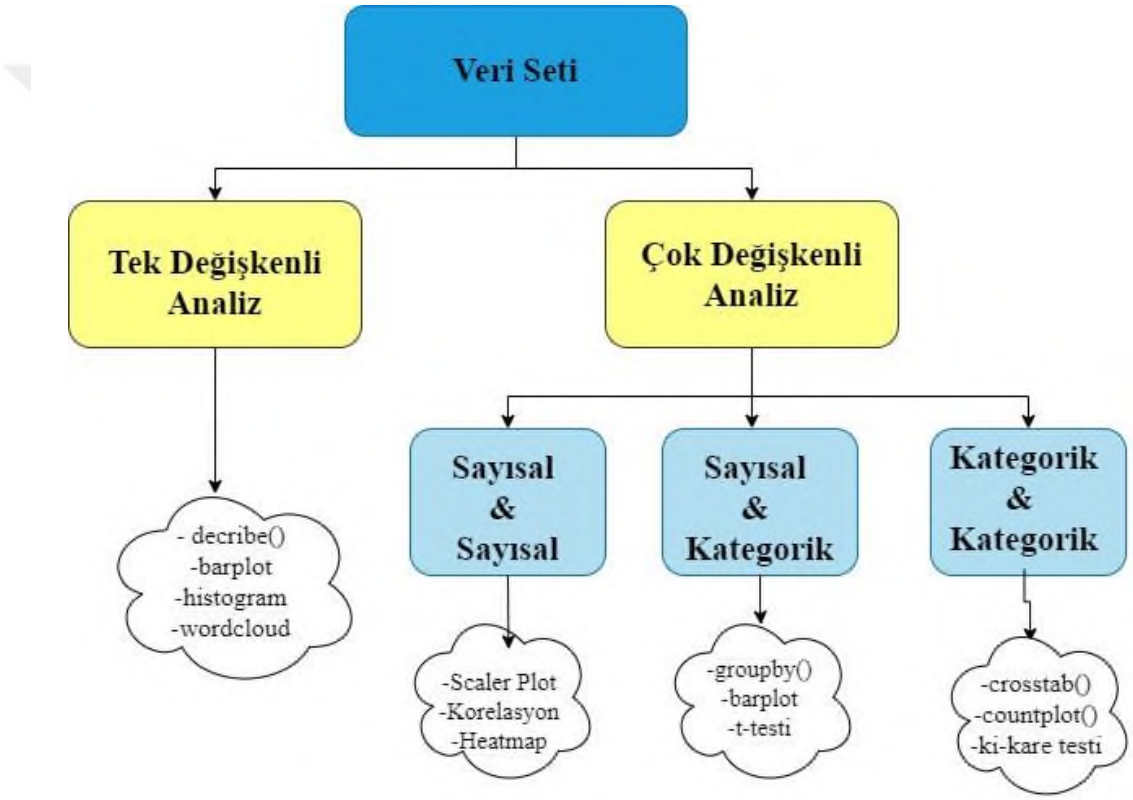
4.2.1 Veri Temizliği

Veri setinden gereksiz bilgilerin, boş ve aykırı değerlerin temizleme işlemi yapıldığı kısımdır. Bu yapılan işlemler modelin daha sağlıklı tahminlerde bulunma olasılığını arttırmış olacaktır. Kullandığımız veri setimizde eksik değer/boş değer bulunmamaktadır.

- Kullanılmayan/gereksiz satırlarda bulunmamaktadır.
- Sütun isimlerinde gerekli düzenlemeler yapılmıştır.
- Sütunlardaki verilerimiz kategorik olarak “Evet/Hayır” bilgisi içermekteydi. Bu verileri “1/0” olarak yeniden tanımlanmıştır.

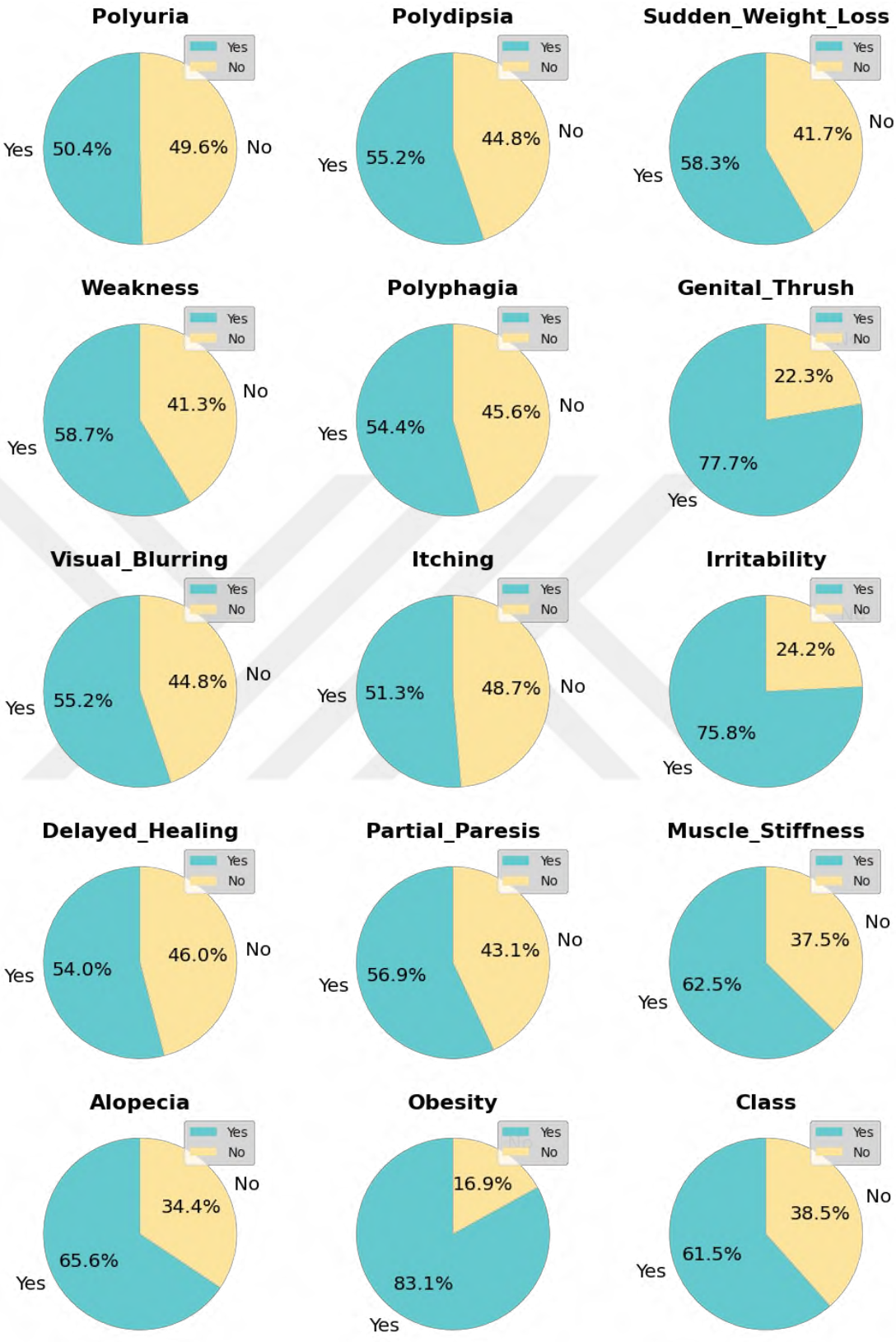
4.2.2 Veri Analizi

Veri setindeki veriler arasında nasıl bir ilişki olduğunu anlamamız için veriler üzerinde detaylı istatiksel teknikler kullandığımız ve sonuçlarını görselleştirdiğimiz aşamadır. Veri keşfi tek değişkenli analiz ve çok değişkenli analiz olmak üzere iki aşamadan oluşmaktadır. Tek değişkenli analiz sadece tek bir değişken ile yapılan analiz işlemidir. Çok değişkenli analiz ise birden fazla değişken arasında yapılan analiz işlemidir. Şekil 4.4'te veri keşif analizini göstermektedir.



Şekil 4.4 Veri keşif analizi (Medium Artar M.A. 2020)

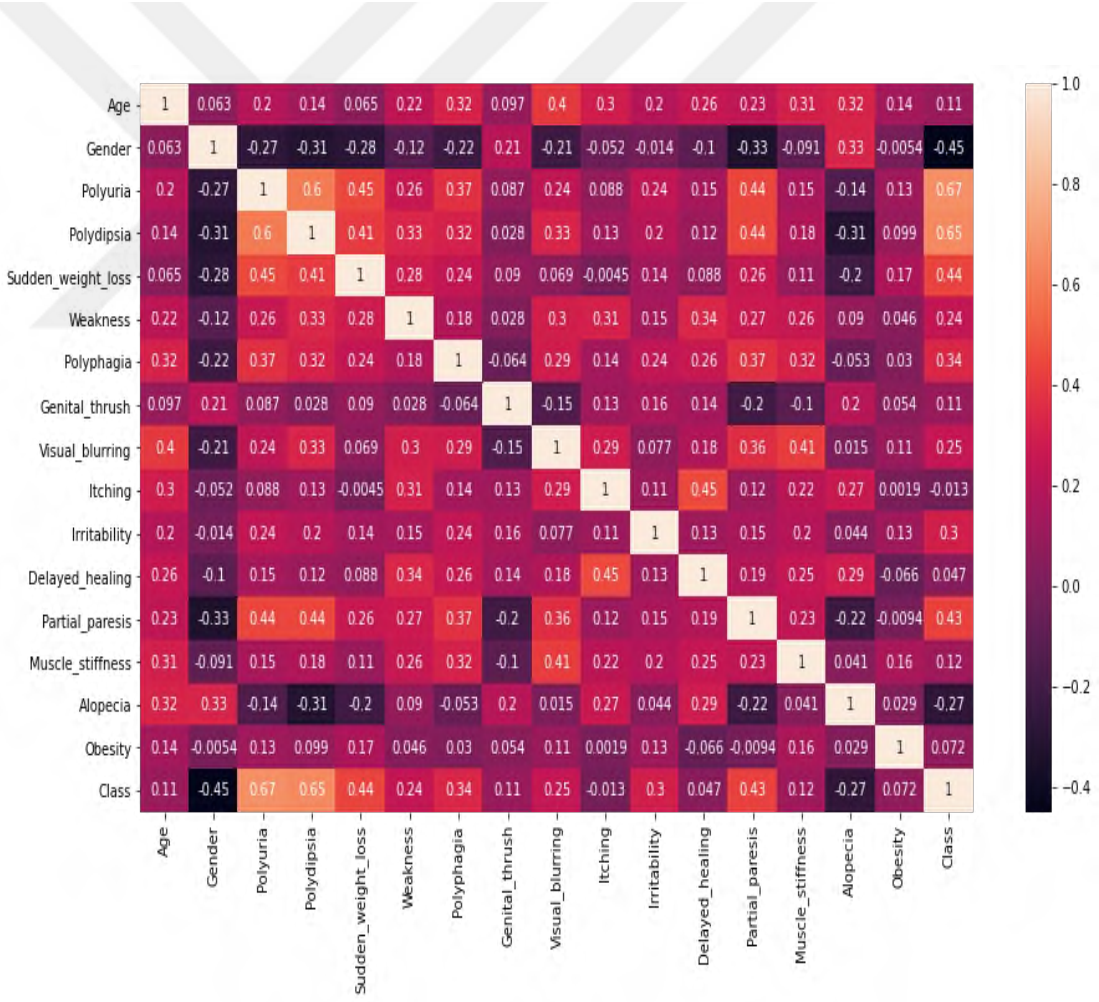
Veri setindeki özniteliklerin diyabetli olup olmadığını grafiksel olarak gösteren dağılım Şekil 4.5'te verilmiştir.



Şekil 4.5 Veri dağılımı

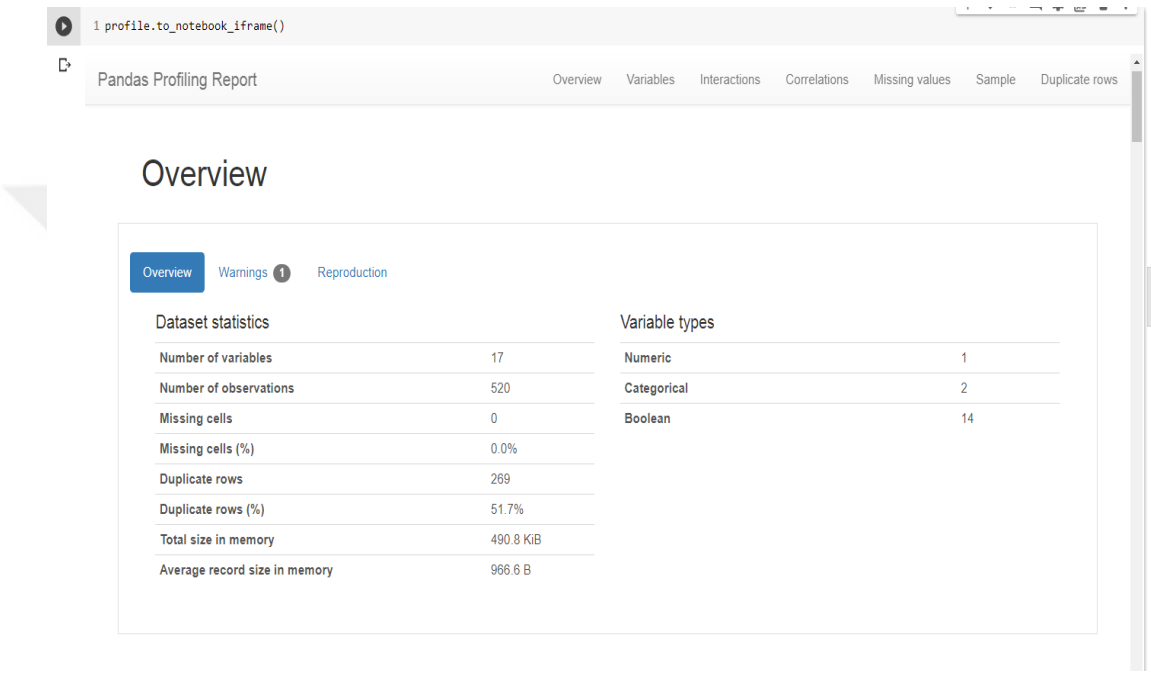
4.3 Korelasyon Matrisi ve Isı Haritası

Tüm öznitelik çiftlerinin korelasyonları hesaplanarak ısı haritası korelasyon grafiği Şekil 4.6'da görselleştirilmiş olarak sunulmuştur. Aşağıdaki ısı haritasında, daha parlak renkler daha fazla korelasyonu gösterirken, daha koyu renkler ise negatif korelasyonu göstermektedir. Korelasyon katsayı değeri (-1) ile (+1) arasında değerler almaktadır. Korelasyon değerimiz 1 değerine ne kadar yakınsa pozitif yönlü bir ilişkiyi, (-1) değerine daha yakınsa negatif yönlü bir ilişkiyi, sıfır ise ilişkinin olmadığını anlamına gelmektedir. Diyabet veri setinde öznitelikler arasından en yüksek korelasyon katsayısına sahip olan Poliüri (0,67) ve Polidipsi (0,65) olduğunu Şekil 4.6'da görülmektedir.



Şekil 4.6 Özellik korelasyonlarının ısı haritası

Veri setimizdeki veri ve modeller hakkında daha fazla bilgi sahibi olabilmek için birçok yöntem mevcuttur. Pandas kütüphanesindeki Profile Report modülünü kullanarak daha kapsamlı ve derinlemesine bir bilgiye ulaşabiliriz. Bu modülde eksik veriler, değişkenler, korelasyon gibi çeşitli verilerin grafiksel olarak sunulduğu HTML bir rapor oluşturur. Bu şekilde fazla vakit kaybetmeden veri keşif analizini yapmış oluruz. Şekil 4.7'de Profile Report modülünün ekran görüntüsü verilmiştir.



Şekil 4.7 Profil Reports

4.4 Öznitelik Seçimleri

Veri setinde 17 öznitelik ve bir sınıf değeri bulunmaktadır. Bu öznitelikler içinde hangisinin modele daha çok katkısı olduğunu tespit edebilmek için öznitelik seçim tekniklerini kullanarak en yüksek skora sahip olanlar seçilip modelin performansı gözlemlenmiştir. Bu çalışmamızda veri setimiz çok değişkenli ve kategorik-kategorik veri tipine sahip olduğu için öznitelik seçimi olarak Ki-Kare (Chi-Square) tekniği tercih edilmiştir. Çizelge 4.2'de verilen özniteliklerden en yüksek skora sahip olan öznitelikten başlayarak bir öznitelik veri listesi oluşturulmuştur. Daha sonra diğer öznitelikleri de ekleyerek toplam dokuz öznitelik veri listesi elde edilmiştir.

Çizelge 4.2 Öznitelik skor listesi

Öznitelik Adı	Skor
Polydipsia	120.785
Polyuria	116.184
Sudden_weight_loss	57.749
Partial_paresis	55.314
Gender	38.747
Irritability	35.334
Polyphagia	33.198
Alopecia	24.402
Age	18.124

Çizelge 4.3'te oluşturulan veri listesi ve içerdikleri öznitelik bilgileri verilmiştir.

Çizelge 4.3 Öznitelik listesi

Veri Liste Adı	Öznitelikler
Veri_Listesi_1	Polydipsia
Veri_Listesi_2	Polydipsia, Polyuria
Veri_Listesi_3	Polydipsia, Polyuria, Sudden_weight_loss
Veri_Listesi_4	Polydipsia, Polyuria, Sudden_weight_loss, Partial_paresis
Veri_Listesi_5	Polydipsia, Polyuria, Sudden_weight_loss, Partial_paresis, Gender
Veri_Listesi_6	Polydipsia, Polyuria, Sudden_weight_loss, Partial_paresis, Gender, Irritability
Veri_Listesi_7	Polydipsia, Polyuria, Sudden_weight_loss, Partial_paresis, Gender, Irritability, Polyphagia
Veri_Listesi_8	Polydipsia, Polyuria, Sudden_weight_loss, Partial_paresis, Gender, Irritability, Polyphagia, Alopecia
Veri_Listesi_9	Polydipsia, Polyuria, Sudden_weight_loss, Partial_paresis, Gender, Irritability, Polyphagia, Alopecia, Age

Çizelge 4.4‘deki sonuçlara bakacak olursak eklenen her öznitelik ile birlikte öznitelik skor puan performansında bir artma meydana gelmiştir. Ve en iyi dokuz öznitelik ile oluşturulan Veri_Listesi_9’da öznitelik skor puanı diğerlerinden daha yüksek performans sonucu vermiştir. Bizim oluşturduğumuz modelde bu dokuz öznitelik (Polydipsia, Polyuria, Sudden_weight_loss, Partial_paresis, Gender, Irritability, Polyphagia, Alopecia, Age) kullanılmıştır.

Çizelge 4.4 Seçilen özniteliklerin performans sonuçları

Veri Liste Adı	Skor
Veri_Listesi_1	0.831
Veri_Listesi_2	0.907
Veri_Listesi_3	0.918
Veri_Listesi_4	0.937
Veri_Listesi_5	0.954
Veri_Listesi_6	0.9642
Veri_Listesi_7	0.9648
Veri_Listesi_8	0.9706
Veri_Listesi_9	0.9815

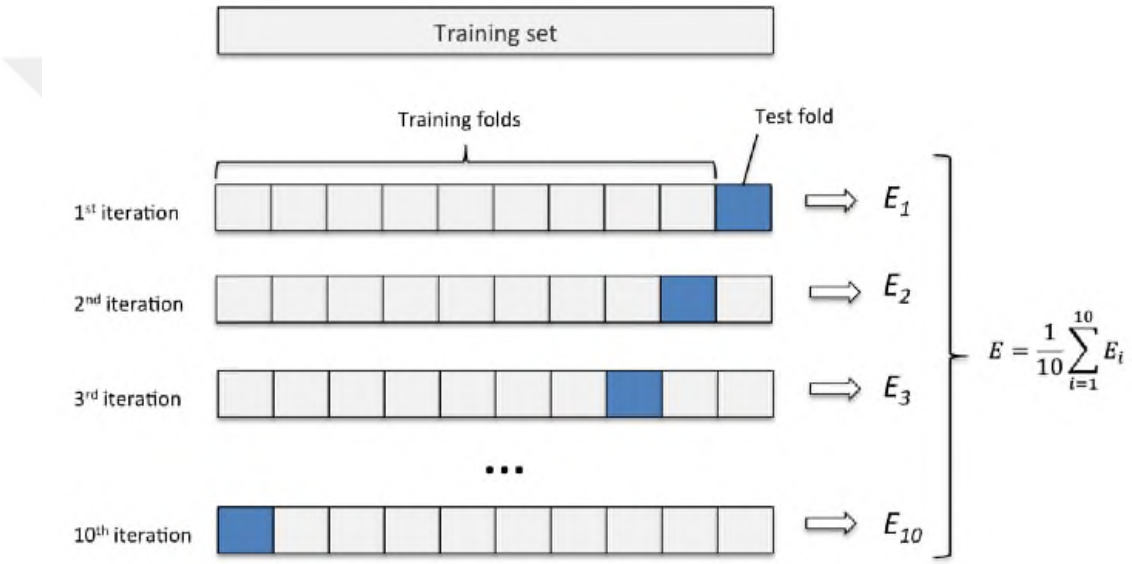
4.5 Veri Setini Bölme ve Doğrulama Yöntemleri

Makine öğrenmesi algoritmaları veri seti üzerinde eğitilirken modelin değerlendirilmesi ve performansını ölçmek için aşağıdaki yöntemler kullanılmaktadır. Aynı zamanda aşırı uyum (overfitting) durumunu engellemek için de tercih edilmektedir.

4.5.1 Çapraz doğrulama yöntemi

Çapraz doğrulama yöntemi veri setini k adet parçalara ayırmaktadır. Her iterasyon adımında k-1 adet set eğitim veri seti olarak ve geriye kalan bir adet ise test veri seti

olarak kullanılmaktadır. Bu şekilde her bir grup test veri seti olana kadar bu iterasyon tekrarlanır. Böylece makine öğrenmesi algoritmaları tarafından görülmeyen veri seti kalmamış olur. Çapraz doğrulama yaparken k değerini genellikle [3,10] aralığında belirlenir. Burada seçilen değer yüksek olursa maliyeti arttırabileceğini de hesaba katmamız gerekmektedir (S. Yadav and Shukla 2016). Bu tez çalışmasında çapraz doğrulama için k=10 olarak ayarlanmıştır. Veri seti 10 eşit parçaya ayrılmıştır. Veri setinin %90'ı eğitim ve %10'u test veri seti olarak ayarlanmıştır. Şekil 4.8'de veri setine uygulanan 10-kat çapraz doğrulama yapısı sunulmuştur.



Şekil 4.8 Çapraz doğrulama (Rosaen 2016)

4.5.2 Hold-out yöntemi

Hold- out tekniği veri setini eğitim ve test veri seti olmak üzere iki ayrı parçaya bölme işlemidir. Makine öğrenmesi algoritmaları eğitim veri seti üzerinde eğitimini tamamlar. Test veri setinde ise eğitim almayan veriler üzerinde modelin performans ölçümünü yaptığımız kısımdır. Hold-out yöntemi bir defa uygulandığı için maliyet hesabı düşüktür (S. Yadav and Shukla 2016). Şekil 4.9'da hold-out diyagramı verilmiştir. Bu çalışmada hold-out tekniğine göre veri setimiz %70 eğitim veri seti, %30 test veri seti olarak ayrılmıştır.

Şekil 4.9 Hold-out diyagramı

4.6 Makine Öğrenmesi Modellerinin Hiper Parametrelerinin Ayarlanması

Bu bölümde, model kurulumu yapabilmek için Bölüm 3'te bahsedilen makine öğrenmesi algoritmalarının nasıl oluşturulduğu hakkında detaylı bilgi verilecektir.

4.6.1 Tek (single) tabanlı makine öğrenmesi algoritmaları parametre ayarları

Diyabet hastalığında teşhis yapabilmek için kullanılan tek (single) tabanlı makine öğrenmesi algoritmalarında model kurulumunda kullanılan parametre ayarlarından bahsedilmiştir.

4.6.1.1 Lojistik regresyon (logistic regression) hiper parametre ayarları

Lojistik regresyon makine öğrenmesi algoritmasındaki parametre ayarları şöyledir; C parametresi ceza kuvveti olarak tanımlanır. C [100, 10, 1.0, 0.1, 0.001] aralığından varsayılan değer olarak C=1.0 atanmıştır. Solver [Newton-cg, lbgs,liblinear, sag, saga] listesinden solver=lbfgs olarak ayarlanmıştır. Daha büyük veri setlerinde "sag" ve "saga" tercih edilebilir. Ceza tipini belirlemek için Penalty [none, l1, l2, elasticnet] listesinden Solver=lbfgs ayarlandığı için penalty=l2 olarak varsayılan değer olarak atanmıştır.

4.6.1.2 Destek vektör makineleri (support vector machines) hiper parametre ayarları

Destek vektör makinelerinde parametre ayarları şöyledir; kernel=[linear, poly, rbf, sigmoid, precomputed] listesinden kernel:=linear olarak ayarlanmıştır. Ceza parametresi

C=1.0 olarak seçilmiştir. Probability=true olarak ayarlanmıştır. Probability değeri true olduğunda sınıf tahmin olasılıkları aktif hale gelmektedir.

4.6.1.3 K-en yakın komşu (k nearest neighbors) hiper parametre ayarları

KNN makine öğrenmesi algoritmasındaki parametre ayarları şöyledir; en yakın komşu sayısı için [1,10] arası parametreler denenmiştir. En iyi sonucu n_neighbors =5 olduğunda alınmıştır. İki komşu arasındaki mesafeyi ölçmek için Öklid fonksiyonunu seçilmiştir. Ağırlık olarak weight=[uniform, distance] listesinden weights = distance olarak ayarlanmıştır. Distance seçimi yakında olan komşulara daha fazla ağırlık verirken, uzaktaki komşulara daha az ağırlık verir.

4.6.1.4 Yapay sinir ağları (artificial neural networks) hiper parametre ayarları

Yapay Sinir Ağları makine öğrenmesi algoritmasındaki parametre ayarları şöyledir; aktivasyon fonksiyonu olarak activation=ReLU ayarlanmıştır. Solver [lbfgs, sgd, adam] listesinden solver=adam, dönem sayısı olarak max_iter=500, gizli katman sayısı hidden_layer_sizes =(13, 13, 13) olarak ayarlanmıştır.

4.6.1.5 Naif bayes (naive bayes) hiper parametre ayarları

Naif Bayes makine öğrenmesi algoritmasındaki parametre ayarları şöyledir; var_smoothing= 1e-9, priors= None, varsayılan değer olarak ayarlanmıştır.

4.6.2 Torbalama ve arttırma topluluk (bagging and boosting ensemble) öğrenmesi modelleri parametre ayarları

Diyabet hastalığında teşhis yapabilmek için kullanılan torbalama ve arttırma topluluk öğrenmesi modellerinde model kurulumunda kullanılan parametre ayarlarından bahsedeceğiz.

4.6.2.1 Karar ağacı (decision tree) hiper parametre ayarları

Karar Ağacı makine öğrenmesi modellerinde parametre ayarları şöyledir; bölünmedeki kaliteyi ölçmek için criterion=[gini,entropi] listesinden criterion=entropi, her düğümdeki bölme seçiminde splitter=[best, random] listesinden splitter=best, bir düğümün bölünmesindeki minimum örnek sayısı min_samples_split=2, ağacın maksimum derinliği için max_depth=5, bir düğümdeki maksimum olması gereken yaprak sayısı max_leaf_nodes=22, en iyi bölünme için dikkat edilmesi gereken maksimum özellik sayısı max_features=7, sınıf ağırlıkları class_weight=[dict,balanced] listesinden class_weight=balanced olarak ayarlamalar yapılmıştır.

4.6.2.2 Rastgele orman (random forest) hiper parametre ayarları

Rastgele Orman topluluk öğrenmesi modellerinde parametre ayarları şöyledir; ormanda tanımlanan ağaç sayısı n_estimators=100, en iyi bölünme için dikkat edilmesi gereken maksimum özellik sayısı max_features=3, bölünmedeki kaliteyi ölçmek için criterion=[gini,entropi] listesinden criterion=gini olarak ayarlamalar yapılmıştır.

4.6.2.3 Adaboost hiper parametre ayarları

Adaboost topluluk öğrenmesi modellerinde parametre ayarları şöyledir; tanımlanan karar ağaçlarının sayısı n_estimators=100, güçlendirme algoritması olarak algorithm=SAMME.R ayarlamaları yapılmıştır.

4.6.2.4 Gradyan arttırma (gradient boosting) hiper parametre ayarları

Gradyan Arttırma topluluk öğrenmesi modellerinde parametre ayarları şöyledir; tanımlanan ağaç sayısı n_estimators=100, en iyi bölünme için dikkat edilmesi gereken maksimum özellik sayısı max_features=3 olarak ayarlamalar yapılmıştır.

4.6.2.5 XGBoost hiper parametre ayarları

XGBoost topluluk öğrenmesi modellerinde parametre ayarları şöyledir; güçlendirici olarak ağaç tabanlı model tanımlamada booster=gbtree, ağacın maksimum derinliği max_depth=4, kayıp fonksiyonu hesaplama parametresi olarak objective=binary:logistic ayarlamaları yapılmıştır.

4.6.3 Yığınlama topluluk (stacking ensemble) öğrenmesi modelleri parametre ayarları

Bu çalışmada diyabet hastalığında teşhis yapabilmek için kullanılan yığınlama (stacking) topluluk öğrenmesi modellerinde model kurulumunda kullanılan parametre ayarlarından bahsedilmiştir.

4.6.3.1 Oylama (voting) hiper parametre ayarları

Oylama topluluk öğrenmesi modellerinde parametre ayarları şöyledir; tahminde bulunmak 3 makine öğrenmesi algoritması (Lojistik Regresyon, Destek Vektör Makineleri, Karar Ağacı) kullanılmıştır. Oylamanın tipini belirlemek için voting=[hard, soft] listesinden voting= soft ayarlamaları yapılmıştır.

4.6.3.2 Yığınlama (stacking) hiper parametre ayarları

Yığınlama topluluk öğrenmesi modellerinde parametre ayarları şöyledir; Seviye-0'da (Lojistik Regresyon, K-En Yakın Komşu, Karar Ağacı, Destek Vektör Makineleri, Naif Bayes) 5 farklı makine öğrenmesi algoritması kullanılmıştır. Seviye-1'de meta learner olarak Lojistik Regresyon makine öğrenmesi algoritmasıyla model kurulmuştur.

4.6.3.3 Harmanlama (blending) hiper parametre ayarları

Harmanlama topluluk öğrenmesi modellerinde parametre ayarları şöyledir; Seviye-0'da (Karar Ağacı, K-En Yakın Komşu, Rastgele Orman, Gradyan Arttırma, Naif Bayes) 5 farklı makine öğrenmesi algoritması kullanılmıştır. Seviye-1'de meta learner olarak Lojistik Regresyon makine öğrenmesi algoritmasıyla model kurulmuştur.

4.6.3.4 Süper öğrenci (super learner) hiper parametre ayarları

Super Learner topluluk öğrenmesi modellerinde parametre ayarları şöyledir; Seviye-0 ve Seviye-1'de çeşitli makine öğrenmesi algoritmalarıyla en iyi sonucu almak için farklı Super Learner model kombinasyonları oluşturularak çalışmalar yapılmıştır. Seviye-1'de meta learner olarak Lojistik Regresyon, Rastgele Orman, Karar Ağacı, K-En Yakın Komşu, Ekstra Karar Ağacı ve Destek Vektör Makineleri ile modeller oluşturulmuştur. Çapraz doğrulama için k (5,10) değerleri kullanılmıştır. En iyi sonucu veren model kombinasyonu ise Seviye-0'da (Lojistik Regresyon, Gradyan Arttırma, Karar Ağacı, Rastgele Orman) 4 farklı makine öğrenmesi algoritmasıyla, Seviye-1'de meta learner olarak Destek Vektör Makineleri seçimi ve çapraz doğrulama için k=10 parametre ayarları yapıldığında alınmıştır.

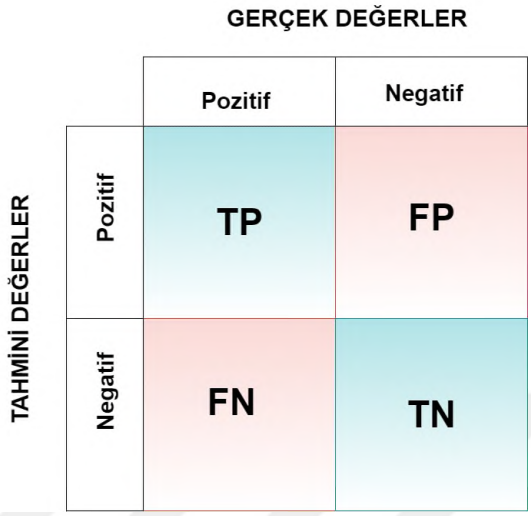
4.7 Veri Seti Değerlendirme Metrikleri

4.7.1 Karışıklık matrisi

Karışıklık Matrisi (Confusion Matrix), sınıflandırma problemlerinde veri setindeki gerçek değerlerini bildiğimiz verilerle tahmin edilen değerlerin doğruluk performansını tanımlamak için kullanılan bir yöntemdir (Stehman 1997). Gerçek değerler ile tahmin edilen değerlerin dört farklı sonucunu görselleştiren bir tablo biçimidir.

Şekil 4.10'da 2 x 2'lik bir sınıflandırma problemi için olan karışıklık matrisini görülmektedir. Farklı sınıflandırma problemleri için N x N'lik bir karışıklık matrisi de

yapılabilmektedir. Karışıklık matrisinde sütunlar gerçek değerleri ifade ederken, satırlar ise tahmin edilen değerleri göstermektedir.



Şekil 4.10 Karışıklık matrisi

- **TP (True Pozitif):** Gerçek değerler ile tahmin edilen değer pozitif ise,
- **FP (False Pozitif):** Gerçek değeri negatif iken tahmin edilen değer pozitif ise,
- **FN (False Negatif):** Gerçek değeri pozitif iken tahmin edilen değer negatif ise,
- **TN(True Negatif):** Gerçek değeri ile tahmin edilen değer negatif ise bu şekilde tanımlanır.

Karışıklık matrisi kullanılarak Doğruluk (Accuracy), Kesinlik (Precision), Duyarlılık (Recall), Özgünlük (Specificity) ve F1-Skor (F1-Score) hesaplanır.

Doğruluk (Accuracy), bir modelin performansı ölçebilmek için en çok tercih edilen ölçüm tekniklerinden biridir. Elde edilen değer [0,1] aralığında bir değer alır ve sonuç 1'e yaklaştıkça modelin performans kalitesinin arttığını gösterir. Ama tek başına bu metriğin kullanımı yeterli değildir. Model Performansı için Doğruluk hesaplanmasının matematiksel gösterimi Denklem (4.1)'de verilmiştir.

$$Doğruluk = \frac{(TP + TN)}{(\ TP\ + FP + TN + FN)} \tag{4.1}$$

Kesinlik (Precision), değerinde diyabetli olarak etiketlediğimiz tahmin değerlerinin, tüm diyabetli olarak etiketlenenlere oranını veren bir metriktir. Bu değerin yüksek olması model performansı açısından önemlidir. Model Performansı için Kesinlik hesaplanmasının matematiksel gösterimi Denklem (4.2)'de verilmiştir.

$$Kesinlik\ = \frac{TP}{(TP + FP)} \tag{4.2}$$

Duyarlılık (Recall) veya geri çağırma olarak bilinir. Diyabetli olarak etiketlediğimiz tahmin değerleri ile veri setindeki tüm diyabetli olanlara oranını veren bir metriktir. Model Performansı için Duyarlılık hesaplanmasının matematiksel gösterimi Denklem (4.3)'te verilmiştir.

$$Duyarlılık\ = \frac{TP}{(TP + FN)} \tag{4.3}$$

Özgünlük (Specificity) değerinde ise diyabetli olmayan olarak etiketlediğimiz tahmin değerleri ile veri setindeki tüm diyabetli olmayanlara oranını veren bir metriktir. Model Performansı için Özgünlük hesaplanmasının matematiksel gösterimi Denklem (4.4)'te verilmiştir.

$$\text{Özgünlük}\ = \frac{\text{TN}}{(\text{TN} + \text{FP})} \tag{4.4}$$

F1-Skor (F1-Score), sınıflandırma işleminin ne kadar iyi olduğunun bir göstergesidir. Duyarlılık (Recall) ve Kesinlik (Precision) değerlerini kullanarak elde ettiğimiz bir metriktir. F1-Skor bu iki değerin harmonik ortalama değeri alınarak hesaplama işlemi yapar. Model Performansı için F1-Skor hesaplanmasının matematiksel gösterimi Denklem (4.5)'te verilmiştir.

$$F1 - Skor = \frac{2 * Precision * Recall}{(Precision + Recall)} \quad (4.5)$$

4.7.2 ROC eğrisi

ROC eğrisi gerçek pozitiflerin oranının (TPR) yani duyarlılık değerinin, yanlış pozitiflere (FPF) olan oranına gösteren eğriye denir. ROC hesaplanması için matematiksel gösterimi Denklem (4.6)'da verilmiştir.

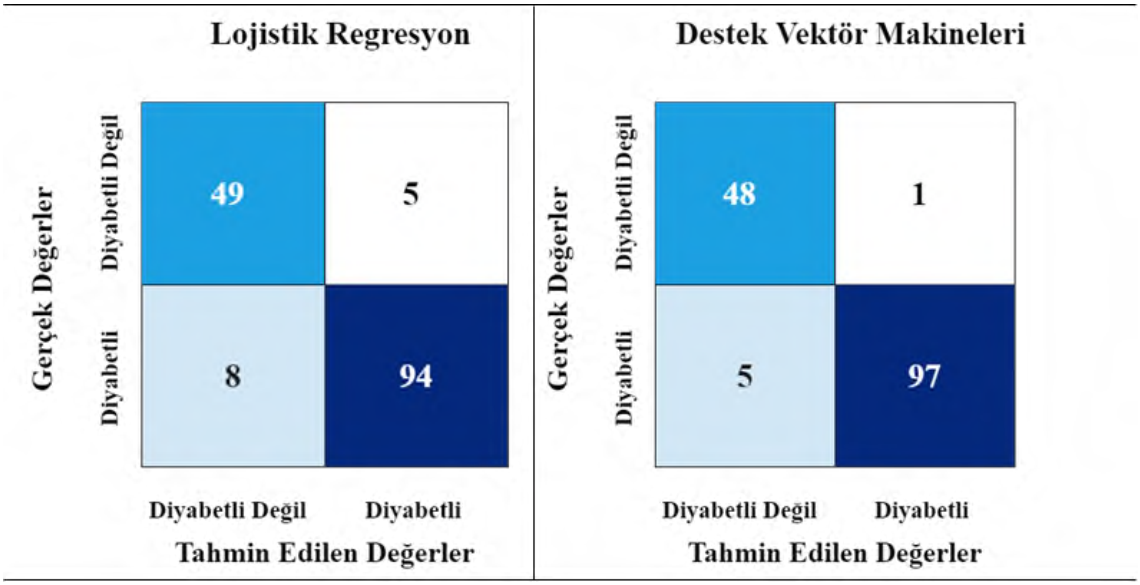
$$TPR = \frac{TP}{TP + FN}, FPR = \frac{FP}{FP + TN} \; ROC = \frac{TPR}{FPR} \quad (4.6)$$

5. BULGULAR VE TARTIŞMA

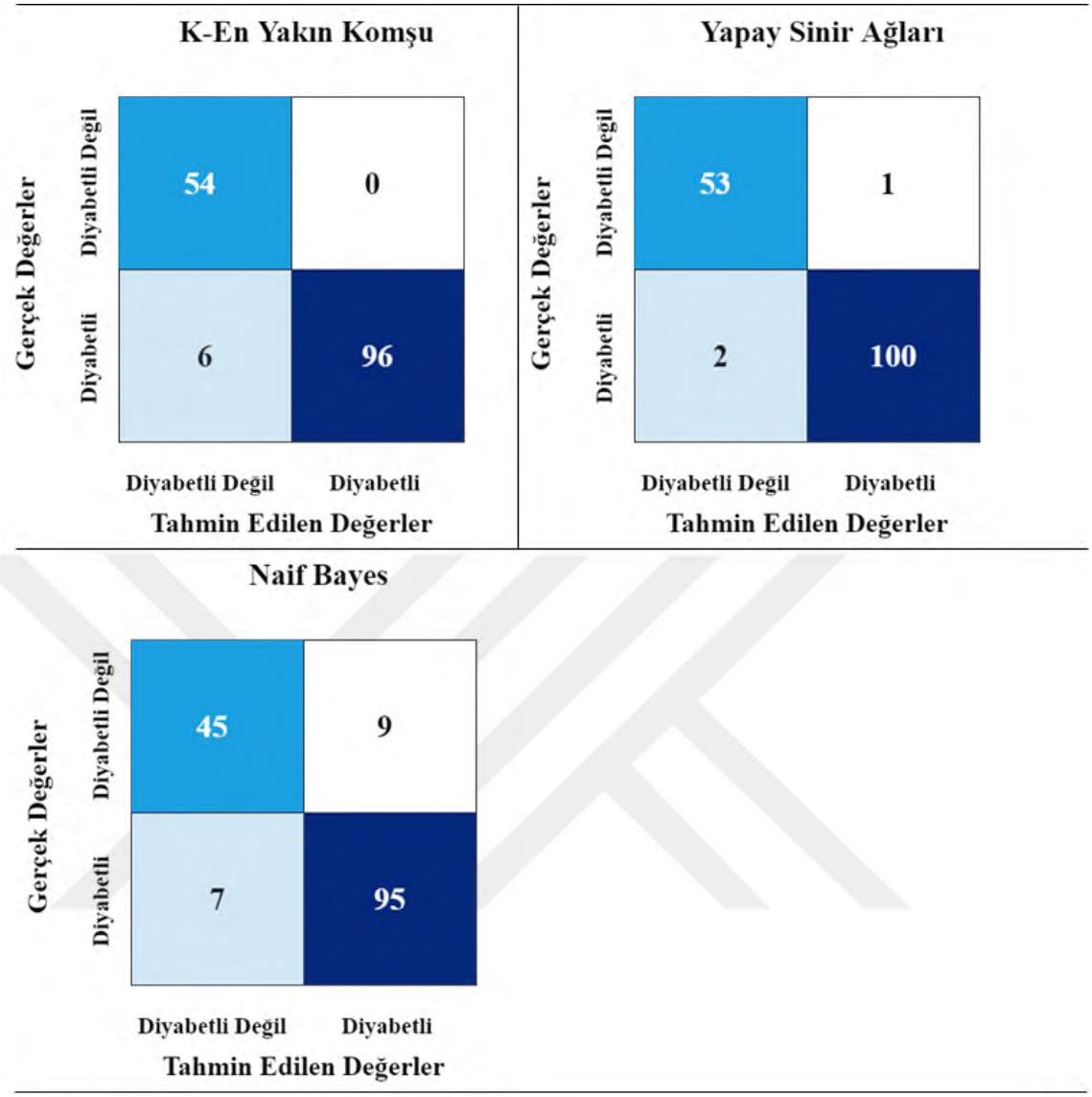
Bu bölümde tez çalışmamızda oluşturmuş olduğumuz modelden elde ettiğimiz sonuçlar hakkında detaylı bilgi verilecektir.

5.1 Tek (Single) Tabanlı Makine Öğrenmesi Modellerinin Performans Sonuçları

Bölüm 3.1'de bahsi geçen tek tabanlı makine öğrenmesi algoritmalarının eğitim aldıktan sonraki karışıklık matris sonuçları Şekil 5.1'de verilmiştir. Lojistik Regresyon (49 + 94) 143 hastayı doğru tahmin ve (8 + 5) 13 hastayı yanlış sınıflandırmıştır. Destek Vektör Makineleri (48 + 97) 145 hastayı doğru tahmin ve (5 + 1) 6 hastayı yanlış sınıflandırmıştır. K-En Yakın Komşu (54 + 96) 150 hastayı doğru tahmin ve (6 + 0) 6 hastayı yanlış sınıflandırmıştır. Yapay Sinir Ağları (53 + 100) 153 hastayı doğru tahmin ve (2 + 1) 3 hastayı yanlış sınıflandırmıştır. Naif Bayes (45 + 95) 140 hastayı doğru tahmin ve (7 + 9) 16 hastayı yanlış sınıflandırmıştır. En yüksek doğru tahmini 153 diyabetli hasta sayısı ile Yapay Sinir Ağları makine öğrenmesi algoritmasıyla elde edilmiştir.



Şekil 5.1 Tek tabanlı makine öğrenmesi algoritmaları karışıklık matris sonuçları



Şekil 5.1 Tek tabanlı makine öğrenmesi algoritmaları karışıklık matris sonuçları (devamı)

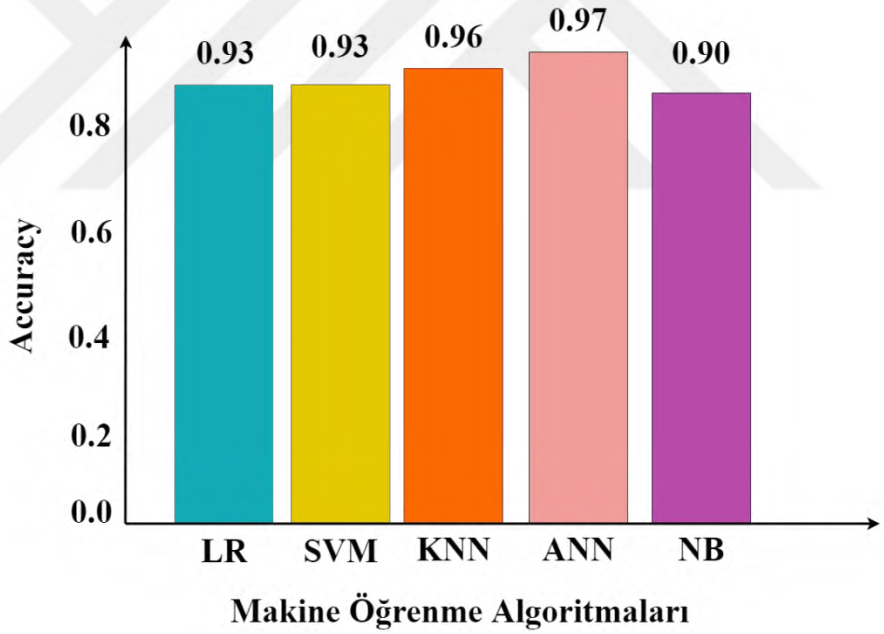
Tek tabanlı makine öğrenmesi algoritmalarının doğruluk (accuracy) performans sonuçları Çizelge 5.1'de sunulmuştur. En yüksek doğruluk performansını %97 Yapay Sinir Ağları makine öğrenmesi algoritmasıyla elde edilmiştir. Diğer algoritmalarda sırasıyla K-En Yakın Komşu %96, Lojistik Regresyon ve Destek Vektör Makineleri %93, Naif Bayes %90 doğruluk performans sonuçları elde edilmiştir. Tek tabanlı makine öğrenmesi algoritmalarının performans sonuçları arasındaki fark %1 ile %7 aralığında değişmektedir. Sonuç performansları arasındaki değerlerin birbirlerine yakın olduğu görülmektedir. Diğer taraftan en düşük performans değeri %90 ile Naif Bayes ile elde edilmiştir.

Çizelge 5.1 Tek tabanlı (single base) makine öğrenmesi algoritmaları doğruluk (accuracy) değerleri

Model Adı	Doğruluk (Accuracy)
LR	0.93
SVM	0.93
KNN	0.96
ANN	**0.97**
NB	0.90

Çizelge 5.1- LR (Lojistik Regresyon), SVM (Destek Vektör Makineleri), KNN (K-En Yakın Komşu), ANN (Yapay Sinir Ağları), NB (Naif Bayes)

Şekil 5.2'de tek tabanlı makine öğrenmesi algoritmalarının doğruluk (accuracy) performans karşılaştırma sonuçları görsel olarak sunulmuştur.



Şekil 5.2 Tek tabanlı makine öğrenmesi algoritmaları performans karşılaştırma sonuçları

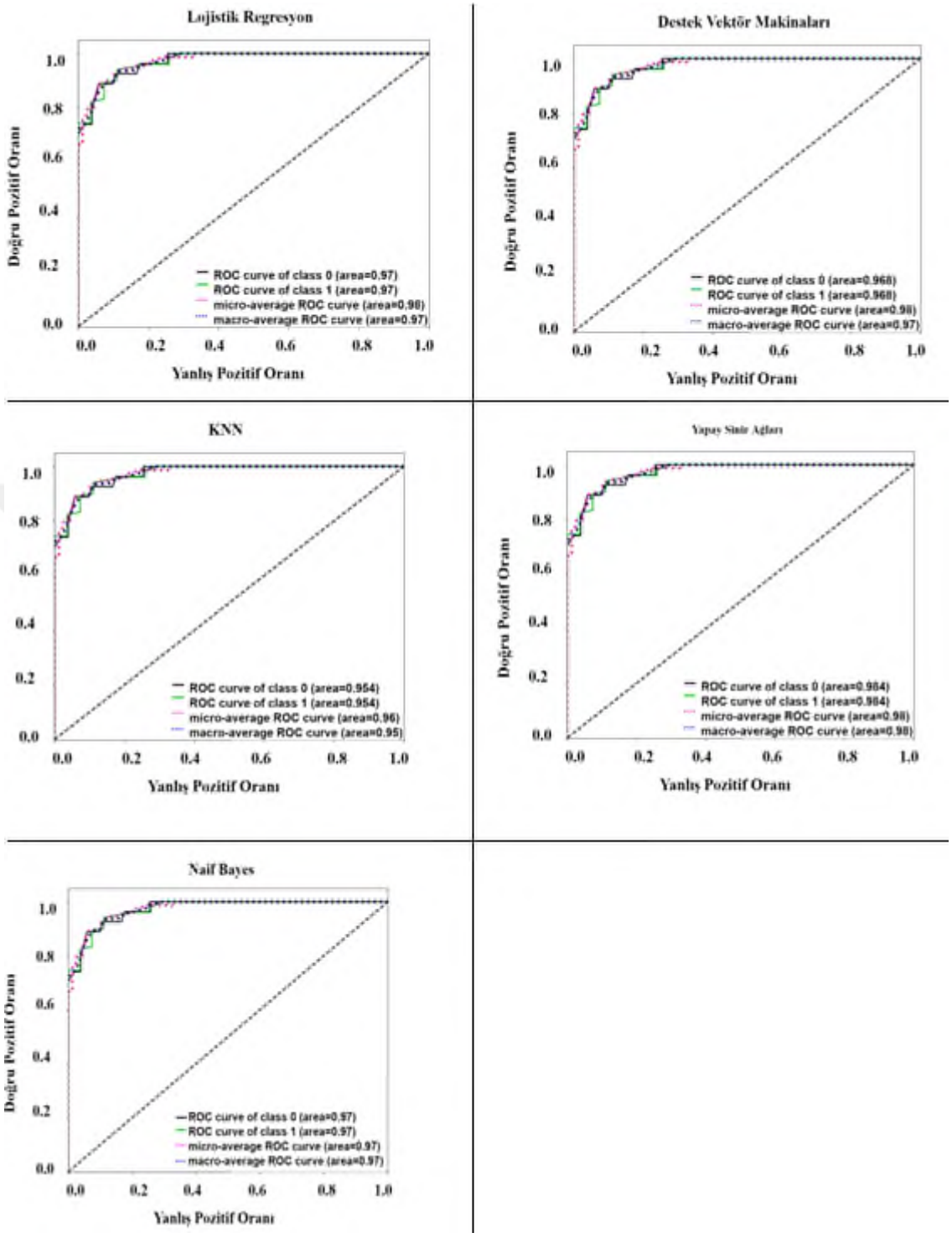
Şekil 5.2- LR (Lojistik Regresyon), SVM (Destek Vektör Makineleri), KNN (K-En Yakın Komşu), ANN (Yapay Sinir Ağları), NB (Naif Bayes)

Tek tabanlı makine öğrenmesi algoritmalarının kesinlik (precision), duyarlılık (recall) ve F1-Skor'a (F1-Score) göre performans sonuçları Çizelge 5.2'de sunulmuştur. En yüksek F1-Skor'a göre performans sonucu %97 ile Yapay Sinir Ağları makine öğrenmesi algoritmalarıyla elde edilmiştir. Diğer algoritmalarda sırasıyla K-En Yakın Komşu %96, Lojistik Regresyon ve Destek Vektör Makineleri %95 performans sonuçları elde edilmiştir. En yüksek kesinlik (precision) değeri ise %97 ile Yapay Sinir Ağları ve K-En Yakın Komşu algoritmasıyla elde edilmiştir. Ayrıca tek tabanlı diğer algoritmaların kesinlik değerleri en az %90 üzeri olduğu görülmektedir. Genel olarak tek tabanlı makine öğrenmesi algoritmalarının performans değerleri %90 değerinin üzerindedir.

Çizelge 5.2 Tek tabanlı (single base) makine öğrenmesi algoritmaları performans sonuçları

Model Adı	F1-Skor (F1-Score)	Kesinlik (Precision)	Duyarlılık (Recall)
LR	0.95	0.94	0.95
SVM	0.95	0.94	0.95
KNN	0.96	0.97	0.94
ANN	0.97	0.97	0.96
NB	0.92	0.91	0.93

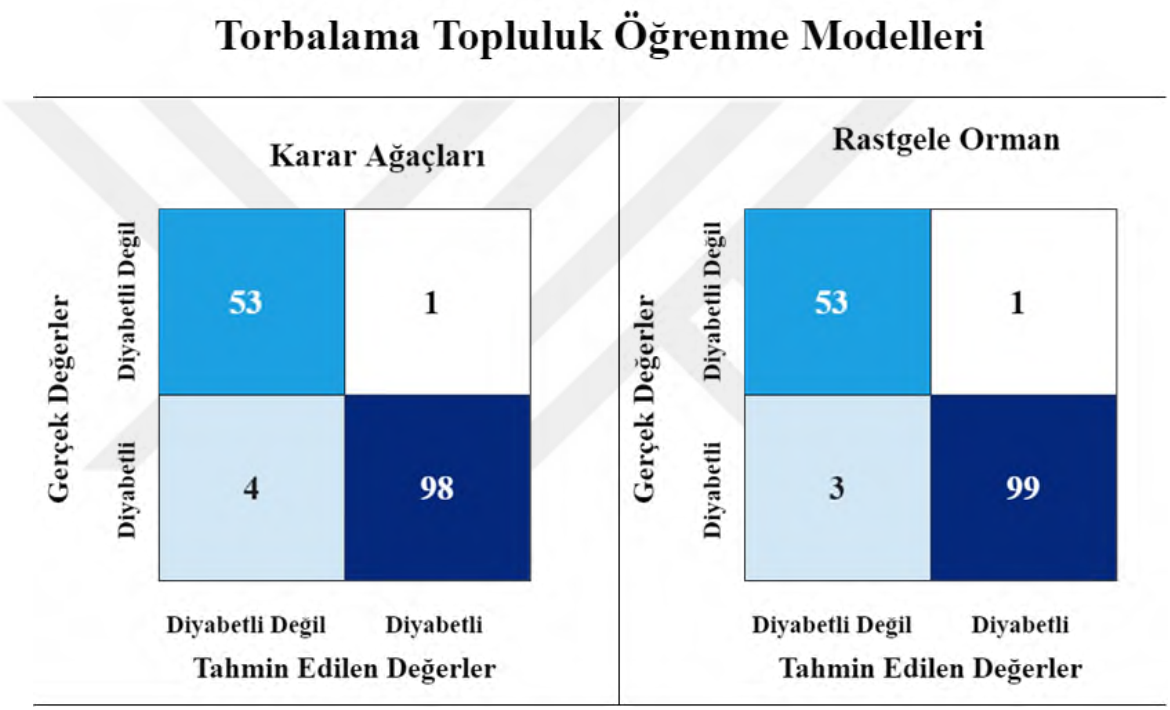
Tek tabanlı makine öğrenmesi algoritmaları ROC eğrileri aşağıdaki Şekil 5.3'te sunulmuştur. En iyi ROC eğri değerleri %98,4 değer ile Yapay Sinir Ağları (ANN) makine öğrenmesi algoritmasıyla elde edilmiştir. Diğer algoritmalarında ROC eğri değerleri %1 ve %3 farkı ile birbirlerine yakın değerler olduğu görülmektedir.



Şekil 5.3 Tek tabanlı makine öğrenmesi algoritmaları ROC eğrileri

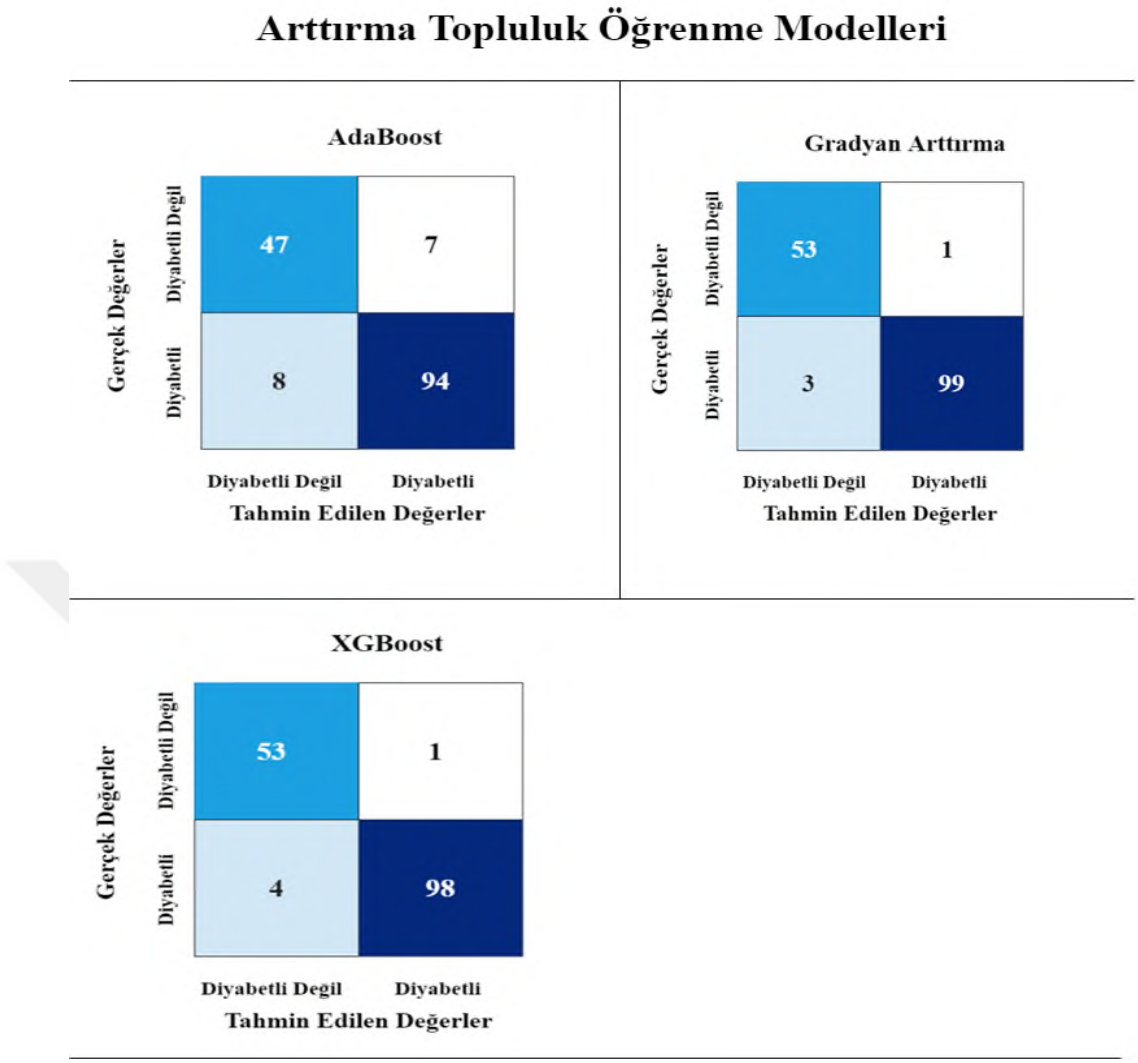
5.2 Topluluk (Ensemble) Öğrenmesi Modellerinin Performans Sonuçları

Bölüm 3.2'de bahsi geçen topluluk öğrenmesi modellerinin eğitim aldıktan sonraki karışıklık matris sonuçları hakkında bilgi verilecektir. Şekil 5.4'te sunulan torbalama (bagging) topluluk öğrenmesi karışıklık matris sonuçlarında Karar Ağaçları (53 + 98) 151 hastayı doğru tahmin ve (4 + 1) 5 hastayı yanlış sınıflandırmış, Rastgele Orman (53 + 99) 152 hastayı doğru tahmin ve (3 + 1) 4 hastayı yanlış sınıflandırmıştır.



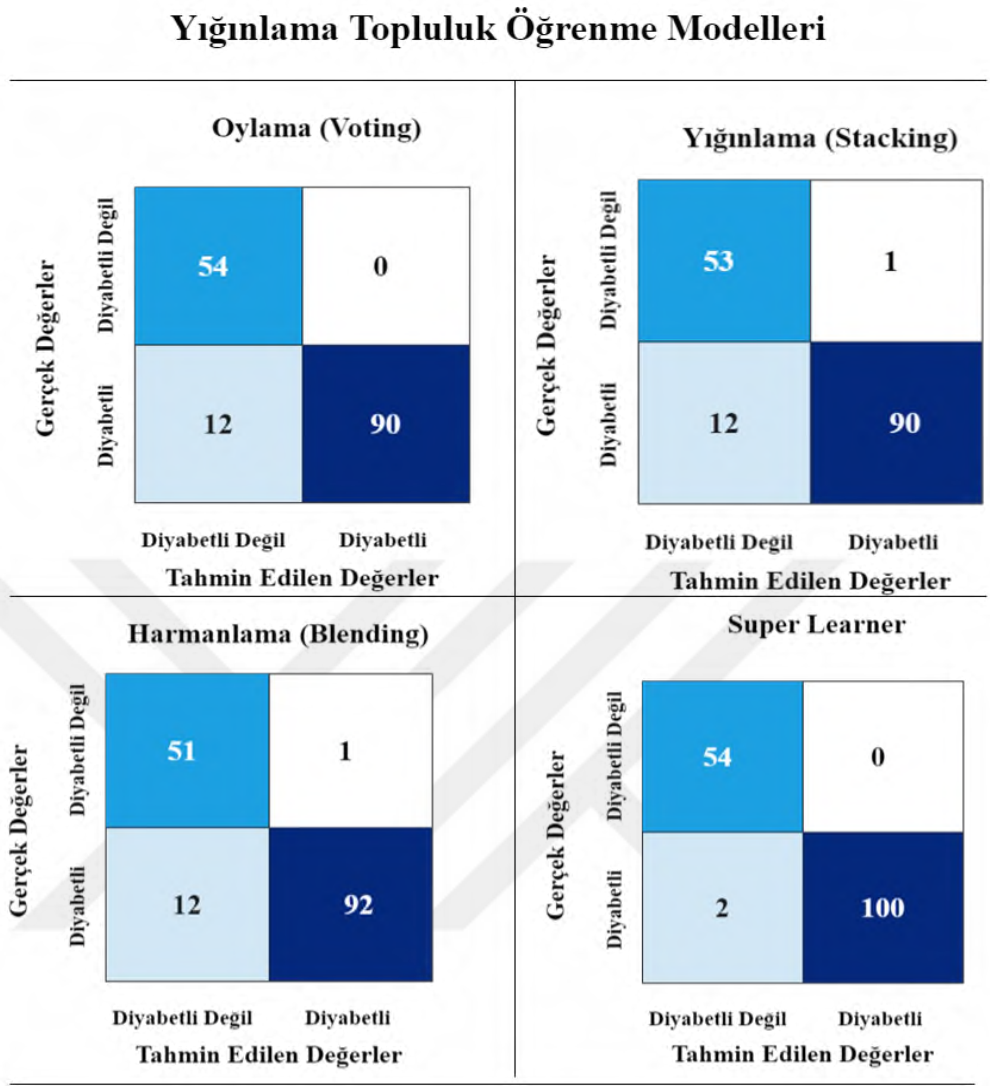
Şekil 5.4 Torbalama topluluk öğrenmesi modelleri karışıklık matrisi sonuçları

Şekil 5.5'te sunulan arttırma (boosting) topluluk öğrenmesi karışıklık matris sonuçlarında AdaBoost (47 + 94) 141 hastayı doğru tahmin ve (8 + 7) 15 hastayı yanlış sınıflandırmıştır. Gradyan Arttırma (53+ 99) 152 hastayı doğru tahmin ve (3 + 1) 4 hastayı yanlış sınıflandırmıştır. XGBoost (53+ 98) 151 hastayı doğru tahmin ve (4 + 1) 5 hastayı yanlış sınıflandırmıştır.



Şekil 5.5 Arttırma topluluk öğrenmesi modelleri karışıklık matrisi sonuçları

Yığınlama (stacking) topluluk öğrenmesi karışıklık matris sonuçlarında Şekil 5.6'da sunulmuştur. Oylama (voting) topluluk öğrenmesi (54 + 90) 144 hastayı doğru tahmin ve (12 + 0) 12 hastayı yanlış sınıflandırmıştır. Yığınlama (stacking) topluluk öğrenmesi (53 + 90) 143 hastayı doğru tahmin ve (12 + 1) 13 hastayı yanlış sınıflandırmıştır. Harmanlama topluluk öğrenmesi (51 + 92) 143 hastayı doğru tahmin ve (12 + 1) 13 hastayı yanlış sınıflandırmıştır. Super Learner topluluk öğrenmesi ise (54 + 100) 154 hastayı doğru tahmin ve (2 + 0) 2 hastayı yanlış sınıflandırmıştır. Bir bütün olarak topluluk öğrenme modellerinin sonuçlarını karşılaştırdığımızda en yüksek doğru tahmini 154 diyabetli hasta sayısı ile Super Learner topluluk öğrenmesi modeliyle elde edilmiştir.



Şekil 5.6 Yığınlama topluluk öğrenmesi modelleri karışıklık matrisi sonuçları

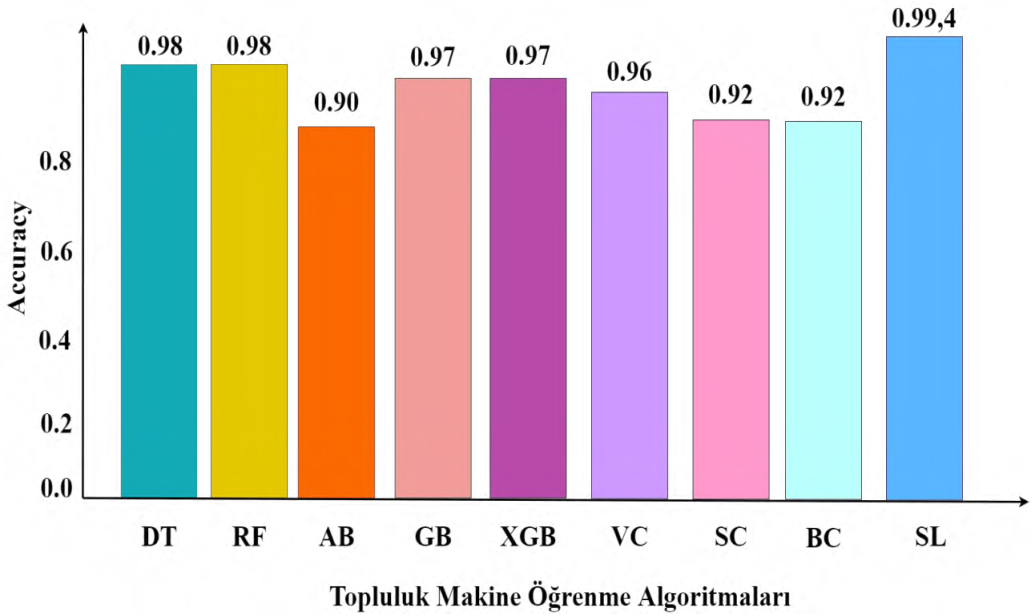
Çizelge 5.3'te topluluk öğrenmesi modellerinin doğruluk (accuracy) performans sonuçları sunulmuştur. En yüksek doğruluk performansı %99,4 ile Super Learner topluluk öğrenmesi algoritmasıyla elde edilmiştir. Diğer algoritmaların doğruluk performans sonuçları arasındaki fark %1 ile %8 aralığındadır. Diğer taraftan en düşük performans değeri ise %90 ile AdaBoost topluluk öğrenmesi algoritmasıyla elde edilmiştir. Genel olarak karşılaştırma yapıldığında yığınlama (stacking) topluluk öğrenmesi modellerinde daha iyi doğruluk performans sonucu elde edilmiştir.

Çizelge 5.3 Topluluk makine öğrenmesi modellerinin doğruluk (accuracy) değerleri

Model Adı	Doğruluk (Accuracy)
DT	0.98
RF	0.98
AB	0.90
GB	0.97
XGB	0.97
VC	0.96
SC	0.92
BC	0.92
SL	**0.99,4**

Çizelge 5.3 DT (Decision Tree),RF (Random Forest), AB (AdaBoost), GB(Gradient Boosting), XGB (eXtreme Gradient Boosting), SC (Stacking Classifier), BC (Blending Classifier) VC (Voting Classifier), SL (Super Learner)

Şekil 5.7'de topluluk öğrenmesi algoritmalarının doğruluk (accuracy) performans karşılaştırma sonuçları görsel olarak sunulmuştur.



Şekil 5.7 Topluluk (ensemble) öğrenmesi modellerinin performans karşılaştırma sonuçları

Topluluk öğrenmesi modellerinin kesinlik (precision), duyarlılık (recall) ve F1-Skor'a (F1-Score) göre performans sonuçları Çizelge 5.4'te sunulmuştur. En yüksek F1-Skor'a göre performans %99 Super Learner topluluk öğrenmesi modeliyle elde edilmiştir. Diğer algoritmaların F1-Skor'a göre performans sonuçları arasındaki fark %1 ile %6 aralığında değişmektedir. En yüksek Kesinlik değeri ise %99 ile Super Learner topluluk öğrenmesi modeliyle elde edilmiştir. Bu değerlere en yakın %98 ile Gradyan Arttırma ve XGBoost topluluk öğrenmesi algoritmaları takip etmiştir. En iyi Duyarlılık performans sonuçları ise %99 ile Super Learner topluluk öğrenmesi algoritmasıyla elde edilmiştir. Genel olarak performans sonuçlarına baktığımızda Super Learner topluluk öğrenmesi modeliyle en iyi performans sonuçları elde edilmiştir.

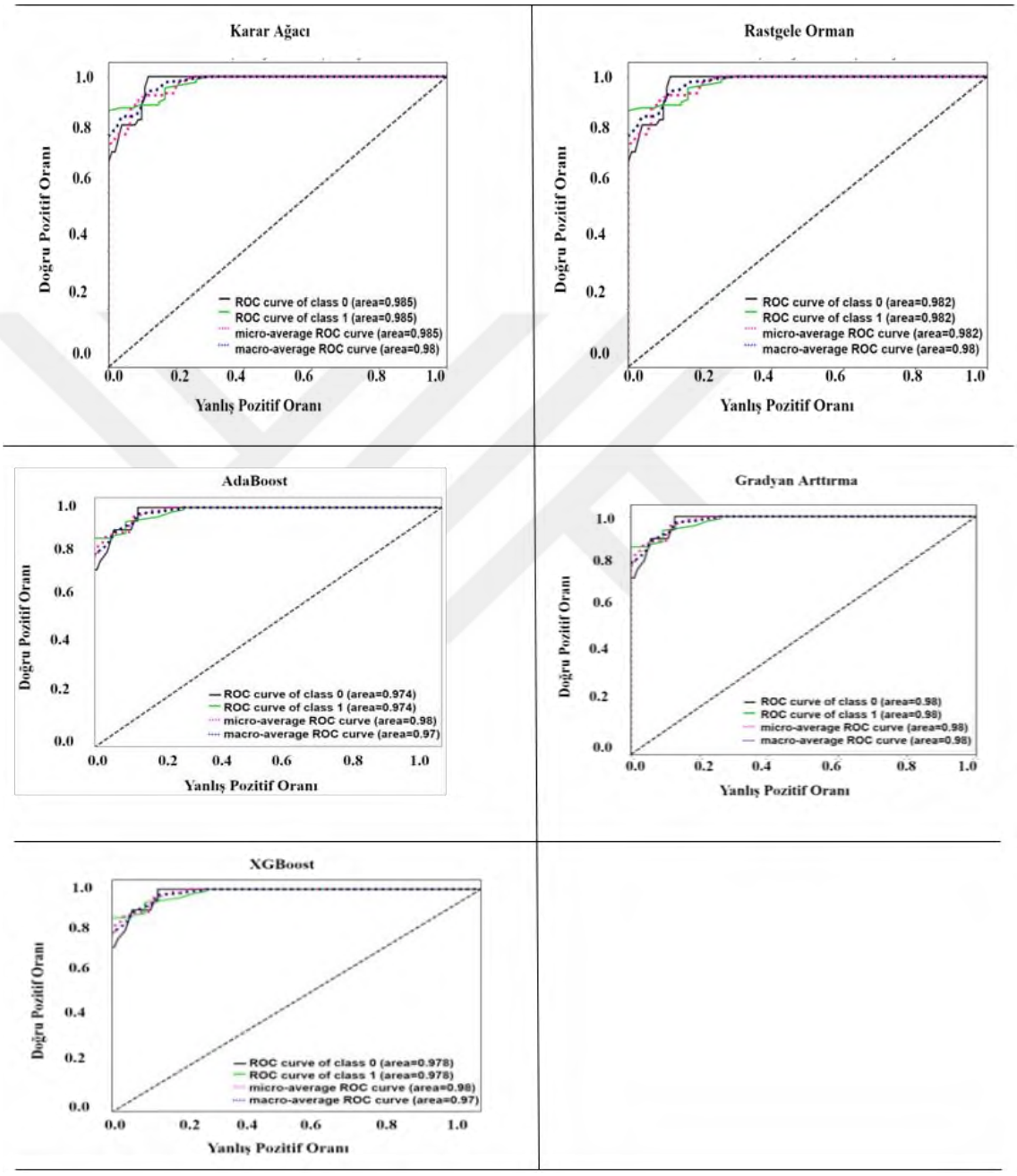
Çizelge 5.4 Topluluk makine öğrenmesi modelleri performans sonuçları

Model Adı	**F1-Skor (F1-Score)**	**Kesinlik (Precision)**	**Duyarlılık (Recall)**
DT	0.93	0.98	0.89
RF	0.94	0.99	0.90
AB	0.93	0.92	0.93
GB	0.98	0.99	0.97
XGB	0.98	0.98	0.96
VC	0.93	0.98	0.89
SC	0.93	0.94	0.92
BC	0.93	0.94	0.93
SL	**0.99**	**0.99**	**0.99**

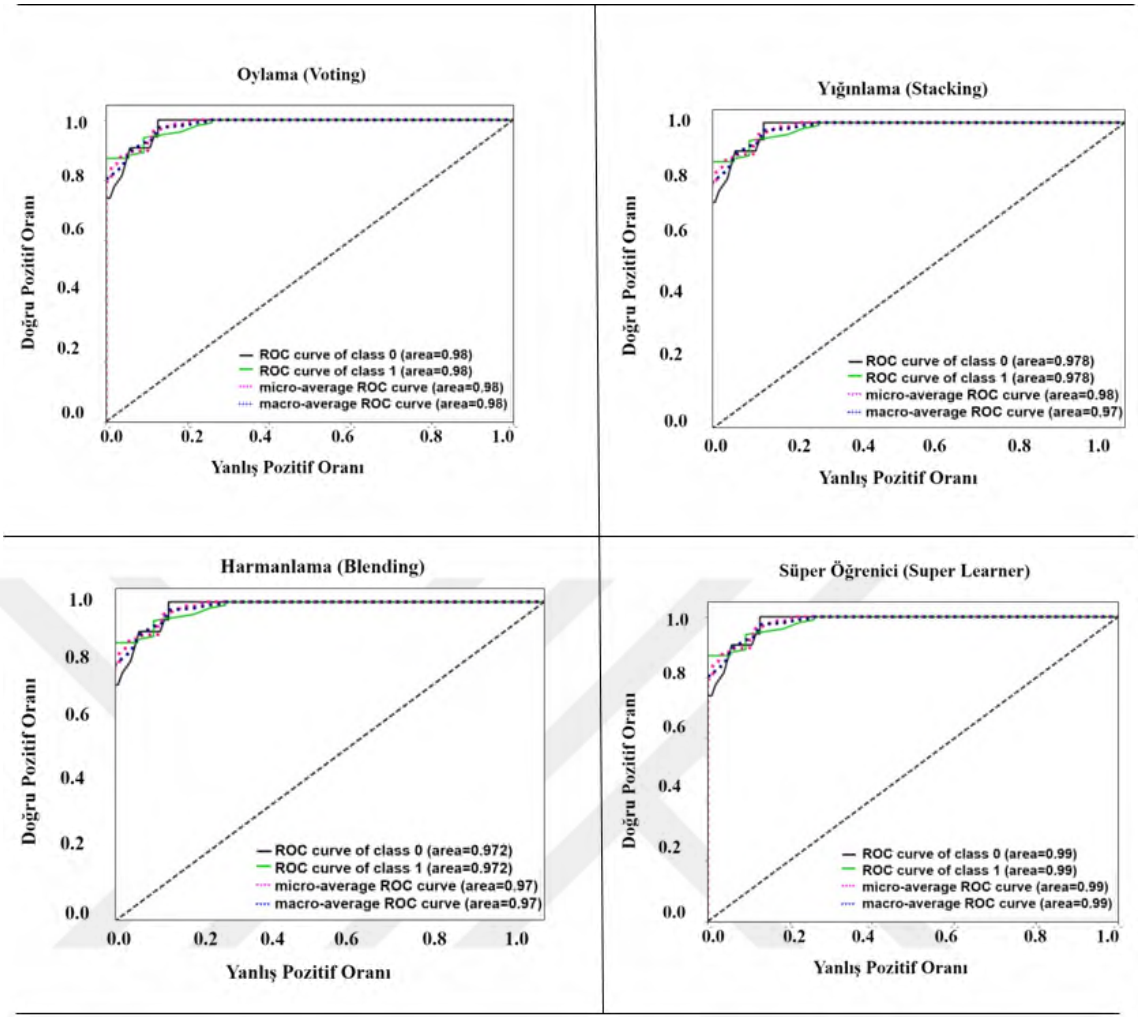
Çizelge 5.4 DT (Decision Tree),RF (Random Forest), AB (AdaBoost), GB(Gradient Boosting), XGB (eXtreme Gradient Boosting), VC (Voting Classifier), SC (Stacking Classifier), BC (Blending Classifier), SL (Super Learner)

Torbalama ve arttırma topluluk öğrenmesi modelleri ROC eğrileri Şekil 5.8'de ve yığınlama topluluk öğrenmesi modelleri ROC eğrileri Şekil 5.9'da sunulmuştur. En iyi

ROC eğri değerleri %99 değer ile Super Learner topluluk öğrenmesi modeliyle elde edilmiştir. Diğer modellerinde ROC eğri değerleri %1 ve %2 farkı ile birbirlerine yakın olduğu görülmektedir.



Şekil 5.8 Torbalama ve arttırma topluluk öğrenmesi modelleri ROC eğrileri



Şekil 5.9 Yığınlama topluluk öğrenmesi modelleri ROC eğrileri

Super Learner topluluk öğrenmesi modelinde en iyi performans sonucu alabilmek için farklı model kombinasyonlarının birleşimlerini deneyip sonuçlarını gözlemledik. Yaptığımız çalışmada oluşturulan bir Super Learner topluluk öğrenmesi modelinin yapısı şöyle oluşturulmuştur; Seviye-0'da Çizelge 5.5'te verilen farklı makine öğrenmesi algoritmaları kullanılmıştır. Seviye-1'de ise meta learner olarak Lojistik Regresyon makine öğrenmesi algoritması ve çapraz doğrulama için k=10 seçilerek parametre ayarları yapıldı. Çizelge 5.5'deki performans sonuçlarına bakıldığında Seviye-0'da (Lojistik Regresyon, K-En Yakın Komşu, Gradyan Arttırma, Karar Ağacı) 4 farklı makine öğrenmesi algoritma seçildiğinde en yüksek doğruluk %96,1ve ROC-AUC skor ise %96,1 performans sonuçlarını elde edilmiştir.

Çizelge 5.5 K:10 Çapraz doğrulama meta learner: lojistik regresyon

Seviye	Temel Makine Öğrenmesi Model Adı	ACC
Seviye-0	Lojistik Regresyon	0.90
Seviye-0	K-En Yakın Komşu	0.91
Seviye-0	Gradyan Arttırma	0.94
Seviye-0	Karar Ağacı	0.94
Seviye-1	**Lojistik Regresyon**	**0.961**
	ROC-AUC skor	**0.961**
Seviye-0	Naif Bayes	0.86
Seviye-0	AdaBoost	0.89
Seviye-0	Lojistik Regresyon	0.90
Seviye-0	AdaBoost	0.89
Seviye-0	K-En Yakın Komşu	0.91
Seviye-1	**Lojistik Regresyon**	**0.955**
	ROC-AUC skor	**0.943**
Seviye-0	Naif Bayes	0.86
Seviye-0	Destek Vektör Makineleri	0.89
Seviye-0	AdaBoost	0.89
Seviye-0	Lojistik Regresyon	0.90
Seviye-1	**Lojistik Regresyon**	**0.929**
	ROC-AUC skor	**0.919**

Bir başka çalışmada oluşturulan Super Learner topluluk öğrenmesi modelinin yapısı şöyledir; Bu modelde Seviye-0 için Çizelge 5.6'da verilen (Lojistik Regresyon, Gradyan Arttırma, Karar Ağacı, Rastgele Orman) 4 farklı makine öğrenmesi algoritması kullanılmıştır. Modelin doğruluk performansını karşılaştırmak için hem 5 kat hem de 10

kat çapraz doğrulama yapılmıştır. Elde edilen sonuçların doğruluk (accuracy) ve ROC-AUC performans değerleri aşağıdaki Çizelge 5.6'da verilmiştir. Çapraz doğrulama da k=5 parametre seçimi yapılan modelde Seviye-1 için meta learner olarak Lojistik Regresyon, Rastgele Orman ve Karar Ağacı olmak üzere 3 farklı makine öğrenmesi algoritmaları seçimi ile parametre ayarları yapılmıştır. Çizelge 5.6'daki performans sonuçlarına bakıldığında meta learner olarak Rastgele Orman makine öğrenmesi algoritması ile en yüksek doğruluk %98,7 ve ROC-AUC skor ise %98,50 performans sonuçlarını elde edilmiştir. Çapraz doğrulama k=10 parametre seçimi ile yapılan diğer modelde ise Seviye-1 için meta learner olarak Lojistik Regresyon, Rastgele Orman, Karar Ağacı, K-En Yakın Komşu, Ekstra Karar Ağaçları ve Destek Vektör Makineleri olmak üzere 6 farklı makine öğrenmesi algoritmaları seçimi ile parametre ayarları yapılmıştır. Çizelge 5.6'daki performans sonuçlarına bakıldığında meta learner olarak Destek Vektör Makineleri ile en yüksek doğruluk %99,4 ve ROC-AUC skor ise %99 performans sonuçlarını elde ettik. Diğer performans sonuçlarına bakıldığında en yakın değerler %98,7 ile Lojistik Regresyon ve %97,4 ile Rastgele Orman ve K-En Yakın Komşu makine öğrenmesi algoritmasıyla elde edilmiştir.

Çizelge 5.6 Super learner (süper öğrenci) topluluk öğrenmesi algoritmaları

Seviye	**Temel Makine Öğrenmesi Model Adı**	**ACC**	
Seviye-0	Lojistik Regresyon	0.90	
Seviye-0	Gradyan Arttırma	0.94	
Seviye-0	Karar Ağaçları	0.94	
Seviye-0	Rastgele Orman	0.95	
Seviye-1	**Meta Learner Adı**	**ACC**	**ROC-AUC**
5 Kat Çapraz Doğrulama	Lojistik Regresyon	0.980	0.980
	Rastgele Orman	**0.987**	**0.985**
	Karar Ağaçları	0.980	0.980

Çizelge 5.6 Super learner (süper öğrenci) topluluk öğrenmesi algoritmaları (devamı)

10 Kat Çapraz Doğrulama	Lojistik Regresyon	0.987	0.985
	Rastgele Orman	0.974	0.976
	Karar Ağacı	0.961	0.966
	K-En Yakın Komşu	0.974	0.976
	Ektsra Karar Ağaçları	0.955	0.961
Seviye-1	**Destek Vektör Makineleri**	**0.994**	**0.990**

Super Learner topluluk öğrenmesi modelinde meta learner olarak genellikle Lojistik Regresyon makine öğrenmesi algoritması seçilir. Bu çalışmada farklı makine öğrenmesi algoritmalarını da meta learner olarak seçip performans sonuçlarını gözlemlemek istedik. Seviye-1'de meta learner olarak doğrusal makine öğrenmesi algoritmalarının daha iyi sonuç verdiği Çizelge 5.6'da görülmektedir.

6. SONUÇ VE ÖNERİLER

Bu bölümde tez çalışmasında oluşturmuş olduğumuz modelden elde ettiğimiz sonuçlar ile gelecekte yapılacak olan çalışmalar için bazı tavsiyeler verilmektedir.

Bu tez çalışmasında hedeflenen; günümüzde yaygın bir hastalık kategorisine giren diyabet hastalığının erken teşhis edilmesine yardımcı olabilmek amacıyla makine öğrenmesi algoritmaları kullanarak daha az hatayla, daha doğru sınıflandırma yapılmasını sağlamak ve başarılı tahmin performansları elde etmektir. Diyabet hastalığının teşhis edilmesi için tek (single) tabanlı makine öğrenmesi algoritmaları ile topluluk (ensemble) öğrenmesi modellerinin performansları karşılaştırılmıştır. Topluluk öğrenmesi modellerinin tek (single) tabanlı makine öğrenmesi algoritmalarından daha iyi sonuç verdiği gözlemlenmiştir. Yığınlama (Stacking) topluluk öğrenmesi modellerinden Süper Öğrenci (Super Learner) topluluk öğrenmesi modeli ise diğer topluluk öğrenmesi modellerinden daha iyi doğruluk tahmin performansı göstermiştir.

Bu çalışmamızda kullandığımız veri seti UCI Machine Learning Reporsitory'den ücretsiz olarak temin edilmiştir. Veri seti Bangladeş, Sylhet'teki Sylhet Diyabet Hastanesinden 520 hasta kaydından elde edilen 16 öznitelik ve bir sınıf değerine sahip örnek veriler içermektedir. Veri setindeki 520 hasta kaydından 320'si diyabet hastası olup, 200'ü diyabet hastası değildir. Veri setinde sınıflandırma doğruluğunu arttırmak ve tahmin performans kalitesini arttırmak amacıyla veri seti eğitim (%70) ve test (%30) veri seti olarak ikiye bölünmüş ve k-katlı çapraz doğrulama teknikleri kullanılmıştır. Öznitelik seçim yöntemlerinden ki-kare tekniği ile veri seti 16 özellikten 9özelliğe (Polydipsia, Polyuria, Sudden_weight_loss, Partial_paresis, Gender, Irritability, Polyphagia, Alopecia, Age) sahip bir veri seti elde edilmiştir. Modelin güvenirliğini arttırmak amacıyla doğruluk, kesinlik, duyarlılık, F1-skor ve ROC eğrisi doğrulama metrikleri kullanılmıştır. Eğitilen modelde en iyi doğruluk tahmin performansını %99,4 ile Super Learner topluluk öğrenmesi modeliyle elde edilmiştir. Model oluşturulurken Seviye-0 çeşitli makine öğrenmesi algoritmaları kullanılarak farklı çalışmalar yapılmıştır. Seviye-1'de ise meta learner yaygın olarak Lojistik Regresyon makine öğrenmesi algoritması seçilir. Fakat bu çalışmada meta learner olarak 6 (Lojistik

Regresyon, Rastgele Orman, Karar Ağacı, K-En Yakın Komşu, Ekstra Karar Ağaçları, Destek Vektör Makineleri) farklı makine öğrenmesi algoritmaları kullanarak sonuç performanslarını karşılaştırdık. Ayrıca k-katlı çapraz doğrulama için de k (5,10) değerleriyle parametre ayarlanıp, sonuçları gözlemledik. Elde edilen çalışma sonuçlarını karşılaştırdığımızda en iyi sonucu Seviye-0'da (Lojistik Regresyon, Gradyan Arttırma, Karar Ağacı, Rastgele Orman) 4 farklı makine öğrenmesi algoritması, Seviye-1'de meta learner Destek Vektör Makineleri ve k-katlı çapraz doğrulama için k=10 olarak parametre ayarları yaptığımız zaman kurulan modelden elde edilmiştir. Kurulan bu modelde %99,4 doğruluk performans sonucu ile Super Learner topluluk öğrenmesi modeliyle elde edilmiştir.

Super Learner topluluk öğrenmesi modelinde temel Seviye-0'da farklı makine öğrenmesi algoritmaları modele dahil edilebilir. Genellikle tek tabanlı makine öğrenmesi algoritmalarının performans ölçümünde hangi makine öğrenmesi algoritmasının kullanılacağı ve parametre ayarları tartışma konusudur. Fakat Super Learner topluluk öğrenmesi modelinde bu problemin çözümü daha kolay hale gelmiştir. Aşırı uyumu önlemek için k=kat çapraz doğrulama kullanılır. Burada seçilecek olan k değeri eğer düşük seçilirse yüksek sapma ve düşük varyansa, k değeri yüksek seçilirse düşük sapma ve yüksek varyansa sebep olmaktadır. Yaygın olarak k=10 olarak kullanılmaktadır. Genel olarak Super Learner topluluk öğrenmesi modelinde aşırı uyumu önlemek, tahmin doğruluk performansını arttırmak için tercih edilen bir modeldir.

Bu tez çalışmasındaki amaç ve hedefler doğrultusundaki istenilen aşamalar başarılı bir şekilde gerçekleştirilmiştir. Bu aşamalar ile birlikte 5 farklı araştırma sorumuzun cevapları şu şekildedir:

1. Diyabet hastalığının tespit edilmesinde Rastgele Orman ve Karar Ağacı torbalama topluluk (bagging ensemble) öğrenmesi modelleri %98 doğruluk tahmin performans sonucu ile tek tabanlı (single base) makine öğrenmesi algoritmalarına (ANN:%97, KNN:%96, LR:%93, SVM:%93, NB:%90) kıyasla daha iyi bir performans göstermişlerdir.

2. Diyabet hastalığının tespit edilmesinde Gradyan Arttırma ve XGBoost arttırma topluluk (boosting ensemble) öğrenmesi modelleri %97 ile genel olarak tek tabanlı (single base) makine öğrenmesi algoritmalarına (ANN:%97, KNN:%96, LR:%93, SVM:%93, NB:%90) kıyasla daha iyi bir performans göstermiştirlerdir. Diğer taraftan tek tabanlı ANN makine öğrenmesi algoritmasıyla performansları aynı oranda elde edilmiştir.

3. Diyabet hastalığının tespit edilmesinde Super Learner yığınlama topluluk (stacking ensemble) öğrenmesi modelleri %99,4 ile tek tabanlı (single base) makine öğrenmesi algoritmalarına (ANN:%97, KNN:%96, LR:%93, SVM:%93, NB:%90) kıyasla daha iyi bir performans göstermişlerdir.

4. Diyabet hastalığının tespit edilmesinde Super Learner yığınlama topluluk (stacking ensemble) öğrenmesi modelleri %99,4 ile topluluk (bagging ensemble) öğrenmesi modelleri (RF:%98, DT:%98) doğruluk tahmin sonucuna kıyasla daha iyi bir performans göstermişlerdir.

5. Diyabet hastalığının tespit edilmesinde Super Learner yığınlama topluluk (stacking ensemble) öğrenmesi modelleri %99,4 ile arttırma topluluk (boosting ensemble) öğrenmesi modellerine (AB:%90, GB:%97, XGBoost: %97) kıyasla daha iyi bir performans göstermişlerdir.

Diyabet hastalığının tespit edilmesinde aynı veri setini kullanılarak yapılan çalışmaları incelediğimizde; Ilyas Özer'in (2020) yaptığı çalışmada LSTM ağı kullanılmıştır. Model 2 ayrı gizli katmanlı bir yapıdan oluşmaktadır. Çalışmada 10 kat çapraz doğrulama tekniği kullanılmıştır. Yapılan çalışma sonucunda %98,65 F1-Skor puanı, %98,43 hassasiyet, %98,9 kesinlik sonuçları elde edilmiştir. Xue ve arkadaşlarının (2020) yaptığı çalışmada ise Destek Vektör Makineleri, Naif Bayes, LightGBM olmak üzere 3 farklı makine öğrenmesi algoritması kullanmışlardır. Veri setine hold-out (80:20) tekniği uygulanmıştır. En yüksek doğruluk performansını %98,07 ile Destek Vektör Makineleri ile edilmiştir. Alpan ve Ilgi'nin (2020) çalışmasında ise Lojistik Regresyon, Naif Bayes, Rastgele Orman, Rastgele Ağaç, Karar Ağacı, K-En Yakın

Komşu, Destek Vektör Makineleri olmak üzere 7 farklı makine öğrenmesi algoritması kullanmışlardır. En yüksek doğruluk performansını %98,07 ile KNN makine öğrenmesi algoritması ile edilmiştir. Yurttakal ve Baş'ın (2021) çalışmasında ise yığınlanmış derin sinir ağı kullanılmıştır. Modelde iki derin sinir ağı yapısından oluşmaktadır. Hiper parametreler optimize olarak Adam, gizli katmanda ve çıktı katmanlarında ise ReLU ve Sotfmax aktivasyon fonksiyonları ve epoch: 300 olarak ayarlanmıştır. 10 kat çapraz doğrulama ve hold-out (70:30) tekniği kullanılmıştır. Elde edilen sonuçlar 10 kat çapraz doğrulama %98,71 ve hold-out %99,36 doğruluk oranları elde edilmiştir. Sadhu ve Jadli'nin (2021) çalışmasında ise Lojistik Regresyon, Naif Bayes, Rastgele Orman, MLP, Karar Ağacı, K-En Yakın Komşu, Destek Vektör Makineleri olmak üzere 7 farklı makine öğrenmesi algoritması kullanmışlardır. En yüksek doğruluk performansını %98,07 ile Rastgele Orman makine öğrenmesi algoritması ile edilmiştir.

Super Learner topluluk öğrenmesi modelini kullanarak diyabet hastalığının tespit edilmesinde Saxena ve diğerleri (2021) bir çalışma yapmışlardır. Bu çalışmada veri seti eğitim (%75) ve test (%25) veri seti olarak ikiye ayrılmıştır. Çapraz doğrulama için k=10 parametre ayarı yapılmıştır. Özellik seçimi için Ekstra Ağaçlar sınıflandırıcısının ortalama azaltma yöntemini kullanarak 12 özellik (Polyphagia, Age, Itching, Visual blurring, Delayed healing, Alopecia, Irritability, Partial paresis, Sudden weight loss, Gender, Polyuria, Polydipsia) kullanmışlardır. Super Learner topluluk öğrenmesi modeli için Seviye-0'da (Lojistik Regresyon, Lineer Diskriminant Analizi, K-En Yakın Komşu, Karar Ağacı, Naif Bayes, Destek Vektör Makinesi, AdaBoost, Gradyan Arttırma, Rastgele Orman ve Ekstra Ağaç Sınıflandırıcı) 10 tane makine öğrenmesi algoritması kullanılmıştır. Seviye-1'de meta learner olarak Gradyan Arttırma makine öğrenmesi algoritması kullanılmıştır. Bu çalışma sonucunda %97 ile en iyi tahmin performansını elde etmişlerdir. Bu çalışmada öznitelik seçim yöntemi olarak seçilen yöntemden farklı bir yöntem uygulansaydı sonuç tahmin performansı daha iyi olabilirdi. Ayrıca Super Learner modelinin Seviye-0'da seçilen temel makine öğrenmesi algoritmalarında farklı seçimler yapılarak tahmin performans sonucunda artış sağlanabilirdi. Biz çalışmamızda Seviye-0'da farklı makine öğrenmesi algoritmalarının birden çok kombinasyonlarını oluşturarak optimum sonuç elde etmeye çalıştık. Seviye-0'da çok sayıda makine öğrenmesi algoritmasının kullanılması doğruluk tahmin

performansının artmasını sağlamadığı için 4 farklı öğrenmesi öğrenme algoritması uygulandığında optimum sonuca ulaşmayı başardık. Çizelge 6.1'de literatürde diyabet tespiti ile yapılmış çalışmalar sunulmaktadır.

Çizelge 6.1 Literatürde diyabet tespiti ile yapılmış çalışmalar

Yazar(lar) Adı	**Yıl**	**Makine Öğrenmesi Algoritmaları**	**Model**	**ACC**
Ilyas Özer	2020	LSTM	LSTM	0,986
Xue ve ark.	2020	SVM, NB, LightGBM	SVM	0,980
Alpan ve Ilgi	2020	LR, NB, RF, RT, DT, KNN, SVM	KNN	0,980
Yurttakal ve Baş	2021	SDNN	DSL	0,9936
Sadhu ve Jadli	2021	LR, NB, RF, MLP, DT, KNN, SVM	RF	0,987
Saxena ve ark.	2021	S0: (LR, LDA, KNN, DT, NB, SVM, AB, GB, RF, ET) S1: GB	SL	0,97
Bu çalışma	**2021**	**S1: (LR, GB, DT, RF)** **S1: SVM**	**SL**	**0,994**

Çizelge 6.1- AB(AdaBoost), DT (Decision Tree), ET (Ekstra Tree), KNN (K-Nearest Neighbor), LDA (Linear Discriminant Analysis), LightGBM (Light Gradient Boost Machine), LSTM (Long Short Term Memory), LR (Logistic Regression), RF (Random Forest), RA (Random Tree), SDNN (Stacked Deep Neural Network), SVM (Support Vector Machines)

Bu tez çalışmasında kullanılan veri seti ile ilgili literatürde yapılan mevcut çalışmalara bakıldığında, veri seti öznitelikleri diyabet tipini tahmin edilmesine imkan tanımıyordu. Bu nedenle gelecekteki çalışmalarımızda diyabet tipini tahmin etmeyi ve keşfetmeyi hedefliyoruz. Bu hedefe ulaşılması diyabetin tahmin doğruluğunu artırabilir. Ayrıca diyabetin nedenlerini ve diyabetten nasıl kaçınılacağını da incelenebilir. Gelecekteki çalışmalarda kritik hastalıklar için tedavi ve iyileşme yöntemleri tasarlanabilir. Farklı öznitelik seçim yöntemleri denenebilir. Hibrit model (örneğin: Deep Belief Network

gibi) oluşturularak modelin doğru tahmin performansı arttırılabilir. Literatürde Super Learner modeli kurulurken Seviye-0 ve Seviye-1'de LSTM makine öğrenmesi algoritmasını kullanarak yapılmış çalışma olmadığından gelecek çalışmalarda bu makine öğrenmesi algoritmasıyla bir çalışma yapılıp sonuçlar gözlemlenebilir.

KAYNAKLAR

Akula, R., Nguyen, N. and Garibay, I. 2019. Supervised Machine Learning based Ensemble Model for Accurate Prediction of Type 2 Diabetes. Conference Proceedings - IEEE Southeastcon, pp. 1-8, Huntsville.

Alcalá-Rmz, V., Galván-Tejada, C. E., García-Hernández, A., Valladares-Salgado, A., Cruz, M., Galván-Tejada, J. I., Celaya-Padilla, J. M., Luna-Garcia, H. and Gamboa-Rosales, H. 2021. Identification of people with diabetes treatment through lipids profile using machine learning algorithms. Healthcare (Switzerland), 9(4): 1-14.

Alghamdi, M., Al-Mallah, M., Keteyian, S., Brawner, C., Ehrman, J. and Sakr, S. 2017. Predicting diabetes mellitus using SMOTE and ensemble machine learning approach: The Henry Ford ExercIse Testing (FIT) project. PLoS ONE, 12(7): 1-15.

Alpan, K. and Ilgi, G. S. 2020. Classification of Diabetes Dataset with Data Mining Techniques by Using WEKA Approach. 4th International Symposium on Multidisciplinary Studies and Innovative Technologies, pp. 1-7, İstanbul.

Alpaydin, E. 2021. Machine Learning, revised and updated edition. MIT Press, Pages 280, Cambridge.

Anteby, R., Lucander, A., Bachul, P. J., Pyda, J., Grybowski, D., Basto, L., Generette, G. S., Perea, L., Golab, K., Wang, L., Tibudan, M., Thomas, C., Fung, J. and Witkowski, P. 2021. Evaluating the Prognostic Value of Islet Autoantibody Monitoring in Islet Transplant Recipients with Long-Standing Type 1 Diabetes Mellitus. Journal of Clinical Medicine, 10: 1-10.

Aronson, D. 2008. Hyperglycemia and the Pathobiology of Diabetic Complications. Advances in Cardiology, 45: 1–16.

Association, A. D. 2010. Diagnosis and Classification of Diabetes Mellitus. Diabetes Care, 33: 62–69.

Ayyüce Kızrak, 2019. Web sitesi. https://towardsdatascience.com/comparison-of-activation-functions-for-deep-neural-networks-706ac4284c8a. Erişim Tarihi: 20.10.2019.

Sharma G. and Hengaju, U. 2020. Performance Analysis of Data Mining Classification Algorithm to Predict Diabetes. Int. J. Advanced Networking and Applications, 12: 4509-4518.

Bailey, T. S., Walsh, J. and Stone, J. Y. 2018. Emerging Technologies for Diabetes Care. Diabetes Technology and Therapeutics, 20: 278–284.

Balasubramanian, K., Dabadghao, P., Bhatia, V., Colman, P. G., Gellert, S. A., Bharadwaj, U., Agrawal, S., Shah, N. and Bhatia, E. 2003. High Frequency of Type 1B (Idiopathic) Diabetes in North Indian Children With Recent-Onset Diabetes. Diabetes Care, 26: 2697–2697.

Barnard, G. A. 1958. Studıes In The Hıstory Of Probabılıty And Statıstıcs: Ix. Thomas Bayes's Essay Towards Solvıng A Problem In The Doctrıne Of Chances: Reproduced with the permission of the Council of the Royal Society from The Philosophical Transactions (1763), 53, 370-418. Biometrika, 45: 293–295.

Bauer, E. and Kohavi, R. 1999. Empirical comparison of voting classification algorithms: bagging, boosting, and variants. Machine Learning, 36: 105–139.

Birjais, R., Mourya, A. K., Chauhan, R. and Kaur, H. 2019. Prediction and diagnosis of future diabetes risk: a machine learning approach. SN Applied Sciences, 1: 1–8.

Breiman, L. 1996. Bagging predictors. Machine Learning, 24: 123–140.

Brownlee, J. 2021. Ensemble Learning Algorithms With Python Make Better Predictions with Bagging, Boosting and Stacking. Machine Learning Mastery, Pages 450, San Francisco.

Burges, C. J. C. 1998. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2: 121–167.

Buyrukoğlu, S. 2021. Early Detectıon Of Alzheımer's Dısease Usıng Data Mınıng: Comparıson Of Ensemble Feature Selectıon Approaches. Konya Journal of Engineering Sciences, 9: 50–61.

Chen, T. and Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, California.

Cohen, J. 2013. Statistical Power Analysis for the Behavioral Sciences. Statistical Power Analysis for the Behavioral Sciences, Pages 567, New York.

Cole, J. B. and Florez, J. C. 2020. Genetics of diabetes mellitus and diabetes complications. Nature Reviews Nephrology 16: 377–390.

Cortes, C. and Vapnik, V. 1995. Support-vector networks. Machine Learning, 20: 273–297.

Cover, T. M. and Hart, P. E. 1967. Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory, 13: 21–27.

Cramer, J. S. 2005. The Origins of Logistic Regression. SSRN Electronic Journal, 4: 119-125.

CS231n, 2020. Web site. https://cs231n.github.io/convolutional-networks/. Erişim Tarihi: 12.10.2020.

Cursos INAOEP, 2017. Web sitesi. https://ccc.inaoep.mx/~emorales/Cursos/NvoAprend/Acetatos/svm2017.pdf. Erişim Tarihi: 10.10.2020.

Dataaspirant, 2020. Web sitesi. https://dataaspirant.com/ensemble-methods-bagging-vs-boosting-difference/. Erişim Tarihi: 10.10.2021.

Deberneh, H. M. and Kim, I. 2021. Prediction of type 2 diabetes based on machine learning algorithm. International Journal of Environmental Research and Public Health, 18(6): 1-14.

Soman, K. P., Loganathan, R., Ajay, V. 2009. Machine Learning with SVM and other Kernel methods. PHI Learning Pvt. Ltd., Page 486, India.

Diabetes Atlas, 2019. Web sitesi. https://www.diabetesatlas.org/. Erişim Tarihi: 12.01.2021.

Diabetes Atlas, 2021. Web sitesi. https://diabetesatlas.org/atlas/tenth-edition/. Erişim Tarihi: 12.12.2021.

Eli Bendersky, 2003. Web sitesi. http://eli.thegreenplace.net/2016/the-softmax-function-and-its-derivative/. Erişim Tarihi: 16.10.2021.

Elish, M. O., Helmy, T. and Hussain, M. I. 2013. Empirical study of homogeneous and heterogeneous ensemble models for software development effort estimation. Mathematical Problems in Engineering, 2013: 1- 21.

Emon, M. U., Keya, M. S., Kaiser, M. S., Islam, M. A., Tanha, T. and Zulfiker, M. S. 2021. Primary Stage of Diabetes Prediction using Machine Learning Approaches.

International Conference on Artificial Intelligence and Smart Systems, pp. 364–367, Coimbatore.

Ergin, E., Akin, S., Kazan, S., Erdem, M. E., Tekce, M. and Aliustaoglu, M. 2013. Lipid Profile of Diabetic Patients: Awareness and the Rate of Treatment Success. The Journal of Kartal Training and Research Hospital, 24: 157–163.

Evert, A. B., Dennison, M., Gardner, C. D., Garvey, W. T., Karen Lau, K. H., MacLeod, J., Mitri, J., Pereira, R. F., Rawlings, K., Robinson, S., Saslow, L., Uelmen, S., Urbanski, P. B. and Yancy, W. S. 2019. Nutrition therapy for adults with diabetes or prediabetes: A consensus report. American Diabetes Association, 42(5): 731–754.

Eatcs, 2003. Web Sitesi. https://eatcs.org/index.php/goedel-prize. Erişim Tarihi: 11.12.2020.

Freedman, D., Pisani, R. and Purves, R. 1998. Statistics. WW Norton and Co Inc (Np), Pages 720, New York.

Freund, Y. 2001. An Adaptive Version of the Boost by Majority Algorithm. Machine Learning, 43: 293–318.

Friedman, J. H. 2001. Greedy function approximation: A gradient boosting machine. Project Euclid, 29(5): 1189–1232.

Friedman, J. H. 2002. Stochastic gradie Freund nt boosting. Computational Statistics and Data Analysis, 38(4): 367–378.

Gavin, J. R., Alberti, K. G. M. M., Davidson, M. B., DeFronzo, R. A., Drash, A., Gabbe, S. G., Genuth, S., Harris, M. I., Kahn, R., Keen, H., Knowler, W. C., Lebovitz, H., Maclaren, N. K., Palmer, J. P., Raskin, P., Rizza, R. A. and Stern, M. P. 2003. Report of the expert committee on the diagnosis and classification of diabetes mellitus. Diabetes Care, 26: 5-20.

Gourieroux, C. and Monfort, A. 1981. Asymptotic properties of the maximum likelihood estimator in dichotomous logit models. Journal of Econometrics, 17(1): 83–97.

Hall, P., Park, B. U. and Samworth, R. J. 2008. Choice of neighbor order in nearest-neighbor classification. Annals of Statistics, 36(5): 2135–2152.

Henry, C. and Hague, W. M. 2015. Type 2 diabetes mellitus. Case Studies in Assisted Reproduction: Common and Uncommon Presentations, 1: 1–6.

Ho, T. K. 1998. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8): 832–844.

Ozer, I. 2020. Uzun Kısa Dönem Bellek Ağlarını Kullanarak Erken Aşama Diyabet Tahmini Early-Stage Diabetes Prediction Using Long Short-Term Memory Networks. Müh. Bil.ve Araş. Dergisi, 2(2): 50–57.

Joshi, R. and Alehegn, M. 2017. Analysis and prediction of diabetes diseases using machine learning algorithm: Ensemble approach IRJET Journal Analysis and prediction of diabetes diseases using machine learning algorithm. International Research Journal of Engineering and Technology, 4(10): 426-436.

Kabir, M. F. and Ludwig, S. A. 2019. Enhancing the Performance of Classification Using Super Learning. Data-Enabled Discovery and Applications, 3: 1–13.

Katsarou, A., Gudbjörnsdottir, S., Rawshani, A., Dabelea, D., Bonifacio, E., Anderson, B. J., Jacobsen, L. M., Schatz, D. A. and Lernmark, A. 2017. Type 1 diabetes mellitus. Nature Reviews Disease Primers, 3: 1–17.

Kaur, H. and Kumari, V. 2020. Predictive modelling and analytics for diabetes using a machine learning approach. Applied Computing and Informatics, 16: 1-11.

Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I. and Chouvarda, I. 2017. Machine Learning and Data Mining Methods in Diabetes Research. Computational and Structural Biotechnology Journal, 15: 104–116.

KI, R. (2007). Diabetes treatment-bridging the divide. The New England Journal of Medicine, 356(15): 1499–1501.

Kothari, R., Kothari, R. and Dong, M. 2001. Decision Trees For Classification: A Review And Some New Results. World Scientific, 4: 169–184.

Kowsher, M., Turaba, M. Y., Sajed, T. and Mahabubur Rahman, M. M. 2019. Prognosis and treatment prediction of type-2 diabetes using deep neural network and machine learning classifiers. 2019 22nd International Conference on Computer and Information Technology (ICCIT), pp. 1-6, Dhaka.

Kuhn, M. and Johnson, K. 2013. Applied Predictive Modeling. Springer, Pages 613, English.

Kumari, S., Kumar, D. and Mittal, M. 2021. An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier. International Journal of Cognitive Computing in Engineering, 2: 40–46.

Lai, H., Huang, H., Keshavjee, K., Guergachi, A. and Gao, X. 2019. Predictive models for diabetes mellitus using machine learning techniques. BMC Endocrine Disorders, 19: 1-9.

Lemaréchal, C. 2012. Cauchy and the Gradient Method. Documenta Mathematica, 2012: 251–254.

Li, G., Shen, M., Li, M. and Cheng, J. 2021. Personal credit default discrimination model based on super learner ensemble. Mathematical Problems in Engineering, 2021: 1-16.

UCI Machine Learning Repository, 2020. Web sitesi. https://archive.ics.uci.edu/ml/datasets/Early+stage+diabetes+risk+prediction+data set. Erişim Tarihi: 01.01.2020.

Mareeswari, V., Saranya, R., Mahalakshmi, R. and Preethi, E. 2017. Prediction of diabetes using data mining techniques. Research Journal of Pharmacy and Technology, 10(4): 1098–1104.

Mason, L., Baxter, J., Bartlett, P. and Frean, M. 2000. Boosting algorithms as gradient descent. Advances in Neural Information Processing Systems, 12: 512–518.

Medium Akca M.F., 2020. Web sitesi. https://medium.com/deep-learning-turkiye/karar-ağaçları-makine-öğrenmesi-serisi-3-a03f3ff00ba5. Erişim Tarihi: 12.10.2021.

Medium Akca M.F., 2020. Web sitesi. Https://Medium.Com/Deep-Learning-Turkiye/Nedir-Bu-Destek-Vektör-Makineleri-Makine-Öğrenmesi-Serisi-2-94e576e4223e. Erişim Tarihi: 12.10.2021.

Medium Artar M.A., 2020. Web sitesi. https://medium.com/@afranur.artar/keşifsel-veri-analizi-2-ed84b898febe . Erişim Tarihi: 12.09.2021.

Michailidis, M. 2017. Investigating machine learning methods in recommender systems Improving prediction for the top K items. Doctor Thesis, University Colloge London, Pages 235, London.

Mishra, S., Chaudhury, P., Mishra, B. K. and Tripathy, H. K. 2016. An implementation of Feature ranking using Machine learning techniques for Diabetes disease prediction. ACM International Conference Proceeding Series, pp. 1-3, Udaipur.

Mujawar, I. K., Jadhav, B. T., Waghmare, V. B. and Patil, R. Y. 2021. Development of diabetes diagnosis system with artificial neural network and open source

environment. 2021 International Conference on Emerging Smart Computing and Informatics, pp. 477–482, Pune.

Nayeem, M. J., Rana, S., Alam, F. and Rahman, M. A. 2021. Prediction of Hepatitis Disease Using K-Nearest Neighbors, Naive Bayes, Support Vector Machine, Multi-Layer Perceptron and Random Forest. 2021 International Conference on Information and Communication Technology for Sustainable Development, pp. 280–284, Dhaka.

Odegua, R. O. 2020. An Empirical Study of Ensemble Techniques (Bagging, Boosting and Stacking). Deep Learning IndabaX, pp. 1-6, Nigeria.

Perveen, S., Shahbaz, M., Guergachi, A. and Keshavjee, K. 2016. Performance Analysis of Data Mining Classification Techniques to Predict Diabetes. Procedia Computer Science, 82: 115–121.

Polikar, R. 2012. Ensemble Machine Learning, Springer, Pages 34, Boston.

Putri, N. K., Rustam, Z. and Sarwinda, D. 2019. Learning Vector Quantization for Diabetes Data Classification with Chi-Square Feature Selection. IOP Conference Series: Materials Science and Engineering, 546: 1-8.

Qin, J., Chen, L., Liu, Y., Liu, C., Feng, C. and Chen, B. 2020. A machine learning methodology for diagnosing chronic kidney disease. IEEE Access, 8: 20991–21002.

Quilan, J. R. 1988. Decision trees and multi-valued attributes. Machine intelligence 11, pp. 305-318, New York.

Ram, A. and Vishwakarma, H. 2021. Diabetes Prediction using Machine learning and Data Mining Methods. IOP Conference Series: Materials Science and Engineering, pp. 1-11, Mathura.

Ramachandran, P., Zoph, B. and Le Google Brain, Q. V. 2018. Searching for activation functions. 6th International Conference on Learning Representations, pp. 1-13, Canada.

Rokach, L. and Maimon, O. 2009. Classification Trees. In Data Mining and Knowledge Discovery Handbook, pp. 149–174, Boston.

Rosaen, K., 2016. Web sitesi. http://karlrosaen.com/ml/learning-log/2016-06-20/. Erişim Tarihi: 10.09.2021.

Sadhu, A. and Jadli, A. 2021. Early-Stage Diabetes Risk Prediction: A Comparative Analysis of Classification Algorithms. International Advanced Research Journal in Science, Engineering and Technology, 8(2): 193-201.

Saxena, S., Mohapatra, D., Padhee, Subhransu, G. and Sahoo, K. 2021. Machine learning algorithms for diabetes detection: a comparative evaluation of performance of algorithms. Evolutionary Intelligence, 1: 1–17.

Schapire, R. E. 1990. The strength of weak learnability. Machine Learning, 5: 197–227.

Schapire, R. E. 2013. Explaining adaboost. In Empirical Inference, pp. 37–52, Berlin Heidelberg.

Shamreen Ahamed, B. and Sumeet Arya, M. 2021. Prediction of Type-2 Diabetes using the LGBM Classifier Methods and Techniques. In Turkish Journal of Computer and Mathematics Education, 12(12): 223-231.

Shannon, C. E. 1948. A Mathematical Theory of Communication. In The Bell System Technical Journal, 27(3): 379-423.

Shatovskaya, T., Repka, V. and Good, A. 2006. Application of the Bayesian networks in the informational modeling. Modern Problems of Radio Engineering, Telecommunications and Computer Science Proceedings of International Conference, pp. 108-108, Ukraine.

Sill, J., Takacs, G., Mackey, L. and Lin, D. 2009. Feature-Weighted Linear Stacking. CoRR, abs/0911.0460: 1-17

Sisodia, D. and Sisodia, D. S. 2018. Prediction of Diabetes using Classification Algorithms. Procedia Computer Science, 132: 1578–1585.

Sollich, P. and Krogh, A. 1995. Learning with ensembles : how over-fitting can be useful, pp. 190-196, Cambridge.

Sreedharan, J., Muttappallymyalil, J., Sharbatti, S. al, Hassoun, S., Safadi, R., Abderahman, I., Hameed, W. A., Ibrahim, A. M., Takana, M. T. and Fouda, A. M. 2015. Incidence of Type 2 Diabetes Mellitus among Emirati Residents in Ajman, United Arab Emirates. Korean Journal of Family Medicine, 36(5): 253–257.

Srinivas, D. S. and Kumar, M. A. 2013. Attrıbute And Informatıon Gaın Based Feature Selectıon Technıque For Cluster Ensemble: Hybrıd Majorıty Votıng Based Varıable Importance Measure. International Journal of Innovative Technology and Research, 1(6): 607-610.

Sugihara, S., Kikuchi, T., Urakami, T., Yokota, I., Kikuchi, N., Kawamura, T. and Amemiya, S. 2021. Residual endogenous insulin secretion in Japanese children with type 1A diabetes. Clinical Pediatric Endocrinology, 30(1): 27–33.

Taşcı, E. and Onan, A. 2017. K-En Yakın Komşu Algoritması Parametrelerinin Sınıflandırma Performansı Üzerine Etkisinin İncelenmesi. Akademik Bilişim Kongresi, s.1-8, Aydın.

Taz, N. H., Islam, A. and Mahmud, I. 2021. A Comparative Analysis of Ensemble Based Machine Learning Techniques for Diabetes Identification. 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques, pp. 1–6, Bangladesh.

Tom M. Mitchell, 1999. Machine learning. McGraw-Hill, Pages 414, New York.

Towardsdatascience, 2019. Web sitesi. https://towardsdatascience.com/understanding-random-forest-58381e0602d2. Erişim tarihi: 12.02.2020.

Töscher, A., Jahrer, M. and Bell, R. M. 2009. The BigChaos Solution to the Netflix Grand Prize. Commendo Research and Consulting, 22: 1-52.

Tsai, C. F., Hsu, Y. F. and Yen, D. C. 2014. A comparative study of classifier ensembles for bankruptcy prediction. Applied Soft Computing, 24: 977–984.

Tukey, J. W. 1977. Exploratory Data Analysis (Book Section). Exploratory Data Analysis, pp. 61–100, New York.

Türkiye Diyabet Vakfı. 2021. Diyabet Tanı ve Tedavi Rehberi, 10. Baskı, Türkiye Diyabet Vakfı, 204 sayfa, İstanbul.

Van Der Laan, M. J., Polley, E. C. and Hubbard, A. E. 2007. Super learner. Statistical Applications in Genetics and Molecular Biology, 6(1): 1-21.

Varian, H. R. 2005. Bootstrap Tutorial. The Mathematica Journal, 9(4): 768–775.

Vijayan, V. V. and Anjali, C. 2016. Prediction and diagnosis of diabetes mellitus - A machine learning approach. 2015 IEEE Recent Advances in Intelligent Computational Systems, pp. 122–127, Trivandrum.

Witten, I. H., Frank, E., Hall, M. A. and Pal, C. J. 2016. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Pages 654, California.

Wolpert, D. H. 1992. Stacked generalization. Neural Networks, 5(2): 241–259.

Xue, J., Min, F. and Ma, F. 2020. Research on Diabetes Prediction Method Based on Machine Learning. Journal of Physics: Conference Series, pp. 1-7, Canada.

Yadav, D. C. and Pal, S. 2021. An Experimental Study of Diversity of Diabetes Disease Features by Bagging and Boosting Ensemble Method with Rule Based Machine Learning Classifier Algorithms. SN Computer Science, 2: 1-10.

Yadav, S. and Shukla, S. 2016. Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. 2016 IEEE 6th International Conference on Advanced Computing, pp. 78–83, Bhimavaram.

Yang, H., Luo, Y., Ren, X., Wu, M., He, X., Peng, B., Deng, K., Yan, D., Tang, H. and Lin, H. 2021. Risk Prediction of Diabetes: Big data mining with fusion of multifarious physical examination indicators. Information Fusion, 75: 140–149.

Yurttakal, A. H. and Baş, H. 2021. Possibility Prediction Of Diabetes Mellitus At Early Stage Via Stacked Ensemble Deep Neural Network. Afyon Kocatepe Üniversitesi Fen Ve Mühendislik Bilimleri Dergisi, 21: 812–819.

Yuvaraj, N. and SriPreethaa, K. R. 2019. Diabetes prediction in healthcare systems using machine learning algorithms on Hadoop cluster. Cluster Computing, 22: 1–9.

Zhang, H. and Li, D. 2008. Naive Bayes Text Classifier. 2007 IEEE International Conference on Granular Computing, pp. 708–708, Fremont.

Zhang, H. and Sheng, S. 2004. Learning weighted naive bayes with accurate ranking. Proceedings - Fourth IEEE International Conference on Data Mining, pp. 567–570, Brighton.

Zhou, Z. H. 2012. Ensemble methods: Foundations and algorithms. Chapman and Hall/CRC, pages 236, New York.

Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y. and Tang, H. 2018. Predicting Diabetes Mellitus With Machine Learning Techniques. Frontiers in Genetics, 9: 1-10.

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı ve Soyadı : Ayşe DOĞRU

Eğitim

Yüksek Lisans	Çankırı Karatekin Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı	2022-Halen
Lisans	Trakya Üniversitesi Mühendislik-Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümü	2006-2010

İş Deneyimi

Yıl	Kurum	Görev
2012-Halen	ÇAKÜ, Bilgi İşlem D.B.	Bilgisayar Mühendisi