

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

ARAÇ MUAYENE VERİLERİ İLE VERİ MADENCİLİĞİ UYGULANMASI

Gizem ÇETİN

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**ANKARA
2021**

Her hakkı saklıdır

ÖZET

Yüksek Lisans Tezi

ARAÇ MUAYENE VERİLERİ İLE VERİ MADENCİLİĞİ UYGULANMASI

Gizem ÇETİN

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Ömer Özgür TANRIÖVER

Araç muayenesi trafikte yer alan motorlu ya da motorsuz araçlar için teknik yeterliliklerinin ölçüldüğü, yolcu ve trafik güvenliğinin sağlanıp sağlanmadığının tespit edildiği sistemdir. Türkiye’de karayolunda seyreden araçların teknik muayenelerini daha etkin ve sağlıklı bir şekilde yapmak ve karayolu trafik güvenliğini sağlamak amacıyla her yıl yaklaşık 6 milyona yakın aracın periyodik muayenesi gerçekleştirilmektedir. Bu tez çalışmasında Ulaştırma ve Altyapı Bakanlığı araç muayene verileri kullanılarak makine öğrenmesi algoritmaları ve derin sinir ağları kullanılarak araç muayene sonucu tahmini yapılmıştır ve veriler üzerinde örnekleme yöntemi kullanılarak makine öğrenmesi algoritmalarının performansları karşılaştırılmıştır. Ayrıca birliktelik kuralları yöntemlerinden apriori algoritması kullanılarak araçların muayene sonucunda birlikte görülen kusurların analizi gerçekleştirilmiştir ve araç kusurları arasında anlamlı ilişkiler bulunmuştur. Çalışmada yaklaşık 7 milyon araç muayene verisi kullanılmıştır. Araç muayene sonucunu tahmin etmek için aracın km, model, yakıt tipi ve muayene sonucu çıkan kusur verileri kullanılmıştır. Makine öğrenme algoritmalarından Lojistik Regresyon (LR), Naive Bayes (NB), Karar Ağaçları (DT), Rastgele Orman (RF), K-En Yakın Komşu (KNN), Gradyan Yükseltme (XG Boost), Ada Boost, Derin Sinir Ağı (DNN) ve Evrimsel Sinir Ağı (CNN) kullanılmıştır. Her bir model AUC, ROC eğrisi, doğruluk, kesinlik, hatırlama ve F1 skor değerleri açısından karşılaştırılmıştır. Makine öğrenme modellerinin hafif kusurlu, ağır kusurlu ve emniyetsiz olarak sınıflandırılan muayene sonucu tahminini yüksek doğrulukla öngörebildiği sonucuna ulaşılmıştır.

Eylül 2021, 101 sayfa

Anahtar Kelimeler: Araç Muayene, Makine Öğrenmesi, Sınıflandırma, Birliktelik Analizi, Tahmin.

ABSTRACT

Master Thesis

APPLICATION OF DATA MINING WITH VEHICLE INSPECTION DATA

Gizem ÇETİN

Ankara University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Asst. Prof. Dr. Ömer Özgür TANRIÖVER

Vehicle inspection is a system in which the technical qualifications of motorized or non-motorized vehicles in traffic are measured and whether passenger and traffic safety are provided or not. In Turkey, periodic inspections of nearly 6 million vehicles are carried out every year in order to carry out technical inspections of vehicles traveling on the road more effectively and healthily and to ensure road traffic safety. In this thesis, vehicle inspection results were estimated by using machine learning algorithms and deep neural networks using the vehicle inspection data of the Ministry of Transport and Infrastructure, and the performances of machine learning algorithms were compared using the sampling method on the data. In addition, using the apriori algorithm, which is one of the association rules methods, the analysis of the defects seen as a result of the inspection of the vehicles was carried out and significant relationships were found between the vehicle defects. Approximately 7 million vehicle inspection data were used in the study. In order to estimate the vehicle inspection result, the vehicle's km, model, fuel and inspection results were used. Logistic Regression (LR), Naive Bayes (NB), Decision Trees (DT), Random Forest (RF), K-Nearest Neighbor (KNN), Gradient Boost (XG Boost), Ada Boost, Deep Neural Network (DNN) from machine learning algorithms and Convolutional Neural network (CNN) was used. Each model was compared in terms of AUC, ROC curve, accuracy, precision, recall and F1 score values. It has been concluded that machine learning models can predict with high accuracy the estimation of lightly defective, severely defective and unsafe inspection result using vehicle inspection data.

September 2021, 101 pages

Keywords: Vehicle Inspection, Machine Learning, Classification, Association Analysis, Prediction.

TEŞEKKÜRLER

Araştırmalarımın her aşamasında bilgi, öneri ve yardımlarını esirgemeyerek çalışmalarına katkıda bulunan ve yön veren danışman hocam Sayın Dr. Öğr. Üyesi Ömer Özgür TANRIÖVER’e, bu süreç boyunca beni destekleyen aileme ve manevi desteklerini eksik etmeyen arkadaşlarıma teşekkür ederim.

Gizem ÇETİN
Ankara, Eylül 2021



İÇİNDEKİLER

TEZ ONAY SAYFASI

ETİK.....	i
ÖZET.....	ii
ABSTRACT	iii
TEŞEKKÜRLER	iv
İÇİNDEKİLER	v
KISALTMALAR DİZİNİ	vii
ŞEKİLLER DİZİNİ	ix
ÇİZELGELER DİZİNİ	xi
1. GİRİŞ.....	1
2. KURUMSAL TEMELLER ve İLGİLİ ÇALIŞMALAR.....	3
3. MATERYAL VE YÖNTEMLER.....	15
3.1 Materyal	16
3.2 Yöntemler.....	17
3.2.1 Araç muayene verileri ve açıklaması.....	17
3.2.2 Veri görselleştirme	21
3.2.3 Veri ön işleme	23
3.2.3.1 Aykırı değerler (Outlier)	24
3.2.3.2 Veri ölçeklendirme.....	28
3.2.3.3 Dönüştürme	28
3.2.3.4 Özellik seçimi (Feature selection)	29
3.2.3.4.1 Özellik önemi (Feature importance)	29
3.2.3.4.2 Korelasyon matrisi ile ısı haritası (Heatmap).....	30
3.2.3.5 Eğitim ve test veri seti oluşturma	32
3.2.3.6 Veri örnekleme	33
3.2.4 Sınıflandırma algoritmaları	35
3.2.4.1 Lojistik Regresyon (LR)	36
3.2.4.2 K-En Yakın Komşu (KNN)	37
3.2.4.3 Naive Bayes (NB).....	38
3.2.4.4 Karar Ağaçları (Decision Tree)	39
3.2.4.5 Rastgele Orman (Random Forest).....	40
3.2.4.6 XG Boost (XGB).....	41
3.2.4.7 Ada Boost	43
3.2.4.8 Derin Sinir Ağları (DNN)	44

3.2.4.9 Evrişimsel Sinir Ağları (CNN)	46
3.2.5 Birliktelik kuralları.....	47
3.2.5.1 Apriori algoritması	47
3.2.6 Model değerlendirmesi	48
3.2.6.1 Karışıklık matrisi	48
3.2.6.2 AUC-ROC eğrisi	49
4. BULGULAR VE TARTIŞMA	51
4.1 Araştırma Bulguları.....	51
4.2. Tartışma	78
4.3 Geçerliliğe Yönelik Tehditler	84
5. SONUÇ	86
KAYNAKLAR	90
EKLER.....	96
EK 1 Araç Muayene ve Kusur Tablolarını Birleştiren Sorgu	96
EK 2 Makine Öğrenmesi Python Kodları.....	97
EK 3 DNN Python Kodları.....	98
EK 4 CNN Python kodları.....	99
EK 5 Araç Muayene Verileri Kullanım İzni Resmi Yazısı	100
ÖZGEÇMİŞ.....	101

KISALTMALAR DİZİNİ

ADABOOST	Adaptive Boosting
AK	Ağır Kusurlu
ANFIS	Adaptive Neural Fuzzy Interference System
ANN	Artificial Neural Network
AUC	Area Under the Curve
AUROC	Area Under the Receiver Operating Characteristic Curve
BP	Back Propagation
CART	Classification and Regression Tree
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DT	Decision Tree
EM	Emniyetsiz
ESP	Electronic Stability Program
FN	False Negative
FP	False Positive
FTA-NNE	Fault Tree Neural Network
GA	Genetik Algoritma
GB	Gigabayt
GP-RBFN	Type -1 Fuzzy System Gaussian Radial Basis Function Network
HF	Hafif Kusurlu
IDE	Integrated Development Environment
IQR	Inter Quantile Range
KM	Kilometre
KNN	K-Nearest-Neighbors
LR	Logistic Regression
LSSVM	Least Squares Support Vector Machine
ML	Machine Learning
MSE	Mean Square Error
NB	Naive Bayes
NN	Neural Network
PCA	Principal Component Analysis

PNN	Probabilistic Neural Network
RBF	Radial Basis Function
RF	Random Forest
RMSE	Root Mean Squared Error
RNN	Tekrarlayan Sinir Ağları
ROC	Receiver Operating Characteristic
SQL	Structured Query Language
SVM	Support Vector Mechine
TDNN	Time Delayed Neural Network
TN	True Negative
TP	True Positive
YSA	Yapay Sinir Ağları
XG BOOST	Extreme Gradient Boosting

ŞEKİLLER DİZİNİ

Şekil 3.1 Sınıflandırma modeli için iş akışı diyagramı	15
Şekil 3.2 Araç muayene verilerinin istatistiksel açıklaması.....	22
Şekil 3.3 Verilerin yakıt tipine göre dağılımı.....	22
Şekil 3.4 Verilerin araç cinsine göre dağılımı.....	23
Şekil 3.5 Verilerin araç model yılına göre dağılımı	23
Şekil 3.6 IQR metodu.....	24
Şekil 3.7 Araç modeli ve muayene sonucuna göre kutu grafiği	26
Şekil 3.8 Model alanının kutu grafiği	27
Şekil 3.9 Aykırı değerler silindikten sonra model alanının kutu grafiği.....	27
Şekil 3.10 Aykırı değerler silindikten sonra KM alanının kutu grafiği	28
Şekil 3.11 Araç muayene verisetinde özellik önemi.....	30
Şekil 3.12 KM ve model alanının ısı haritası	31
Şekil 3.13 Tüm özneliklerin ısı haritası	32
Şekil 3.14 Eğitim ve test verisi	33
Şekil 3.15 Araç muayene verilerinin muayene sonucuna göre dağılımı.....	34
Şekil 3.16 Az örnekleme	34
Şekil 3.17 Çok örnekleme	35
Şekil 3.18 Uzaklık fonksiyonları.....	37
Şekil 3.19 KNN adımları (Anonymous. 2021g)	38
Şekil 3.20 Karar ağacı yapısı	39
Şekil 3.21 XG Boost algoritmasının karar ağaçlarından evrimi (Anonymous 2021j)....	43
Şekil 3.22 Derin Öğrenme, makine öğrenmesi ve yapay zeka arasındaki ilişki.....	44
Şekil 3.23 Yapay sinir ağı modeli	45
Şekil 3.24 Yapay sinir ağı mimarisi (Zafari vd., 2013)	46
Şekil 3.25 Evrimsel Sinir Ağı katmanları (Anonymous 2018)	46
Şekil 3.26 Karışıklık matrisi	49
Şekil 4.1 Lojistik Regresyon karışıklık matrisi.....	52
Şekil 4.2 Lojistik Regresyon Tüm Veri ROC Eğrisi grafiği.....	52
Şekil 4.3 Lojistik Regresyon Az ve Çok Örnekleme ROC Eğrisi grafiği.....	53
Şekil 4.4 KNN karışıklık matrisi.....	55
Şekil 4.5 KNN Tüm Veri ROC Eğrisi grafiği.....	55

Şekil 4.6 KNN Az ve Çok Örnekleme ROC Eğrisi grafiği	55
Şekil 4.7 Naive Bayes karışıklık matrisi	57
Şekil 4.8 Naive Bayes Tüm Veri ROC Eğrisi grafiği	57
Şekil 4.9 Naive Bayes Az ve Çok Örnekleme ROC Eğrisi grafiği.....	58
Şekil 4.10 Karar Ağacı karışıklık matrisi.....	59
Şekil 4.11 Karar Ağacı Tüm Veri ROC Eğrisi grafiği.....	60
Şekil 4.12 Karar Ağacı Az ve Çok Örnekleme ROC Eğrisi grafiği.....	60
Şekil 4.13 Rastgele Orman karışıklık matrisi	62
Şekil 4.14 Rastgele Orman Tüm Veri ROC Eğrisi grafiği.....	62
Şekil 4.15 Rastgele Orman Az ve Çok Örnekleme ROC Eğrisi grafiği	62
Şekil 4.16 XG Boost karışıklık matrisi	64
Şekil 4.17 XG Boost Tüm Veri ROC Eğrisi grafiği	64
Şekil 4.18 XG Boost Az ve Çok Örnekleme ROC Eğrisi grafiği	65
Şekil 4.19 Ada Boost karışıklık matrisi	66
Şekil 4.20 Ada Boost Tüm Veri ROC Eğrisi grafiği	67
Şekil 4.21 Ada Boost Az ve Çok Örnekleme ROC Eğrisi grafiği	67
Şekil 4.22 Kullanılan Derin Sinir Ağı modeli.....	69
Şekil 4.23 DNN karışıklık matrisi.....	70
Şekil 4.24 DNN kayıp grafiği	70
Şekil 4.25 DNN doğruluk grafiği.....	70
Şekil 4.26 DNN Tüm Veri ROC Eğrisi grafiği.....	71
Şekil 4.27 DNN Az ve Çok Örnekleme ROC Eğrisi grafiği.....	71
Şekil 4.28 Oluşturulan CNN modeli katmanları	73
Şekil 4.29 Sınıflandırma algoritmalarının ROC Eğrisi grafiği	79
Şekil 4.30 Sınıflandırma algoritmaları doğruluk değerleri	83

ÇİZELGELER DİZİNİ

Çizelge 3.1 Araç muayene verileri açıklaması.....	18
Çizelge 3.2 ARAC_MUAYENE tablosu.....	18
Çizelge 3.3 KUSUR tablosu	18
Çizelge 3.4 Kusur Açıklama tablosu örnek veri	19
Çizelge 3.5 Kusur grupları	20
Çizelge 3.6 Araç muayene örnek veri	20
Çizelge 3.7 Araç muayene verisi alanları ve veri tipleri.....	21
Çizelge 3.8 Araç muayene verilerinin istatistiksel açıklaması.....	21
Çizelge 4.1 Lojistik Regresyon sınıflandırma raporu	53
Çizelge 4.2 KNN sınıflandırma raporu	56
Çizelge 4.3 Naive Bayes sınıflandırma raporu	58
Çizelge 4.4 Karar Ağacı sınıflandırma raporu	61
Çizelge 4.5 Rastgele Orman sınıflandırma raporu	63
Çizelge 4.6 XG Boost sınıflandırma raporu.....	65
Çizelge 4.7 Ada Boost sınıflandırma raporu.....	68
Çizelge 4.8 DNN sınıflandırma raporu	72
Çizelge 4.9 Apriori algoritmasında kullanılan veri seti	73
Çizelge 4.10 Apriori algoritması birliktelik kuralları	75
Çizelge 4.11 Apriori algoritması sonucu çıkan kurallar	76
Çizelge 4.12 Sınıflandırma algoritmalarının sonuçlarının karşılaştırılması	78

1. GİRİŞ

Türkiye’de uluslararası standartlarda gerçekleştirilen periyodik araç muayenesi hizmeti verilmektedir. 197 sabit, 3 motosiklet ve 73 gezici istasyonu ile Türkiye’nin tamamında hizmet verilen yılda 6 milyona yakın aracın periyodik muayenesi gerçekleştirmektedir. Amaç karayolunda seyreden motorlu ve motorsuz araçların teknik muayenelerini daha etkin ve sağlıklı bir şekilde yapmak, bu suretle araçlar bakımından karayolu trafik güvenliğini sağlamaktır. Araç muayene istasyonlarında periyodik araç muayenesi, zorunlu muayene, tadilat sonrası muayene, yola elverişlilik muayenesi, egzoz gazı emisyon ölçümü hizmetleri verilmektedir.

Araç muayeneleri 2918 sayılı Karayolları Trafik Kanunu, Karayolları Trafik Yönetmeliği, Sanayi ve Ticaret Bakanlığınca yürürlüğe konulan Tip Onay Yönetmelikleri, Araçların İmal Tadil ve Montajı Hakkında Yönetmelik ile bu yönetmelikte yapılan düzenlemeler dikkate alınarak belirli sınıflara göre ve belirli hususları kapsayan kontrol ve testleri içerir. Araç muayene sonuçları kusurlu, hafif kusurlu, ağır kusurlu ve emniyetsiz olarak değerlendirilir. Muayene sonuçları bilgisayar ağında da yer alacak bir Araç Muayene Raporuna işlenir. Bu araç muayene raporu ana merkez bilgisayar ağının yanı sıra Ulaştırma ve Altyapı Bakanlığı, Maliye Bakanlığı ve Emniyet Genel Müdürlüğü bilgisayar ağlarına çevrimiçi olarak elektronik ortamda aktarılır.

Araçların muayenesi, sadece yetkili tek kuruluş olan TÜVTÜRK’te yapılan 2918 sayılı Karayolları Trafik Kanununun 34. maddesi gereğince zorunlu olan, trafiğe çıkan motorlu ya da motorsuz araçların yolcu ve trafik güvenliğine uygun nitelikleri taşıyıp taşımadığının kontrol edilmesidir. Araç muayenesinin amacı trafikteki taşıtların kusurlarının giderilerek bunların sebep olacağı kazaların önüne geçilmesidir.

Son yıllarda araç bakım ve onarımında birçok tespit sistemi geliştirilmiştir. Bu sistemler araçlardaki çeşitli arızaları teşhis etmek için kullanılır. Bu tezin amacı karayolunda seyreden motorlu ve motorsuz araçların teknik muayenelerini daha etkin ve sağlıklı bir şekilde yapmak ve araçların güvenliğini sağlamaktır. Araç kusurlarının birliktelik analizi gerçekleştirilerek araçlarda çıkabilecek hataların erken tespitlerinde yararlı olacaktır.

Bilgisayarların bilgi saklama kapasiteleri artmasıyla bilgilerin kaydedildiği alanların sayısı da bununla birlikte artmaktadır. Bu sebepten dolayı elimizdeki verilerin analizi ve sonucu bu verilerden tahmin etme tekniklerinin önemi karar vericiler, yöneticiler için giderek artmaktadır.

Veriler belirli bir amaç ile işlendiği zaman bir anlam ifade etmeye başlamaktadır. Bu sebepten dolayı büyük miktardaki verileri işleyebilen yöntemleri kullanabilmenin büyük önemi vardır. Bu verileri bilgiye veya anlamlı bir hale dönüştürme işlemleri veri madenciliği teknikleri ile yapılabilmektedir. Günümüzde veri madenciliği işletmeler ve kurumlar için çok büyük önem taşımaktadır. Bu kadar büyük verilerin analizi, bu analiz neticesinde anlamlı bilgi elde etme ve elde edilen bilgiyi yorumlama işlemi insan yeteneği ve ilişkisel veri tabanlarının yapabileceklerinin sınırını aşmaktadır. Veri madenciliği yöntemleri kullanılarak, kurum ve kuruluşların daha etkili kararlar almasına yönelik karar destek sistemlerinde gerekli olan eğilimlerin ve davranış kalıplarının ortaya çıkarılması mümkün olmaktadır. Veri madenciliğinin en önemli özelliklerinden biri veri kümeleri arasındaki benzer eğilimlerin ve davranış kalıplarını ortaya çıkarmasıdır.

Veriye dayanan tekniklerin hızlı bir şekilde büyümesi ve gelişmesiyle birlikte modern otomotiv endüstrisi için hata tespiti ve durum izleme yöntemlerini geliştirmek için yeni imkanlar ortaya çıkmaktadır. Modern yazılım teknolojisinin ve otomotiv elektronik teknolojisinin gelişmesi ile otomobilin hata problemleriyle başa çıkmak kolaylaşmıştır. Hata tespit doğruluğunu ve bakım kalitesini artırmak için otomobil arıza tespit yöntemini incelemek çok önemlidir.

Bu tez çalışmasında Ulaştırma ve Altyapı Bakanlığı Araç Muayene verileri ile makine öğrenmesi algoritmaları ve derin sinir ağları kullanılarak araç muayane sonucu tahmini yapılmıştır ve kullanılan modellerin performansı değerlendirilmiştir. Ayrıca birliktelik kuralları yöntemlerinden apriori algoritması kullanılarak araçların muayene sonucu birlikte görülen kusurların analizi gerçekleştirilmiştir.

2. KURUMSAL TEMELLER ve İLGİLİ ÇALIŞMALAR

Das ve arkadaşları (2019) tarafından yapılan çalışmada Amerika Birleşik Devletlerinin farklı eyaletlerinde araç muayene düzenleme programlarının etkinliğini belirleme amaçlanmıştır. Çalışmada 2011-2016 yılları arası Ulusal Karayolu Trafik Güvenliği İdaresi'nin (NHTSA) araç şikayet veritabanı ve Ölüm Analizi Raporlama Sistemi (FARS) veritabanı olmak üzere iki veritabanı kullanılmıştır. Çalışmada, periyodik muayene programlarının araç durumunu iyileştirdiği ve güvenliği iyileştirdiği fakat güvenlik muayene programlarının çarpışma sonuçları üzerindeki etkisi konusunda araştırmaların sonuçsuz kaldığından bahsedilmiştir. Ayrıca çalışmada NHTSA şikayet verileri gibi daha yeni veri kaynaklarını kullanarak araç muayenesi olan ve olmayan eyaletler arasındaki bu farklılıklar ele alınmıştır. Das ve arkadaşları araç kusurlarının modellerini anlamak için NHTSA şikayet verilerinin metinsel içerikleri üzerinde metin madenciliği ve konu modellemesi yaparak ve devlet tarafından yürütülen araç muayenelerinin güvenlik etkinliğini araştırmak için istatistiksel anlamlılık testi uygulayarak iki önemli katkıda bulunmuştur. Bu çalışmada, araç şikayetlerinin azaltılması için araç güvenliği denetimlerinin sürdürülmesine ihtiyaç olduğu sonucuna varmıştır.

Martín-delosReyes ve arkadaşları (2021) tarafından yapılan çalışmada periyodik motorlu araç muayenelerinin karayolu çarpışmaları ve yaralanmaları üzerindeki etkisini belirlemek amaçlanmıştır. Çalışmada Medline, Web of Science ve Scopus veri tabanlarından 5065 adet çalışmanın sadece altısı kriterleri karşılayarak inceleme için seçilmiştir. İncelenen çalışmalarda karayolu kazaları ile geçerli bir araç muayene sertifikasının yokluğu arasında önemli bir ilişki olduğunu gösteren vaka-kontrol çalışması dışında, diğer çalışmalar çarpışma oranlarında küçük bir azalma (yaklaşık %9) göstermiştir. Ayrıca çalışmada periyodik muayeneler ile karayolu kazaları ve/veya çarpışmalardan kaynaklanan yaralanmalar arasındaki ilişkinin nedenselliğini değerlendirmeyi mümkün kılan yöntemleri kullanan çalışmaların az sayıda olması gibi önemli konuyu gündeme getirmektedir.

Talonen ve arkadaşları (2012) tarafından yapılan çalışmada araç muayenelerinde reddedilme nedenlerinin temel bileşen analizine (PCA) dayalı ağ görselleştirmesini tanıtmışlardır. Bu çalışma ile farklı reddetme nedenleri ile arabalar arasındaki

bağımlılıklar, yapılan ağ görselleştirmesi ile verimli bir şekilde incelenebilir. Çalışmada Finlandiya'da denetlenen binek otomobillerin verileri kullanılmıştır. Muayene reddi lastikler, frenler, direksiyon ve kontrol cihazları gibi 13 farklı sınıfa ayrılmıştır. Araştırmada Gephi görselleştirme aracı kullanılmıştır. Ağ düzeni ForceAtlas2 (FA2) yöntemine dayanmaktadır. Veri görselleştirme için iki farklı yöntemi test edilmiştir. PCA kullanılarak 2 boyutlu bir uzaya yansıtılmıştır. Kullanılan verilerin etkileşimli olarak keşfedilebileceği Google Motion Chart kullanılarak görselleştirilmiştir. PCA görselleştirmesinin bir dereceye kadar veri keşfini kolaylaştırdığı bulunmuştur. Araç muayene verilerini keşfetmek, düzinelerce tablodan ziyade Gephi'deki veya bir tarayıcıdaki görselleştirmeleri kullanarak daha hızlıdır. Çalışmanın amacı, tüm bu verileri en yaygın ret nedenlerini içeren tek bir görselleştirmede toplamaktır. Verileri analiz ederken tamamen sürücülere bağlı olan bazı reddetme nedenleri çıkarılarak kullanıcıların kendi sonuçlarını çıkarmasına yardımcı olmuştur.

Dejun vd. (2011) tarafından yapılan çalışmada SVM (Support Vector Machine) algoritmasına dayalı otomobil motorunun hata tespiti yapılmaktadır. AMES'im yazılımı tarafından motor simülasyon datalarından özellik çıkarımı yapılmaktadır. Bu hata karakteristik parametreleri istatistiksel özellikleri sahiptir ve spesifik fiziksel bir anlamları vardır. PCA (Principal Component Analysis) ile veri fazlalıkları ve boyutları azaltılmaktadır. Daha sonra bu verilere SVM (Support Vector Machine) uygulanarak normalize edilmiştir. Bu makale doğru hata sınıflandırılmasını amaçlar. Çünkü SVM'in sınıflandırma performansı iyidir ve genelleme kabiliyeti vardır. Özellik uzayında sınıflandırıcının genelleme kabiliyetini en üst düzeye çıkarmak için en uygun hiperplane parametresi belirlenir. SVM tabanlı sınıflandırıcının YSA (Yapay Sinir Ağları) tabanlı sınıflandırıcılardan daha iyi genelleme özelliklerine sahip olduğu iddia edilmektedir. Buna ek olarak SVM tabanlı sınıflandırma doğrudur, çünkü etkinliği, sınıflandırılmış varlıkların özellik sayısına bağlı değildir. Bu özellik arıza tespitinde çok faydalıdır. Çünkü arıza sınıflandırmasının temelini oluşturan özellik sayısı sınırlı değildir. MATLAB/SIMULINK ve AMES'im kullanılarak elde edilen sonuçlar ile bu doğrulanmıştır. SVM tabanlı hata tespit metodu Yapay Sinir Ağlarına göre daha iyi performans göstermiştir. Sonuçlar SVM'nin az sayıda örnek olması durumunda doğru hata sınıflandırmasını başarabileceğini göstermektedir. Bunu aksine PNN (Probabilistic Neural Network) ve RBF (Radial Basis Function) ağlarının eğitim prosedürleri doğru

sınıflandıramaz. Sonuçlar Yapay Sinir Ağlarının (ANN) çok sayıda eğitim örneği gerektirdiğini göstermektedir. SVM temelli sınıflandırıcının ANN sınıflandırıcılarına kıyasla daha iyi performans gösterdiği iddia edilmektedir.

Bo (2010) tarafından yapılan çalışmada LSSVM ile otomobil motor hata tespiti yapılmaktadır. LSSVM lineer olmayan programlama probleminin yerine lineer bir denklem seti kullanan değiştirilmiş bir SVM yöntemidir. SVM makine öğrenmesi yöntemi iyi genelleme yapma özelliğine sahiptir. SVM aşırı öğrenme problemini çözebilecek veri sınıflandırması için güçlü bir araçtır. Çalışmada kullanılan LSSVM hata tespit modeli, normal durum, düşük dereceli kaza sonucu çıkan yangın ve ciddi kaza sonucu çıkan yangın da dahil olmak üzere otomobil motorunun üç durumu vardır. Otomobil motorundaki bu üç durumu tanımak için iki LSSVM modelini kullanılır. Elde edilen sonuçlar kullanılan LSSVM yönteminin otomobil motorunun arıza teşhisinde etkin bir şekilde yorumlanabileceğini göstermektedir. Otomobil motorunun LSSVM tespit modeli eğitim verileri tarafından elde edilir ve tespit sonuçları da test verilerinden elde edilir.

Gang (2016) tarafından yapılan çalışmada gelişmiş yapay sinir ağlarına dayanan otomobil hata tespit sistemi geliştirilmiştir. Motor arıza tipi ve arıza belirtileri analizlerine göre, 11 tip motor arıza bulgusu ve arıza sebepleri çıkarılmıştır. Hata bilgileri normalizasyon sonrası yapay sinir ağı eğitim örneği olarak alınır. Artificial fish swarm algoritmasına dayanan optimize edilmiş BP (Back Propagation) yapay sinir ağı otomobil motor arızası tespitinde kullanılmıştır. BP sinir ağı pek çok endüstriyel probleme uygulanmıştır. BP sinir ağı uygulama sürecinde büyük avantaj gösterir, fakat aynı zamanda hatalar da ortaya çıkmıştır. BP sinir ağı algoritmasının dezavantajları herhangi bir doğrusal ve ya doğrusal olmayan fonksiyon haritalaması için çok tabanlı bir BP sinir ağı bunu anlayabilir. Fakat pratikte, farklı spesifik uygulama alanları nedeniyle, başlangıç parametresinin rastgele olması, BP sinir ağı da eğitim sürecindeki spesifik problemler için çözüm bulamayabilir ve yerel optimuma düşer. İkinci problem olarak, öğrenme hızının seçim standardının belirsiz olmasıdır. Son problem, gizli katman nöron sayısını belirlerken teorik bir standart yoktur. Çalışmada BP sinir ağının eksiklikleri için Artificial fish swarm BP sinir ağını optimize etmek için kullanılır, bu da BP sinir ağının performansını artırır. Eğer gizli katmanda yeteri kadar nöron olmazsa ağı eğitilemez. Eğer gizli katmanda çok fazla nöron

varsa ağ bu sefer test verilerini ezberler. Deney sonuçları gelişmiş BP yapay sinir ağının iyi genelleme kabiliyetine sahip olduğunu ve farklı arıza tiplerini etkili bir şekilde ele alabileceğini göstermektedir. Sonuçlar doğru ve güvenilir olup otomobil arıza tespit sistemi tasarımı için önemli bir referans sağlayabilir.

Youjun vd. (2011) tarafından yapılan çalışmada Fault Tree – Neural Network'üne (FTA-NNE) dayanan bir çeşit arıza tespit uzman sistemi üzerinde çalışılmaktadır. Çalışmada, motor, şasi sistemi ve elektrikli cihazların arıza tespiti için FTA-NNE tabanlı araç içi arıza tespit uzman sistemi öne sürülmüştür. Uzman sistem, fault tree analizine dayalı ve yapay sinir ağına dayalı olmak üzere iki kısma ayrılmıştır. Üretim arızaları, fault tree analizi uzman kurallarını oluşturmak için kullanılır. Arıza tespit uzman sistemlerinin tespit nesnesi, tipik bir arıza motor, şasi sistemi ve elektrik ekipmanını kapsar. Uzman sistemin bilgi tabanı Fault Tree - Neural Network'e dayalı bilgi tabanı (FTAKB) ve Neural Network tabanlı bilgi tabanı (NNEKB) olmak üzere iki kısma ayrılmıştır. FTAKB, Fault Tree analizi (FTA) yöntemiyle oluşturulan tanı kurallarını temsil eder ve bilgi tabanının ana bölümüdür. NNEKB, bilgi tabanının tamamlayıcı bir parçası olarak, Neural Network tabanlı yöntemler tanı modelin, temsil eder. Son olarak Windows CE'deki EVC ++ programlama dili, daha iyi insan-makine etkileşimli olan arabalar için arıza tespit uzman sistemini geliştirmek için kullanılmıştır.

Kong vd. (2011) çalışmasında titreşim (vibration) parametresinin arıza tespit problemini çözmek için motor hata tespitine Adaptive Neural Network tabanlı bulanık çıkarım sisteminin uygulanmasını önermektedir. Gradient Descent Genetik Algoritması ve Neural Network öğrenme algoritması kullanılarak öğrenme hızlandırılır. Adaptive Neural Network tabanlı bulanık çıkarım sisteminin (ANFIS) süreci, örnek veriyi edinme ve inceleme girdi değişkeninin üyelik fonksiyon sayısını belirleme ANFIS'in başlangıç yapısını belirleme ve ANFIS'i eğitmeyi içermektedir. ANFIS arıza tespit modeli girdi arayüzünü optimize etmek için entropi yönteminde bilgi füzyonu yöntemini benimser. Tüm aktivasyon fonksiyonları Gauss türüne aittir. Gradient descent algoritması öğrenmeyi hızlandırmak için parametreleri optimize eder. Simülasyon sonuçlarının karşılaştırmalı bir çalışması, Back Propagation algoritmasında yanlış teşhisde bulunduğunu göstermektedir. Tüm test verilerini doğru bir şekilde sınıflandıramaz. ANFIS'in eğitim süresinde ve eğitim hatasında Back Propagation algoritmasına göre

biraz daha düşük olmasına rağmen ANFIS'in kararlı çıktı sonuçları ve test verilerini tanıma oranının Back Propagation algoritmasından daha iyi olduğu sonucuna varılabilir. Ayrıca hem ANFIS'in hemde belirsizlik tanıma sisteminin çıktı sonuçları istikrarlı olmasına rağmen, resim tanıma oranı değerlerinden daha iyidir. Adaptive Neural Network tabanlı bulanık çıkarım sistemine (ANFIS) dayanılarak, deneyler bu sistemin kararlılık, tanıma oranı ve uyum yeteneği açısından bireysel neural network ve fuzzy modellenen sistemlerden daha üstün olduğunu göstermektedir.

Murphey ve Chen (2010) tarafından yapılan çalışmada araç arıza teşhisi ve bakımı için geliştirilmiş bir multi-agent sistemi (MAS-VFD & HM) önerilmektedir. MAS-VFD & HM sistemi sinyal tespit ajanlarından, özel durum ajanlarından ve araç izleme ajanlarından oluşmaktadır. Bir sinyal ajanı arıza tespitinden veya sinyal arızalarının karmaşıklığına bağlı olarak tek bir sinyal veya birden fazla sinyal kullanarak belirli bir sinyalin izlenmesinden sorumludur. Özel durum ajanları, spesifik bileşen arızalarını saptamak için eğitilmektedir. Bütün bu ajanlar otonomdur ve sonuçlarını araç sistemi ajanına bildirmektedir. Önerilen MAS-VFD & HM sistemi ölçeklenebilir, çok yönlü ve bir araç sisteminde birden fazla arıza gibi karmaşık problemleri işleme yeteneğine sahiptir. Önerilen MAS-VFD & HM sistemi, araçların karmaşık mühendislik tespit problemlerine uygulanabilir. Sinyal ajanları ve özel durum ajanları araç sağlığını incelemek veya arıza tespitini yapmak için eğitilebilmektedir. MAS-VFD & HM sistemi üç seviyede algılama düzeyi sağlar: Sinyal bölümü arızası, sinyal arızası ve bileşen arızası. MA-VFD & HM ölçeklenebilir. Çünkü yeni ajanlar kolayca eğitilebilir, sisteme eklenebilir ve ajanlar bağımsız olarak çalışabilir. Çok yönlüdür çünkü tüm sinyallerin sağlık izlemesi veya arıza tespit bulunması gerektirmez. MAS-VFD & HM sistemi ayrıca birden fazla arıza tespit etme özelliğine sahiptir. Bu nitelikler MAS-VFD & HM sistemini karmaşık mühendislik tespit ve sağlık izleme problemlerinin geniş bir alanını çözmekte güçlü kılmaktadır.

Satoh vd. (2003) tarafından yapılan çalışmada bulanık sistemlerin sinir ağlarının avantajlarından yararlanılarak hızlı ve doğru Sugeno'nun tip-1 bulanık sistemi (Tip – 1 bulanık sistemi) Gauss radyal tabanlı fonksiyon ağı (GP - RBFN) ve zaman gecikmeli sinir ağının (TDNN) kombinasyonu ile uygulanmaktadır. General Parameter Learning (GP) ve adaptasyon kullanarak otomobil şanzımanlarının hata tespiti için RBF ve TDNN

kombinasyonu uygulanmaktadır. Bulanık sistemlerin ve neural networklerin avantajlarını olarak type – 1 fuzzy system Gaussian Radial Basis Function Network (GP - RBFN) ve Time Delayed Neural Network kombinasyonu birlikte uygulanmaktadır. Bu sistem fata tespit modeline uygulanmıştır. Çalışmada, daha hızlı bir bulanık sinir ağının oluşturulması için kullanılan algoritmalarından farklı bir genel parametre (GP) öğrenme ve uyarlama algoritmasını önermektedir. Kullanılan ağ modeli Gauss Radyal Temelli Fonksiyon Sinir Ağı (GP-RBFN) ve Zaman Gecikmeli Sinir Ağlarının birlikte kullanılmasına dayanmaktadır. Geliştirilen GP-TRFN, GPRBFN'den daha hızlı ve daha doğrudur. Bir otomobil transmission gear (şanzıman dişli) sistemi için arıza bulma deneysel olarak uygulanmaktadır. İyi bir algılama performansı elde edilmektedir.

Luckow vd. (2016) çalışmasında özellikle bilgisayar görme alanı içinde derin öğrenme için farklı otomotiv kullanım durumlarını anlatılmaktadır. Derin sinir ağlarını uygulamak, eğitmek ve için kütüphaneler ve altyapılardaki (örneğin, GPU'lar ve bulutlar) mevcut durum incelenmektedir. Özellikle, üretim tesislerinde görsel denetim süreci ve sosyal medya verilerinin analizi gibi CNN (Convolutional Neural Network) ve bilgisayar görme kullanımı durumlarına odaklanmaktadır. Bu makalenin ana katkısı, farklı araç özelliklerini öğrenip otomatik olarak tanımamıza olanak tanıyan bir otomotiv veri setinin oluşturulmasıdır. Veri toplama ve süreç desteği için bir mobil uygulama kullanan derin öğrenme uygulamasını depolama ve eğitim için Amazon tabanlı cloud backend tanımlanır. Eğitim için, farklı sinir ağı mimarileri ve çerçeveleri ile birlikte bulut ve kurum içi altyapıları (birden çok GPU da dahil) değerlendirmektedir. Hem eğitim zamanlarını hem de sınıflandırıcının doğruluğunu değerlendirilmektedir. Son olarak, eğitilmiş sınıflandırıcının gerçek dünya ortamındaki imalat sürecindeki etkisini göstermektedir. Makine öğrenimi ve derin öğrenme, araç, gelişmiş sürüş yardım sistemleri (ADAS), otonom araç sürüşü ve aracın dışındaki otomotiv alanlarında geliştirme, imalat, satış ve satış sonrası süreçlerde birçok potansiyel uygulamaya sahiptir. Bu çalışmada, farklı frameworkler (Tensorflow, Caffè ve Torch) ve ağ mimarileri (AlexNet, GoogLeNet ve Inception) kullanan bulut ve kurum içi altyapı üzerinde uygulanan iki kullanım durumunda derin öğrenmenin kullanımını göstermektedir.

Derin Öğrenme teknikleri, otomotiv endüstrisinde birçok kullanım alanına uygulanabilmektedir. Örneğin, bilgisayarla görme, derin öğrenme sistemlerinin son

zamanlarda çarpıcı biçimde geliştirildiği bir alandır. Araç ve şerit tespiti için konvolusyonel sinir ağları (CNN) kullanılmaktadır. Pomerleau (Yao ve Pan, 2010) sinir ağlarını, bir kamera, bir lazer uzaklık ölçer ve gerçek bir şoförden gelen girişleri gözlemleyerek araç sürmekte otomatik olarak eğitmek için kullanmaktadır. Deep Learning'in çeşitli zorlukları vardır. Büyük miktarda veri, hesaplama ve veri için yeni araçlar ve altyapılar gerektirir. Mevcut model mimarileri ve transfer öğreniminin otomotiv alanındaki bilgisayar görme problemlerini çözmek için uygulanabileceğini göstermektedir. Eğitim sınıflandırıcısının, gerçek dünya kullanımında %85'lik bir doğruluk elde edilmektedir.

Kong vd. (2011) tarafından yapılan çalışmada otomobil motorunun bir arıza teşhis modeli oluşturmak için uyarlamalı sinir ağı tabanlı bulanık çıkarım sistemi (ANFIS) (adaptive neural network-based fuzzy inference system) uygulanmaktadır. Çalışmada ANFIS ile birlikte gradient descent genetic algoritması ve neural network öğrenme algoritmasının sistem parametrelerinin optimizasyonu kullanılmıştır. Gradient descent genetik algoritması, öğrenmeyi hızlandırmak için parametrelerini optimize etmektedir. Tespit modelinin motor test verileri ile doğrulanmasıyla, doğruluğun % 98,99'a ulaştığı, eğitim hatasının 0,02038'e düştüğü saptanmıştır. Deney gradient descent genetic algoritmasının, tek bir kromozomun yerel optimizasyonunu ve bir gruptaki kromozomların küresel optimizasyonunu destekleyebilen hızlı bir algoritma olduğunu göstermektedir. Uyarlamalı sinir ağı (adaptive neural network) tabanlı bulanık çıkarım sisteminin (ANFIS) bulanık tespit süreci, örnek veriyi edinme ve inceleme, girdi değişkeninin üyelik fonksiyonu sayısını ve tipini belirleme, genişli fonksiyonuyla ANFIS'in başlangıç yapısını belirleme ve ANFIS'i eğitmeyi içermektedir. ANFIS arıza teşhis modeli girdi ara yüzünü optimize etmek için entropi yönteminde bilgi füzyonu metodunu benimser. Tüm aktivasyon fonksiyonları Gauss tipine aittir. Simülasyon sonuçları, BP algoritmasının yanlış tespitte bulunduğunu göstermektedir. Tüm test verilerini doğru bir şekilde sınıflandıramamaktadır. Çıktı sonuçları, ortalama doğruluk oranı % 98,99 ile kararlı değildir. ANFIS'in eğitim süresi ve eğitim hatasında BP'ye göre biraz daha düşük olmasına rağmen ANFIS istikrarlı çıktı sonuçlarına sahiptir ve test verilerini tanıma oranı BP'den daha iyidir.

Lili vd. (2010) tarafından yapılan çalışmada araç tanıma elektrikle kontrol edilen sisteminden gelen veri akışının analizi, sensör ve çalıştırıcı arızasının açık şekilde ayrılabilceğini ve tanıma için sinir ağıları ve D-S kanıtlı bilgi birleştirme yöntemini kullandığını göstermektedir. Sonuç, hata tanımada sınıflandırmaya dayalı yöntemin sadece hassaslığı arttırmakla kalmayıp aynı zamanda bir dereceye kadar verimliliği de artırabileceğini göstermektedir. Otomobilin elektrikle kumandalı sisteminin Araç Üzerindeki Teşhisi (OBD), özellikle arıza olduğunda, araç çalışma durumunu kaybeder. Scanner (tarayıcı) sayesinde sensör veya aktüatörün giriş / çıkış değerinin kaydedicisi sürekli olarak okunabilir. Bu veri akışıdır (dataflow). Veri akışı parametrelerinin değeri ve durumu olmak üzere iki türü vardır. Değer parametrelerinin, sensörü veya aktüatörün değerini, basıncını, sıcaklığını, zamanını veya hızını gösteren birimi ve değişen aralığı vardır. Durum parametreleri genellikle açık veya kapalı, yüksek veya düşük, evet veya hayır vb. gibi özellikle de anahtar veya valf için performans durumunu gösterir. Bu nedenle, tespit modu ve tanı için kullanılan veriler, algılama parametresinin özelliklerini bilmelidir. Örneğin, sensör ve aktüatör hatasını ele alındığında, çalışma parametrelerin veri akışının ilişkisini ve tespit hassasiyeti problemini incelemektedir. Veri akışı analizinden, sensör ve aktüatör arızası açıkça ayrılabilir. Neural Network tanıma sonucu bunu doğrulanmaktadır. Bu yöntem akıllı sistemlere uygulanabilir ve OBD(Otomobilin elektrikle kumandalı sisteminin araç üzerindeki teşhisi) veri akışından gelen son arızayı doğrudan tanıyabilmektedir. Deneyler, sınıflandırma tespitinin sadece tanıma hassasiyetini arttırmasının yanı sıra tespitin etkili bir şekilde yapılabileceğini göstermektedir.

Mao vd. (2007) tarafından yapılan çalışmada GP'ye dayalı korelasyon boyutu hesaplama teorisi ele almaktadır. Ancak korelasyon boyutu hesaplamak zordur. Bu çalışmada korelasyon integralinin ikinci türevine dayalı olarak bölge tanıma ve korelasyon boyut hesaplamasını ölçeklendirmek için bir yöntem sunulmuştur. Teorik olarak, korelasyon integralinin ölçekleme bölgesinin ikinci türevi sıfırdır; korelasyon integralinin ölçeklendirme bölgesine karşılık gelen korelasyon integralinin ikinci türevi eğrisinde sürekli sıfır bölgesini aramaktadır. Bu yöntemin geçerliliği Lorenz attractor analizi ile doğrulanmıştır.

Zheng vd. (2004) tarafından yapılan çalışmada ESP sisteminde sensör hata tespitinde Support Vector Machines (SVM) uygulaması ele alınmıştır. ESP sistemindeki algılayıcı tahmin modelleri, destek vektör makineleri (SVM) regresyon algoritması ile oluşturulmaktadır. DAGSVM sınıflandırma algoritması, sensör arıza izolasyonunu karşılar. Araştırmalar ESP sisteminde sensör hata tespitine SVM uygulaması etkili ve uygulanabilir. Otomotiv sisteminin gelişimi, özellikle emniyetli sürüş için artan sayıda sensör, aktüatör ve mikro-elektronik işlemciyi göstermektedir. ESP (Elektronik Stability Programı), emniyetli sürüş için otomotiv kararlılığı sağlayan dinamik bir kontrol sistemidir. ESP sürücünün kritik sürüş koşullarında sürücüye destek vermesi, özellikle de sürücünün bunun için doğrudan tepki vermemesi durumunda kullanılır. ESP sensörleri, otomotiv dinamikleri ve sistem durumu hakkında hayati bilgiler sağlamaktadır. Böylece ESP kontrol cihazı, otomobili istikrarlı bir şekilde tutmak için aktif olarak kontrol edebilir. Başka bir deyişle, bir otomobilin başarılı bir şekilde kontrolü, sensörlerin güvenilirliğine kesinlikle bağlıdır. Bununla birlikte, genel olarak elektronik bileşenler, mekanik bileşenlere göre daha düşük güvenilirliğe ve oldukça farklı arıza davranışlarına sahiptir. Bir arızayı önlemek ve otomotiv güvenliğini ve emniyet performansını geliştirmek için ESP'nin önemli sensörleri için algılama ve tespit gereklidir. SVM, özellikle pattern recognition'dan tahminlee alanlarında daha fazla dikkat çekmiştir. SVM algoritması, iyi genelleme performansı, küresel olarak en optimal çözümü sunar. SVM yapısal riski en aza indirme (SRM) ilkesine dayanan genelleme hatasının (eğitim hatası ve bir güven aralığı) üst sınırını en aza indirmeyi hedeflemektedir. Bu prensip temeli sadece eğitim hatasını en aza indirgeyen geleneksel deneysel risk minimizasyonu (ERM) ilkesinden farklıdır. SVM'lerin bir diğer kilit noktası özelliği, SVM'lerin lineer ikinci dereceden programlama probleminin çözüm bulabilmesidir. Bu makalede SVM'lerin doğrusal olmayan sınıflandırma ve regresyon tahmin algoritmaları sunulmuştur. Çalışmada, ESP sisteminde SVM'lerin üç anahtar sensör arıza teşhisine uygulanması incelenmektedir. Araştırma sonucunda, ESP sistemindeki sensör arıza teşhisine SVM'lerin uygulanmasının etkili ve uygulanabilir olduğu göstermektedir. Bu çalışma, ESP sisteminin emniyetini ve güvenilirliğini artırmaya katkıda bulunur.

Weijie vd. (2014) tarafından yapılan çalışmada Wavelet Neural Network'lerine dayalı otomobil motor hata tespiti yapılmıştır. Bu çalışmada motor titreşim sinyallerinin karakterleri dalgacık paketi teknolojisi kullanılarak çıkarılır. Dalgacık sinir ağları

(Wavelet Neural Network) modeli dalgacık çerçeve (wavelet frame) teorisi ve sinir ağı teknolojisi temel alınarak oluşturulmuştur. Daha sonra wavelet nöronunu seçmek ve optimize etmek için çok çözünürlüklü (multiresolution) analiz kullanılır. Deneysel sonuçlar Wavelet Neural Network'lerine dayanan otomobil motor arıza teşhis modelinin motor arızasını etkili bir şekilde teşhis edebildiğini göstermektedir. Otomobilin gücü, motorun çalışma durumları otomobilin çalışmasını doğrudan etkiler. Otomobil motorundan gelen arızalar çok fazladır ve çok zaman harcamaktadır. Motorun çalışma durumu zamanla doğru bir şekilde değerlendirilebiliyorsa, arıza durumu ayırt edilebilir. Otomobil motorunu parçalara ayırmadan arıza nedeni bulunabilir. Şüphesiz, otomobillerin güvenilirliğini ve güvenliğini artıracaktır. Otomobil motoru çalışırken, titreşim sinyali, itme ve gürültü, otomobil motorunun gerçek zamanlı çalışma durumlarını gösterebilen ve arıza teşhisinin önemli bir bilgi kaynağı olan arızalar hakkında bol miktarda bilgi içerir (Hu vd. 1998, Hang vd. 2002). Çok bileşene sahip pistonlu motor, karmaşık hareketler, belirsiz ve kötü koşullarda çalışma, geniş frekans aralıklı çoklu darbeler içeren hareketler motor yüzeyinin titreşiminde birleştirilir ve aktarılırlar. Gürültü onlara eklendiğinde, motor yüzeyinin titreşim sinyalini daha karmaşık hale getirir. Kapsamlı hata durumlarının özelliklerinin nasıl çıkarılacağı, hatayı başarıyla teşhis etmek için önemlidir. Dalgacık sinir ağları (Wavelet Neural Network), son zamanlarda dalgacık analiz (wavelet analysis) teknolojisinin büyük buluşuna bağlı yeni bir sinir ağı modelidir. Dalgacık analizi ve sinir ağlarının kendi kendine öğrenen zaman frekansının özelliğini kullanır. Motor titreşim sinyalini işlemek ve analiz etmek için dalgacık sinir ağı (Wavelet Neural Network) uygulamak, tespit doğruluğunu artırabilmektedir. Sadece akademik alanda değil aynı zamanda motor arıza teşhisi konusundaki uygulamada da çok önemli bir anlam taşımaktadır. Bu makalede, otomobil motor arıza teşhisine uygulanabilen yeni dalgacık sinir ağları aritmetiği sunulmuştur. Otomobil motor titreşim sinyalinin karakteristik değerleri dalgacık paketleri (wavelet packets) ile çıkarılır. 25 gizli düğüm, 5 giriş düğümü ve 3 çıkış düğümü bulunan dalgacık sinir ağlarının modeli oluşturulmuştur. Otomobil motorundaki valf boşluk arızasının simülasyonu ile dalgacık sinir ağlarına dayanan otomobil motor arıza teşhis modelinin otomobil motor arızasını etkin bir şekilde teşhis edebildiğini göstermektedir.

Kher vd. (2001) tarafından yapılan çalışmada Neural Network'ler kullanarak otomobil motorunun hata tespiti ele alınmıştır. Son yıllarda araç bakım / onarımında bir dizi tespit

sistemi geliştirilmiştir. Bu sistemler araçtaki çeşitli arızaları teşhis etmek için kullanılır. Bu çalışmada sinir ağına dayalı olarak hataların tespiti için bir teşhis sistemi tasarımı yapılmaktadır. Geliştirilen sistemler motor için bir hata tablosuna dayanmaktadır. Böyle bir tespit modülü sistemin kullanımını arttırmayı amaçlar. Otomobil motorundaki arızalar çeşitli nedenlerden dolayı olabilir. Arızalar aralıklarla meydana gelen arızalar veya kalıcı arızalar olarak adlandırılabilir. Otomobil motoru gibi dinamik bir sistem söz konusu olduğunda, araç kullanım sırasında performansı etkileyen araç özellikleri zamanla değişebilir. Sapma (deviation) muhtemel arızaların erken evrelerde göstergesi olabilir ve bu tür arızalar ilk hatalar olarak adlandırılır. İlk hatalar aşamalı olarak artabilir ve sonuç olarak büyük bir arızaya veya kalıcı arızaya neden olabilir. Bir arızanın ortaya çıkması, normal değerlerinden sapma gösteren çeşitli sistem parametreleri sebep olur. Çalışmada çeşitli araç parametreleri ile arızalar arasındaki ilişkiyi belirten tipik bir arıza tablosu verilmiştir. Bir arızanın meydana geldiğinde değerlerinde sapma elde etmek için referans değerlere ihtiyaç duyulmaktadır. Bunlar araç özelliklerinden elde edilir ve araca özgüdür. Çalışmada üç adet sinir ağı kullanılmıştır. Parametre değerinin saptanma miktarı arıza türüne bağlıdır. Arıza bulma yaklaşımı, erken arızanın erken tespitinde yararlı olacaktır.

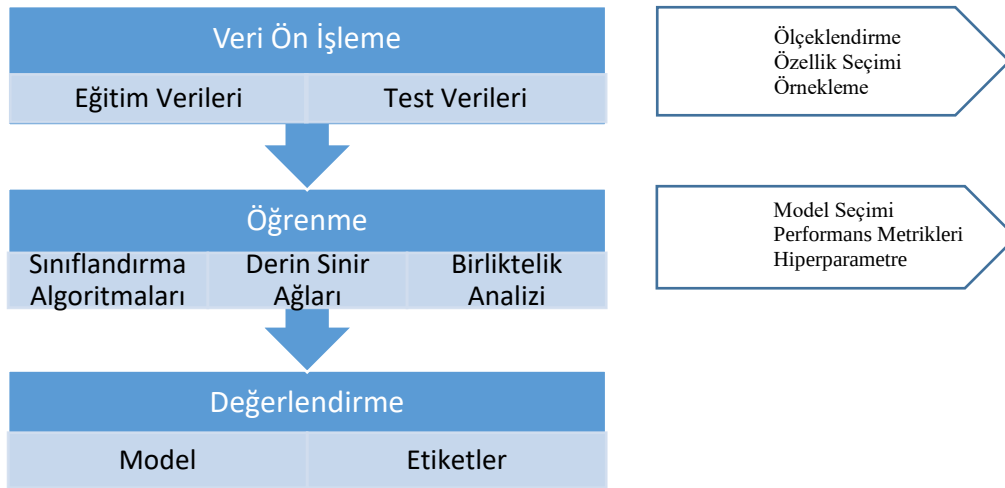
Gao vd. (2010) tarafından yapılan çalışmada Elman Neural Network ile Genetik Algoritmalar kullanılarak motor hata tespiti yapılmaktadır. Hata tespit yöntemleri, motorlu sürüş sistemlerinde emniyetli ve güvenilir bir çalışma elde etmek için çok önemlidir. Olası arızaları bulmak için gelişmiş algılama teknikleri uygulayarak önemli ölçüde bakım büyük maliyetleri olabilmektedir. Bununla birlikte, geleneksel motor arıza bulma yaklaşımları sıklıkla belirgin motor modelleri ile çalışmak zorundadır. Buna ek olarak, çoğu deneyseldir veya uygulanamazlar ve bu nedenle zamanla değişen durumlarda kullanılamazlar. Bu makalede, bu tür zorlukların üstesinden gelmek için bir Elman sinir ağı tabanlı motor arıza bulma modeli önerilmektedir. Motor arızaları, sinyal hatasının beklenen değişikliklerden saptanır. Elman sinir ağı için Genetik Algoritma (GA) destekli eğitim stratejisi, doğruluğu artırmak ve daha iyi tespit performansı elde etmek için ileri sürülmüştür.

AC ve DC motorlar, çeşitli endüstriyel uygulamalarda yoğun bir şekilde uygulanmaktadır. Geliştirilmiş emniyet ve güvenlik ile bu arızalar için uygun bir erken teşhis stratejileri ile başarılabilir. Ayrıca, gizli arızaları bulmak için etkili tespit

yöntemleri uygulayarak büyük bakım maliyetinden tasarruf edilir. Motor sürücü gözlemlene, arıza tespiti ve teşhisi, bu nedenle çok önemlidir, ancak endüstriyel mühendislik alanının zorlu konularındandır. Bazı klasik motor arıza teşhis ve tespit yöntemleri mevcuttur. Geçmiş yıllarda, klasik motor arıza tespit stratejilerinin karşılaştığı zorluklardan kurtulmak için yumuşak hesaplama (soft computing) kullanılmaktadır. Aslında, sinir ağları, bulanık mantık ve genetik algoritmalar (GA) içeren yumuşak hesaplama (soft computing), insan zihninin belirsiz ve kesin olmayan koşullarda akıl yürütme ve öğrenme gibi olağanüstü kabiliyetine paralel akıllı bilgi işleme için yeni bir yaklaşım olarak düşünülmektedir. Bu yöntem gerçek dünya sorunlarına karşı kesin olmayan veya optimal olmayan fakat ekonomik ve rekabetçi çözümler edinmede etkilidir. Bazı genel amaçlı motor arıza teşhis çözümleri, zor doğrusal olmayan tanı problemlerini etkili bir şekilde ele alabilen sinir ağları teknikleri kullanılarak incelenmiştir. Bu çalışmada, motor arıza bulma sisteminde sinir ağlarının (Elman Neural Network) ve genetik algoritmaları birlikte kullanılmıştır. Bu melez öğrenme algoritması Elman sinir ağı optimizasyonu için büyük avantaja sahiptir. Dolayısıyla, eğitim sırasında yerel minimuma sıkışma olasılığı azaltılır. Gerçek dünya verileriyle deneylerde, bir otomobil vites dişli sisteminde verimliliğini doğrulanmaktadır.

3. MATERYAL VE YÖNTEMLER

Bu bölümde, tez çalışmasında kullanılan materyal ve yöntemlerden bahsedilmiştir. İlk aşamada kullanılan farklı materyaller, ikinci aşamada ise tahmin ve birliktelik analizi için uygulanan yöntemler açıklanmıştır. Yöntemler bölümü veri ön işleme aşamasını, sınıflandırma modeli ve birliktelik analizi için kullanılan algoritmaları ve son olarak modellerin değerlendirilmesinde kullanılan ölçümleri içermektedir. Bu tezin amacı belirli periyotlarla muayeneye giren aracın muayene bilgilerinden yola çıkarak araç muayene sonucunu tahmin etmektir. Makine öğrenmesi ve derin sinir ağı modelleri kullanılarak aracın kilometresi, modeli, yakıt tipi vb. özellikleri ve muayene sonucunda çıkan kusurları göz önüne alınarak araç muayene sonucunun tahmini ile muayene işlem sürecine yardımcı olmak amaçlanmıştır. Makine öğrenme algoritmaları olarak LR (Lojistik Regresyon), KNN (K-Nearest Neighbors), NB (Naive Bayes), DT (Decision Tree), RF (Random Forest), XG Boost, Ada Boost, DNN (Deep Neural Network) ve CNN (Convolutional Neural Network) kullanılmıştır. Ayrıca araçların muayene sonucuna göre görülen kusurların analizini gerçekleştirmek için birliktelik analizi yöntemlerinden Apriori algoritması kullanılmıştır. Araç kusurları üzerinde yapılan birliktelik analizi ile araçlarda çıkabilecek hataların erken tespitinde yararlı olacaktır. Şekil 3.1’de tezde kullanılan yöntemlerin aşamaları verilmiştir.



Şekil 3.1 Sınıflandırma modeli için iş akışı diyagramı

3.1 Materyal

Kullanılan sistem, Windows 10 Enterprise, işlemci AMD Ryzen 7 2.20 GHz, RAM 16 GB (gigabayt) ve sabit disk 500 GB'tır. Bu bölümde kullanılan farklı malzemeler tanımlanmaktadır. Programlama dili olarak Anaconda altında Python versiyon 3.6.5 kullanılmıştır. Derin öğrenme algoritmaları için Tensorflow kütüphanesi kullanılmıştır.

Python: Guido van Rossum tarafından 1990 yılında geliştirilen nesne yönelimli, yüksek seviyeli bir programlama dilidir. Python hemen hemen her türlü platformda çalışabilmektedir (Windows, Linux, Unix, Mac, Symbian). Veri analizi, makine öğrenimi, derin öğrenme ve daha bir çok alan üzerinde çalışmamıza imkan veren birçok kütüphanesi mevcuttur.

Scikit – learn: Makine öğrenmesi ve veri bilimi için kullanılan en yaygın Python kütüphanelerinden biridir. Scikit-learn ayrıca veri işleme, model seçimi, regresyon, sınıflandırma ve kümeleme analizi algoritmalarını içermektedir.

Anaconda: Veri bilimi ve benzeri uygulamalar için Python'da geliştirme yapmak isteyenlere hazırlanmış tümleşik bir Python dağıtım uygulamasıdır. Anaconda veri bilimi derin öğrenme gibi konularda kullanılan kütüphaneleri barındırır. Farklı kütüphaneler eklenebilir ve güncellenebilir. Ayrıca Jupiter Notebook, Spyder gibi çeşitli araçları içermektedir. Bu tez çalışmasında Jupiter ve Spyder araçları kullanılmıştır (Anonymous 2021c).

Spyder: Python dilinde geliştirme yapmak için kullanılan açık kaynak kodlu bir IDE'dir (Integrated Development Environment). Spyder, NumPy IPython, SciPy ve matplotlib gibi Python kütüphanelerinin desteği ile sayısal bir hesaplama ortamıdır.

Jupyter Notebook: Herhangi bir web tarayıcısı üzerinde çalışabilen metin belgesi formatında kodları oluşturmayı, düzenlemeyi ve çalıştırmayı sağlayan bir web uygulamasıdır (Anonymous 2021d).

Tensorflow: Google tarafından geliştirilen açık kaynak kodlu derin öğrenme alanında en çok kullanılan kütüphanelerden birisidir. Tensorflow kütüphanesi ile derin öğrenme tabanlı yapay zeka uygulamaları geliştirilebilir (Anonymous 2021e)

3.2 Yöntemler

Bu tez çalışması, Ulaştırma ve Altyapı Bakanlığı araç muayene verileri ile veri madenciliği yöntemleri kullanılarak tahminleme ve birliktelik analizi yapılmıştır. Araç muayene verileri, Oracle veri tabanında ilişkisel veri tabanı kurallarına uygun olarak tutulmaktadır. Veriler text ve excel dosyası formatında elde edilmiş olup daha sonra üzerinde çalışmak üzere SQL Server veri tabanına aktarılmıştır.

3.2.1 Araç muayene verileri ve açıklaması

Bu çalışmada kullanılan veriler 2015, 2016 ve 2017 yılları Ocak, Şubat ve Mart aylarına ait araçların muayene bilgilerini içermektedir. Toplam 7.392.283 veri kullanılmıştır. Tahminleme modelinde veride bulunan ve kategorik bir alan olan MUAYENE_SONUC_KODU (AK, HK, EM) alanı kullanılmıştır.

Ağır Kusurlu (AK): Muayeneye giren aracın ağır kusurlu çıktığını, muayeneyi geçemediğini ve düzeltilmesi gereken kusurları olduğunu göstermektedir. Aracın trafiğe çıkması yasaklanır.

Hafif Kusurlu (HK): Muayeneye giren aracın hafif kusurlu çıktığını, muayeneyi geçtiğini ve aracın bazı bölgelerinde problem oluşturmayacak şekilde kusuru olduğu göstermektedir. Aracın trafiğe çıkabileceği anlamına gelir.

Emniyetsiz (EM) : Muayeneye giren aracın sonucunun emniyetsiz çıktığını ifade eder. Trafik içerisinde halk sağlığını ve güvenliğini tehdit eden bir yapıda olduğu ve kusurlarının giderilmesi gerektiği anlamına gelmektedir. Muayene ağır kusur sonucundan daha fazla sorun bulması halinde emniyetsiz sonucu çıkmaktadır.

Bu değişken için “HK” olan veri 0, “AK” olan veri 1 ve “EM” olan veri 2 olarak kodlanmıştır. Aynı düzenlemeler Label Encoder kullanılarak kategorik olan tüm alanlar için uygulanmıştır. Veri seti ile ilgili detaylı bilgi çizelge 3.1’de gösterilmiştir.

Çizelge 3.1 Araç muayene verileri açıklaması

Öznitelikler	Veri Tipi	Açıklama
MUAYENE_ID	Nümerik	Aracın muayene id'si
CINSI	Kategorik	Aracın Cinsi
MODEL	Nümerik	Aracın Model Yılı
YAKIT_TIPI	Kategorik	Aracın Yakıt Türü
KM (Kilometre)	Nümerik	Kilometre
KULLANIM_AMACI	Kategorik	Aracın Kullanım Amacı
TESCİL_SEKLI	Kategorik	Aracın Tescil Şekli
DAHİLİ_ARAC_ID	Nümerik	Aracın id'si
MUAYENE_SONUC_KODU	Kategorik	Aracın Muayene Sonucu
KUSUR	Kategorik	Muayene sonucunda çıkan kusurlar

Çalışmada kullanılan veriler, araç muayene bilgilerinin olduğu ve çizelge 3.2’de gösterilen SQL Server veri tabanına aktarılan ARAC_MUAYENE tablosu ve muayene sonucunda çıkan kusur bilgilerinin tutulduğu ve çizelge 3.3’de gösterilen KUSUR tablosu olmak üzere iki ayrı tabloda tutulmaktadır. Excel elektronik tablolarında bulunan veriler SQL Server veri tabanına aktarılmıştır. ARAC_MUAYENE ve KUSUR tablosu MUAYENE_ID foreign key alanı ile ilişkilendirilmiştir. KUSUR tablosu ve çizelge 3.4’te verilen Kusur Açıklama tablosu UBAK_KODU alanı ile ilişkilendirilmiştir.

Çizelge 3.2 ARAC_MUAYENE tablosu

ID	MUAYENE_ID	CINSI	MODEL	YAKIT_TIPI	KM	KULLANIM_AMACI	TESCİL_SEKLI	DAHİLİ_ARAC_ID	MUAYENE_TUR	MUAYENE_TARİH	MUAYENE_SON_RAKIT_TARİH	MUAYENE_SONUC_KODU	MUAYENE_HİZMET_TIPI
55372861	56072812	TIC2	1992	DİZ	247746	YUK	TIC	1769410533	GM	2015-03-05	1950-01-01	AK	GM
55372864	56072815	HTA1	1996	BEN	35765	YOL	HUS	1778034465	GM	2015-03-05	1950-01-01	EM	GM
55372865	56072816	HTA1	1993	BEN	222245	YOL	HUS	1778034406	GM	2015-03-05	2017-03-06	HK	GM
55372866	56072817	HTA1	1995	B+LPG	157489	YOL	HUS	38017409	GM	2015-03-05	2017-03-06	HK	GM-MT

Çizelge 3.3 KUSUR tablosu

ID	MUAYENE_ID	UBAK_KODU
344652336	56249484	5.12.4
344652337	56249484	5.12.3
344652338	56249484	6.7.8
344652339	56249484	5.16.1
344652374	56249489	5.3.2

Çizelge 3.4 Kusur Açıklama tablosu örnek veri

KODU	ACIKLAMA	KUSUR_GRUBU
1	Aracın Tespiti	
1.1	Şasi No./ Fabrika Tip Etiketi/Diğer Etiketler/Araç Ruhsatı	
1.1.4	Araç şasi No: Şasi gövdesi uygunsuz değiştirilmiş	Emniyetsiz
1.1.6	Motor No: bulunamıyor.	Hafif Kusur
1.1.7	Motor No: orijinal değil	Hafif Kusur
1.1.8	Motor No: yok	Hafif Kusur
1.1.9	Motor No: yanlış	Hafif Kusur
1.1.11	Fabrika tip etiketi: yok	Hafif Kusur
1.1.12	Fabrika tip etiketi: okunamıyor	Hafif Kusur
1.1.13	Araç Ruhsatı: okunamıyor	Ağır Kusur

Araç muayene verileri, makine öğrenmesi algoritmasında kullanılmak üzere SQL Server veri tabanında SQL (Structured Query Language) sorguları ile birleştirilip anlamlı hale getirilmiştir. KUSUR ve ARAC_MUAYENE olarak iki ayrı tabloda ve yıllarına göre tutulan veriler EK-1’de verilen sql sorgusu ile birleştirilip toplam 7.392.283 veri elde edilmiştir.

Kusur Açıklama tablosunda 1694 tane kusur türü olduğundan ve bu kusurların hepsini makine öğrenmesi algoritmaları ile analiz etmek hem güç hem de yeterli bilgisayar donanımına sahip olunmaması sebebi ile veri toplam 15 kusur grubu altında incelenmiştir. Örneğin veri setinde bulunan “1” alanı için; çizelge 3.5’de verilen “1” alan adlı “Aracın Tespiti” kusur grubunda altında bulunun 47 kusurdan muayene sonucunda bunlardan herhangi birinde kusuru bulunan araç için “1” değeri, bu grupta kusuru bulunmayan araç için “0” değeri verilmiştir. Aynı düzenlemeler 15 adet kusur grubu için uygulanmıştır. Çizelge 3.5’de 15 adet kusur grubunun açıklamaları verilmiştir.

Araç muayene veri setinde bulunan aracın kilometresi, modeli, yakıt tipi vb. özellikleri ve muayene sonucunda çıkan kusurları gruplanıp toplamda 15 grup altında toplanarak elde edilen kusurlar baz alınarak muayene sonucunu tespit etmek için sınıflandırmaya dayalı bir yöntem geliştirilmiştir. Kusur verilerinin detayına bakılmadan üst kusur

grupları kullanılmıştır. Bu çalışmada kusurlar arasındaki ilişkileri çıkarabilmek için kusur tahminleme de yapılabilir.

Çizelge 3.5 Kusur grupları

ALAN ADI	AÇIKLAMA
1	Aracın Tespiti ile ilgili kusurlar
2	Fren Sistemi ile ilgili kusurlar
3	Direksiyon Sistemi ile ilgili kusurlar
4	Görüş Alanı ile ilgili kusurlar
5	Aydınlatma ile ilgili kusurlar
6	Dingiller/Tekerlekler/Lastikler/Süspansiyon ile ilgili kusurlar
7	Şase/karoseri - bağlı parçalar ile ilgili kusurlar
8	Römork çeki sistemi ile ilgili kusurlar
9	Yükleme kasası ile ilgili kusurlar
10	Üst yapı ile ilgili kusurlar
11	Motor/Tahrik Debriyaj/Vites değiştirme ile ilgili kusurlar
12	Motosiklet, Motorlu Bisiklet ile ilgili kusurlar
13	Diğer Donanımlar ile ilgili kusurlar
14	Çevre Kirliliği: Gürültü oluşumu/ Parazit önleme/Yağ, Hidrolik, Sıvı kaçaqları ile ilgili kusurlar
15	İlave kontroller ile ilgili kusurlar

Çizelge 3.6’da makine öğrenmesi algoritmalarında kullanılan araç muayene verilerinin örnek 5 satırı verilmiştir.

Çizelge 3.6 Araç muayene örnek veri

CINSI	MOD EL	YAKIT _TIPI	KM	KULLAN IM _AMACI	TESCİL _SEKLI	MUAYENE _SONUC _KODU	1 ...	1 1 1 3 4 5
HTA1	2002	BEN	1332 02	YOL	HUS	HK	1 ...	0 0 1
HTA1	1985	BEN	5481 2	YOL	HUS	AK	0 ...	0 1 1
HTA3	2005	DIZ	2419 97	YUK	HUS	HK	1 ...	0 0 1
HTA3	2013	DIZ	1677 0	YUK	HUS	HK	0 ...	0 0 0
HTA1	2006	DIZ	1714 61	YOL	HUS	AK	1	0 1 1

Araç muayene verilerindeki alanlar ve veri tipleri çizelge 3.7’de verilmiştir.

Çizelge 3.7 Araç muayene verisi alanları ve veri tipleri

Data	Kolon Adı	Veri Tipi
0	MUAYENE_ID	int64
1	CINSI	object
2	MODEL	int64
3	YAKIT_TIPI	object
4	KM	int64
5	KULLANIM_AMACI	object
6	TESCIL_SEKLI	object
7	DAHILI_ARAC_ID	int64
8	MUAYENE_SONUC_KODU	object
9	1	int64
10	2	int64
11	3	int64
12	4	int64
13	5	int64
14	6	int64
15	7	int64
16	8	int64
17	9	int64
18	10	int64
19	11	int64
20	12	int64
21	13	int64
22	14	int64
23	15	int64

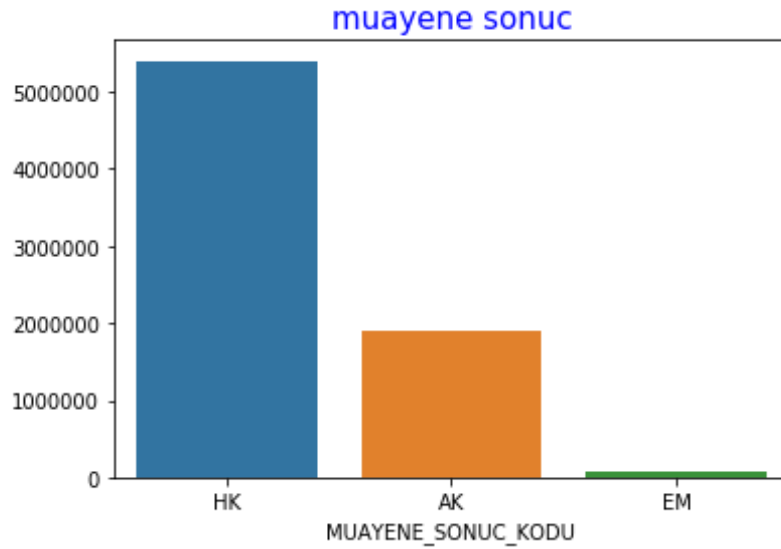
Çizelge 3.8’de araç muayene veri setinde nümerik olan alanların ortalaması (mean), maksimum, minimum, standart sapma (std) gibi istatistiksel değerleri verilmiştir.

Çizelge 3.8 Araç muayene verilerinin istatistiksel açıklaması

	MODEL	KM	1	2	...	14	15
count	7.392283e+06	7.392283e+06	7.392283e+06	7.392283e+06	...	7.392283e+06	7.392283e+06
mean	2.004239e+03	1.569095e+05	3.743728e-01	1.889976e-01	...	4.733333e-01	5.349533e-01
std	8.162943e+00	1.644958e+05	4.839606e-01	3.915068e-01	...	4.992884e-01	4.987768e-01
min	1.900000e+03	0.000000e+00	0.000000e+00	0.000000e+00	...	0.000000e+00	0.000000e+00
25%	1.900000e+03	5.929100e+04	0.000000e+00	0.000000e+00	...	0.000000e+00	0.000000e+00
50%	2.006000e+03	1.232080e+05	0.000000e+00	0.000000e+00	...	0.000000e+00	1.000000e+00
75%	2.011000e+03	2.043180e+05	1.000000e+00	0.000000e+00	...	1.000000e+00	1.000000e+00
max	2.018000e+03	8.352402e+07	1.000000e+00	1.000000e+00	...	1.000000e+00	1.000000e+00

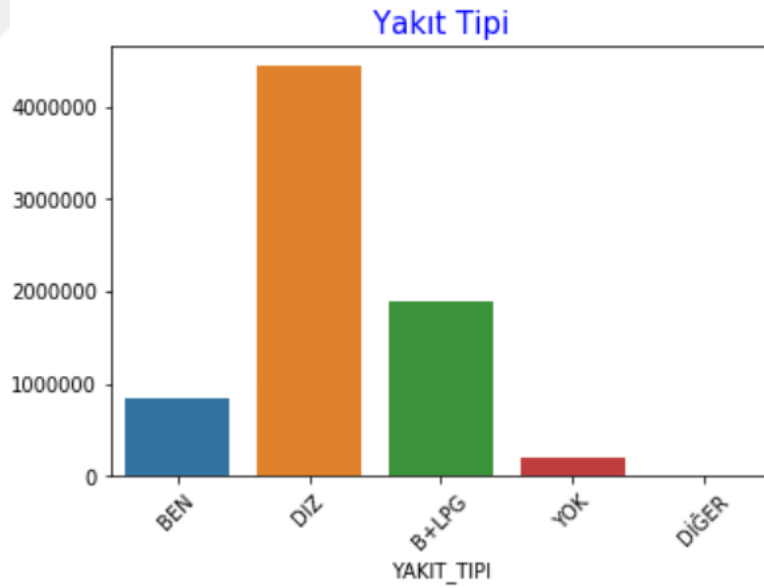
3.2.2 Veri görselleştirme

Araç muayene verileri Python’da seaborn ve matplotlib kütüphaneleri kullanılarak görselleştirilmiştir. Şekil 3.2’de verilen grafik, verilerin muayene sonucuna göre dağılımını göstermektedir. Araç muayene veri setinde en çok Hafif Kusurlu sınıfına mensup araçlar bulunmaktadır. Buradan veri setinin araç muayene sonucuna göre dengesiz dağıldığı ortaya çıkmaktadır.



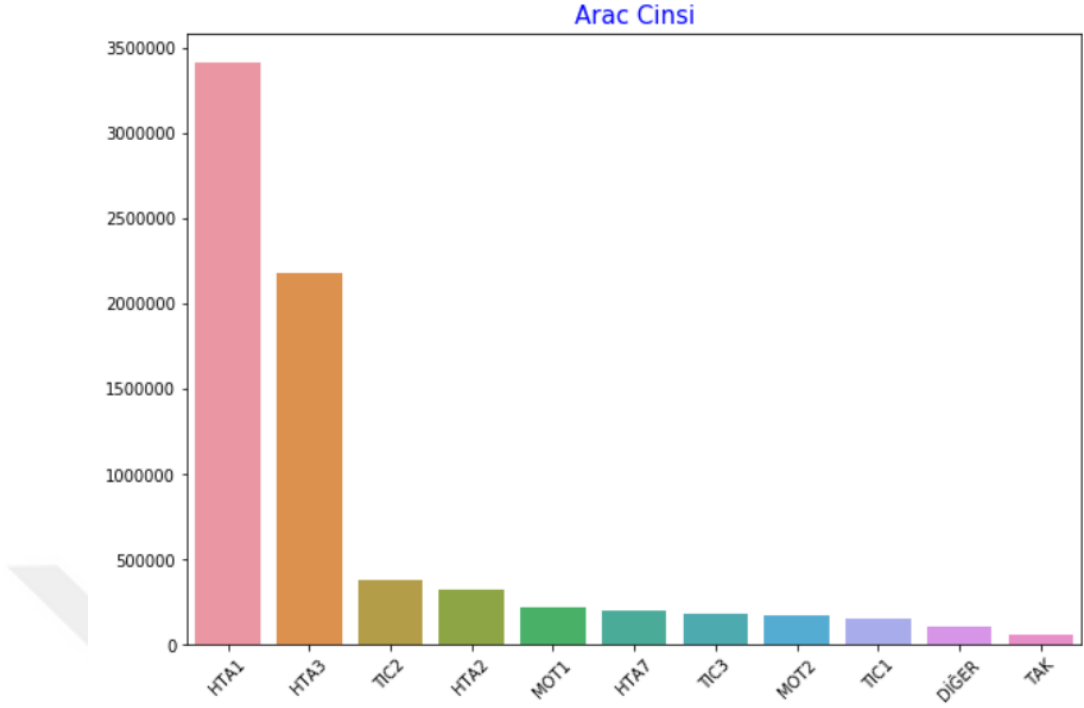
Şekil 3.2 Araç muayene verilerinin istatistiksel açıklaması

Şekil 3.3’de verilen grafik verilerin yakıt tipine göre dağılımını göstermektedir. Kullanılan araç muayene verilerinde en çok Dizel yakıt tipine sahip araçlar bulunmaktadır. Daha sonra Benzin ve Benzin + LPG yakıt türüne sahip araçlar bulunmaktadır.



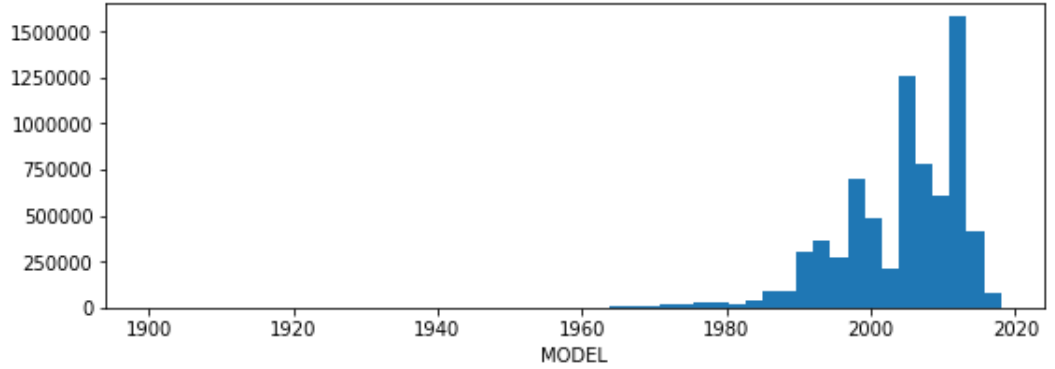
Şekil 3.3 Verilerin yakıt tipine göre dağılımı

Şekil 3.4’de verilen görselde ise verilerin araç cinsine göre dağılımı gösterilmektedir. Veri setinde en çok “HTA1” kodlu Otomobil türü araçların olduğu görülmektedir. Daha sonra “HTA3” kodlu Kamyonet cinsi araçlar vardır.



Şekil 3.4 Verilerin araç cinsine göre dağılımı

Şekil 3.5’de verilen grafikte verinin MODEL alanına göre dağılımı histogram türü grafik ile gösterilmiştir. Araçların en çok 2000 ve 2020 yılları arası modellerde dağılım gösterdiği tespit edilmektedir.



Şekil 3.5 Verilerin araç model yılına göre dağılımı

3.2.3 Veri ön işleme

Veri ön işleme, makine öğrenmesi modellerini uygulamadan önce veri kümesi üzerinde yapılan düzenleme, eksik verilerin tamamlanması, dönüştürme, birleştirme, temizleme, boyut indirgeme, normalleştirme gibi işlemlerdir. Veri ön işleme aşaması zaman alan bir süreçtir. Makine öğrenmesi sürecinin en önemli aşamalarından birisidir. Kurulan makine

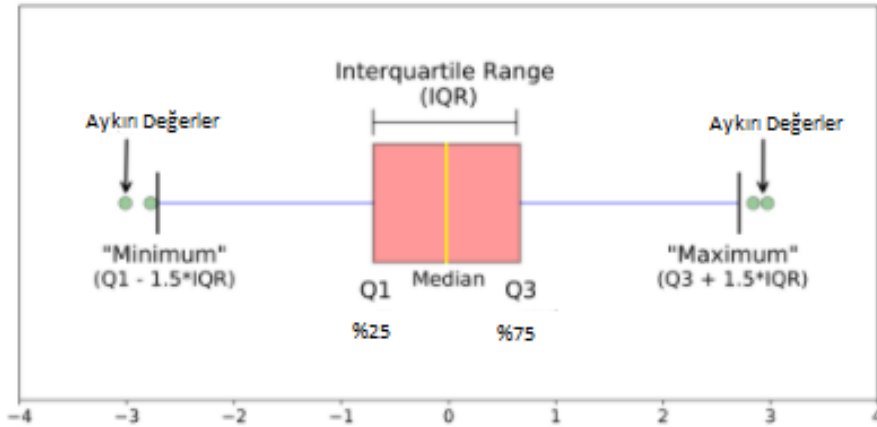
öğrenmesi modellerinden daha iyi sonuç elde edebilmek için verilerin ön işlemden geçmesi gerekmektedir.

Araç muayene verileri veri madenciliği yöntemleri ile analiz edilmeden önce gereksiz olan “DAHILI_ARAC_ID, MUAYENE_ID” gibi birincil ve yabancı anahtar alanları veriden çıkarılmıştır. Verideki “MUAYENE_SONUC_KODU” alanına göre tahminleme yapılacağı için bu alan da veri setinden kaldırılmıştır. Veri seti, veri madenciliği algoritmaları ile analizinin yapılacağı gerekli alanların olduğu ve “MUAYENE_SONUC_KODU” alanın olmadığı x_data ve sınıflandırma algoritmalarının sonuçlarının karşılaştırılabileceği ve sadece “MUAYENE_SONUC_KODU” alanının olduğu y_data şeklinde ayrılmıştır.

3.2.3.1 Aykırı değerler (Outlier)

Aykırı değer, verilerin geri kalanından aralık dışı olan değerdir. Aykırı değerleri tespit etmek için çok çeşitli yöntemler vardır. Bu çalışmada Araç muayene veri setinde aykırı değerleri tespit etmek için Kutu grafiği ve IQR (Inter Quantile Range) yöntemi kullanılmıştır.

Inter Quartile Range (IQR) yöntemi kullanarak da aykırı değerler tespit edilebilir. IQR yöntemi veri setindeki değişimleri bulur. Bu yöntemle göre $-1,5 \times \text{IQR}$ ile $1,5 \times \text{IQR}$ aralığının dışındaki herhangi bir değer, aykırı değerler olarak değerlendirilir (Anonymous 2021b).



Şekil 3.6 IQR metodu

Şekil 3.12’de IQR yönteminin görseli verilmiştir.

Q1, verilerin 1. çeyrek / 25. yüzdelik dilimini temsil eder.

Q2, verilerin 2. çeyrek / medyan / 50. yüzdelik dilimini temsil eder.

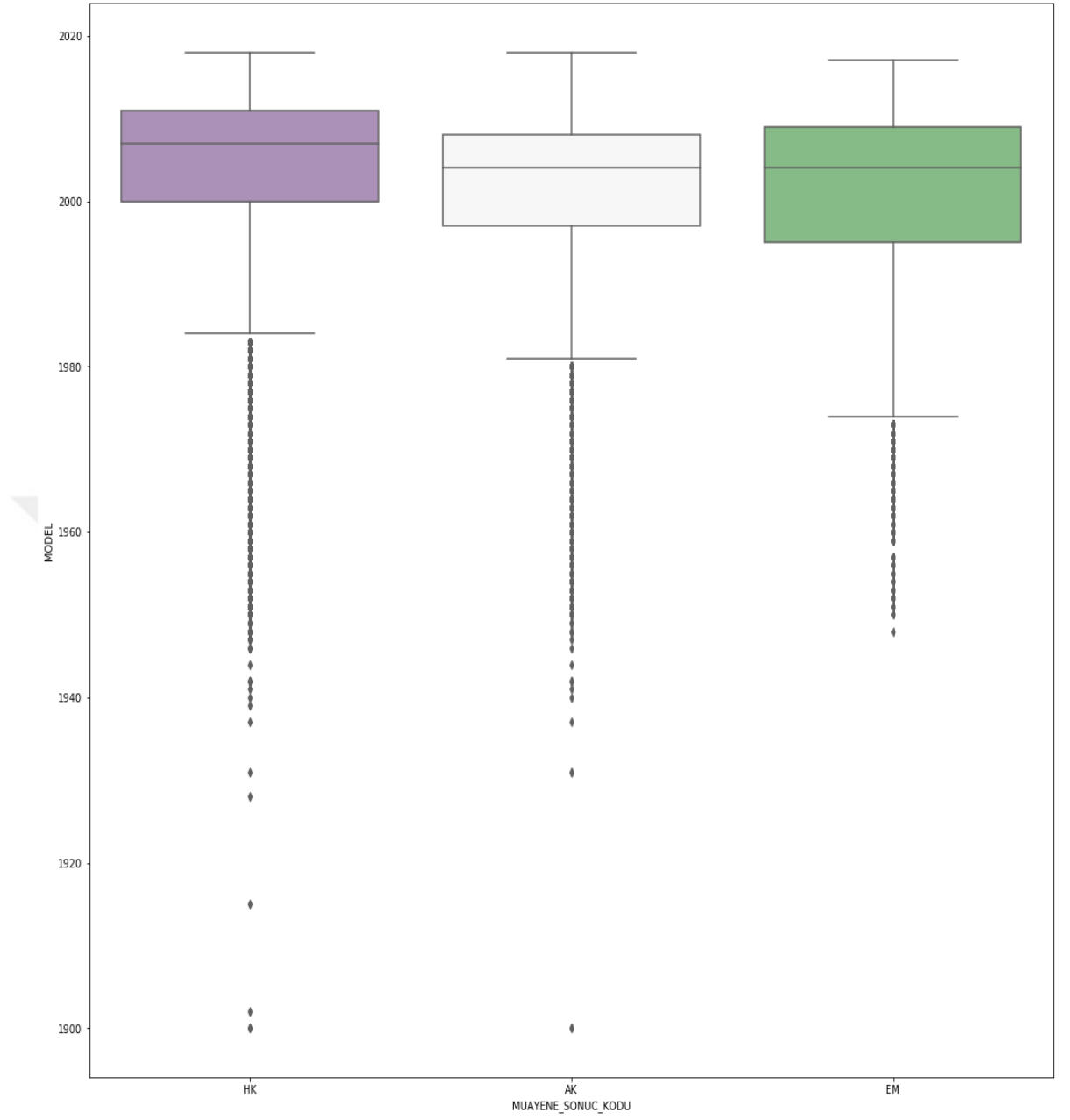
Q3, verilerin 3. çeyrek / 75. yüzdelik dilimini temsil eder.

$(Q1 - 1.5 * IQR)$ veri setindeki en küçük değeri temsil eder ve $(Q3 + 1.5 * IQR)$ veri setindeki en büyük değeri temsil eder (Şekil 3.6).

Aykırı değerleri tespit etmek için veri görselleştirme grafiklerinden de faydalanılır. Bu çalışmada KM ve MODEL alanlarında bulunan aykırı değerleri tespit etmek için Kutu grafiği (Box Plot) kullanılmıştır.

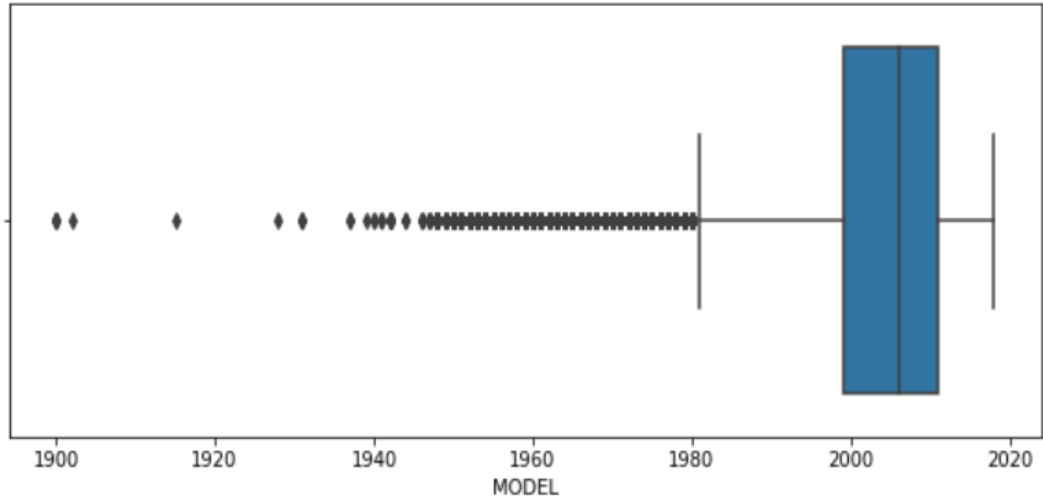
Kutu Grafiği (Box Plot): Kutu grafiği, verilerdeki değerlerin nasıl bir yayılım gösterdiğini verir. Grafikteki kutu ne kadar uzun olursa veriler o kadar dağılmış olur, kutu ne kadar küçük olursa veriler de o kadar az dağılmış olmaktadır. Bir kutu grafiğindeki aykırı değerler, kutu grafiğinin bıyıklarının dışında kalan veri kümesi olarak tanımlanır.

Şekil 3.7’de ki grafikte araç modeli ve araç muayene sonucuna göre box plot grafiğini gösterilmektedir. Grafik, araç modellerinin muayene sonucuna göre nasıl yayıldığını gösterir. Her muayene sonucu için araç model yılı median değeri hemen hemen aynıdır. Grafiğe göre hafif kusurlu araçlarda aykırı değerlerin daha fazla olduğu ortaya çıkmıştır. Kutucukların dışındaki noktalar aykırı değerleri göstermektedir. Örneğin şekil 3.7’ye göre muayene sonucu hafif kusurlu ve ağır kusurlu çıkan ve yaklaşık 1900 modelinde olan 3 araç tespit edilmiştir ve tüm aykırı değerler veri setinden çıkarılmıştır.



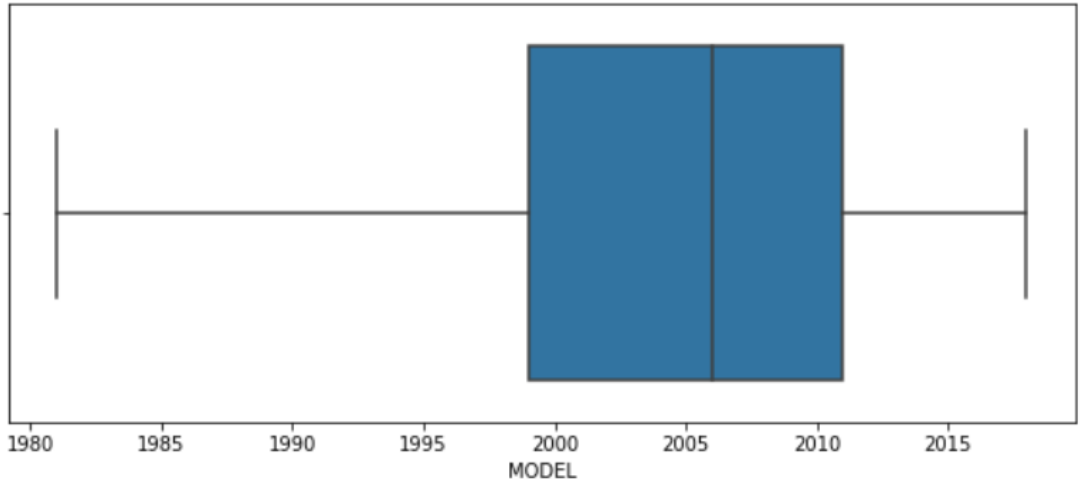
Şekil 3.7 Araç modeli ve muayene sonucuna göre kutu grafiği

Şekil 3.8’de araç model alanı için IQR metodu uygulamadan önceki aykırı değerleri gösteren kutu grafiği verilmiştir.



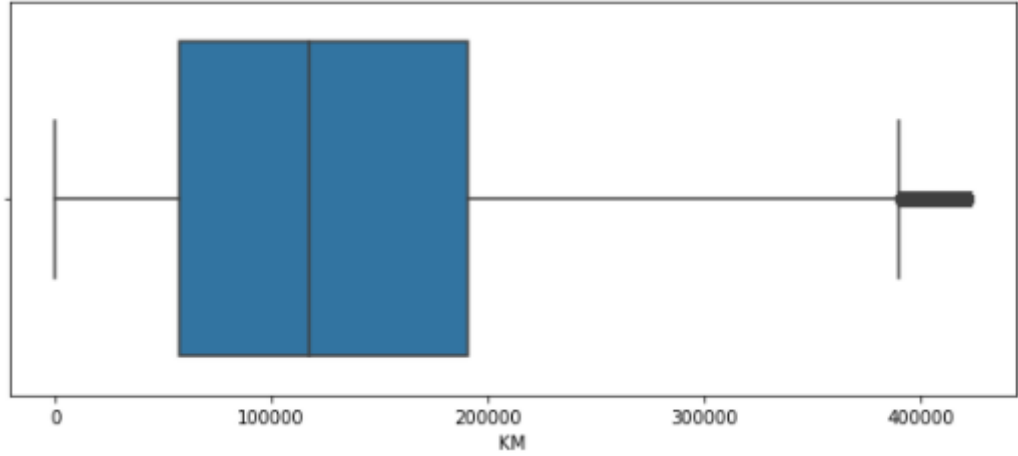
Şekil 3.8 Model alanının kutu grafiği

Şekil 3.9 Model alanı için IQR metodu uyguladıktan sonra kutu grafiğini gösterir. Model alanı için 7 milyon veriden yaklaşık 100 bin aykırı değer veri setinden çıkarılmıştır.



Şekil 3.9 Aykırı değerler silindikten sonra model alanının kutu grafiği

Şekil 3.10 KM alanına IQR metodunu uyguladıktan sonra kutu grafiğini gösterir. KM alanı için 7 milyon veriden yaklaşık 300 bin aykırı değer veriden çıkarılmıştır.



Şekil 3.10 Aykırı değerler silindikten sonra KM alanının kutu grafiği

3.2.3.2 Veri ölçeklendirme

Veri ön işleme aşamasında araç muayene verilerine Python'da en çok kullanılan iki yöntem uygulanmıştır: MinMaxScaler ve Standard Scaler.

MinMaxScaler: Verideki değişkenleri normalleştirmenin bir yolu Min-Max Scaler'dır. Bunu yaparak, tüm öznitelikler $[0,1]$ aralığına dönüştürülür, yani bir değişkenin minimum ve maksimum değeri sırasıyla 0 ve 1 olacaktır.

Standard Scaler: Bu ölçekleme yöntemi için verilerin normal dağılıma uygun dağıldığı kabul edilir. Verilerin ortalama değeri 0 olacak ve standart sapma 1 olacak şekilde yeniden veriler ölçeklenir.

3.2.3.3 Dönüştürme

Scikit-learn kütüphanesinin preprocessing alt kütüphanesinin altında, veri ön hazırlaması aşamasında kullanılan iki encoder (kodlayıcı) vardır. Label Encoder ve OneHot Encoder, kategorik değişkenlerin tahminlerde kullanılabilmesi için makine öğrenmesi algoritmalarına uygun bir forma dönüştürüldüğü bir süreçte kullanılmaktadır. Bu iki yöntem araç muayene verilerinde kategorik olan CINSI, YAKIT_TIPI, KULLANIM_AMACI, TESCIL_SEKLI, MUAYENE_SONUC_KODU alanları için nümerik değerlere dönüştürme işlemi için kullanılmıştır.

Label Encoder: Veriyi birebir sayısallaştırmaya yarar. Yani kategorik her veriye sayısal bir değer atar.

OneHotEncoder: Yine aynı kütüphanede bulunan kodlayıcı bu kategorik verilerin binarizasyonunu gerçekleştirmemizi sağlar. Mevcut değere “1” verip mevcut olmayanlara “0” vermektedir.

3.2.3.4 Özellik seçimi (Feature selection)

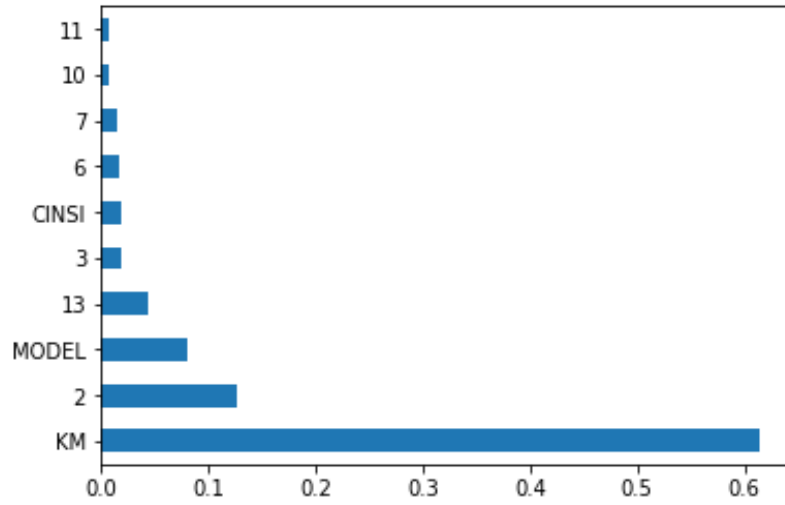
Özellik Seçimi, makine öğrenmesi modelinin performansını önemli ölçüde etkileyen temel kavramlardan birisidir. Makine öğrenimi modelleri eğitmek için kullanılan veri özelliklerinin, elde edebilen performans üzerinde büyük bir önemi vardır. İlgili veya kısmen ilgili özellikler, model performansını olumsuz yönde etkileyebilir. Özellik seçimi ve veri temizleme işlemi, model tasarımının ilk ve en önemli adımlarından biridir.

Özellik Seçimi, tahmin etmeye çalışılan değişkene veya sonucana en çok etkileyen özellikleri otomatik veya manuel olarak seçilen süreçtir. Verilerde alakasız özellikler olması, modellerin doğruluk oranını azaltmasına sebep olabilir ayrıca modelin alakasız özelliklere göre öğrenmesine sebep olabilir (Anonymous 2021a). Özellik seçimi yöntemleri şunlardır: Özellik Önemi, Korelasyon Matrisi ve Isı Haritası.

3.2.3.4.1 Özellik önemi (Feature importance)

Modelin özellik önemini (feature importance) kullanarak veri kümesinin her bir özelliğinin önemi elde edilebilir. Özellik önemi, verilerin her bir özelliği için bir puan verir; puan ne kadar yüksekse, özellik çıktı değişkenine göre o kadar önemli veya alakalı olur. Özellik önemi, Ağaç Tabanlı Sınıflandırıcılar ile birlikte gelen dahili bir sınıftır, veri kümesinin en iyi 10 özelliğini çıkarmak için Ekstra Ağaç Sınıflandırıcı (Extra Tree Classifier) kullanılmıştır.

Şekil 3.11’de Ekstra Ağaç Sınıflandırıcı kullanılarak veri setinde en iyi 10 özellik verilmiştir. Bu grafiğe göre makine öğrenmesi modelleri için en iyi özellik KM alanı olduğu tespit edilmiştir. Daha sonraki en iyi özelliğin 2 numaralı kusur grubu olan Fren Sistemi olduğu ortaya çıkmıştır.



Şekil 3.11 Araç muayene verisetinde özellik önemi

3.2.3.4.2 Korelasyon matrisi ile ısı haritası (Heatmap)

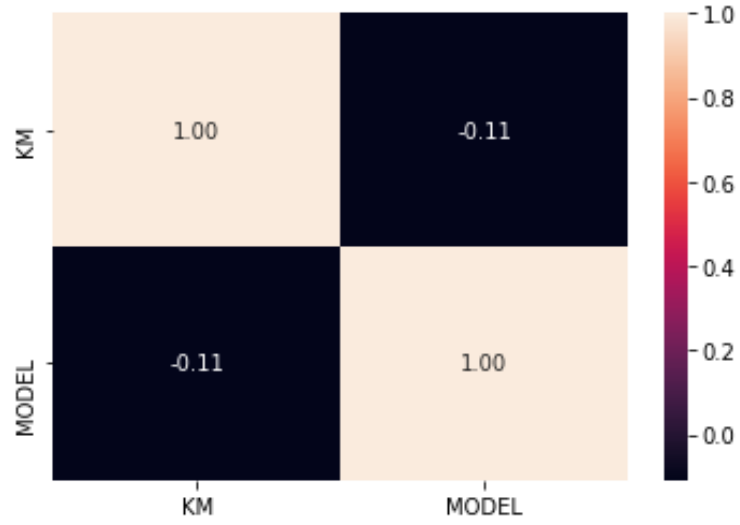
Korelasyon, özelliklerin birbirleriyle veya hedef değişkenle nasıl ilişkili olduğunu belirtir. Korelasyon pozitif (özellğin bir değerindeki artış hedef değişkenin değerini artırır) veya negatif (özellğin bir değerindeki artış hedef değişkenin değerini azaltır) olabilir.

Isı haritası, hangi özelliklerin hedef değişkenle en çok ilişkili olduğunu belirlemeyi kolaylaştırır, seaborn kütüphanesini kullanarak ilişkili özelliklerin ısı haritasını çizilmiştir.

Şekil 20'de verilen ısı haritası grafiğinde KM ve MODEL alanlarının birbiri ile korelasyonunu göstermektedir. Isı haritası grafikleri sadece sayısal değerler üzerinde çalışır. Korelasyon değerleri -1 ile 1 arasındadır.

- 1'in anlamı, birbiriyle pozitif korelasyon gösteren iki değişkendir.
- 0'ın anlamı, değişkenler arasında korelasyon olmamasıdır.
- -1'in anlamı, iki değişkenin birbirleriyle negatif ters korelasyon göstermesidir.

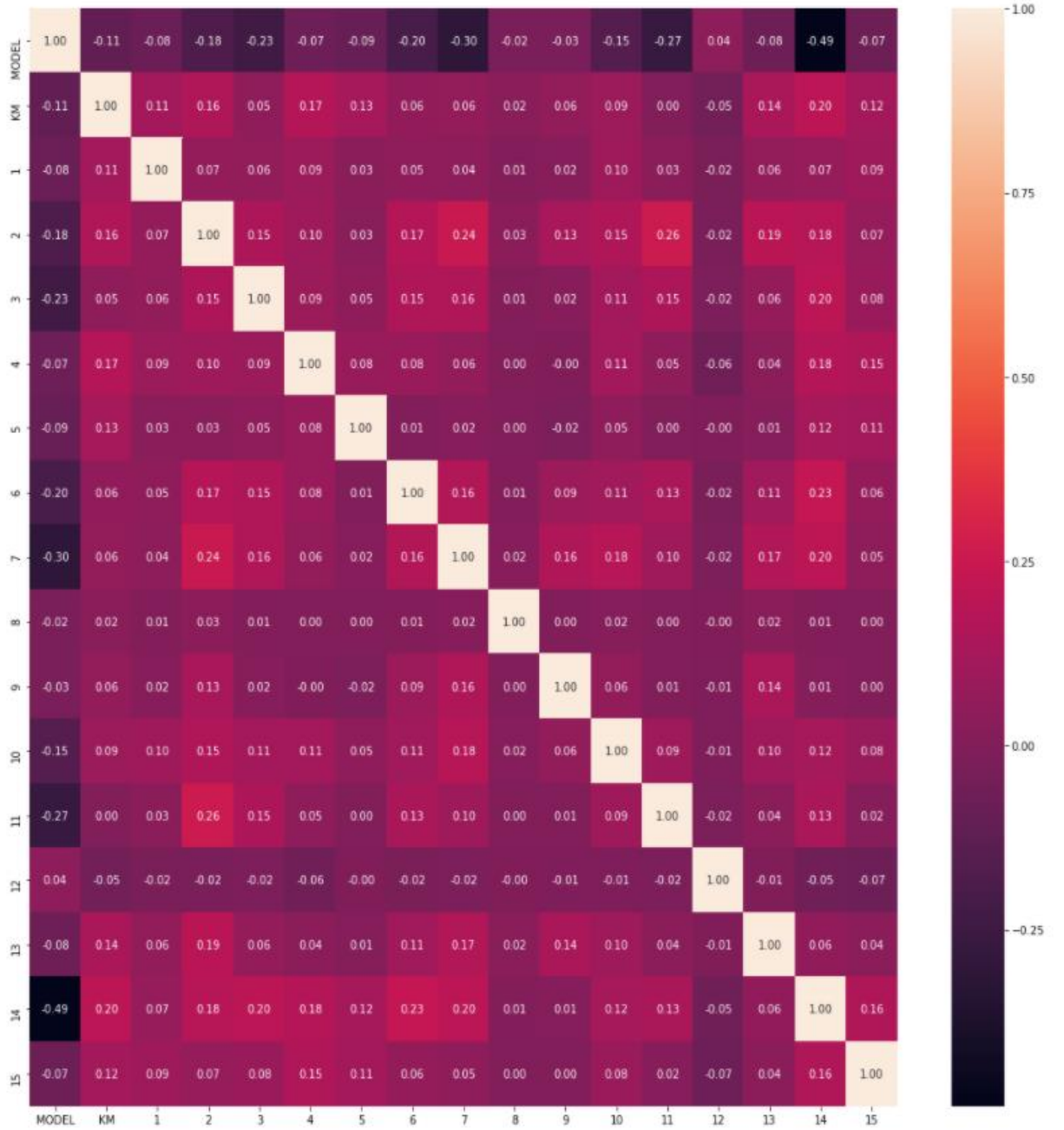
KM ve Model alanı sayısal değerlerdir. Şekil 3.12'ye göre KM ile MODEL arasında 0.11'lik ters korelasyon vardır. Yani KM arttıkça MODEL yılı düşmektedir.



Şekil 3.12 KM ve model alanının ısı haritası

Isı haritasındaki renklerin farklılığı, sınıflandırma raporundaki en etkili öznitelikleri gösterir, en düşük değer daha koyu bir renk alırken, en parlak renkler daha yüksek değere sahiptir ve diğer özellikler arasında en etkilidir.

Tüm nümerik özniteliklerin birbiri ile ilişkisini şekil 3.13’de verilmiştir. Isı haritası sadece nümerik veri tipini kabul eder. Bu sebepten dolayı veride nümerik olan KM, MODEL ve kusur alanları (1,2,...15) kullanılmıştır. Grafiğe göre en yüksek korelasyon 14. Kusur grubu yani Çevre Kirliliği: Gürültü oluşumu/ Parazit önleme/Yağ, Hidrolik, Sıvı kaçaqları ile ilgili kusurlar ve MODEL alanı 0.5 korelasyona sahiptir. Yani bir aracın model yılı arttıkça çevreye verdiği kirlilik oranında artış yaşanmaktadır. -0.3 korelasyona sahip olan MODEL alanı ile 7. (Şase/karoseri - bağlı parçalar ile ilgili kusurlar) ve 11. (Motor/Tahrik Debriyaj/Vites değiştirme ile ilgili kusurlar) kusur grupları birbiriyle ters ilişkili ortaya çıkmıştır. Yani aracın model yılı arttıkça bu kusurların çıkma olasılığı azalmaktadır. Araç muayene verisindeki diğer alanların birbiriyle olan korelasyonu düşük çıkmıştır.



Şekil 3.13 Tüm özneliliklerin ısı haritası

3.2.3.5 Eğitim ve test veri seti oluşturma

Veriler sınıflandırma algoritmaları uygulanmadan önce şekil 3.14'deki gibi eğitim ve test olarak ikiye ayrılmıştır. Verinin %70'i olan 5.174.598 veri X_train adında bağımsız değişkenlere sahip olan ve veri madenciliği algoritmaları ile veriyi eğitmek kullanılan veri setine ayrılmıştır. Verinin %30'u ise 2.217.685 veri Y_train adında bağımlı değişkene sahip olan test veri setine ayrılmıştır.



Şekil 3.14 Eğitim ve test verisi

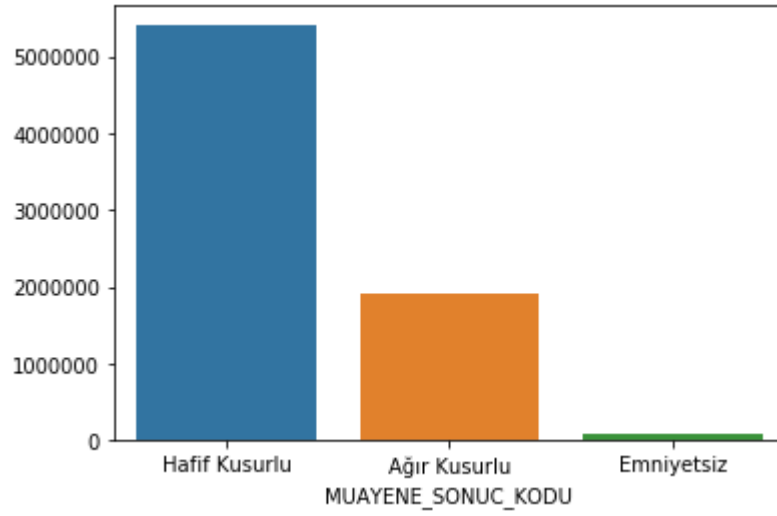
3.2.3.6 Veri örnekleme

Veri setindeki sınıf dengesizliği, makine öğreniminde, özellikle sınıflandırma problemlerinde yaygın bir problemdir. Dengesizlik durumunda, model doğruluğu oranı büyük oranda etkilenir.

Çoğu makine öğrenimi algoritması, her sınıftaki örnek sayısı yaklaşık olarak eşit olduğunda en iyi sonucu verir. Bunun nedeni, çoğu algoritmanın doğruluğu en üst düzeye çıkarmak ve hataları azaltmak için tasarlanmış olmasıdır. Bununla beraber, eğer veriler dengesiz ise, sadece çoğunluk sınıfını tahmin ederek oldukça yüksek bir doğruluk elde edilebilir, fakat çoğu zaman model azınlık sınıfını yakalamada başarısızdır. Bu gibi durumlarda yeniden örnekleme teknikleri kullanılır.

Örnekleme tekniği dengesiz veri kümeleriyle başa çıkmak için yaygın olarak kullanılan bir yöntemdir. Çoğunluk sınıfından örneklerin çıkarılmasından (az örnekleme) veya azınlık sınıfından daha fazla örnek eklenmesinden (çok örnekleme) oluşur.

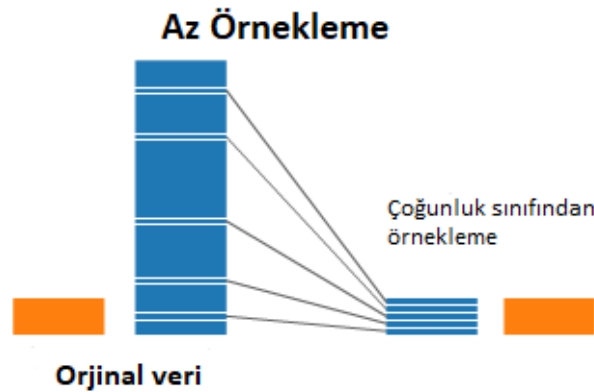
Bu tez çalışmasında kullanılan araç muayene verilerinin sınıfları dengesiz dağıldığından dolayı veri örnekleme yöntemi kullanılmıştır. Şekil 3.15’de verilen grafikte araç muayene verilerinin 5.396.831 tanesi hafif kusurlu, 1.907.238 tanesi ağır kusurlu, 88.214 tanesi emniyetsiz sınıfına mensuptur. Hafif kusurlu sınıfı çoğunluk sınıfıdır, emniyetsiz ve ağır kusurlu sınıfta azınlık sınıfıdır. Bu sebepten dolayı veriler dengeli değildir. Veriler üzerinde az örnekleme ve çok örnekleme yöntemleri kullanılarak makine öğrenmesi modelleri üzerinde karşılaştırma yapılmıştır.



Şekil 3.15 Araç muayene verilerinin muayene sonucuna göre dağılımı

Az Örnekleme (Under Sampling): Çoğunluk sınıfının bazı verilerin veri setinden çıkarılması olarak tanımlanabilir (Şekil 3.16). Bu işlem çoğunluk ve azınlık sınıf dengelenene kadar yapılır.

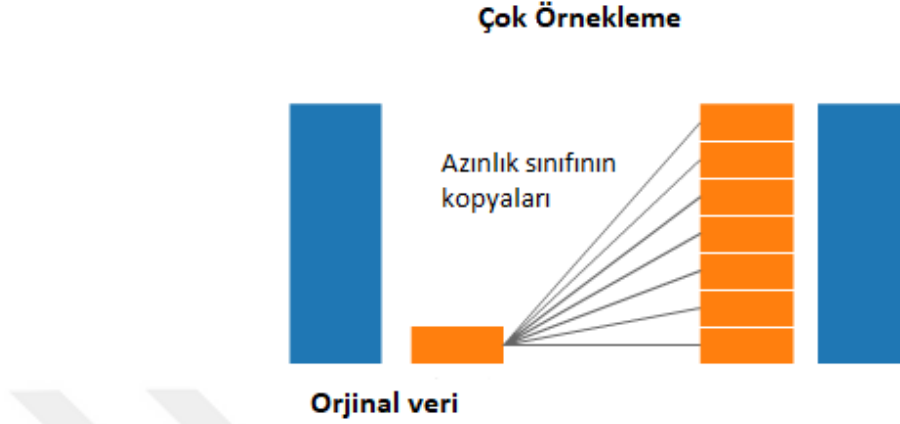
Milyonlarca satır veriye sahip olduğunda az örnekleme iyi bir seçim olabilir. Ancak az örneklemenin bir dezavantajı, değerli olabilecek bilgilerin de veriden çıkarılarak bilgi kaybına sebep olmasıdır. Eğitim veri seti çok büyük olduğunda eğitim verisi örneklerinin sayısını azaltarak çalışma süresini ve depolama sorunlarını iyileştirmeye yardımcı olabilir. Az örnekleme yaparken göz önünde bulundurulması gereken bir husus, test verisinde fazla uydurmaya ve zayıf genellemeye neden olabilmesidir.



Şekil 3.16 Az örnekleme

Çok Örnekleme (Over Sampling): Azınlık sınıfına daha fazla kopya eklemek olarak tanımlanabilir (Şekil 3.17). Çalışmak için çok veri olmadığında çok örnekleme iyi bir

seçim olabilir. Az örneklemeden farklı olarak, bu yöntem bilgi kaybına yol açmaz. Az örneklemeden daha iyi performans gösterir. Azınlık sınıfı verileri çoğaltıldığı için fazla uydurma olasılığını artırır.



Şekil 3.17 Çok örnekleme

3.2.4 Sınıflandırma algoritmaları

Sınıflandırma, makine öğrenmesine dayanan klasik bir veri madenciliği yöntemidir. Temel olarak sınıflandırma, bir veri setindeki her bir öğeyi önceden tanımlanmış sınıflar veya gruplardan birine sınıflandırmak için kullanılır.

Sınıflandırma yöntemi karar ağaçları, doğrusal programlama ve yapay sinir ağı gibi matematiksel teknikleri kullanır.

Sınıflandırmada, veri öğelerini gruplara nasıl sınıflandıracağını öğrenebilen yazılımlar hazırlanır. Örneğin, “şirketten ayrılan çalışanların tüm geçmiş kayıtları göz önüne alındığında, mevcut çalışanların gelecekte çıkma olasılığını tahmin edebileceği” uygulamada sınıflandırma yöntemi uygulanabilir. Bu durumda, çalışanın kayıtlarını “ayrılmak” ve “kalmak” olmak üzere iki gruba ayrılır. Daha sonra veri madenciliği ile çalışanlar gruplara göre sınıflandırılabilir (Kalyani, 2012, s.440)

Bu tez çalışmasında yedi adet veri madenciliği sınıflandırma tekniği kullanılmıştır ve bu algoritmalar Python programlama dili ile uygulanmıştır. Bu yedi algoritma, Lojistik Regresyon, KNN, Naive Bayes, Decision Tree, Random Forest, XGBoost ve Ada Boost sınıflandırıcısını içermektedir. Araç muayene verileri ile farklı sonuçlar ve doğruluk seviyeleri elde edilmiştir.

3.2.4.1 Lojistik Regresyon (LR)

Lojistik regresyon, verileri tanımlamak ve bir bağımlı ikili değişken ile bir veya daha fazla nominal, sıra, aralık veya oran düzeyinde bağımsız değişken arasındaki ilişkiyi açıklamak için kullanılır. Lojistik regresyon, tahmine dayalı, bağımlı değişken iki farklı değer aldığı durumlarda yapılacak uygun bir regresyon analizidir. Kategorik veya sayısal verilerin sınıflandırılmasında kullanılır (Anonymous 2021f). Verideki karmaşık ilişkileri anlamak için kullanılan yararlı bir istatistiksel tekniktir (Connelly, 2020).

En basit lineer denklem $ax+b$ olup, a doğrunun eğimi ve b y eksenindeki kesişme noktasıdır. Regresyon, bir değişkenin ortalama değeri ile diğerinin karşılık gelen değerleri arasındaki korelasyonun hesaplanmasıdır. Bu öğrenme algoritmasında, x girdi özelliklerinden çıktı etiketi y'yi tahmin etmek için doğrusal model ve regresyon modeli birleştirilir.

Doğrusal bir regresyon formülü aşağıdaki gibidir.

$$y=w[0].x[0]+w[1].x[1]+\dots+w[n].x[n]+b$$

Bu, n özellikli bir verinin regresyon versiyonudur. $w[0]$, $w.x+b$ denkleminin eğimine karşılık gelen bir ağırlık parametresidir; burada b, bias parametresine karşılık gelen y kesişimidir. Tek bir özelliğe sahip bir veri seti için aşağıdakilerin çizgi denklemini verir:

$$y= w.x+b$$

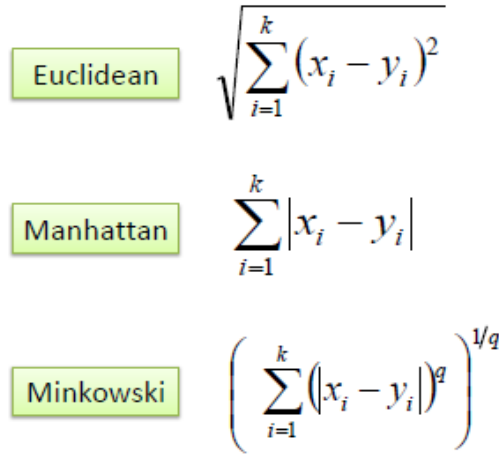
Özellik sayısı arttıkça parametrenin ağırlığı (n,m) şeklinde matrisler olarak alınır; burada n örnek sayısı ve m özellik sayısıdır. Böylece tahmin, tüm özelliklerin ağırlıklar ve yanlılıklar ile ağırlıklı toplamı ile yapılır.

Doğrusal regresyon modelinde tahmin ile etiket arasındaki mesafe maliyet fonksiyonu ile ölçülür. Öğrenme algoritmasının amacı, maliyeti en aza indirmenin bir yolunu bulmaktır. Optimizasyon, eğitim adımı parametreler ayarlanarak bu mesafelerin minimizasyonunun sağlandığı bir süreçtir.

3.2.4.2 K-En Yakın Komşu (KNN)

K-En Yakın Komşu algoritması temel mantığı basit, uygulaması kolay gözetimli öğrenme algortimalardan biridir. Hem regresyon hem de sınıflandırma problemlerinin çözümünde kullanılmaktadır. KNN algoritması, T. M. Cover ve P. E. Hart tarafından 1967 yılında önerilmiştir.

Algoritmadaki k, veriye en yakın olan k adet komşu sayısını belirtmektedir. Yeni bir veri geldiğinde bu verinin her bir nokta ile arasındaki uzaklık hesaplanır ve k sayıda kendine en yakın olan komşu sayısına bakılarak sınıflandırılır. KNN algoritmasında genellikle uzaklık fonksiyonu için şekil 3.18’de verilen Öklid (Euclidean), Manhattan, Minkowski fonksiyonları kullanılır (Taşdelen, 2019).



The diagram displays three distance functions, each with a label in a green box and its corresponding mathematical formula. The background features a large, light gray 'X' shape.

- Euclidean**:
$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$
- Manhattan**:
$$\sum_{i=1}^k |x_i - y_i|$$
- Minkowski**:
$$\left(\sum_{i=1}^k (|x_i - y_i|)^q \right)^{1/q}$$

Şekil 3.18 Uzaklık fonksiyonları

KNN gürültülü verilere karşı dirençli olması, basit ve uygulaması kolay olması avantajları sebebi ile popüler bir algoritmadır. Bunun yanında veri boyutunun büyük olması, öznelik sayısının fazla olması ile birlikte bellek kullanımında artış ve artan işlem maliyetleri dezavantajlarıdır. Algoritmadaki k değerinin belirlenmesinde bir kriter yoktur seçim deneme yanılma yolu ile yapılmaktadır (Laribi, 2018).

KNN algoritmasının adımları (Şekil 3.19) aşağıdaki gibidir:

- İlk önce k parametresi değeri belirlenir. Bu k parametresi verilen bir noktaya en yakın komşuların sayısıdır. Örneğin k değeri 3 olsun. Böylece en yakın 3 komşuya göre sınıflandırma işlemi yapılacaktır.

- Veri kümesine katılacak olan yeni bir verinin, mevcut verilere göre uzaklığı uzaklık fonksiyonları yardımıyla hesaplanır.
- Uzaklıklara göre en yakın k komşular bulunur. Veriler öznitelik değerlerine göre k adet komşuların sınıfına atanır.
- Seçilen sınıf, tahmin edilen gözlem değerinin sınıfı olarak kabul edilir. Böylece yeni veri etiketlenmiş (label) olur.



Şekil 3.19 KNN adımları (Anonymous. 2021g)

3.2.4.3 Naive Bayes (NB)

Naive Bayes algoritması gözetimli öğrenmeye dayalı bir sınıflandırma yöntemidir. Bu teknik Bayes teoremine dayalı olasılıksal bir modeldir. Bayes teoremi formül (1)'de verilmiştir; buradaki y , sınıf değişkenidir ve X , m özellikleri için $X = (x_1, x_2, \dots, x_m)$ olarak tanımlanan özellik kümesidir (Tüysüzoğlu, 2021).

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)} \quad (1)$$

Bu teoreme göre bir öznitelik değerinin belirli bir sınıf üzerindeki etkisinin, diğer özniteliklerin değerlerinden bağımsız olduğunu varsayar. Bu varsayım, sınıf koşullu bağımsızlık olarak adlandırılır (Han ve Kamber, 2000).

Naive Bayes algoritmasının avantajı, uygulaması kolay ve küçük veri kümeleriyle eğitilebilir olması, alakasız özniteliklere duyarlı olmamasıdır. Dezavantajları ise, her

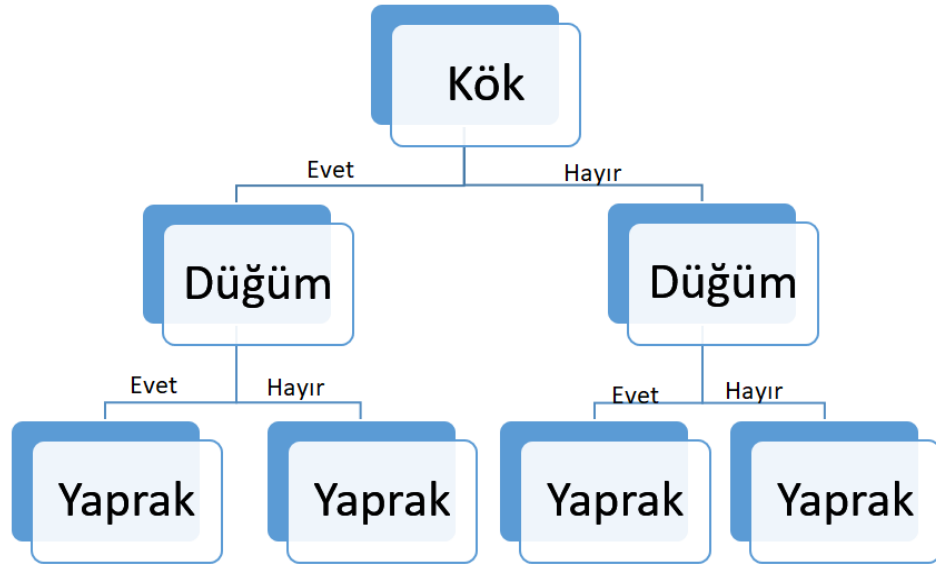
özelliğın bağımsız ve hesaplamasının yoğun olmasıdır. Naive Bayes sınıflandırma algoritması, çoğu veri sınıflandırma probleminde daha iyi çalışır.

3.2.4.4 Karar Ağaçları (Decision Tree)

Denetimli bir öğrenme yöntemi olan karar ağacı, J. Ross tarafından Sydney Üniversitesi'nde geliştirilen bir veri madenciliği tekniğidir (Fayez, 2018). Karar ağaçları, regresyon ve sınıflandırma problemlerinde kullanılan, ağaç tabanlı algorithmadan birisidir (Anonymous 2021h).

Makine öğrenmesi yöntemlerinden biri olan karar ağacı, tümevarım yöntemi ile sınıflandıran bir karar verme algoritmasıdır. Büyük miktardaki veri setini basit karar verme adımları ile daha küçük parçalara ayıran bir sınıflandırma algoritmasıdır. Güvenirliliği, kolay yorumlanması ile sınıflandırma algoritmaları arasındaki popülerliğini korumaktadır (Çelik ve Altunaydın, 2018).

Karar ağaçları algoritmasındaki amaç; bilinmeyen bir veri modelinin tahmin edilmesi için, elimizdeki verilerin özelliklerinden basit kurallar çıkararak bu kuralları öğrenen bir model kurulmasıdır (Atak, 2019).



Şekil 3.20 Karar ağacı yapısı

Karar ağaçlarında ilk hücreye kök (root) denir. Gelen her bir veri kökteki koşula göre “Evet” veya “Hayır” olarak sınıflandırılır (Şekil 3.20).

Kök hücrelerin altında hücelere düğüm denir. Gelen her bir veri düğümler yardımıyla sınıflandırılır. Modeldeki düğüm sayısı arttıkça modelin karmaşıklığı da artar.

Karar ağacının en alt tabakasında yapraklar (leaf) bulunmaktadır. Yapraklar, bize sonucu vermektedir.

3.2.4.5 Rastgele Orman (Random Forest)

Rastgele orman algoritması hem regresyon hem de sınıflandırma problemleri için uygulanabilen denetimli bir veri madenciliği tekniğidir. Tin Km Ho tarafından 1995 yılında geliştirilmiştir.

Yüksek doğruluk oranı için genellikle rastgele orman sınıflandırıcı algoritması kullanılır. Karar Ağaçları (DT) ve Rastgele Orman (RF) arasındaki fark, Rastgele Orman sınıflandırıcısının birkaç karar ağacı oluşturmak için özellikleri ve gözlemleri rastgele seçmesi ve sonra sonuçların ortalamasını almasıdır. Başka bir fark ise, karar ağaçlarının en büyük problemlerinden olan aşırı öğrenerek veriyi ezberlemesidir. Rastgele Orman, hem özniteliklerin hemde veri setinin rassal olarak onlarca, yüzlerce farklı alt kümelerini oluşturarak bunları eğitir ve bu alt kümeleri kullanarak daha küçük ağaçlar oluşturup daha sonra alt ağaçları birleştirerek ezberlemeyi önler. Yani rastgele orman modelinde farklı veri kümeleri üzerinde eğitim gerçekleştiği için karar ağaçlarının en büyük sorunu olan aşırı uyumu azaltır.

Rastgele orman algoritmasının diğer bir özelliği ise veri setindeki özniteliklerin ne kadar önemli olduğunu vermesidir (Anonymous 2021).

Rastgele Orman algoritmasının çalışma mantığı aşağıda verilmiştir.

- İlk önce kullanıcı tarafından 2 parametre tanımlanması istenir. Bunlar, oluşturulacak ağaç sayısı (N) ve ağaç yapısındaki her düğümde kullanılacak değişken sayısı m'dir (Breiman, 2001).
- Parametreler belirlendikten sonra eğitim veri kümesinin 2/3'ü öğrenme verisi, 1/3'ü test verisi olarak belirlenmektedir.
- Oluşturulan her bir ağaç için budama (bootstrap) tekniği kullanılarak önyüklemeli örneklem oluşturulur. Örneklemeler öğrenme ve test verileri olarak ayrılır ve daha sonra her bir örneklem için ağaç gelişimleri başlar.

- Her düğümde m değişkenleri tüm değişkenler arasından rastgele olarak seçilir ve bu değişkenler kullanılarak en iyi dallanma belirlenir. Bunun için CART (Classification and Regression Tree) algoritması kullanılır (He, vd., 2015; Gislason, vd., 2006)
- CART algoritması ağaçtaki en iyi dalı bulmak için formül (2) de verilen GINI indeksini kullanır (Gislason, vd., 2006). Denklemden verilen T eğitim veri kümesini, C_i pikselin ait olduğu sınıfı, $f(C_i, T)/|T|$ seçilen pikselin C_i sınıfına ait olma olasılığını göstermektedir. (Gislason, vd., 2006). GINI indeksi her bir düğümdeki örneklerin homojenliğini ölçer.
- Algoritma, her düğümde rastgele olarak seçilen değişkenler için GINI indeksini hesaplar. GINI indeksinin küçük olduğu değişken seçilir ve diğer düğüme geçer. GINI indeksi 0 ise düğüm tamamı ile homojendir ve dallanma biter (Gislason, vd., 2006). Böylece ağaçlar oluşturulur.

$$\sum_{j \neq i} (f(C_i, T)/|T|)(f(C_j, T)/|T|) \quad (2)$$

Rastgele orman algoritması avantajları ise büyük veri kümelerinde etkilidir ve yüksek doğruluk oranı sağlar. Ayrıca özellikle veri kümesinde eksik değerler olduğunda doğruluğu korumada iyidir. Rastgele Orman algoritması sınıflandırıcı kategorik veriler için modellenebilir. Ayrıca bu algoritma sınıflandırma problemlerindeki aşırı uyum problemine önleyebilmektedir.

3.2.4.6 XG Boost (XGB)

XG Boost (Extreme Gradient Boost), 2014 yılında Chen tarafından geliştirilmiştir. Günümüzde en popüler makine öğrenme algoritmalarından birisidir. XGBoost diğer makine öğrenme tekniklerinden daha iyi çözümler sağladığı bilinmektedir.

Algoritmanın en önemli avantajları yüksek tahmin elde edebilmesi, aşırı uyumu önleyebilmesi, eksik verileri yönetebilmesi ve hızlı olmasıdır. Daha iyi sonuçlar elde edebilmek için daha az kaynak kullanarak donanım ve yazılım optimizasyon yöntemleri uygulanmıştır. Karar ağacı temelli modellerin en iyisi olarak gösterilmektedir (Anonymous 2021i).

XG Boost, tüm kolektif öğrenme yöntemleri arasından en güçlü modellerden birisidir. Kolektif Öğrenme genel performansı iyileştirmek için birden fazla modeli birleştirir.

Kolektif öğrenme yöntemleri:

- Torbalama (bagging),
- Yığma (stacking),
- Yükseltme (boosting)

Torbalama yöntemi her seferinde veri setinden farklı bir alt küme olarak eğitir ve sonra tüm baz modellerin sonuçlarının ortalamasını alarak bir kolektif öğrenme modeli oluşturur. Baz modeller oluşturulurken veri kümesinin farklı altkümeleri seçilir.

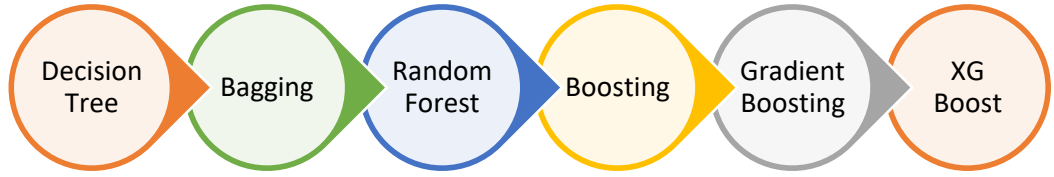
İlk baz model oluşturulurken seçilen altkümeler herhangi bir örnek ikinci model oluşturulurken de seçilebilir. Örneğin Rastgele Orman (Random Forest) bagging tekniği kullanır.

Yükseltme tekniğinde veri kümesinin bir alt kümesini alınır ve yine o alt kümenin üzerine bir model kurulur. Fakat daha sonra kurulan başka bir baz model ilkinden bağımsız değil, tersine ilk modeli üzerine geliştirilir.

Torbalama yöntemi her bir baz model sonuca eşit etki eder fakat boosting yönteminde ilk olarak hatalı tahmin edilen verilere daha fazla ağırlık verilip bir sonraki baz modelin bu hataları düzeltmesi amaçlanır.

Yığma yöntemi ise diğer iki kolektif öğrenme modelinin tersine veri setinin altkümesi değil veri setinin tamamı üzerinde kurulmuş farklı modelleri birleştirmektedir. Boosting yönteminde baz modeller kendinden önceki baz modelleri geliştirirken, stacking tekniğinde her bir baz model kendi içinde en iyi sonuca ulaşmaya çalışmaktadır.

XG Boost modeli diğer kolektif öğrenme modellerinden daha hızlıdır. Paralel işleme prensibiyle çalışır. Bu sayede büyük veri setlerinde en kullanışlı modellerdendir. XGBoost çoğunlukla diğer kolektif öğrenme modellerinden daha iyi sonuç vermektedir. Şekil 3.21’de XG Boost algoritmasının evrimi verilmiştir.



Şekil 3.21 XG Boost algoritmasının karar ağaçlarından evrimi (Anonymous 2021j).

3.2.4.7 Ada Boost

Adaptive Boosting'in kısaltması olan Ada Boost algoritması, zayıf sınıflandırıcıları bir araya getirerek daha güçlü bir sınıflandırıcı oluşturan Freund ve Schapire (1997) tarafından önerilmiş bir topluluk sınıflandırıcı yöntemidir. Algoritma içerisinde farklı makine öğrenmesi algoritmalarını birleştirerek ya da karşılaştırarak olası en iyi sonucun elde edilmesi amaçlanır (Freud ve Schapire, 1997). Başlangıçta her bir örnek veri için eşit bir dağılımla başlar ve sınıflandırma performansına göre en iyi sınıflayıcı bulunur. Daha sonra ağırlıklar güncellenerek yanlış sınıflandırılan örneklerle yoğunlaşılır. Böylece iterasyon sonucunda en güçlü sınıflayıcılar birleştirilerek güçlü bir sınıflayıcı oluşturulmuş olur ve sınıflandırma başarısı artırılır (Başar vd., 2016).

Ada Boost, modelin performansını artırmak için diğer öğrenme algoritmaları ile birlikte kullanılabilir. Bazı problemlerde, diğer öğrenme algoritmalarına göre daha çok ezberleme problemine daha az duyarlı olabilmektedir. Öğrenme algoritmaları, bazı problem türlerinde diğerlerinden daha iyi uyma eğilimindedir ve tipik olarak, bir veri setinde en iyi performansı elde etmeden önce ayarlanması gereken birçok farklı parametre ve konfigürasyona sahiptir. Ada Boost kullanıma hazır en iyi sınıflandırıcı olarak adlandırılmaktadır (Anonymous 2021n).

Ada Boost algoritmasının uygulanması kolaydır. Zayıf sınıflandırıcıların hatalarını yinelenmeli olarak düzelterek ve zayıf öğrencileri birleştirerek doğruluğu artırmayı hedeflemektedir. Ada Boost ile birçok temel sınıflandırıcılar kullanılabilir. Ada Boost modeli aşırı uyuma duyarlı değildir. Ada Boost dezavantajları, gürültü verilere duyarlı olmasıdır ve aykırı değerlerden etkilenmektedir. Ada Boost algoritması, XGBoost modeline ile karşılaştırıldığında daha yavaştır (Anonymous 2021o).

3.2.4.8 Derin Sinir Ağları (DNN)

Derin öğrenme, insan beyninin işleyişini, verileri işleme ve karar vermede işleminde kullanmak için modeller oluşturarak taklit eden yapay bir zeka işlevidir.

Derin öğrenme, makine öğrenmesinin bir alt dalıdır. Makine öğrenmesi ise yapay zekanın bir alt dalıdır. Yapay zeka, derin öğrenme ve makine öğrenmesi arasındaki ilişki şekil 3.22'deki gibi görselleştirilebilir (Goodfellow vd., 2006).

Makine öğrenmesi algoritmalarında doğru tahmin yapmanın yolu daha fazla veri verilerek öğretmek iken, derin öğrenme yöntemlerinde algoritma bunu verileri işleyerek öğrenebilir. Basitçe, özellik çıkarma, makine öğrenmesi yönteminde insan tarafından yapılırken, derin öğrenme yönteminde bunu kendi kendine çözer (Yaraş, 2020).



Şekil 3.22 Derin Öğrenme, makine öğrenmesi ve yapay zeka arasındaki ilişki

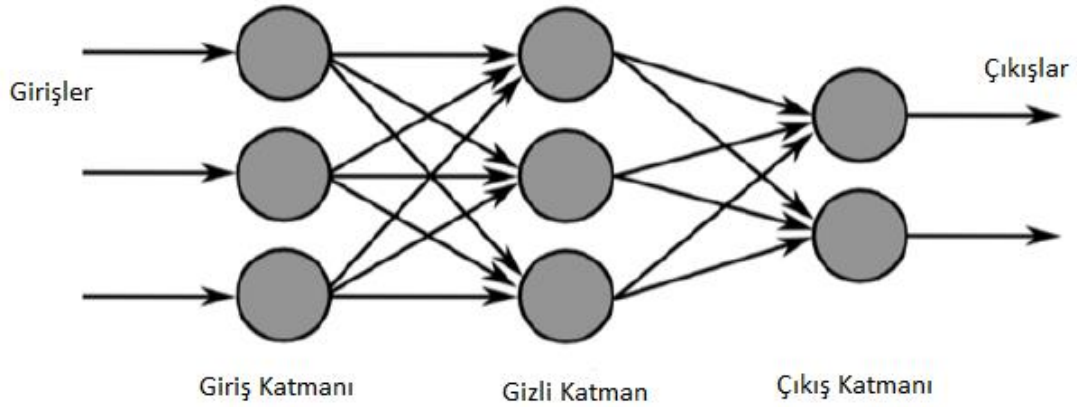
Derin sinir ağları yapay sinir ağının çok katmanlı biçimidir. En basit yapay sinir ağı modeli olarak bilinen tek katmanlı yapay sinir ağı doğrusal olmayan problemlerin çözümünde yetersiz kalması nedeniyle çok katmanlı yapay sinir ağları geliştirilmiştir.

Derin sinir ağları, birden çok gizli katmana sahip olan yapay sinir ağlarıdır. Buradaki derinlik gizli katman sayısı artırılarak sağlanmaktadır (Olgun, 2020),

Derin sinir ağlarının, Tekrarlayan Sinir Ağları (RNN), Evrişimsel Sinir Ağları (CNN), gibi farklı derin öğrenme mimarileri vardır. Evrişimsel Sinir Ağları, görüntü, ses, video

işlemede büyük başarılar sağlarken; Tekrarlayan Sinir Ağları, metin ve konuşma gibi ardışık verilerde başarı göstermektedir.(Lecun vd., 2015).

Yapay Sinir ağlarında aynı insan beyni gibi nöronlardan oluşmaktadır. Tüm nöronlar birbiriyle bağlantılıdır. Basit bir yapay sinir ağı giriş katmanı ile başlar. Giriş katmanında giriş verileri sayısı kadar nöron bulunur. Giriş katmanı gizli katman ile bağlanır. Gizli katmanda girdi verilerimiz üzerinde matematiksel hesaplamalar yapar. Bu gizli katman ise çıkış katmanına bağlanır. Çıkış katmanı çıkış tahminlerini vermektedir. Yapay sinir ağlarında en az bir gizli katman bulunur. Basit bir yapay sinir ağı modeli şekil 3.23’de gösterilmektedir.

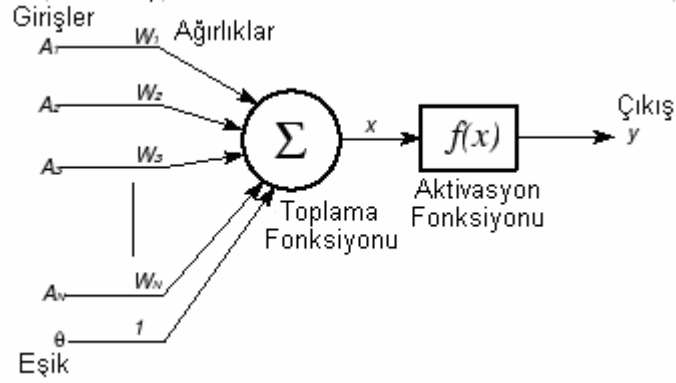


Şekil 3.23 Yapay sinir ağı modeli

Giriş katmanına gelen veriler rastgele bir değer verilen ağırlıklar ile çarpılır. Her bir nöron girişi, o girişe atanan bir ağırlığa sahiptir Bu ağırlıklar modelin eğitim süresince güncellenmektedir (Yılmaz, 2019).

Eğitilen yapay sinir ağı modeline verileri öğretebilmek için kullanılan fonksiyonlara aktivasyon fonksiyonları denir. Yapay sinir ağındaki her bir nöron, bir diğer nöronun çıktısını girdi sinyali olarak alır, aktivasyon fonksiyonu ile işler ve sonucu diğer nörona vererek çıktı sinyali üretir (Taylor, 2005). Aktivasyon fonksiyonları, nöronun aktivasyonunu sağlayan bir fonksiyondur. Sinyallerin iletilmesi ve işlenebilmesi için eşik değerinin aşılması gerekmektedir; aksi takdirde iletim gerçekleşemez (Olgun, 2020). Şekil

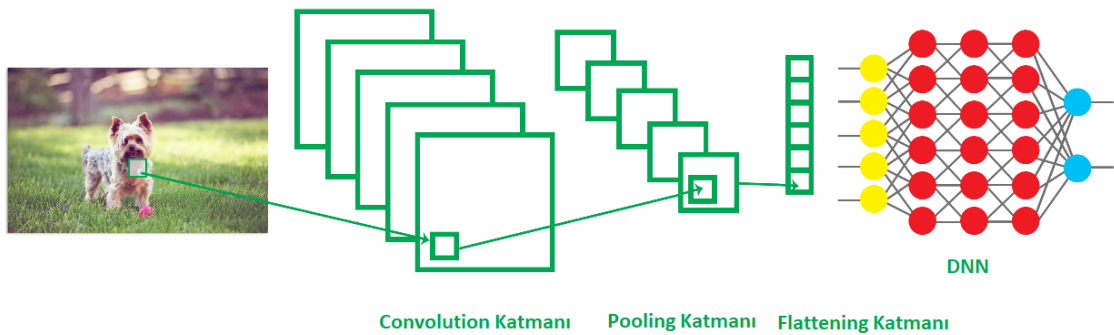
3.24’de bir yapay sinir ağı yapısı verilmektedir. En çok kullanılan aktivasyon fonksiyonları ReLU(Rectified Linear Unit), softmax, sigmoid fonksiyonlarıdır.



Şekil 3.24 Yapay sinir ağı mimarisi (Zafari vd., 2013)

3.2.4.9 Evrişimsel Sinir Ağları (CNN)

Evrişimsel Sinir ağları olarak bilinen CNN görüntü tanıma, sınıflandırma, kümeleme problemlerinde kullanılan derin yapay sinir ağlarıdır. CNN yapısı gereği girdi olarak resim veya videoları alır. Şekil 3.25’de gösterildiği gibi 4 katmandan oluşmaktadır: Convolutional Katmanı, Pooling (Havuzlamam) Katmanı, Flattening Katmanı ve Fully Connected Katmanı. Bu katmanlarda, farklı işlemlerden geçen resim derin sinir ağları modeline verilir.



Şekil 3.25 Evrişimsel Sinir Ağı katmanları (Anonymous 2018)

Convolutional (Evrişimsel) Katmanı: Bu katman CNN algoritmasının yapı taşıdır. Bu katmanda girdi olarak alınan resim bir filtreleme işleminden geçirilerek resmin özellikleri saptanır.

Pooling (Havuzlama) Katmanı: Bu katmanda boyutsallık azaltılır. Bu sayede resimde saptanan gereksiz özellikler çıkartılarak önemli özelliklere odaklanılır. CNN’de en çok max (maximum) ve average (ortalama) pooling teknikleri kullanılmaktadır.

Flattening (Tam Bağlantılı) Katmanı: Bu katmanda veri sinir ağına girmeden önce veriler hazırlanır. Derin sinir ağları, giriş verilerini tek boyutlu dizi şeklinde alır. Bu katmanda Convolutional ve Pooling katmanından gelen matrisler şeklindeki, veriler tek boyutlu dizi haline getirilir.

Fully Connected (Tam Bağlantılı) Katmanı: Bu katman Evrişimdel Sinir Ağlarının en son katmanıdır. Girdi verileri klasik derin sinir ağı modeline verilir. Sinir ağı modeli ile öğrenme işlemi gerçekleşir.

3.2.5 Birliktelik kuralları

Veri madenciliğinin en önemli tekniklerinden biri olan birliktelik kuralı operasyonel veritabanlarındaki veya diğer veri havuzlarındaki öğeler arasında ilginç korelasyonları, sık kalıpları, birliktelik davranışlarını veya ilişkilendirmeleri ortaya çıkarmayı amaçlamaktadır (Kotsiantis ve Kanellopoulos, 2006).

Birliktelik kural çıkarımı yöntemine genellikle market sepeti uygulaması denilmektedir. Örneğin, müşteriler alışveriş yaparken satın aldıkları ürünler arasındaki bağlantıyı ve birlikteliği bularak müşterilerin satın alma alışkanlıklarını analiz etmektedir (Örneğin yumurta alan müşteri ekmek de alır ya da peynir alan müşteri zeytin de alır gibi). Bu birliktelik-ilişki bağlantılarının keşfedilmesi, müşterilerin hangi ürünleri bir arada aldıkları bilgisini ortaya çıkarır ve satıcılar bu bilgi ışığında kazançlı satışlar yapabilir ve etkili satış stratejileri geliştirebilmektedirler.

3.2.5.1 Apriori algoritması

Birliktelik kuralı veri madenciliği yöntemleri içinde Apriori algoritması en fazla bilinen ve uygulanan yöntemdir. Birliktelik kuralları algoritması olan Apriori algoritması Agrawal, Imielinski ve Swami tarafından 1993 yılında geliştirilmiştir (Agrawal, Imielinski ve Swami, 1993).

Apriori algoritması, veri kümesi içerisindeki herhangi bir olayın meydana gelme durumunu, diğer olayların meydana gelme olasılığına bakarak tahmin etmektedir (Elabiad, 2013, s.18).

Birliktelik kurallarında kullanılan temel kavramlar aşağıda verilmiştir.

Öğeler kümesi (itemset): Bir veya daha çok öğeden oluşan veri kümesidir.

Destek sayısı (support count): Öğeler kümesinin veri kümesinde görülme sıklığını ifade etmektedir.

Destek(support): Verideki bağıntının ne kadar sık görüldüğünü tanımlar, öğeler kümesinin içinde bulunduğu birlikteliklerin toplam birliktelik sayısına oranıdır. Destek($A \Rightarrow B$) şeklinde gösterilmektedir.

Güven (confidence): Öğeler arasındaki birlikteliklerin doğruluğunu ifade eder. X ürünü almış bir kişinin Y ürününü alma olasılığını vermektedir. Güven($X \Rightarrow Y$) şeklinde gösterilmektedir. Güven değerinin 1 ya da %100 olması kesin kural olduğunu ifade etmektedir. Örneğin A ve B'nin olduğu ve % 90 güvene (confidence) sahip bir birliktelik kuralında, A'yı içeren ürün kümelerinin % 90'ı B'yi de içermektedir (Keleş ve Kaya, 2014).

3.2.6 Model değerlendirme

Bir makine öğrenmesi algoritması veri setine uyguladıktan sonraki adım, modelin ne kadar etkili olduğunu bulmaktır. Literatürde makine öğrenmesi algoritmalarının başarısının ölçülmesi için farklı metrikler/ölçütler önerilmiştir. Sıklıkla kullanılan ölçütler şunlardır: Karışıklık Matrisi, Doğruluk, Hatırlama, ROC, AUC, gibi.

Bu tez çalışmasında uygulanan makine öğrenmesi algoritmalarını değerlendirmek için karışıklık matrisi, doğruluk, kesinlik, hatırlama, F1 skor, AUC değerleri kullanılmıştır.

3.2.6.1 Karışıklık matrisi

Bir sınıflandırma problemi için bir performans ölçümüdür, gerçekleşen ve tahminlenen değerleri göstermektedir. Gerçek Pozitif (True Positive) ve Gerçek Negatif (True Negative) modelin doğru olarak tahminlediği, Yanlış Pozitif (False Positive) ve Yanlış

Negatif (False Negative) ise modelin yanlış olarak tahminlediği alanlardır. Şekil 3.26’da karışıklık matrisi tablosu verilmiştir.

True Positive (TP) & True Negative (TN): Sınıfların doğru tahmin edildiği rakamı verir. False Negative (FN) & False Positive (FP): Sınıfların birbirleri ile olan yanlış tahmin adetlerini verir.

Tahminlenen (Predicted)	
Gerçekleşen (Actual)	True Positives (TP)
	False Negatives (FN)
Gerçekleşen (Actual)	False Positives (FP)
	True Negatives (TN)

Şekil 3.26 Karışıklık matrisi

3.2.6.2 AUC-ROC eğrisi

Makine öğrenmesi işlemlerinde performans ölçümü önemlidir. Bu sebepten bir sınıflandırma probleminin performansının ölçümünde AUC - ROC eğrisinden yararlanılır. Herhangi bir sınıflandırma modelinin performansını ölçmek için en önemli değerlendirme ölçütlerinden birisidir. Özellikle dengesiz dağılan veri setlerinde, makine öğrenmesi algoritmalarının performansını ölçmek için en çok kullanılan ölçümlerden birisidir. Modelin tahminlemede ne kadar iyi olduğunu açıklamaktadır.

Tipik bir ROC eğrisinde X eksenini Yanlış Pozitif Oran (FPR) ve Y eksenini Gerçek Pozitif Oran (TPR) değerlerini göstermektedir. Eğri Altındaki Alan (AUC), bir sınıflandırıcının sınıfları ayırt etme yeteneğinin ölçüsüdür. Eğri altında kalan alan ne kadar büyükse, makine öğrenme modelleri o kadar iyi verilen sınıfları ayırt edebilir. AUC için ideal değer 1'dir (Anonymous 2021k).

Doğruluk (Accuracy): Doğruluk değeri makine öğrenmesi algoritmasının veride doğru sınıflandırdığı örnek sayısının (TP+TN) toplam veri sayısına oranıdır.

Kesinlik (Precision): Pozitif olarak tahmin etmek istenilen değerlerin gerçekten kaç tanesinin pozitif olduğunu göstermektedir. Bu durumda Kesinlik değerinin yüksek olması

model seçiminde önemli bir kriterdir. Kesinlik, modelin ne kadar doğru olduğu ile ilgili dir. Kesinlik değeri hesaplaması formül (3)'de verilmiştir.

$$Kesinlik (Precision) = \frac{TP}{TP+FP} \quad (3)$$

Duyarlılık-Hatırlama (Recall): Pozitif olarak tahmin edilmesi gereken ifadelerin ne kadarını pozitif olarak tahmin edildiğini gösteren değerdir. Mümkün olduğunca yüksek olması gereklidir. Örneğin; Test setinde (lojistik regresyon için) ağır kusurlu araç varsa ve Lojistik Regresyon modelimiz bunu (0.01) %1 oranında belirleyebilir. Duyarlılık değeri hesaplaması formül (4)'de verilmiştir.

$$Hatırlama (Recall) = \frac{TP}{TP+FN} \quad (4)$$

F1 Score: Bu değer bize Kesinlik (Precision) ve Duyarlılık (Recall) değerlerinin harmonik ortalamasını göstermektedir. Maksimum 1 ve minimum 0 değerine sahiptir. Genel olarak, modelinizin kesinliğinin ve sağlamlığının bir ölçüsüdür (Anonymous, 2021m). F1 Score değeri hesaplaması formül (5)'de verilmiştir.

$$F1 = 2 * \frac{precision*recall}{precision+recall} \quad (5)$$

Dengesiz dağılan verilerde, çalışmada kullanılan araç muayene verilerindeki gibi hafif kusurlu verileri tahmin etmek önemli ise doğruluk (accuracy) değeri, ağır kusurlu verileri tahmin etmek önemli ise recall değerine bakmakta fayda vardır.

Support (Destek): Bu sınıfta yer alan gerçek yanıtın örnek sayısıdır.

4. BULGULAR VE TARTIŞMA

Bu bölümde, yapılan çalışmanın sonucunda elde edilen bulgular verilmiştir. Bu tezde makine öğrenmesi algoritmaları kullanılarak araç muayene sonucu tahmin edilmiştir. Tahmin modeli için araç muayene verilerine yedi adet makine öğrenme algoritması (LR, NB, KNN, DT, RF, XGBoost), derin öğrenme algoritmalarından derin sinir ağları (DNN) ve CNN uygulanmıştır. Birliktelik analizi için apriori algoritması kullanılmıştır.

Araç muayene veri setinde sınıflar dengesiz dağıldığından dolayı veriler üzerinde örnekleme işlemi yapılmıştır. İki çeşit örnekleme yöntemi uygulanmıştır: Az örnekleme (under sampling) ve çok örnekleme (over sampling). Az örnekleme yapılarak hafif kusurlu ve ağır kusurlu sınıfının bazı verileri veri setinden çıkarılarak sınıflar dengeli hale getirilmiştir. Emniyetsiz sınıfı sayısı kadar diğer sınıfların sayısı indirgenerek toplamda 264.642 adet veri elde edilmiştir. Çok örnekleme yapılarak azınlık sınıf olan ağır kusurlu ve emniyetsiz sınıfları verileri çoğaltılmıştır. Çok öğrenme ile sınıflar dengelenerek toplamda 161.90.493 veri elde edilmiştir. Tüm araç muayene verileri, az örnekleme ve çok örnekleme yapılarak elde edilen verilere yedi adet makine öğrenme algoritması (LR, NB, KNN, DT, RF, XGBoost) ve DNN uygulayarak sonuçlar karşılaştırılmıştır.

Bu tezde ayrıca Apriori algoritması uygulanarak birliktelik analizi yapılmıştır. Araç muayene verilerinde birlikte sık görülen kusurların analizi yapılmıştır.

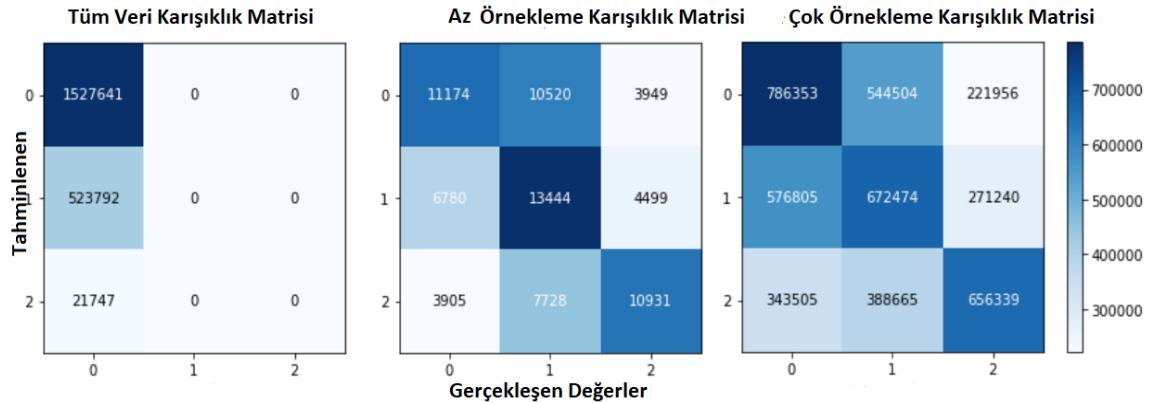
Bu bölümde araştırma sonuçları, tartışma ve çalışmanın geçerliliğinin tehdidi ele alınmıştır. Araştırma Bulguları bölümünde kullanılan makine öğrenme algoritmaları sonuçları verilmiştir. Tartışma bölümde algoritma sonuçları karşılaştırılıp bu sonuçlar tartışılmıştır.

4.1 Araştırma Bulguları

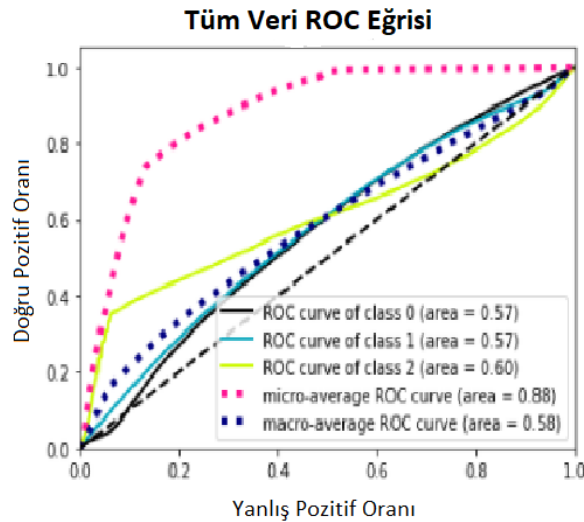
Lojistik Regresyon: Lojistik Regresyon çok değişkenli verileri gruplandırmada ideal bir yöntem olduğu için ve çok sınıflı problemlerde de kullanılabildiği için tercih edilmiştir. Lojistik regresyon, varsayılan olarak, iki sınıflı sınıflandırma problemlerinde kullanılmaktadır. Fakat çok terimli (multinomial) lojistik regresyon, çok sınıflı sınıflandırma problemleri için de kullanılan lojistik regresyonun bir uzantısıdır. Bu tezde scikit-learn Python makine öğrenimi kitaplığını kullanarak çok terimli (multinomial) bir

lojistik regresyon modeli geliştirilip değerlendirilmiştir. Lojistik Regresyon sınıfı kullanılarak "multi_class" parametresini "multinomial" olarak ve "solver" parametresini "lbfgs" olarak çok terimli lojistik regresyonu destekleyen bir çözücüye ayarlayıp çok terimli lojistik regresyon için yapılandırılmıştır.

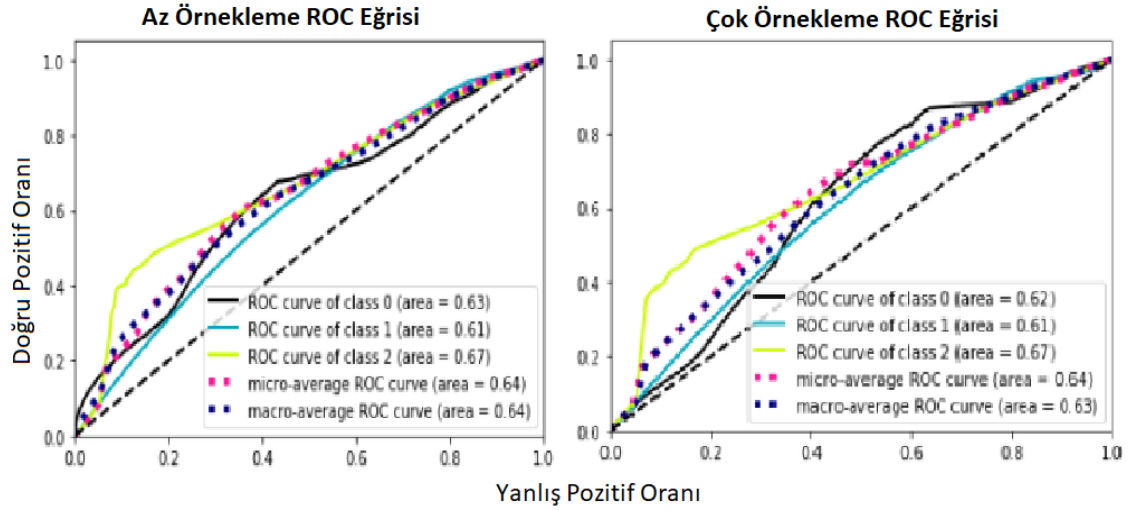
Veri madenciliği algoritmalarından çok terimli Lojistik Regresyon araç muayene verilerin tümüne (7.392.283 adet) uygulandığında muayene sonucunu %73.69 oranında doğru bildiği tespit edilmiştir. Araç muayene verileri üzerinde az örnekleme yapılarak elde edilen verilere (264.642 adet) Lojistik Regresyon uygulandığında muayene sonucunu %48.74 oranında doğru bildiği tespit edilmiştir. Ayrıca çok örnekleme uygulayarak elde edilen veriler (161.90.493 adet) Lojistik Regresyon ile araç muayene sonucu %47.41 oranında doğru bildiği tespit edilmiştir. Tüm araç muayene verilerinin, az örnekleme ve çok örnekleme yapılarak Lojistik Regresyon ile elde edilen karışıklık matrisi şekil 4.1’de gösterilmektedir.



Şekil 4.1 Lojistik Regresyon karışıklık matrisi



Şekil 4.2 Lojistik Regresyon Tüm Veri ROC Eğrisi grafiği



Şekil 4.3 Lojistik Regresyon Az ve Çok Örneklem ROC Eğrisi grafiği

Şekil 4.2’de ve şekil 4.3’de tüm araç muayene verisi, az örneklem ve çok örneklem ile elde edilen verilerin Lojistik Regresyon sonucu ROC Eğrisi grafiği verilmektedir. Lojistik Regresyon ROC Eğrisi grafiğine göre AUC değerleri düşük olması nedeniyle sınıfları tahmin edebilme yeteneği iyi olduğu söylenemez. Tüm veri ile elde edilen ROC eğrisinde Hafif kusurlu (0) sınıfı için AUC değeri 0.57, Ağır kusurlu (1) sınıfı için AUC değeri 0.57 ve Emniyetsiz (2) sınıfı için AUC değeri 0.60’dır. Az örneklem ile elde edilen ROC eğrisinde Hafif kusurlu (0) sınıfı için AUC değeri 0.63, Ağır kusurlu (1) sınıfı için AUC değeri 0.61 ve Emniyetsiz (2) sınıfı için AUC değeri 0.67’dir. Çok örneklem ile elde edilen ROC eğrisinde Hafif kusurlu (0) sınıfı için AUC değeri 0.62, Ağır kusurlu (1) sınıfı için AUC değeri 0.61 ve Emniyetsiz (2) sınıfı için AUC değeri 0.67’dir. Az örneklem ve çok örneklem yapılarak elde edilen ROC eğrisinde AUC değerleri hemen hemen aynıdır ve tüm veri ile elde edilen AUC değerlerinden daha yüksektir. Az örneklem ve çok örneklem yapılarak Lojistik Regresyon modelinin sınıfları tahminleme oranı artmıştır.

Çizelge 4.1 Lojistik Regresyon sınıflandırma raporu

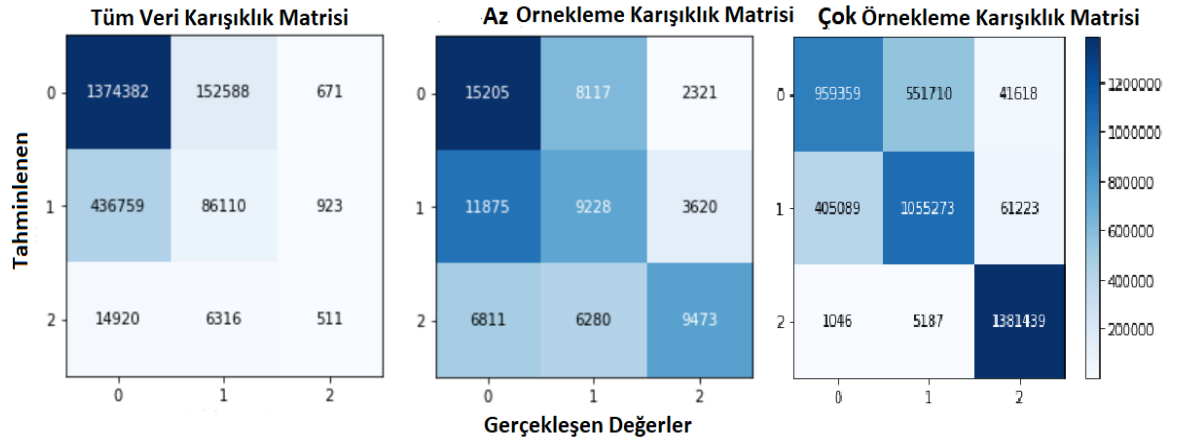
Sınıf	Kesinlik			Duyarlılık			F1-Skor			Destek		
	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ
0 (HK)	0.74	0.51	0.46	1.00	0.44	0.51	0.85	0.47	0.48	1527641	25643	1552813
1 (AK)	0.00	0.42	0.42	0.00	0.54	0.44	0.00	0.48	0.43	523792	24723	1520519
2 (EM)	0.00	0.56	0.57	0.00	0.48	0.47	0.00	0.52	0.52	21747	22564	1388509
Avg/ Total	0.54	0.50	0.48	0.74	0.49	0.47	0.63	0.49	0.48	2073180	72930	4461841

*TV: Tüm Veri, AÖ: Az Örneklem, ÇÖ: Çok Örneklem

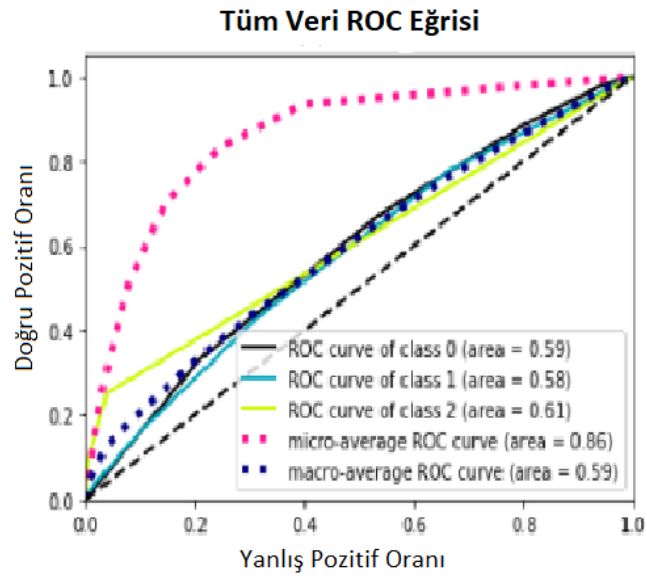
Çizelge 4.1’de Lojistik Regresyonun sınıflandırma raporu sonuçları verilmiştir. Tüm veri ile elde edilen sonuçlarda Hafif Kusurlu sınıfı için kesinlik değeri 0.74, duyarlılık değeri 1.00 ve F1 skoru değeri 0.85 olarak çıkmıştır. Ağır kusurlu ve Emniyetsiz sınıfları için duyarlılık, kesinlik ve F1 skor değeri 0 olarak çıkmıştır. Az örnekleme ile elde edilen sonuçlarda Hafif kusurlu sınıfı için kesinlik değeri 0.51, duyarlılık değeri 0.44 ve F1 skoru değeri 0.47 olarak çıkmıştır. Ağır kusurlu sınıfı için kesinlik değeri 0.42, duyarlılık değeri 0.54 ve F1 skoru değeri 0.48 olarak çıkmıştır. Emniyetsiz sınıfı için kesinlik değeri 0.56, duyarlılık değeri 0.48 ve F1 skoru değeri 0.52 olarak çıkmıştır. Çok örnekleme ile elde edilen sonuçlar ile az örnekleme ile elde edilen sonuçlar hemen hemen aynıdır.

Hem sınıflandırma raporu ve hem de karmaşıklık matrisine göre lojistik regresyon tüm veride Ağır Kusurlu ve Emniyetsiz sınıflarını tahmin etmede başarısızdır. Tüm veri ile elde edilen sonuçlarda Ağır Kusurlu ve Emniyetsiz sınıflarının hepsini hafif kusurlu olarak sınıflandırmıştır. Az örnekleme ve çok örnekleme yöntemleri ile elde edilen verilerde ise lojistik regresyon sınıfları tahmin etmede başarısı hemen hemen aynıdır.

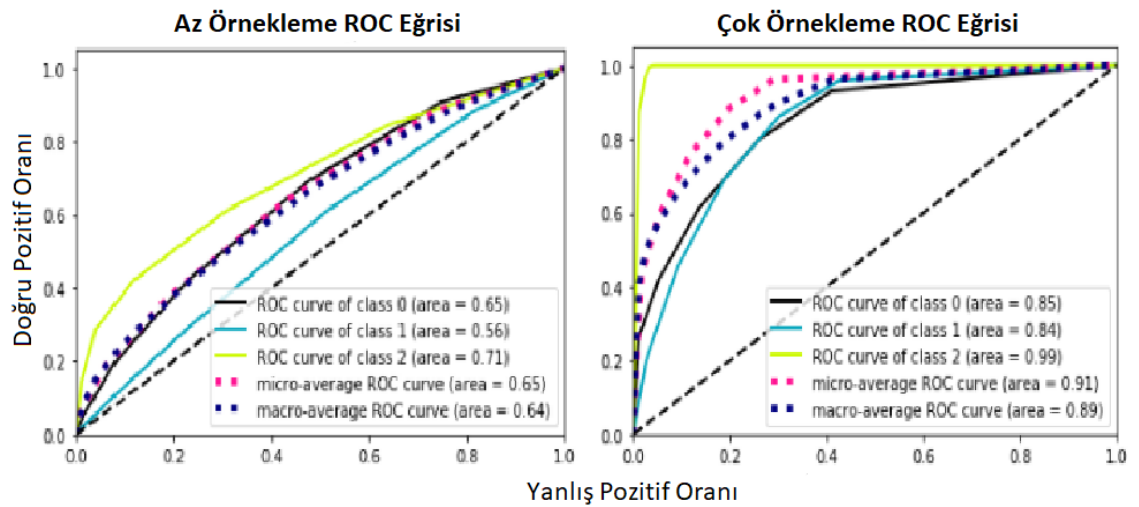
K-En Yakın Komşu (KNN): KNN algoritması, uygulaması kolay olması, gürültülü verilere karşı dirençli olması ve eğitim aşamasına sahip olmaması sebebiyle kullanılmıştır. Fakat büyük verilerde kullanıldığında fazla bellek alanına gereksinim duyması sebebiyle çalışması uzun sürmektedir. Makine öğrenmesi algoritmalarından KNN sınıflandırıcısında en yakın komşu sayısı olan k parametresi 5 olarak belirlenip araç muayene verilerine uygulanmıştır. Tüm veriler ile araç muayene sonucunu %70.47 oranında doğru bildiği tespit edilmiştir. Az örnekleme yapılarak elde edilen verilere KNN uygulandığında muayene sonucunu %46.49 oranında doğru bildiği tespit edilmiştir. Çok örnekleme uygulayarak muayene sonucunu %76.11 oranında doğru bildiği tespit edilmiştir. Tüm verilerin, az örnekleme ve çok örnekleme yapılarak KNN ile elde edilen karmaşıklık matrisi şekil 4.4’de gösterilmektedir.



Şekil 4.4 KNN karışıklık matrisi



Şekil 4.5 KNN Tüm Veri ROC Eğrisi grafiği



Şekil 4.6 KNN Az ve Çok Örneklem ROC Eğrisi grafiği

Şekil 4.5’de ve şekil 4.6’da verilen ROC Eğrisine göre tüm veride ve az örnekleme yapılarak elde edilen veride AUC değerleri düşük olduğu için KNN algoritmasının sınıfları tahmin edebilme yeteneği iyi olduğu söylenemez. Tüm veri ROC eğrisinde Hafif kusurlu (0) sınıfı için AUC değeri 0.59, Ağır kusurlu (1) sınıfı için AUC değeri 0.58 ve Emniyetsiz (2) sınıfı için AUC değeri 0.61’ dir. Az örnekleme yapılarak elde edilen ROC eğrisinde ise Hafif kusurlu (0) sınıfı için AUC değeri 0.65, Ağır kusurlu (1) sınıfı için AUC değeri 0.56 ve Emniyetsiz (2) sınıfı için AUC değeri 0.71’dir. Çok örnekleme yapılarak elde edilen ROC eğrisinde ise Hafif kusurlu (0) sınıfı için AUC değeri 0.85, Ağır kusurlu (1) sınıfı için AUC değeri 0.84 ve Emniyetsiz (2) sınıfı için AUC değeri 0.99’dur. KNN algoritması ile en yüksek AUC değerleri çok örnekleme yapılan veriden elde edilmiştir.

Çizelge 4.2 KNN sınıflandırma raporu

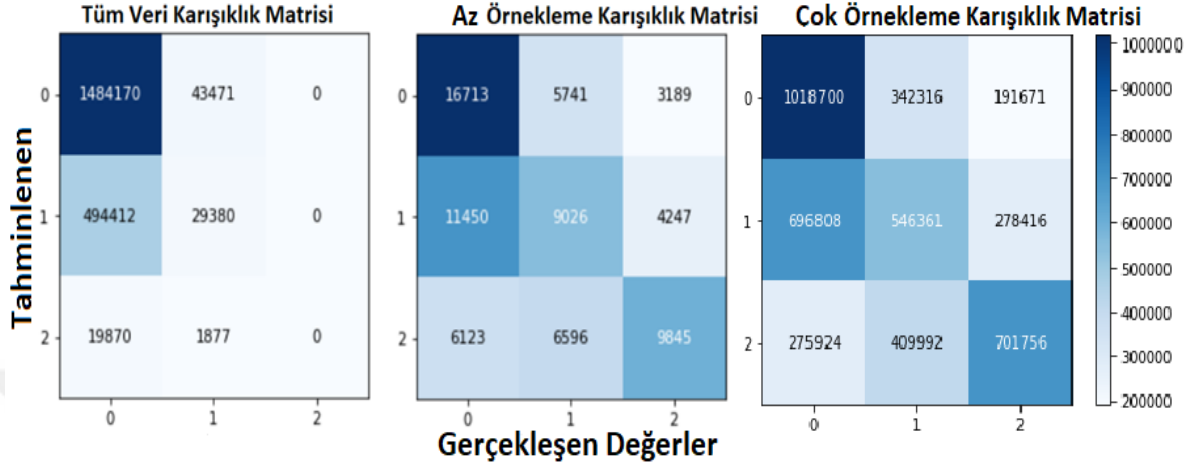
Sınıf	Kesinlik			Duyarlılık			F1-Skor			Destek		
	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ
0 (HK)	0.75	0.45	0.51	0.87	0.59	0.66	0.81	0.51	0.57	1527641	25643	1552687
1 (AK)	0.33	0.39	0.42	0.20	0.37	0.36	0.25	0.38	0.39	523792	24723	1521585
2 (EM)	0.20	0.61	0.60	0.04	0.42	0.51	0.07	0.50	0.55	21747	22564	1387672
Avg/ Total	0.64	0.48	0.51	0.69	0.46	0.51	0.66	0.46	0.50	2073180	72930	4461944

*TV: Tüm Veri, YÖ: Az Örnekleme, AÖ: Çok Örnekleme

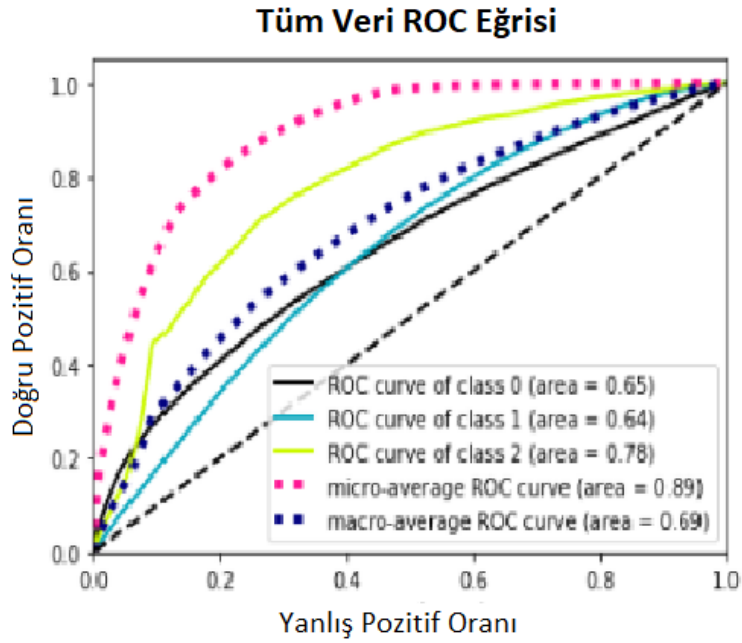
Çizelge 4.2’de KNN algoritması sınıflandırma raporu sonuçları verilmektedir. Tüm veri ile elde edilen sonuçlarda Hafif Kusurlu sınıfı için kesinlik değeri 0.75, duyarlılık değeri 0.87 ve F1 skoru değeri 0.81 olarak çıkmıştır. Az örnekleme ve çok örnekleme ile elde edilen sonuçlar tüm veri ile elde edilen sonuçlara göre Hafif kusurlu sınıfı için daha düşük çıkmıştır fakat Ağır kusurlu ve Emniyetsiz sınıfı için daha yüksek çıkmıştır. Emniyetsiz sınıfını tahmin edebilme başarısı en yüksek değer çok örnekleme yapılan veriden elde edilmiştir. Tüm veride Ağır kusurlu ve Emniyetsiz sınıfları için duyarlılık, kesinlik ve F1 skor değerleri Hafif Kusurlu sınıfına göre daha düşük çıkmıştır.

Naive Bayes: Naive Bayes algoritması yüksek boyutlu verilerde iyi performans göstermesi ve her özellik birbirinden bağımsız kabul edildiği için lojistik regresyon gibi modellerden daha iyi performans göstermesi sebebinden dolayı tercih edilmiştir. Naive Bayes araç muayene verilerine uygulandığında muayene sonucunu tüm veri için %73.01 oranında doğru bildiği tespit edilmiştir. Az örnekleme yapılarak elde edilen veride muayene son

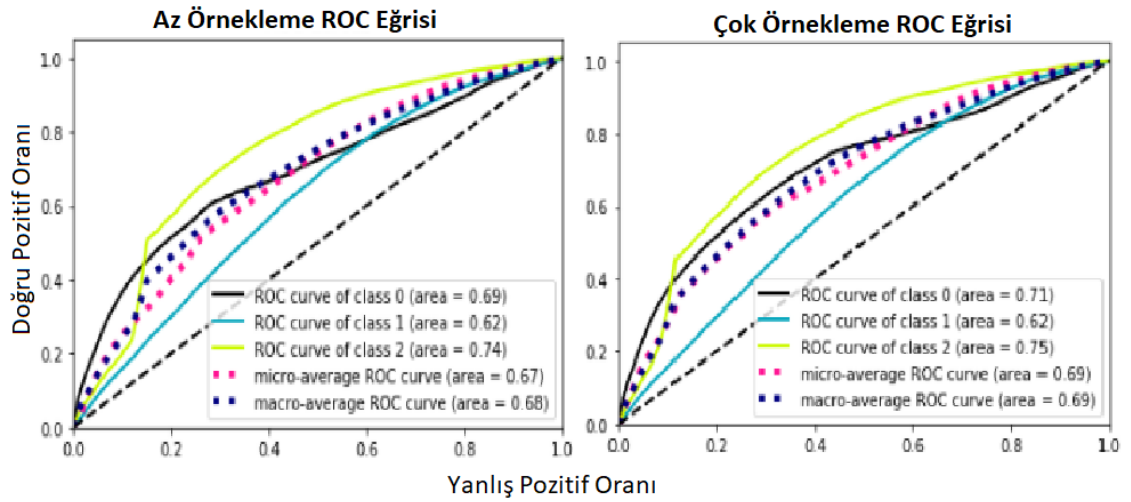
ucunu %48.79, çok örnekleme yapılarak elde edilen veride ise %76.11 oranında doğru b
ildiği tespit edilmiştir. Tüm veride, az örnekleme ve çok örnekleme yapılarak elde edilen
veride Naive Bayes algoritması sonucu çıkan karışıklık matrisi şekil 4.7’de verilmiştir.



Şekil 4.7 Naive Bayes karışıklık matrisi



Şekil 4.8 Naive Bayes Tüm Veri ROC Eğrisi grafiği



Şekil 4.9 Naive Bayes Az ve Çok Örneklem ROC Eğrisi grafiği

Şekil 4.8’de ve şekil 4.9’da verilen ROC Eğrisi grafiğinde Naive Bayes algoritmasının sınıfları tahmin edebilme başarısı tüm veride ve az örneklem ile elde edilen verilerde Lojistik Regresyon ve KNN algoritmasına göre daha iyi olduğu söylenebilir. Tüm veri ROC eğrisinde Hafif kusurlu (0) sınıfı için AUC değeri 0.65, Ağır kusurlu (1) sınıfı için AUC değeri 0.64 ve Emniyetsiz (2) sınıfı için AUC değeri 0.78’dir. Az örneklem yapılarak elde edilen ROC eğrisinde ise Hafif kusurlu (0) sınıfı için AUC değeri 0.69, Ağır kusurlu (1) sınıfı için AUC değeri 0.62 ve Emniyetsiz (2) sınıfı için AUC değeri 0.74’dür. Çok örneklem yapılarak elde edilen ROC eğrisinde ise Hafif kusurlu (0) sınıfı için AUC değeri 0.71, Ağır kusurlu (1) sınıfı için AUC değeri 0.62 ve Emniyetsiz (2) sınıfı için AUC değeri 0.75’dir. Çok örneklem yapılarak elde edilen AUC değerleri tüm veri ve az örnekleme göre daha yüksek çıkmıştır.

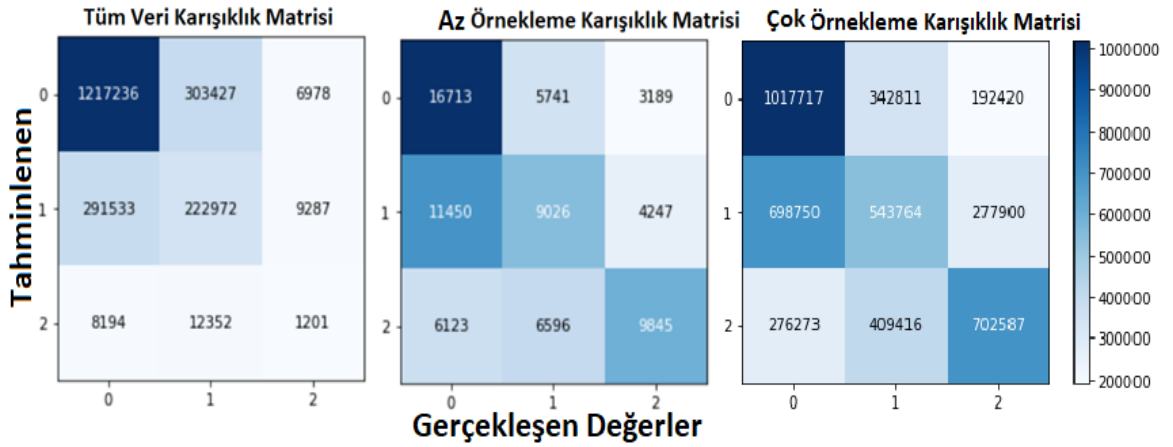
Çizelge 4.3 Naive Bayes sınıflandırma raporu

Sınıf	Kesinlik			Duyarlılık			F1-Skor			Destek		
	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ
0 (HK)	0.74	0.49	0.51	0.97	0.65	0.66	0.84	0.56	0.57	1527641	25643	1552687
1 (AK)	0.39	0.42	0.42	0.06	0.37	0.36	0.10	0.39	0.39	523792	24723	1521585
2 (EM)	0.00	0.57	0.60	0.00	0.44	0.51	0.00	0.49	0.55	21747	22564	1387672
Avg/ Total	0.65	0.49	0.51	0.73	0.49	0.51	0.65	0.48	0.50	2073180	72930	4461944

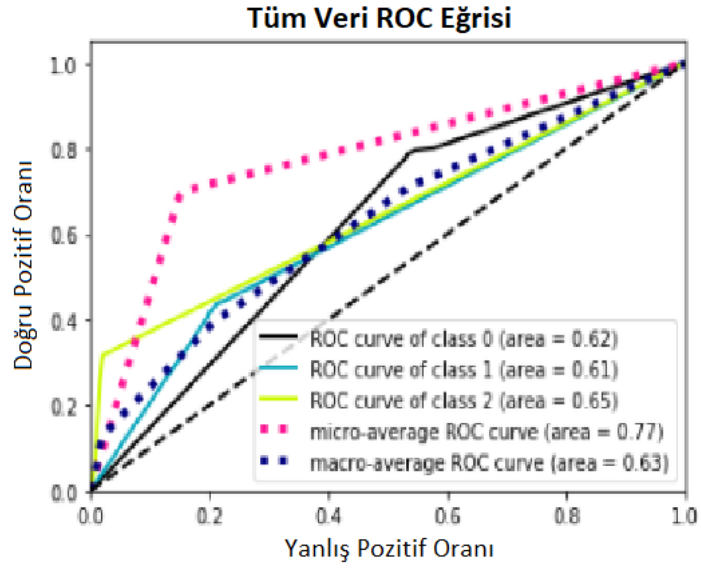
*TV: Tüm Veri, YÖ: Az Örneklem, AÖ: Çok Örneklem

Çizelge 4.3'te Naive Bayes algoritmasının sınıflandırma raporu sonuçları gösterilmektedir. Tüm veri ile elde edilen sonuçlarda Hafif Kusurlu sınıfı için kesinlik değeri 0.74, duyarlılık değeri 0.97 ve F1 skoru değeri 0.84 olarak çıkmıştır. Tüm veride Naive Bayes algoritması karmaşıklık matrisine ve sınıflandırma raporuna göre emniyetsiz sınıfını hiç doğru tahmin edememiştir. Fakat az örnekleme ve çok örnekleme ile elde edilen sonuçlar da emniyetsiz sınıfını doğru tahmin edebilme başarısı daha fazladır. Az örnekleme ve çok örnekleme ile elde edilen sonuçlar tüm veri ile elde edilen sonuçlara göre Hafif kusurlu sınıfı için daha düşük diğer sınıflar için daha yüksek çıkmıştır.

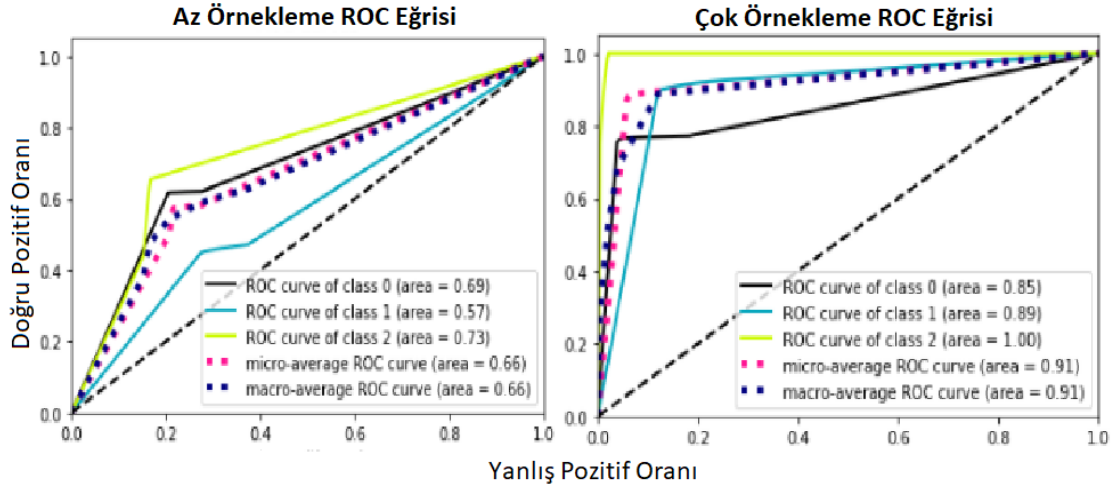
Karar Ağacı (Decision Tree): Karar Ağaçları algoritması hem sayısal hemde kategorik verileri işleyebilmesi ve çok sınıflı problemleri ele alabilmesinden dolayı tercih edilmiştir. Karar Ağacı araç muayene verilerine uygulandığında muayene sonucunu %69.53 oranında doğru bildiği tespit edilmiştir. Az örnekleme yapılarak elde edilen verilere uygulandığında muayene sonucunu %48.79, çok örnekleme yapılarak elde edilen verilerde ise %50.75 oranında doğru bildiği tespit edilmiştir. Tüm verilerin, az örnekleme ve çok örnekleme yapılarak elde edilen verilerin Karar Ağaçları algoritması sonucu çıkan karışıklık matrisi şekil 4.10'da verilmiştir.



Şekil 4.10 Karar Ağacı karışıklık matrisi



Şekil 4.11 Karar Ağacı Tüm Veri ROC Eğrisi grafiği



Şekil 4.12 Karar Ağacı Az ve Çok Örneklemeye ROC Eğrisi grafiği

Şekil 4.11’de ve şekil 4.12’de Karar Ağacı algoritmasına ait ROC eğrisi grafiği verilmiştir. Tüm veri için ROC eğrisinde Hafif kusurlu (0) sınıfı için AUC değeri 0.62, Ağır kusurlu (1) sınıfı için AUC değeri 0.61 ve Emniyetsiz (2) sınıfı için AUC değeri 0.65 dir. Az örnekleme yapılarak elde edilen ROC Eğrisinde ise Hafif kusurlu (0) sınıfı için AUC değeri 0.69, Ağır kusurlu (1) sınıfı için AUC değeri 0.57 ve Emniyetsiz (2) sınıfı için AUC değeri 0.73’dür. Çok örnekleme yapılarak elde edilen ROC Eğrisinde ise Hafif kusurlu (0) sınıfı için AUC değeri 0.85, Ağır kusurlu (1) sınıfı için AUC değeri 0.89 ve Emniyetsiz (2) sınıfı için AUC değeri 1.00’dır. Karar Ağacı algoritması için en yüksek AUC değerleri çok örnekleme yapılan veriden elde edilmiştir.

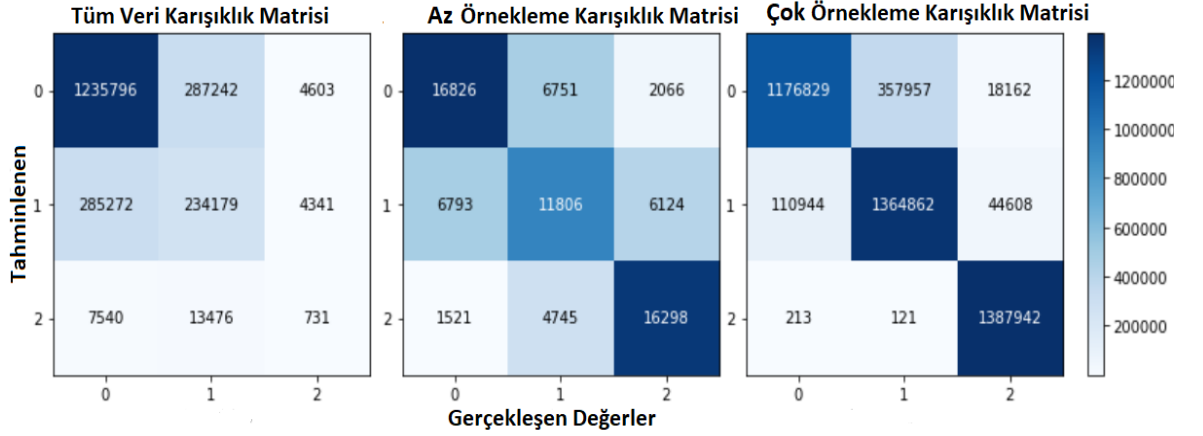
Çizelge 4.4 Karar Ağacı sınıflandırma raporu

Sınıf	Kesinlik			Duyarlılık			F1-Skor			Destek		
	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ
0 (HK)	0.80	0.49	0.51	0.80	0.65	0.66	0.80	0.56	0.57	1527641	25643	1552948
1 (AK)	0.41	0.42	0.42	0.43	0.37	0.36	0.42	0.39	0.39	523792	24723	1520414
2 (EM)	0.07	0.57	0.60	0.06	0.44	0.51	0.06	0.49	0.55	21747	22564	1388276
Avg/ Total	0.70	0.49	0.51	0.70	0.49	0.51	0.70	0.48	0.50	2073180	72930	4461638

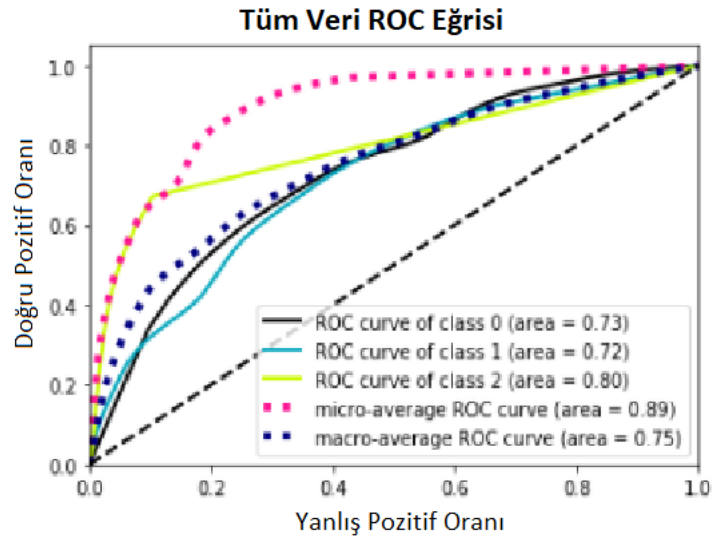
*TV: Tüm Veri, YÖ: Az Örnekleme, AÖ: Çok Örnekleme

Çizelge 4.4’de Karar Ağacı algoritmasının sınıflandırma raporu sonuçları verilmiştir. Tüm veri ile elde edilen sonuçlara göre Hafif kusurlu sınıfı için kesinlik, duyarlılık ve F1 skoru değeri 0.80 olarak çıkmıştır. Ağır kusurlu sınıfı için kesinlik değeri 0.41, duyarlılık değeri 0.43 ve F1 skoru değeri 0.42 olarak çıkmıştır. Emniyetsiz sınıfı için ise kesinlik değeri 0.07, duyarlılık değeri 0.06 ve F1 skoru değeri 0.06 olarak çıkmıştır. Az örnekleme ile elde edilen sonuçlar tüm veri ile elde edilen sonuçlara göre Hafif kusurlu sınıfı için daha düşük çıkmıştır. Fakat Emniyetsiz sınıfında tüm veri ile elde edilen sonuçlardan daha yüksek çıkmıştır.

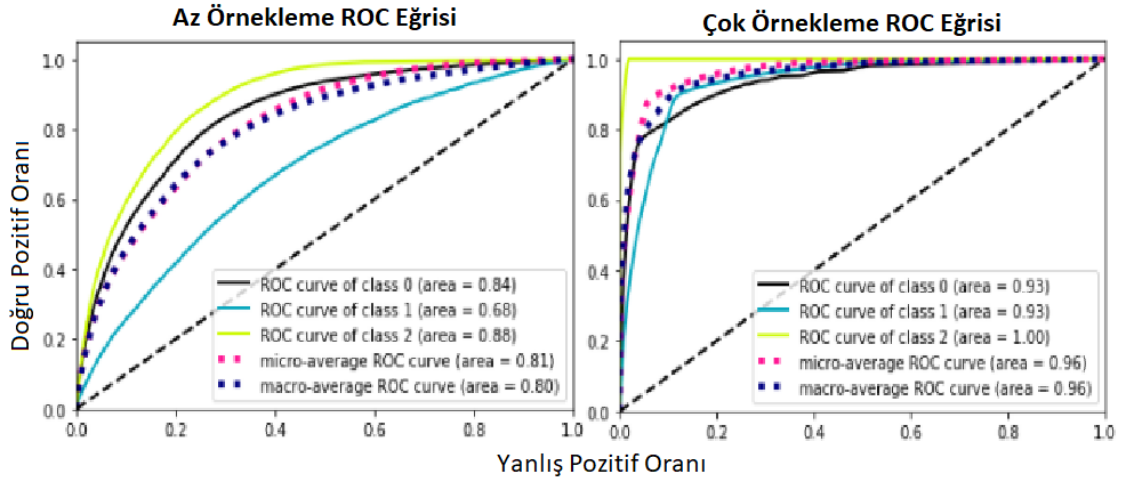
Rastgele Orman (Random Forest) : Random Forest algoritması aşırı uyum (overfitting) problemini önleyebilmesi sebebiyle kullanılmıştır. Rastgele Orman araç muayene verilerine uygulandığında muayene sonucunu %70.94 oranında doğru bildiği tespit edilmiştir. Az örnekleme yapılarak elde edilen verilere Random Forest uygulandığında muayene sonucunu %61.61, çok örnekleme yapılarak elde edilen verilerde ise %88.08 oranında doğru bildiği tespit edilmiştir. Tüm verilerin, az örnekleme ve çok örnekleme yapılarak elde edilen verilerin Rastgele Orman algoritması sonucu çıkan karışıklık matrisi şekil 4.13’de gösterilmektedir.



Şekil 4.13 Rastgele Orman karışıklık matrisi



Şekil 4.14 Rastgele Orman Tüm Veri ROC Eğrisi grafiği



Şekil 4.15 Rastgele Orman Az ve Çok Örneklem ROC Eğrisi grafiği

Şekil 4.14’de ve şekil 4.15’de verilen ROC eğrisi grafiğine göre Rastgele Orman algoritmasının sınıfları tahmin edebilme yeteneği Karar Ağaçları algoritmasına göre daha iyi o

lduğu söylenebilir. Hafif kusurlu (0) sınıfı için AUC değeri 0.73, Ağır kusurlu (1) sınıfı için AUC değeri 0.72 ve Emniyetsiz (2) sınıfı için AUC değeri 0.80 dır. Az örnekleme yapılarak elde edilen ROC eğrisinde ise Hafif kusurlu (0) sınıfı için AUC değeri 0.84, Ağır kusurlu (1) sınıfı için AUC değeri 0.68 ve Emniyetsiz (2) sınıfı için AUC değeri 0.88’dir. Çok örnekleme yapılarak elde edilen ROC eğrisinde ise Hafif kusurlu (0) sınıfı için AUC değeri 0.93, Ağır kusurlu (1) sınıfı için AUC değeri 0.93 ve Emniyetsiz (2) sınıfı için AUC değeri 1.00’dır. Çok örnekleme ile elde edilen sonuçlar tüm veri ve az örnekleme ile elde edilen sonuçlardan daha yüksek çıkmıştır.

Çizelge 4.5 Rastgele Orman sınıflandırma raporu

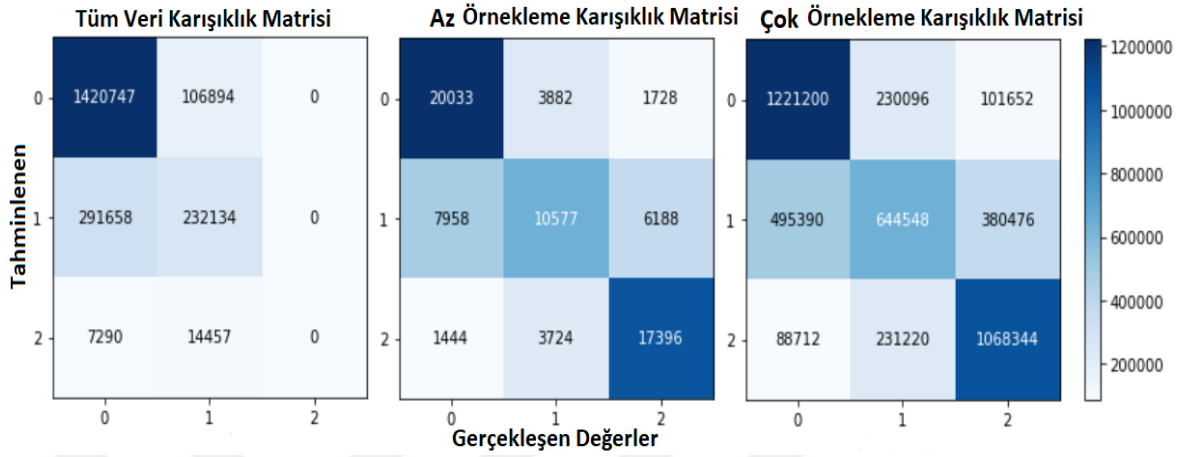
Sınıf	Kesinlik			Duyarlılık			F1-Skor			Destek		
	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ
0 (HK)	0.81	0.67	0.91	0.81	0.66	0.76	0.81	0.66	0.83	1527641	25643	1552948
1 (AK)	0.44	0.51	0.79	0.45	0.48	0.90	0.44	0.49	0.84	523792	24723	1520414
2 (EM)	0.08	0.67	0.96	0.03	0.72	1.00	0.05	0.69	0.98	21747	22564	1388276
Avg/ Total	0.71	0.61	0.89	0.71	0.62	0.88	0.71	0.61	0.88	2073180	72930	4461638

*TV: Tüm Veri, YÖ: Az Örnekleme, AÖ: Çok Örnekleme

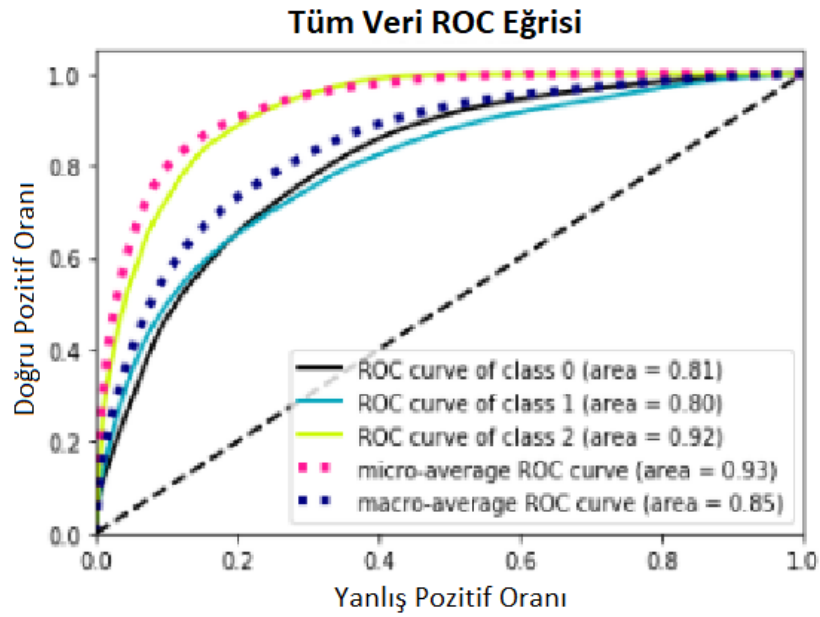
Çizelge 4.5’de Rastgele Orman algoritmasının sınıflandırma raporu sonuçları verilmiştir. Tüm veride Hafif Kusurlu sınıfı için kesinlik, duyarlılık ve F1 skoru değeri 0.81 olarak çıkmıştır. Ağır kusurlu sınıfı için kesinlik değeri 0.44, duyarlılık değeri 0.45 ve F1 skoru değeri 0.44 olarak çıkmıştır. Emniyetsiz sınıfı için ise kesinlik değeri 0.08, duyarlılık değeri 0.03 ve F1 skoru değeri 0.05 olarak çıkmıştır. Rastgele Orman algoritmasının sınıflandırma raporundaki kesinlik, duyarlılık ve F1 skor değerleri Karar Ağaçları algoritmasına göre daha iyi sonuçlar çıkmıştır. Çok örnekleme ile elde edilen sonuçlar tüm veri ve az örnekleme ile elde edilen sonuçlara göre daha iyi çıkmıştır. Ayrıca çok örneklemenin emniyetsiz sınıflarını tahmin etmedeki başarısı diğerlerine göre daha iyidir.

XG Boost: XG Boost algoritması yüksek tahmin gücü elde edebilmesi, aşırı öğrenmeye karşı dirençli olması, karar ağacı tabanlı algoritmaların en iyisi olması ve hızlı çalışması sebebiyle tercih edilmiştir. XG Boost tüm araç muayene verilerine uygulandığında muayene sonucunu %79.73 oranında doğru bildiği tespit edilmiştir. Az örnekleme yapılarak elde edilen verilere XG Boost uygulandığında muayene sonucunu %65.82, çok

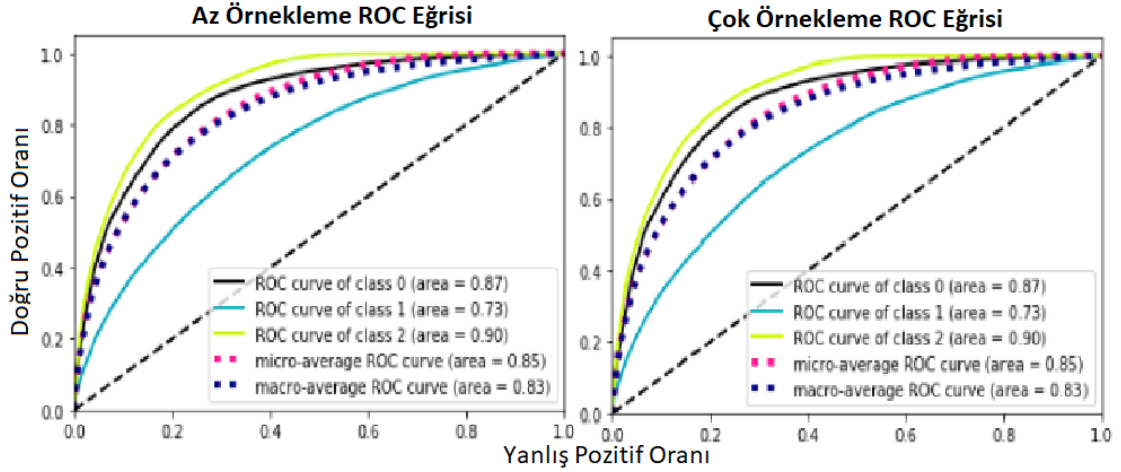
örnekleme yapılarak elde edilen verilerde ise %65.76 oranında doğru bildiği tespit edilmiştir. Tüm verilerin, az örnekleme ve çok örnekleme yapılarak elde edilen verilerin XG Boost algoritması sonucu çıkan karışıklık matrisi şekil 4.16’da gösterilmektedir.



Şekil 4.16 XG Boost karışıklık matrisi



Şekil 4.17 XG Boost Tüm Veri ROC Eğrisi grafiğı



Şekil 4.18 XG Boost Az ve Çok Örneklem ROC Eğrisi grafiği

Şekil 4.17’de ve şekil 4.18’de verilen ROC Eğrisi grafiğine göre XG Boost algoritması sınıfları tahmin edebilme yeteneği diğer algoritmalara göre daha iyi olduğu ortaya çıkmıştır. AUC değerleri diğer algoritmalara göre daha yüksektir. Tüm veri ile elde edilen sonuçlara göre Hafif kusurlu (0) sınıfı için AUC değeri 0.81, Ağır kusurlu (1) sınıfı için AUC değeri 0.80 ve Emniyetsiz (2) sınıfı için AUC değeri 0.92’dir. Az örneklem yapılarak elde edilen ROC Eğrisinde ise Hafif kusurlu (0) sınıfı için AUC değeri 0.87, Ağır kusurlu (1) sınıfı için AUC değeri 0.73 ve Emniyetsiz (2) sınıfı için AUC değeri 0.90’dır. Çok örneklem yapılarak elde edilen ROC Eğrisinde ise Hafif kusurlu (0) sınıfı için AUC değeri 0.87, Ağır kusurlu (1) sınıfı için AUC değeri 0.73 ve Emniyetsiz (2) sınıfı için AUC değeri 0.90’dır. Çok örneklem ve az örneklem ile elde edilen AUC değerleri aynı çıkmıştır. Çok örneklem ve az örneklem AUC değerleri tüm veri ile elde edilen AUC değerlerinden daha yüksek çıkmıştır.

Çizelge 4.6 XG Boost sınıflandırma raporu

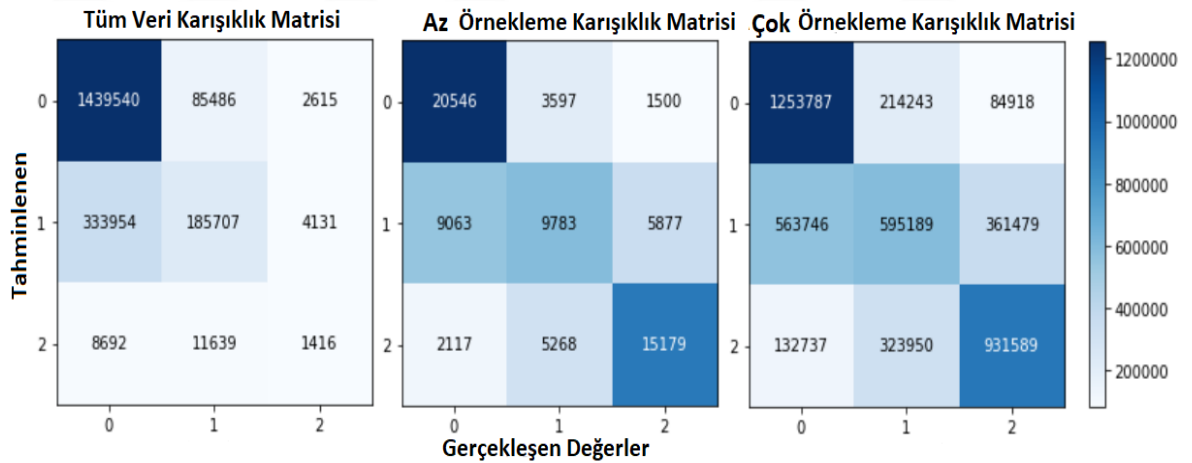
Sınıf	Kesinlik			Duyarlılık			F1-Skor			Destek		
	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ
0 (HK)	0.83	0.68	0.68	0.93	0.78	0.79	0.88	0.73	0.73	1527641	25643	1552948
1 (AK)	0.66	0.58	0.58	0.44	0.43	0.42	0.53	0.49	0.49	523792	24723	1520414
2 (EM)	0.00	0.69	0.69	0.00	0.77	0.77	0.00	0.73	0.73	21747	22564	1388276
Avg/ Total	0.77	0.65	0.51	0.80	0.66	0.66	0.78	0.65	0.65	2073180	72930	4461638

*TV: Tüm Veri, YÖ: Az Örneklem, AÖ: Çok Örneklem

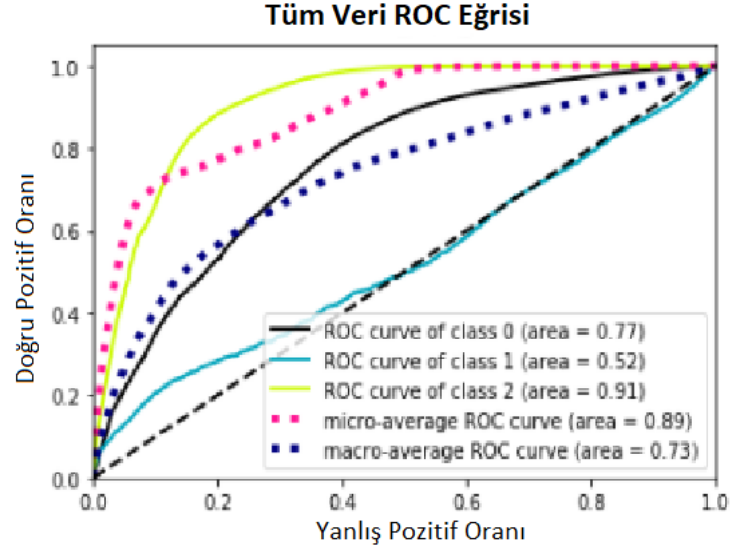
Çizelge 4.6’da XG Boost algoritmasının sınıflandırma raporu sonuçları verilmiştir. Hafif kusurlu sınıfı için kesinlik değeri 0.83, duyarlılık değeri 0.93 ve F1 skoru değeri 0.88 ola

rak çıkmıştır. Ağır kusurlu sınıfı için kesinlik değeri 0.66, duyarlılık değeri 0.44 ve F1 s koru değeri 0.53 olarak çıkmıştır. Tüm veri ile elde edilen sonuçlara göre XG Boost algoritması karmaşıklık matrisi ve sınıflandırma raporundan da anlaşılacağı üzere Emniyetsiz sınıfına ait verileri hiç doğru tahmin edememiştir. Ama Hafif Kusurlu ve Ağır Kusurlu sınıflarını doğru tahmin etme başarısı diğer algoritmalara göre daha yüksek çıkmıştır. Az örnekleme ve çok örnekleme ile elde edilen sonuçlar aynı çıkmıştır. Ayrıca az örneklemenin ve çok örneklemenin emniyetsiz sınıfını tahmin etmedeki başarısı tüm veri ile elde edilen sonuçlara göre daha iyidir.

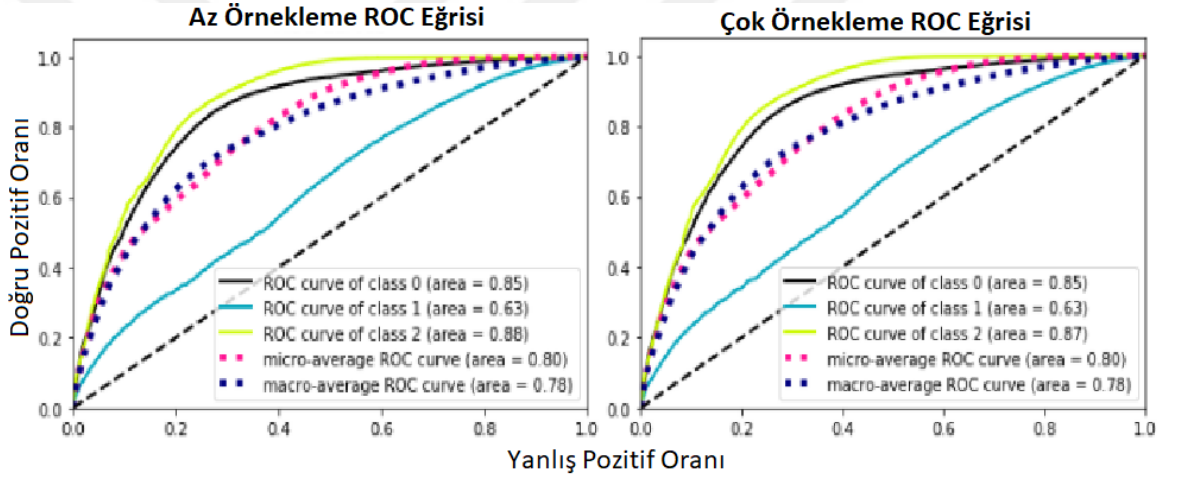
Ada Boost: Ada Boost algoritması aşırı öğrenmeye karşı dirençli olması ve yüksek doğruluk elde edebilmesi sebebiyle kullanılmıştır. Ada Boost araç muayene verilerine uygulandığında muayene sonucunu %78.46 oranında doğru bildiği tespit edilmiştir. Az örnekleme yapılarak elde edilen verilere Ada Boost algoritması uygulandığında muayene sonucunu %62.40, çok örnekleme yapılarak elde edilen verilerde ise %62.32 oranında doğru bildiği tespit edilmiştir. Tüm verilerin, az örnekleme ve çok örnekleme yapılarak elde edilen verilerin Ada Boost algoritması sonucu çıkan karışıklık matrisi şekil 4.19’da gösterilmektedir.



Şekil 4.19 Ada Boost karışıklık matrisi



Şekil 4.20 Ada Boost Tüm Veri ROC Eğrisi grafiği



Şekil 4.21 Ada Boost Az ve Çok Örneklemeye ROC Eğrisi grafiği

Şekil 4.20’de ve şekil 4.21’de Ada Boost algoritmasının ROC eğrisi grafiği verilmiştir. Tüm veri ile elde edilen sonuçlara göre Hafif kusurlu (0) sınıfı için AUC değeri 0.77, Ağır kusurlu (1) sınıfı için AUC değeri 0.52 ve Emniyetsiz (2) sınıfı için AUC değeri 0.91’dir. Az örneklemeye yapılarak elde edilen ROC eğrisinde ise Hafif kusurlu (0) sınıfı için AUC değeri 0.85, Ağır kusurlu (1) sınıfı için AUC değeri 0.63 ve Emniyetsiz (2) sınıfı için AUC değeri 0.88’dir. Çok örneklemeye yapılarak elde edilen ROC eğrisinde ise Hafif kusurlu (0) sınıfı için AUC değeri 0.85, Ağır kusurlu (1) sınıfı için AUC değeri 0.63 ve Emniyetsiz (2) sınıfı için AUC değeri 0.87’dir. Çok örneklemeye ve az örneklemeye ile elde edilen AUC değerleri aynı çıkmıştır. Çok örneklemeye ve az örneklemeye AUC değerleri tüm veri ile elde edilen AUC değerlerinden daha yüksek çıkmıştır.

Çizelge 4.7 Ada Boost sınıflandırma raporu

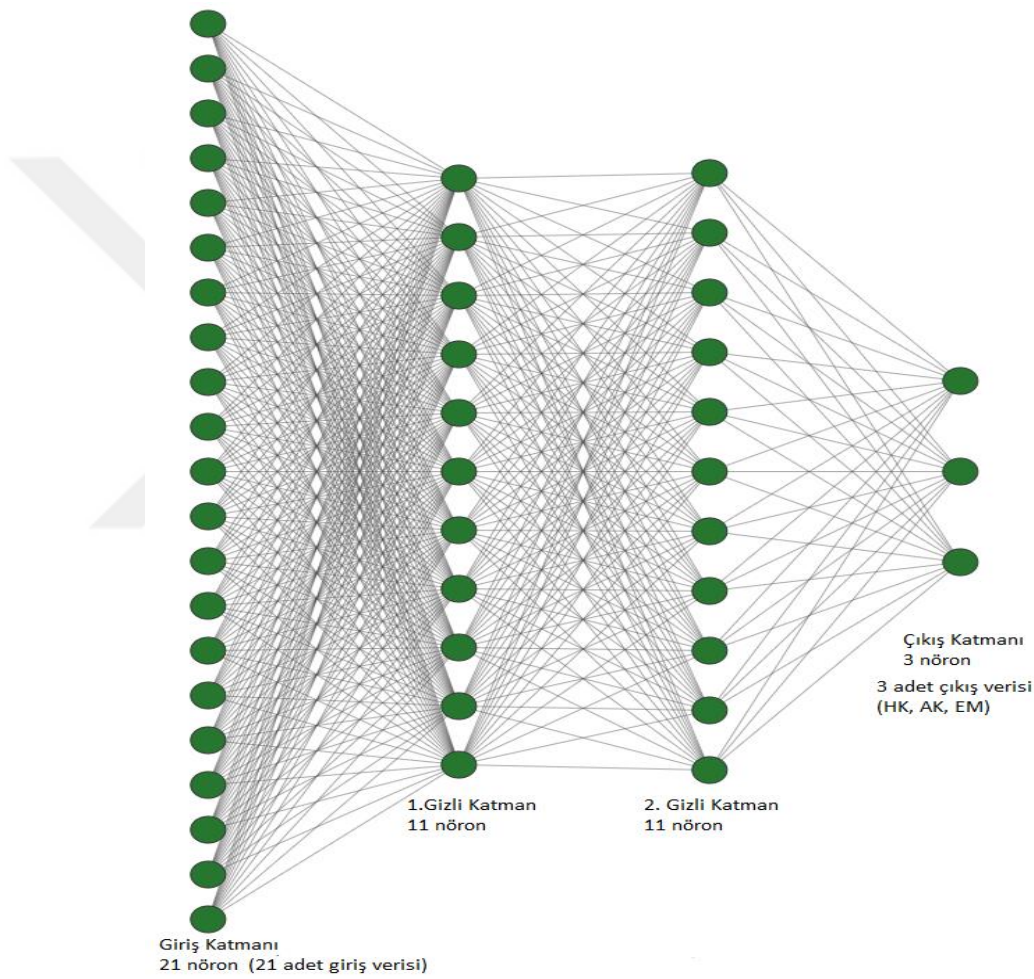
Sınıf	Kesinlik			Duyarlılık			F1-Skor			Destek		
	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ
0 (HK)	0.81	0.65	0.64	0.94	0.80	0.81	0.87	0.72	0.72	1527641	25643	1552948
1 (AK)	0.66	0.52	0.53	0.35	0.40	0.39	0.46	0.45	0.45	523792	24723	1520414
2 (EM)	0.17	0.67	0.68	0.07	0.67	0.67	0.09	0.67	0.67	21747	22564	1388276
Avg/ Total	0.76	0.61	0.61	0.78	0.62	0.62	0.76	0.61	0.61	2073180	72930	4461638

*TV: Tüm Veri, YÖ: Az Örnekleme, AÖ: Çok Örnekleme

Çizelge 4.7’de Ada Boost algoritmasının sınıflandırma raporu sonuçları verilmiştir. Tüm veri ile elde edilen sonuçlara göre Hafif Kusurlu sınıfı için kesinlik değeri 0.81, duyarlılık değeri 0.94 ve F1 skoru değeri 0.87 olarak çıkmıştır. Ağır kusurlu sınıfı için kesinlik değeri 0.66, duyarlılık değeri 0.35 ve F1 skoru değeri 0.46 olarak çıkmıştır. Emniyetsiz sınıfı için kesinlik değeri 0.17, duyarlılık değeri 0.07 ve F1 skoru değeri 0.09 olarak çıkmıştır. Ada Boost, XG Boost’un hiç doğru tahmin edemediği Emniyetsiz sınıfına ait verileri doğru tahmin edebilme kabiliyeti vardır. Az örnekleme ve çok örnekleme ile elde edilen sonuçlar tüm veri ile elde edilen sonuçlara göre daha iyi çıkmıştır. Veride çok örnekleme ve az örnekleme yapılarak emniyetsiz sınıfını doğru tahmin edebilme oranı artmıştır.

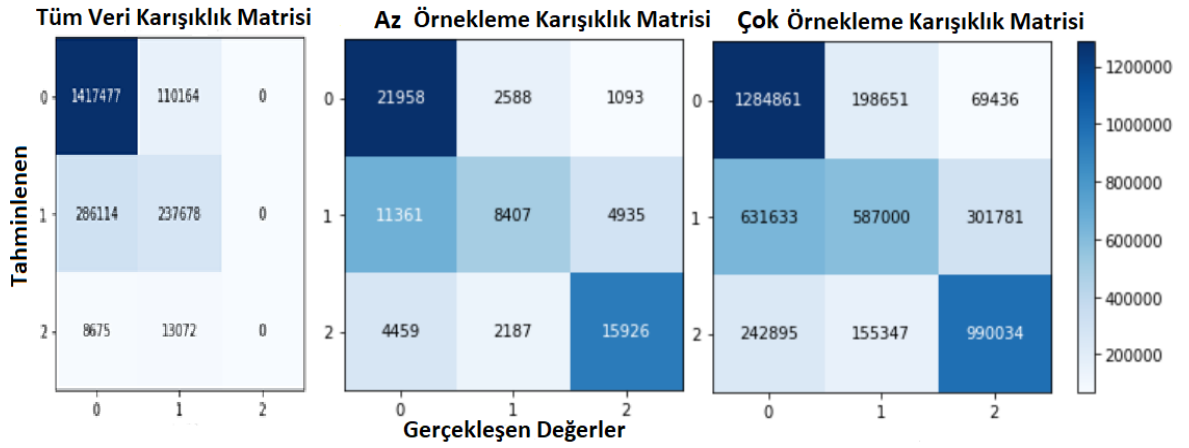
Derin Sinir Ağı (DNN) : Derin Sinir Ağlarının öğrenme ve genelleme kabiliyeti olması, eksik veya gürültülü veriyle çalışabilmesi, paralel çalışma prensibine sahip olması nedeniyle bu tezde kullanılmıştır. Bu çalışmada, derin öğrenme kütüphanesi olan keras yardımı ile sinir ağı oluşturulmuştur. 1 adet giriş katmanı, 2 adet gizli katman ve 1 adet çıkış katmanı olmak üzere toplam 4 katmanlı, tam bağlantılı bir ağ oluşturulmuştur. İlk iki katmanda ReLU aktivasyon fonksiyonu ve çıkış katmanında sigmoid aktivasyon fonksiyonu kullanılmıştır. Adam optimizasyon yöntemi kullanılmıştır. Adam yöntemi diğer yöntemlere kıyasla daha başarılı olduğu için kullanılmıştır. Adam optimizasyon yöntemi iyi işleyen bir yöntemdir ve öğrenme oranını ayarlama ihtiyacını ortadan kaldırmaktadır. Epok sayısı 5 olarak alınmıştır çünkü epok sayısını artırdıkça doğruluk ve kayıp değerinde çok fazla değişim olmamaktadır. Kayıp (loss) fonksiyonu olarak ise categorical crossentropy kullanılmıştır. Öğrenme oranı ise optimizasyon algoritmalarının varsayılan olarak aldığı değerler kullanılmıştır.

Ağ çıkış değerleri 0 (Hafif Kusurlu), 1 (Ağır Kusurlu) ve 2 (Emniyetsiz)'dir. Toplamda 7.304.069 veri ve 21 adet öznitelik kullanılmıştır. İlk katman, giriş değişkenleri için 21 adet nörona sahiptir. Gizli katmanlar 11 adet nöron dan ve çıkış katmanı 3 adet nörondan oluşmaktadır. İlk iki katmanda relu aktivasyon fonksiyonunu ve çıkış katmanı için softmax fonksiyonu kullanılmıştır. Ağ çıkışı 3 adet nörona sahip olduğu için softmax aktivasyon fonksiyonu kullanılmıştır. Şekil 4.22'de oluşturulan sinir ağı diyagramı gösterilmektedir.



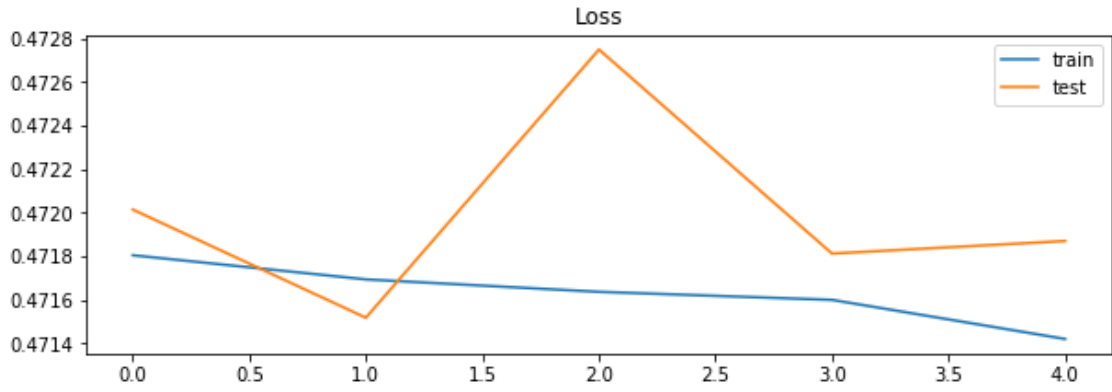
Şekil 4.22 Kullanılan Derin Sinir Ağı modeli

DNN modeli araç muayene verilerine uygulandığında muayene sonucunu %79.86 oranında doğru bildiği tespit edilmiştir. Az örnekleme yapılarak elde edilen verilere DNN uygulandığında muayene sonucunu %59.18, çok örnekleme yapılarak elde edilen verilerde ise %59.53 oranında doğru bildiği tespit edilmiştir. Tüm verilerin, az örnekleme ve çok örnekleme yapılarak elde edilen verilerin DNN sonucu çıkan karışıklık matrisi şekil 4.23'de verilmiştir.

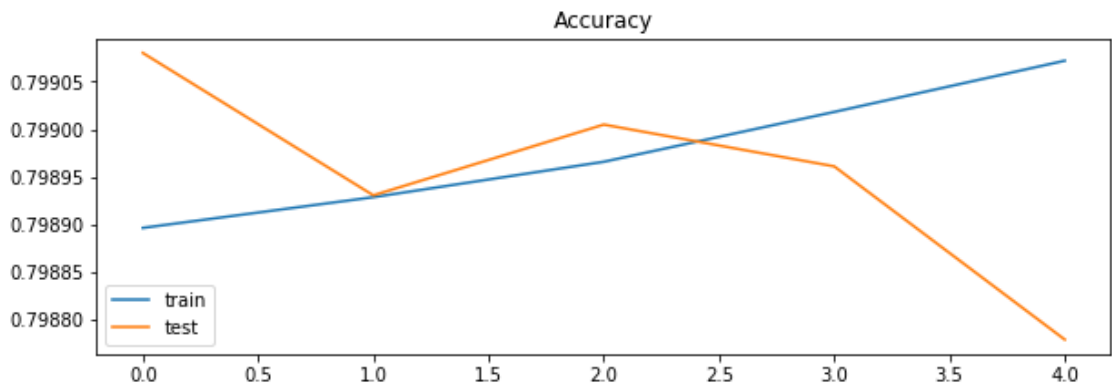


Şekil 4.23 DNN karışıklık matrisi

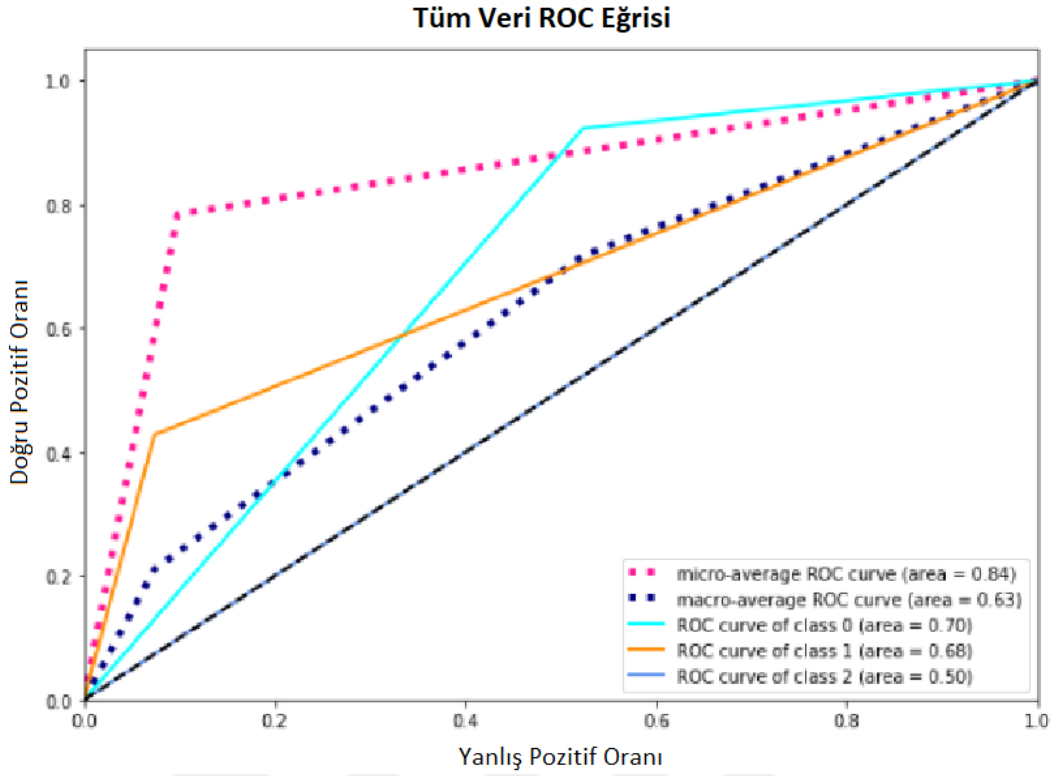
Tüm veriye uygulanan DNN modelinin eğitim ve test verisinin epok adımlarına göre doğruluk ve kayıp grafiği şekil 4.24 ve şekil 4.25’de verilmiştir.



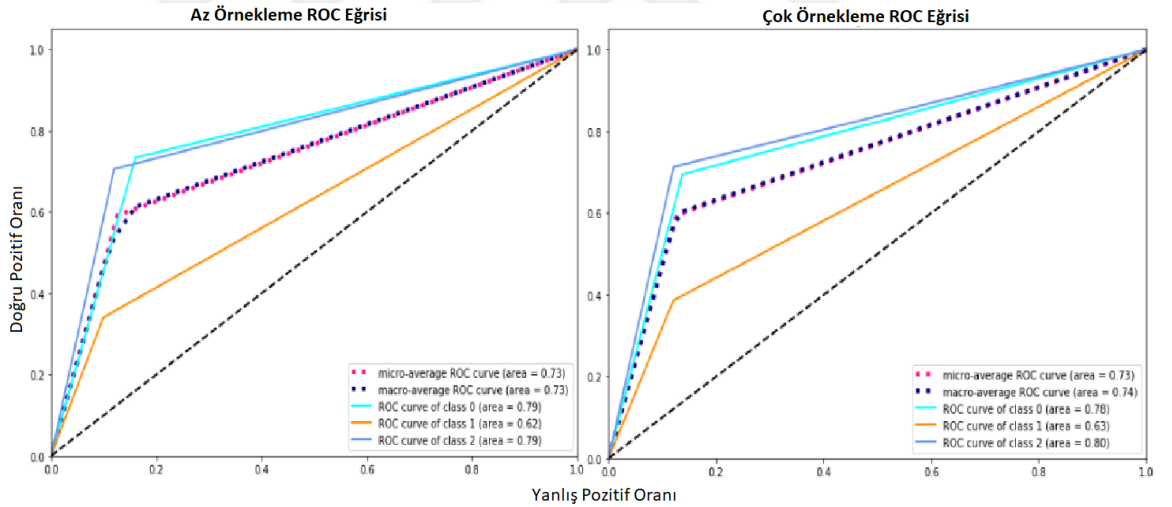
Şekil 4.24 DNN kayıp grafiği



Şekil 4.25 DNN doğruluk grafiği



Şekil 4.26 DNN Tüm Veri ROC Eğrisi grafiği



Şekil 4.27 DNN Az ve Çok Örneklem ROC Eğrisi grafiği

Şekil 4.26’da ve şekil 4.27’de DNN modelinin ROC Eğrisi grafiği verilmiştir. Tüm veri ile elde edilen sonuçlara göre Hafif kusurlu (0) sınıfı için AUC değeri 0.70, Ağır kusurlu (1) sınıfı için AUC değeri 0.68 ve Emniyetsiz (2) sınıfı için AUC değeri 0.50’dir. Az örneklem yapılarak elde edilen ROC eğrisinde ise Hafif kusurlu (0) sınıfı için AUC değeri 0.79, Ağır kusurlu (1) sınıfı için AUC değeri 0.62 ve Emniyetsiz (2) sınıfı için AUC değeri 0.79’dur. Çok örneklem yapılarak elde edilen ROC eğrisinde ise Hafif kusurlu (0) sınıfı için AUC değeri 0.78, Ağır kusurlu (1) sınıfı için AUC değeri 0.63 ve Emniyetsiz (2)

) sınıfı için AUC değeri 0.80'dir. Çok örnekleme ve az örnekleme ile elde edilen AUC değerleri hemen hemen aynı çıkmıştır. Çok örnekleme ve az örnekleme AUC değerleri tüm veri ile elde edilen AUC değerlerinden daha yüksek çıkmıştır.

Çizelge 4.8 DNN sınıflandırma raporu

Sınıf	Kesinlik			Duyarlılık			F1-Skor			Destek		
	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ	TV	AÖ	ÇÖ
0 (HK)	0.84	0.71	0.73	0.92	0.73	0.69	0.87	0.72	0.71	1527641	25639	1552948
1 (AK)	0.66	0.64	0.62	0.45	0.34	0.39	0.54	0.44	0.48	523792	24703	1520414
2 (EM)	0.00	0.73	0.73	0.00	0.71	0.71	0.00	0.72	0.72	21747	22572	1388276
Avg/ Total	0.78	0.69	0.69	0.79	0.59	0.60	0.78	0.63	0.63	2073180	72914	4461638

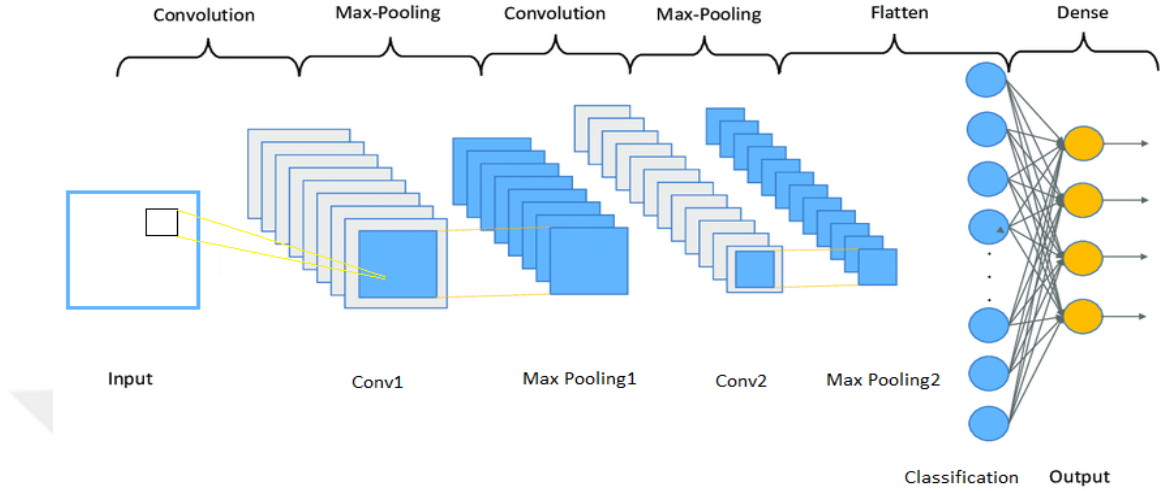
*TV: Tüm Veri, YÖ: Az Örnekleme, AÖ: Çok Örnekleme

Çizelge 4.8'de DNN modelinin sınıflandırma raporuna göre sonuçları gösterilmektedir. Tüm veri ile elde edilen sonuçlara göre Hafif Kusurlu sınıfı için kesinlik değeri 0.84, duyarlılık değeri 0.92 ve F1 skoru değeri 0.87 olarak çıkmıştır. Ağır Kusurlu sınıfı için kesinlik değeri 0.66, duyarlılık değeri 0.45 ve F1 skoru değeri 0.54'tür. Ağır kusurlu sınıfı için duyarlılık kesinlik ve F1 skor değerleri Hafif Kusurlu sınıfına göre daha düşük çıkmıştır. Bunun sebebi veri kümesinin yaklaşık %70'ini Hafif kusurlu verilerin oluşturmasıdır. Emniyetsiz sınıfı verilerini sınıflandıramadığı için kesinlik, duyarlılık ve F1 skoru değerleri 0'dır. Az örnekleme ve çok örnekleme ile elde edilen sonuçlar hemen hemen aynı çıkmıştır. Veride çok örnekleme ve az örnekleme yaparak emniyetsiz sınıfını doğru tahmin edebilme oranı artmıştır.

CNN: Bu çalışmada, derin öğrenme kütüphanesi olan keras yardımı ile CNN modeli oluşturulmuştur. Araç muayene verilerine CNN uygulanmadan önce çıkış verileri için One Hot Encoding ile kategorik değişkenleri ikili (binary) olarak adlandırma işlemi gerçekleştirilmiştir. Araç muayene verileri normalize edilmiştir.

Ağ çıkış değerleri 0 (Hafif Kusurlu), 1 (Ağır Kusurlu) ve 2 (Emniyetsiz)'dir. Toplamda 7.304.069 veri ve 21 adet öznitelik kullanılmıştır. Oluşturulan CNN modelinde ikişer adet Convolution ve Max Pool katmanı kullanılmıştır (Şekil 4.28). Convolution katmanı için giriş değeri 21'dir ve bu katmanlarda relu aktivasyon fonksiyonu kullanılmıştır. Çıkış katmanında ise 3 tane etiket (HK, AK, EM) olduğu için 3 adet sinir node'u

oluşturulmuştur ve softmax aktivasyon fonksiyonu kullanılmıştır. 10 epeoch sonucunda %80 doğruluk ve 0.4675 kayp değerine ulaşılmıştır.



Şekil 4.28 Oluşturulan CNN modeli katmanları

Apriori Algoritması: Apriori algoritması birliktelik kuralları için kullanılan en popüler ve klasik algoritmadır. Bu çalışmada 2015 Mart ayında muayenesinde sonucu Ağır Kusurlu çıkan araçların kusur verileri kullanılmıştır. Her muayene için bir satır olarak tutulan kusur verileri EK-2’deki sorgu ile muayenede çıkan kusurlar çizelge 4.9’da verildiği gibi i tek satırda birleştirilmiştir. Çizelge 4.9’da MUAYENE_ID alanı muayeneye giren araç a veril tekil alanı ve UBAK_KODU ise muayene sonucunda araçta çıkan kusur kodların ı göstermektedir. Apriori birliktelik analizi ile muayene sonucunda çıkan hatalarda sık geçen kusurlar bulunmuş olup diğer kusurlar ile arasında ilişkiler aranmıştır. 2015 Mart a yına ait veriler 232.717 adet muayene kusur verilerini içermektedir ve bu veri seti içerisi nde toplamda 1457 farklı kusur tipi vardır.

Çizelge 4.9 Apriori algoritmasında kullanılan veri seti

MUAYENE_ID	UBAK_KODU
54674047	15.16.3,5.5.4,2.5.3,5.12.11,14.7.1,5.3.2,5.3.3
54703171	6.7.8,6.7.14,5.12.16
55412354	6.16.5,10.5.1,15.5.6,5.3.1,15.8.1,15.5.1
55555430	4.1.13,4.1.4,5.15.1,4.3.7,5.3.3,5.4.9,15.16.2,5.14.8,1.1.20,5.14.20,6.16.2
55901349	14.1.23,14.7.2,1.1.30,5.14.7,5.11.1,14.1.6,14.1.27,5.11.3,4.1.10,14.3.1

Apriori modelinde minumum destek (support) parametresi için %20 eşik değeri seçilmiştir. Çizelge 4.10'da support değeri %20'den büyük değerler için birliktelik kuralları listelenmektedir.

Apriori algortitmasında kullanılan parametreler aşağıda verilmiştir (Anonymous 2021p).

- **min_support:** float (varsayılan: 0.5). Minumum destek değeri 0.2 seçilmiştir.
- **use_colnames:** bool (default: False). Varsayılan değeri “Yok” seçilmiştir. Veri setinin sütun dizinlerini kullanır. Varsayılan değer kullanılmıştır.
- **max_len:** int (varsayılan: None). Oluşturulan öge kümelerinin maksimum uzunluğunu ifade eder. Yok ise (varsayılan) tüm olası öge kümeleri değerlendirilir. Varsayılan değer kullanılmıştır.
- **verbose:** int (varsayılan: 0) 1'e eşit veya büyükse ve low_memory “Evet” ise yineleme sayısını gösterir. Eğer 1'e eşit ve low_memory “Yok” ise kombinasyon sayısını gösterir. Varsayılan değer kullanılmıştır.
- **low_memory:** bool (varsayılan: False). Evet ise, min_support üzerindeki kombinasyonları aramak için bir yineleyici kullanır. Bu durum yalnızca bellek kaynakları sınırlıysa büyük veri kümesi için kullanılmalıdır, çünkü bu uygulama yaklaşık varsayılandan 3-6 kat daha yavaş çalışır. Varsayılan değer kullanılmıştır.

Support değeri %20'nin üzerinde 25 kural listelenmiştir. Apriori algoritması sonucunda 14 adet tekli kural, 11 adet ikili kural ortaya çıkmıştır. Örneğin çizelge 4.10'a göre 14.1.23 kodlu “Gaz Sızdırmazlık Raporu: Yok” kusuru tüm 2015 Mart ayı muayene verilerinin %26'sında, 5.3.3 kodlu “Kısa huzmeli far: yönü hatalı” kusuru tüm 2015 Mart ayı muayene verilerinin %68'inde, 14.7.1 kodlu “Motor: yağ kaçakları var” kusuru tüm 2015 Mart ayı muayene verilerinin %42'sinde, 5.3.3 kodlu “Kısa huzmeli far: yönü hatalı” kusuru ve 4.7.1 kodlu “Motor: yağ kaçakları var” kusuru tüm verilerin %33'ünde beraber görülmektedir yorumu yapılabilir.

Çizelge 4.10 Apriori algoritması birliktelik kuralları

	Destek (Support)	Itemset (Öğeler)	Length (Uzunluk)
0	0.264317	14.1.23	1
1	0.218179	14.1.6	1
2	0.426037	14.7.1	1
3	0.340985	14.7.2	1
4	0.242281	15.16.3	1
5	0.247816	15.8.1	1
6	0.202465	15.8.3	1
7	0.241792	5.11.1	1
8	0.229794	5.12.1	1
9	0.313320	5.3.1	1
10	0.381717	5.3.2	1
11	0.689034	5.3.3	1
12	0.318468	5.4.10	1
13	0.277895	5.4.9	1
14	0.210367	5.3.3, 14.1.23	2
15	0.308937	14.7.1, 14.7.2	2
16	0.336250	5.3.3, 14.7.1	2
17	0.272950	5.3.3, 14.7.2	2
18	0.215910	15.16.3, 15.8.1	2
19	0.263595	5.3.3, 5.3.1	2
20	0.312852	5.3.2, 5.3.3	2
21	0.215687	5.3.2, 5.4.9	2
22	0.285557	5.4.10, 5.3.3	2
23	0.228741	5.3.3, 5.4.9	2
24	0.248001	5.3.3, 14.7.1, 14.7.2	3

Python’da “Association rules” fonksiyonu kullanılarak çizelge 4.11’de kurallar tablosu oluşturulmuştur. %20 minumun confidence değeri seçilmiştir. Yani araç muayenesinde X kusuru çıktığında Y kusurunda çıkması %20 ve üzeri olduğu söylenebilir. Toplam 25 kural oluşmuştur.

Örneğin 7. index değerindeki kural için;

- 5.3.3. kodlu kusurun “Kısa huzmeli far: yönü hatalı” görülme olasılığı %68’dir.
- 14.7.2 kodlu kusurun “Şanzıman: yağ kaçakları var” görülme olasılığı %34’tür.
- Her iki kusurunda **support** değerinin %27 ölçüldüğü görülmektedir. Yani her iki kusur birlikte %27 olasılıkla görülmektedir.
- 14.7.2 kodlu “Şanzıman: yağ kaçakları var” kusuru görülenlerin (**confidence**) %80’inde 5.3.3. kodlu “Kısa huzmeli far: yönü hatalı” kusuru da görülmektedir.

- 14.7.2 kodlu Şanzıman: yağ kaçakları olan araçların bu kusuru olmayan araçlara göre muhtemelen **(lift)** %1.16 kat daha fazla 5.3.3. kodlu “Kısa huzmeli far: yönü hatalı” kusuru görülecektir.
- “Kısa huzmeli far: yönü hatalı” ve “Şanzıman: yağ kaçakları var” kusurunun birlikte görülmeleri ile birbirlerinden bağımsız olarak görülmeleri **(leverage)** 0.037 fazla olduğu ortaya çıkmıştır.
- Bu iki kusurun birbiriyle olan korelasyonları **(conviction)** 1.09 olarak görülmektedir.

Çizelge 4.11 Apriori algoritması sonucu çıkan kurallar

	Antecedents	Consequents	Antecedent support	Consequents support	Support	Confidence	Lift	Leverage	Conviction
0	5.3.3	14.1.23	0.689034	0.264317	0.210367	0.305307	1.155081	0.028244	1.059005
1	14.1.23	5.3.3	0.264317	0.689034	0.210367	0.795890	1.155081	0.028244	1.523521
2	14.7.1	14.7.2	0.426037	0.340985	0.308937	0.725143	2.126612	0.163665	2.397663
3	14.7.2	14.7.1	0.340985	0.426037	0.308937	0.906015	2.126612	0.163665	6.106959
4	5.3.3	14.7.1	0.689034	0.426037	0.336250	0.488001	1.145444	0.042696	1.121025
5	14.7.1	5.3.3	0.426037	0.689034	0.336250	0.789250	1.145444	0.042696	1.475521
6	5.3.3	14.7.2	0.689034	0.340985	0.272950	0.396133	1.161733	0.037999	1.091326
7	14.7.2	5.3.3	0.340985	0.689034	0.272950	0.800474	1.161733	0.037999	1.558521
8	15.16.3	15.8.1	0.242281	0.247816	0.215910	0.891155	3.596035	0.155869	6.910606
9	15.8.1	15.16.3	0.247816	0.242281	0.215910	0.871252	3.596035	0.155869	5.885305
10	5.3.3	5.3.1	0.689034	0.313320	0.263595	0.382557	1.220976	0.047706	1.112134
11	5.3.1	5.3.3	0.313320	0.689034	0.263595	0.841295	1.220976	0.047706	1.959390
12	5.3.2	5.3.3	0.381717	0.689034	0.312852	0.819592	1.189479	0.049836	1.723680
13	5.3.3	5.3.2	0.689034	0.381717	0.312852	0.454044	1.189479	0.049836	1.132479
14	5.3.2	5.4.9	0.381717	0.277895	0.215687	0.565044	2.033297	0.109609	1.660179
15	5.4.9	5.3.2	0.277895	0.381717	0.215687	0.776144	2.033297	0.109609	2.761967
16	5.4.10	5.3.3	0.318468	0.689034	0.285557	0.896658	1.301325	0.066122	3.009087
17	5.3.3	5.4.10	0.689034	0.318468	0.285557	0.414431	1.301325	0.066122	1.163879
18	5.3.3	5.4.9	0.689034	0.277895	0.228741	0.331974	1.194600	0.037262	1.080952
19	5.4.9	5.3.3	0.277895	0.689034	0.228741	0.823120	1.194600	0.037262	1.758061
20	5.3.3, 14.7.1	14.7.2	0.336250	0.340985	0.248001	0.737550	2.162998	0.133345	2.511009
21	5.3.3, 14.7.2	14.7.1	0.272950	0.426037	0.248001	0.908596	2.132670	0.131714	6.279391
22	14.7.2, 14.7.1	5.3.3	0.308937	0.689034	0.248001	0.802754	1.165042	0.035132	1.576537
23	5.3.3	14.7.2, 14.7.1	0.689034	0.308937	0.248001	0.359925	1.165042	0.035132	1.079659
24	14.7.1	5.3.3, 14.7.2	0.426037	0.272950	0.248001	0.582111	2.132670	0.131714	1.739818

Apriori algoritması sonucu çıkan kurallara göre;

- Apriori algoritması sonucu birlikte çıkan kurallara göre en çok 5.3.3. kodlu “Kısa huzmeli far: yönü hatalı” kusuru ve 14.7.1 kodlu “Motor: yağ kaçakları var” kusuru birlikte %33 **(support)** olasılıkla görülmektedir (Kural 4 ve 5).

- 2015 Mart ayı araç muayene verilerinde en çok %68.90 olasılığı ile 5.3.3. kodlu “Kısa huzmeli far: yönü hatalı” kusuru görülmüştür.
- Kural 21’e göre 5.3.3. kodlu “Kısa huzmeli far: yönü hatalı” kusuru ve 14.7.2 kodlu “Şanzıman: yağ kaçakları var” kusuru görülenlerin (**confidence**) %90’ında 14.7.1 kodlu “Motor: yağ kaçakları var” kusuru da görülmektedir.
- Kural 3’e göre 14.7.2 kodlu “Şanzıman: yağ kaçakları var” kusuru görülenlerin (**confidence**) %90’ında 14.7.1 kodlu “Motor: yağ kaçakları var” kusuru da görülmektedir.
- 5.4.10 kodlu “Uzun huzmeli far: yönü hatalı” kusuru görülenlerin (**confidence**) %89.66’sında 5.3.3 kodlu “Kısa huzmeli far: yönü hatalı” kusuru da görülmektedir. (Kural 16)
- 15.16.3 kodlu “İlk yardım çantası: yok” kusuru görülenlerin (**confidence**) %89.11’inde 15.8.1 kodlu “Yangın söndürme tüpü: yok” kusuru da görülmektedir. (Kural 8)
- Birbiriyle olan korelasyonları (**conviction**) en yüksek olan kusurlar: Conviction 6.91 değeri ile 15.16.3 kodlu “İlk yardım çantası: yok” kusuru ve 15.8.1 kodlu “Yangın söndürme tüpü: yok” kusurudur (Kural 8). Daha sonra conviction 6.27 değeri ile 5.3.3 kodlu “Kısa huzmeli far: yönü hatalı”, 14.7.2 kodlu “Şanzıman: yağ kaçakları var” ve 14.7.1 kodlu “Motor: yağ kaçakları var” kusurudur (Kural 21).
- 14.7.2 kodlu “Şanzıman: yağ kaçakları var” kusuru ve 14.7.1 kodlu “Motor: yağ kaçakları var” kusurunun birlikte görülmeleri ile birbirlerinden bağımsız olarak görülmeleri (**leverage**) 0.163 fazla olduğu ortaya çıkmıştır (Kural 2 ve 3).
- “Kısa huzmeli far: yönü hatalı” ve “Şanzıman: yağ kaçakları var” kusurunun birlikte görülmeleri ile birbirlerinden bağımsız olarak görülmeleri (**leverage**) 0.037 fazla olduğu ortaya çıkmıştır.
- İlk yardım çantası olmayan araçların olan araçlara göre muhtemelen (**lift**) 3.59 kat daha fazla 15.8.1 kodlu “Yangın söndürme tüpü: yok” kusuru görülecektir (Kural 8). Yine aynı şekilde yangın söndürme tüpü olmayan araçların olan araçlara göre muhtemelen (**lift**) 3.59 kat daha fazla “İlk yardım çantası: yok” kusuru görülecektir (Kural 9).

Sonuç olarak kusurların birlikte görülme olasılıklarına bakılarak araç muayeneyesi daha etkili bir şekilde yapılabilir. Araçlarda birlikte görülen kusurlara bakılarak araç muayenesi iyileştirilebilir ve araçlarda çıkabilecek hataların erken tespitlerinde yararlı olabilir.

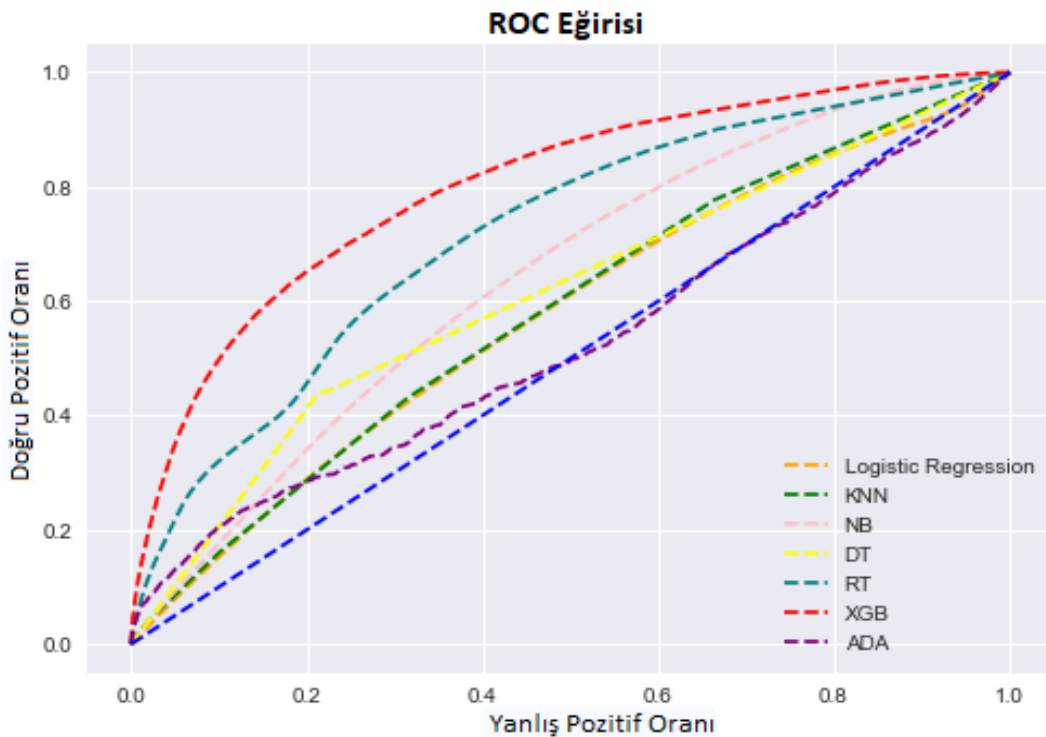
4.2. Tartışma

Bu tez çalışmasında araç muayene veri setine uygulanan makine öğrenmesi algoritmalarının performans karşılaştırması yapılmıştır. Bu tezin amacı belirli periyotlarla muayeneye giren aracın bilgilerinden yola çıkarak araç muayene sonucunu tahmin etmektir. Makine öğrenmesi modelleri ve derin sinir ağı modelleri kullanılarak verideki en önemli özellikler göz önüne alınarak araç muayene sonucunun tahmini ile muayene işlem sürecine yardımcı olmak hedeflenmektedir.

Çizelge 4.12 Sınıflandırma algoritmalarının sonuçlarının karşılaştırılması

Model	Doğruluk	Kesinlik			Duyarlılık			F1 Score			AUC		
		Sınıf 0 HK	Sınıf 1 AK	Sınıf 2 EM	Sınıf 0 HK	Sınıf 1 AK	Sınıf 2 EM	Sınıf 0 HK	Sınıf 1 AK	Sınıf 2 EM	Sınıf 0 HK	Sınıf 1 AK	Sınıf 2 EM
Tüm araç muayene verisi													
LR	%73.69	0.74	0.00	0.00	1.00	0.01	0.00	0.85	0.00	0.00	0.59	0.58	0.61
KNN	%70.47	0.75	0.33	0.20	0.87	0.20	0.04	0.81	0.25	0.07	0.59	0.58	0.61
NB	%73.01	0.74	0.39	0.00	0.97	0.06	0.00	0.84	0.10	0.00	0.65	0.64	0.78
DT	%69.53	0.80	0.41	0.07	0.80	0.43	0.06	0.80	0.42	0.06	0.62	0.61	0.65
RF	%70.94	0.81	0.44	0.08	0.81	0.45	0.03	0.81	0.44	0.05	0.73	0.72	0.80
XGB	%79.73	0.83	0.66	0.00	0.93	0.44	0.00	0.88	0.53	0.00	0.81	0.80	0.92
ADA	%78.46	0.81	0.66	0.17	0.94	0.35	0.07	0.87	0.46	0.09	0.77	0.52	0.91
DNN	%79.86	0.84	0.66	0.00	0.92	0.45	0.00	0.87	0.54	0.00	0.70	0.68	0.50
Az Örnekleme													
LR	%48.74	0.51	0.42	0.56	0.44	0.54	0.48	0.47	0.48	0.52	0.65	0.56	0.71
KNN	%46.49	0.45	0.39	0.61	0.59	0.37	0.42	0.51	0.38	0.50	0.65	0.56	0.71
NB	%48.79	0.49	0.42	0.57	0.65	0.37	0.44	0.56	0.39	0.49	0.69	0.62	0.74
DT	%48.79	0.49	0.42	0.57	0.65	0.37	0.44	0.56	0.39	0.49	0.69	0.57	0.73
RF	%61.61	0.67	0.51	0.67	0.66	0.48	0.72	0.66	0.49	0.69	0.84	0.68	0.88
XGB	%65.82	0.68	0.58	0.69	0.78	0.43	0.77	0.73	0.49	0.73	0.87	0.73	0.90
ADA	%62.40	0.65	0.52	0.67	0.80	0.40	0.67	0.72	0.45	0.67	0.85	0.63	0.88
DNN	%59.18	0.71	0.64	0.73	0.73	0.34	0.71	0.72	0.44	0.72	0.79	0.62	0.79
Çok Örnekleme													
LR	%47.41	0.46	0.42	0.57	0.51	0.44	0.47	0.48	0.43	0.52	0.62	0.61	0.67
KNN	%76.11	0.51	0.42	0.60	0.66	0.36	0.51	0.57	0.39	0.55	0.85	0.84	0.99
NB	%76.11	0.51	0.42	0.60	0.66	0.36	0.51	0.57	0.39	0.55	0.71	0.62	0.75
DT	%50.75	0.51	0.42	0.60	0.66	0.36	0.51	0.57	0.39	0.55	0.85	0.89	1.00
RF	%88.08	0.91	0.79	0.96	0.76	0.90	1.00	0.83	0.84	0.98	0.93	0.93	1.00
XGB	%65.76	0.68	0.58	0.69	0.79	0.42	0.77	0.73	0.49	0.73	0.87	0.73	0.90
ADA	%62.32	0.64	0.53	0.68	0.81	0.39	0.67	0.72	0.45	0.67	0.85	0.63	0.87
DNN	%59.53	0.73	0.62	0.73	0.69	0.39	0.71	0.71	0.48	0.72	0.78	0.63	0.80

Araç muayene verileri için kullanılan makine öğrenmesi algoritmalarının sonuçları Çizelge 4.12’de verilmiştir. Tüm araç muayene verisinde sınıflandırma algoritmalarının doğruluk değerleri şunlardır: LR için %73.69, KNN için %70.47, NB için %73.01, DT için %69.53, RF için %70.94, XGB için %79.73, ADA için %78.46, DNN için %79.86 ve CNN için %80’dir. Az örnekleme yapılarak elde edilen araç muayene verisinde sınıflandırma algoritmalarının doğruluk değerleri şunlardır: LR için %48.74, KNN için %46.49, NB için %48.79, DT için %48.79, RF için %61.61, XGB için %65.82, ADA için %62.40, DNN için %59.18’dir. Çok örnekleme yapılarak elde edilen araç muayene verisinde sınıflandırma algoritmalarının doğruluk değerleri şunlardır: LR için %47.41, KNN için %76.11, NB için %76.11, DT için %50.75, RF için %88.08, XGB için %65.76, ADA için %62.32, DNN için %59.53’dür. Tüm veride araç muayene sonuç tahmini doğruluk değeri en yüksek CNN, DNN ve XG Boost algoritmasından elde edilmiştir. Az örnekleme ile araç muayene sonuç tahmini doğruluk değeri en yüksek XG Boost algoritmasından elde edilmiştir. Çok örnekleme ile araç muayene sonuç tahmini doğruluk değeri en yüksek Random Forest algoritmasından elde edilmiştir.



Şekil 4.29 Sınıflandırma algoritmalarının ROC Eğrisi grafiği

Şekil 4.29’da verilen tüm makine öğrenmesi algortimalarının ROC Eğrisi grafiğinde AUC değeri en yüksek XG Boost algoritmasından elde edilmiştir. Tüm veri ile yapılan

algoritma karşılaştırmalarında en iyi sonuç XG Boost modelinden elde edilmiştir. XG Boost, Ada Boost ve DNN araç muayene sonucu tahmininde en iyi sonucu veren modellerdir.

Araştırma bulgularına bakıldığında Lojistik Regresyonun tüm veride, az örnekleme ve çok örnekleme ile elde edilen verilerde AUC değerleri düşük olduğu için sınıfları tahmin edebilme beceresi iyi olduğu söylenemez. Ayrıca hem sınıflandırma raporuna hem de karmaşıklık matrisine göre Lojistik Regresyon ağır kusurlu ve emniyetsiz sınıflarını tahmin etmede başarısızdır. Bu sınıflara ait verilerin hepsini tüm veride hafif kusurlu olarak sınıflandırmıştır. Az örnekleme ve çok örnekleme yapılarak ağır kusurlu ve emniyetsiz sınıflarını tahmin etme başarısı artırılmıştır. Çok terimli (multinomial) lojistik regresyon kullanılmasına rağmen sınıfları azınlık sınıfları tahmin etme başarısı düşüktür

KNN algoritması için tüm veride ve yeaztersiz örnekleme ile elde edilen verilerde sınıfları tahmin edebilme beceresi iyi olduğu söylenemez. Fakat çok örnekleme ile elde edilen veride ROC Eğrisi grafiğine bakıldığında AUC değerleri yüksektir ve sınıfları tahmine edebilme yeteneği yüksektir. Sınıflandırma raporuna göre ağır kusurlu ve emniyetsiz sınıfları için duyarlılık, kesinlik ve F1 sokur değerleri hafif kusurlu sınıfına göre daha düşük çıkmıştır. KNN algortiması büyük verilerde bellek sorununa sebep olduğu için diğer algoritmalara göre çalışması uzun sürmüştür.

Naive Bayes algoritması sınıfları tahmin edebilme beceresi Lojistik Regresyon ve KNN algoritmasına göre daha iyi olduğu söylenebilir. Fakat sınıflandırma raporuna göre az ve çok örnekleme ile elde edilen sonuçlar karşılaştırıldığında KNN algoritması ile aynı çıkmıştır. Tüm veride Naive Bayes algoritması karmaşıklık matrisine göre emniyetsiz sınıfını hiç doğru tahmin edememiştir. Ayrıca Ağır kusurlu sınıfını tahmine etme başarısı da düşüktür.

Rastgele Orman algoritmasının sınıfları tahmin edebilme beceresi AUC değerine bakıldığında Karar Ağaçları algoritmasına göre daha iyi olduğu söylenebilir. Rastgele Orman algoritması karar ağaçlarının en büyük sorunu olan aşırı uyumu azaltmaktadır. Rastgele Orman algoritmasının sınıflandırma raporundaki kesinlik, duyarlılık ve F1 skor değerleri Karar Ağaçları algoritmasına göre daha iyi sonuçlar çıkmıştır. Rastgele Orman algoritma

ası çok örnekleme ile elde edilen veride en %88.08 ile en yüksek doğruluk seviyesine ulaşmıştır.

Araştırma bulgularına göre XG Boost algoritması sınıfları tahmin edebilme yeteneği diğer algoritmalara göre daha iyi olduğu ortaya çıkmıştır. XG Boost algoritması ile yüksek tahmin elde edebilir, aşırı uyumu önleyebilir, hızlıdır ve büyük veri setlerinde en kullanışlı modellerdendir. XG Boost algoritmasının AUC değeri diğer algoritmalara göre daha yüksektir. XG Boost algoritması karmaşıklık matrisi ve sınıflandırma raporundan da anlaşılacağı üzere emniyetsiz sınıfına ait verileri doğru tahmin edememiştir. Fakat hafif kusurlu ve ağır kusurlu sınıflarını doğru tahmin etme başarısı diğer algoritmalara göre daha yüksektir. Çok örnekleme ve az örnekleme ile elde edilen doğruluk değerleri ise %65 oranı ile benzer çıkmıştır.

Ada Boost modeli XG Boost modeli ile karşılaştırıldığında doğruluk, kesinlik, duyarlılık ve F1 Score değerleri hemen hemen birbirine yakındır. Ada Boost modelinde XG Boost'un hiç doğru tahmin edemediği emniyetsiz sınıfına ait verileri doğru tahmin edebilme kabiliyeti vardır. Çok örnekleme ve yetersiz örnekleme ile elde edilen sonuçlar ise benzer çıkmıştır. XG Boost algoritması sınıfların AUC değeri Ada Boost algoritmasına göre daha yüksektir.

Ağır kusurlu sınıfını tüm veride en çok %45 ile Random Forest ve DNN, %44 ile XG Boost algoritması belirleyebilmiştir (Çizelge 4.12). Modellerin emniyetsiz sınıfını belirlemede başarısı düşüktür. Az örneklemede ağır kusurlu sınıfını en çok %48 ile RF, emniyetsiz sınıfını %77 ile XG Boost belirleyebilmiştir. Çok örneklemede ise ağır kusurlu sınıfını %90 ile Random Forest, emniyetsiz sınıfını %100 ile yine Random Forest belirleyebilmiştir.

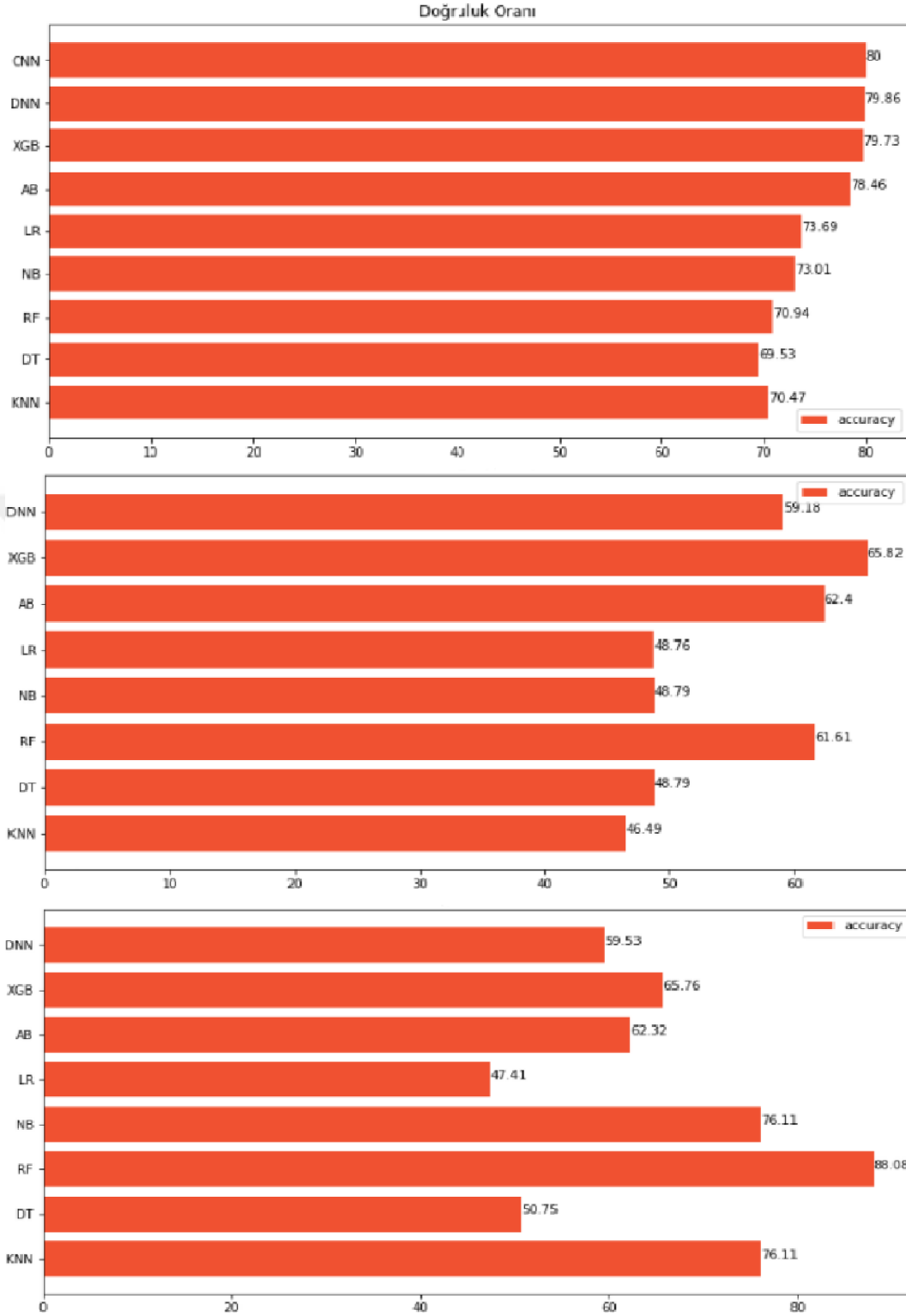
DNN modeli, öğrenme ve genelleme kabiliyeti olması ve elimizde büyük bir veri seti olması sebebiyle kullanılmıştır. DNN modeli tüm araç muayene verilerine uygulandığında muayene sonucunu %79.86, az örnekleme ile %59.18 ve çok örnekleme ile %59.53 oranında doğru bildiği tespit edilmiştir. Tüm veride ağır kusurlu sınıfı için duyarlılık kesinlik ve F1 score değerleri hafif kusurlu sınıfına göre daha düşük çıkmıştır. Emniyetsiz sınıfına ait verileri sınıflandıramadığı için kesinlik, duyarlılık ve F1 skoru değerleri 0'dır. Çok örnekleme ve az örnekleme ile elde edilen sonuçlar ise benzer çıkmıştır ve sınıfları tah

min etme oranı çok örnekleme ve az örnekleme ile elde edilen verilerde daha yüksek çıkmıştır.

CNN modeli tüm araç muayene verilerine uygulandığında muayene sonucunu %80 oranında doğru bildiği tespit edilmiştir. CNN, fotoğraf verilerinde daha iyi sonuç vermektedir. Fakat araştırma bulgularına göre CNN, fotoğraf dışındaki verilerde kullanılabilir. Araç muayene verileri, fotoğraf verilerindeki gibi üç boyutlu hale çevirilip CNN modeline verilmiştir. DNN ve CNN algoritmalarının doğruluk değerleri yüksek olmasına rağmen kayıp değeri epok sayısı ve katman sayısı artırılarak düşürülememiştir.

Bu çalışmada muayeneye giren araçların muayene sonuçları tahmin edilmeye çalışılmıştır. Araç muayene verilerini sınıflandırmak amacıyla veri madenciliği sınıflandırma tekniklerini uygulamak için Python programlama dili kullanılmıştır. Python kullanılmasının sebebi açık kaynak kodlu, ücretsiz, öğrenmesi kolay, tüm işletim sistemleri ile uyumlu olmasıdır. Veri analizi, makine öğrenimi, derin öğrenme ve daha bir çok alan üzerinde çalışmamıza imkan veren birçok kütüphanesi mevcuttur.

Bu tez çalışmasında tüm araç muayene verilerinde en iyi sonuç %80 doğruluk değeri ile CNN, %79.73 doğruluk değeri ile XG Boost, %79.86 doğruluk değeri ile DNN ve %79.46 doğruluk değeri ile Ada Boost algoritmasından elde edilmiştir. Az örnekleme ile elde edilen verilerde en iyi sonuç %65.82 doğruluk değeri ile XG Boost, %62.40 doğruluk değeri ile Ada Boost algoritmasından elde edilmiştir. Çok örnekleme ile elde edilen verilerde en iyi sonuç %88.08 doğruluk değeri ile Random Forest algoritmasından elde edilmiştir (Şekil 4.30).



Şekil 4.30 Sınıflandırma algoritmaları doğruluk değerleri

Bir veri madenciliği yöntemi olan birliktelik kuralları genel olarak; veriler arasında bir arada sık olarak görülen kalıpları veya birliktelik davranışlarını ortaya çıkarmaktadır. Yapılan bu tez çalışmasında araç muayene verileri veri madenciliği yöntemlerinden

Apriori birliktelik kuralları kullanılarak sonuçlar değerlendirilmiştir. Apriori algoritması araç muayene veri kümesinde yer alan temel örüntüleri keşfetmek için kullanılmıştır. Araç muayene sonucunda çıkan kusurlar arasında anlamlı ilişkiler bulmak ve bu ilişkilerden faydalanarak araç sürücülerine ve araç muayene sürecine katkı sağlamak amaçlanmıştır. Birliktelik kuralları sonucuna göre kısa huzmeli far yönü hatası %68 oranla en çok görülen kusur olmuştur. Daha sonra ise %42 oranla motorda yağ kaçaklarının olması kusuru en çok görülen kusurdur. Bu iki kusurun birlikte görülme olasılığı ise %33'dür. Şanzımanda yağ kaçakları görülen araçların %90'ında motorda yağ kaçakları olduğu da görülmektedir. Birlikte en çok görülen kusurlar %30 olasılıkla Şanzıman: yağ kaçakları var ve Motor: yağ kaçakları var kusurudur. Uzun huzmeli far yönü hatalı olan araçların %89'unda kısa huzmeli far yönünün de hatalı olduğu görülmektedir. İlk yardım çantası olmayan araçların %89'unda yangın söndürme tüpünün de olmadığı görülmektedir. Ayrıca bu iki kusur birbiriyle olan korelasyonları en yüksek olan kusurlardır. İlk yardım çantası olmayan araçların, olan araçlara göre %3.6 kat daha fazla yangın söndürme tüpü olmadığı görülmüştür. Kısa huzmeli far: yönü hatalı, Şanzıman: yağ kaçakları var ve Motor: yağ kaçakları var kusurları da birbiriyle korelasyonları yüksek olan kusurlardır.

4.3 Geçerliliğe Yönelik Tehditler

İç geçerlilik: Bu çalışmada iç geçerlilik ile ilgili tehdit, kullanılan araç muayene veri setinin dengesiz dağılmasıdır. Çalışmada kullanılan yaklaşık 7 milyon verinin 5 milyonu hafif kusurlu, 2 milyonu ağır kusurlu, çok az bir kısmı da emniyetsiz sınıfına ait verilerdir. Veri setinden yaklaşık 100 bin aykırı değer çıkarılmıştır. Veriler dengesiz dağıldığı için doğruluk değeri %80 üzerine çıkamamıştır. Çalışmada 2015, 2016 ve 2017 yılları Ocak, Şubat ve Mart aylarına ait araçların muayene bilgileri kullanılmıştır. Her aya ait veri yaklaşık 1 milyon kayıttır. Kullanılan bilgisayarın donanım özelliklerinin yetersiz kalmasından dolayı daha fazla veri ile çalışılamaması iç geçerlilik tehditi olarak görülmüştür.

Dış geçerlilik: Dış geçerlilik sonuçların ne kadar genelleştirilebilir olduğunu ifade eder. Bu çalışmada sınırlı sayıda veri ile çalışılmıştır. Bir sonraki çalışmada diğer yılların verileride eklenebilir. Ayrıca araç muayene veris etinde 1694 tane kusur türü olduğundan ve bu kusurların hepsini makine öğrenmesi algoritmaları ile analiz etmek hem güç hem

de yeterli bilgisayar donanımına sahip olunmaması sebebi ile veri toplam 15 kusur grubu altında incelenmiştir. Bir sonraki çalışmada tüm kusur tipleri kullanılarak detaylı bir çalışma yapılabilir. Araç muayene verilerinin Apriori algoritması ile birliktelik analizinde ise yeterli bilgisayar belleği olmamasından dolayı sadece 2015 Mart ayı verileri ile çalışılmıştır. Daha sonraki çalışmalarda daha fazla veri ile birliktelik analizi yapılabilir.

Yapı Geçerliliği: Yapı geçerliliği, çalışmanın amaçlanan yapıyı ölçüp ölçmediği, gözlemler veya ölçümler temelinde yapılan çıkarımların uygunluğudur. Bu çalışmada araç muayene sonucu tahmin etmek amaçlanmıştır. Makine öğrenmesi algoritmaları ile oluşturulan modellerin %70'in üzerinde muayene sonucunu doğru tahmin ettiği gözlemlenmiştir. Sonraki çalışmalarda özellikle DNN ve CNN modelleri geliştirilerek daha iyi doğruluk değerlerine ulaşılabilir.

5. SONUÇ

Bu tez çalışmasında Ulaştırma ve Altyapı Bakanlığı araç muayene verileri ile makine öğrenmesi algoritmaları ve derin sinir ağları kullanılarak araç muayene sonucu tahmini yapılmıştır ve kullanılan modellerin performansı değerlendirilmiştir. Ayrıca araç muayene sonucunda çıkan kusurların birliktelik analizi gerçekleştirilmiştir.

Bu tezin amacı karayolunda seyreden motorlu ve motorsuz araçların teknik muayenelerini daha etkin ve sağlıklı bir şekilde yapmak ve araçların güvenliğini sağlamaktır. Yapılan araştırmalar neticesinde araçların hata tahmini probleminde daha önceden yapılan çalışmalardan faydalanılarak araç muayene sonucunu doğru tahmin edebilmek amaçlanmıştır. Ayrıca kusur verileri üzerinde birliktelik analizi yapıp araçlarda çıkabilecek hataların önceden keşfedilerek çıkabilecek arızaların erken tespitlerinde yararlı olacaktır. Birliktelik analizi ile araç muayene sonucunda çıkan kusurlar arasında anlamlı ilişkiler bulmak ve bu ilişkilerden faydalanarak araç sürücülerine ve araç muayene sürecine katkı sağlamak amaçlanmıştır.

İkinci bölümde bu konu üzerinde daha önceden yapılan çalışmalara yer verilmiştir. Tez konusu ile ilgili yapılan literatür araştırmasında daha çok otomobil arıza tespit sistemi ve otomobil motorunun hata tespiti üzerine yapılan çalışmalar yer almaktadır. Ayrıca bu çalışmalarda genellikle araç servislerine ait daha çok teknik bilgi içeren veriler ile araç arıza tespit sistemi uygulamaları üzerine çalışılmıştır. İncelenen çalışmalarda yapay sinir ağları ve bulanık mantık otomobil arıza teşhis alanında çok kullanılan bir yöntemlerdir. Son zamanlarda uzman sistemler (expert system), sinir ağları (neural network), doğrusal olmayan PCA yöntemi ve bulanık mantık yaklaşımları da dahil olmak üzere çeşitli veriye dayanan hata tanıma algoritmaları önerilmiştir. İnsan uzmanlığını içeren uzman sistem (expert system) ve bulanık mantık yaklaşımları bu alanda başarıyla uygulanmıştır.

Türkiyede ise araç muayene verileri üzerinde makine öğrenmesi uygulanması ve muayene sonucunda çıkan kusurların birliktelik analizi çalışması daha önce gerçekleştirilmemiştir. Bu yapılan çalışma ile araç kusurları arasında anlamlı ilişkiler bulunmuştur. Bizim çalışmamız ile daha benzer olan Das ve arkadaşlarının yaptığı çalışmasında araç muayene verileri ile kaza çarpışma verileri birlikte kullanılıp metin madenciliği modeli gerçekleştirilerek araç muayenelerinin güvenlik etkinliğini

araştırılmıştır. Bu yapılan çalışmada ise sadece araç muayene verileri üzerinde çalışılmıştır. Elimizde kaza çarpışma verileri olmadığından dolayı hem araç muayene verisi hem de kaza çarpışma verileri ile ilgili bir çalışma yapılamamıştır. Bu tez çalışmasında diğer çalışmalardan farklı olarak daha çok veri ile (yaklaşık 7 milyon veri) çalışılmıştır. Diğer yıllarında verileri elde edilerek daha kapsamlı bir çalışma yapılabilir. Ayrıca kullanılan veri setinde araç muayenesinde çıkan araç kusurları daha detaylı bilgiler içermektedir. Bu tez çalışmasında 1694 adet kusur türü vardır fakat bunları analiz etmek hem güç hem de yeterli bilgisayar donanımına sahip olunmaması sebebi ile 15 kusur sınıfı altında incelenmiştir.

Periyodik araç muayenesi araç bakımı güvenliği ve verimliliği için çok önemlidir. Araç güvenliğini ve verimliliğini artırmak için araçta kusurlar ve arızalar meydana gelmeden önce muayene ve bakım yaptırmak oldukça önemlidir. Bu tez çalışmasında aracın kilometresi, modeli, yakıt tipi vb. özellikleri ve muayene sonucunda çıkan kusurları gruplanıp toplamda 15 grup altında toplanarak elde edilen kusur grupları baz alınarak muayene sonucunun ağır kusurlu, hafif kusurlu veya emniyetsiz olarak tespit etmek için sınıflandırmaya dayalı bir yöntem geliştirilmiştir. Kusur verilerinin detayına bakılmadan üst veri kusur grupları kullanılmıştır. Bu çalışmada kusurlar arasındaki ilişkileri çıkarabilmek için kusur tahminleme de yapılabilir. Ayrıca sınıflandırma tabanlı modellerin performansını iyileştirmek ve uygulanabilirliğini artırmak için Derin Sinir Ağları (DNN) ve Evrişimli Sinir Ağları (CNN) ile araç muayene sonucu tespiti için derin öğrenmeye dayalı bir yaklaşım önerilmiştir. Bu çalışma derin öğrenme tabanlı yöntemlerin araç muayene sonucu tespit etmede etkili olduğunu göstermektedir.

Tahmin modeli için araç muayene verilerine makine öğrenme algoritmalarından Lojistik Regresyon (LR), Naive Bayes (NB), Karar Ağaçları (DT), Rastgele Orman (RF), K-En Yakın Komşu (KNN), Gradyan Yükseltme (XG Boost), Ada Boost ve derin öğrenme algoritmalarından Derin Sinir Ağı (DNN) ve Evrişimsel Sinir Ağı (CNN) uygulanarak araç muayene sonucu tahmini yapılmıştır. Araç muayene veri setinde sınıflar dengesiz dağılmıştır. Araç muayene verisinin yaklaşık %70'ini hafif kusurlu sınıfı, %20'sini ağır kusurlu sınıfı ve %10'unu emniyetsiz sınıfı oluşturmaktadır. Sınıfların dengesiz dağılmasından dolayı araç muayene verisi üzerinde örnekleme işlemi yapılmıştır. Az örnekleme yapılarak hafif kusurlu ve ağır kusurlu sınıfının bazı verileri veri setinden

çıkarılmış olup 264.642 adet veri elde edilmiştir ve sınıflar dengeli hale getirilmiştir. Çok örnekleme yapılarak azınlık sınıf olan ağır kusurlu ve emniyetsiz sınıfının verileri çoğaltılıp 161.90.493 veri elde edilmiştir. Tüm araç muayene verileri (yaklaşık 7 milyon veri), az örnekleme ve çok örnekleme işlemi yapılarak elde edilen verilere makine öğrenmesi algoritmaları (LR, NB, KNN, DT, RF, XG Boost) ve Derin Sinir Ağları (DNN) uygulanmıştır. Her bir model AUC, ROC eğrisi, doğruluk, kesinlik, hatırlatma ve F1 skor değerleri açısından karşılaştırılmıştır. Makine öğrenme modellerinin araç muayene verilerini kullanarak hafif kusurlu, ağır kusurlu ve emniyetsiz muayene sonucu tahmininin yüksek doğrulukla öngörebildiği sonucuna ulaşılmıştır. En yüksek doğruluk değerine çok örnekleme ile elde edilen verilere Rastgele Orman algoritması uygulanarak %88.08 doğruluk seviyesine ulaşılmıştır. Tüm veri ile elde edilen sonuçlara göre azınlık sınıf olan ağır kusurlu ve emniyetsiz sınıflarını tahmin etme oranı düşük çıkmıştır. Bunun sebebi verilerin dengesiz dağılmasıdır. Fakat az örnekleme ve çok örnekleme yapılarak ağır kusurlu ve emniyetsiz sınıflarını tahmin etme oranı artırılmıştır. Azınlık sınıflarını tahmin etme oranı en yüksek XG Boost, Ada Boost, Derin Sinir Ağı ve Random Forest algoritmalarından elde edilmiştir. Ayrıca çok örnekleme ile elde edilen sonuçlar az örnekleme ile elde edilen sonuçlara göre daha başarılı çıkmıştır.

Ayrıca bu tez çalışmasında birliktelik kuralları yöntemlerinden Apriori algoritması kullanılarak araçların muayene sonucuna göre görülen kusurların analizi gerçekleştirilmiştir. Apriori algoritması araç muayene veri kümesinde yer alan temel örüntüleri keşfetmek için kullanılmıştır. Apriori algoritması ile araç muayene verileri üzerinden istatistiksel teknikler ile tahmin edilemeyen çıkarımlar yapılmıştır ve bu sayede ileriye dönük analizler yapmak veya kararlar almak mümkün olacaktır. Birliktelik kuralları sonucuna göre kısa huzmeli far yönü hatası %68 oranla en çok görülen kusur olmuştur. İkinci en çok görülen kusur ise %48 oranla motorda yağ kaçaklarının olması kusuru olmuştur. Şanzımanında yağ kaçakları görülen araçların %90'ında motorunda da yağ kaçakları olduğu görülmüştür. Ayrıca bu iki kusur %30 olasılıkla birlikte en çok görülen kusurlardandır. Uzun huzmeli far yönü hatalı olan araçların %89'unda kısa huzmeli far yönünün de hatalı olduğu görülmüştür. İlk yardım çantası olmayan araçların %89'unda yangın söndürme tüpünün de olmadığı görülmüştür. Ayrıca bu iki kusur birbiriyle korelasyonları en yüksek olan kusurlardır. Araç muayene verisinde çıkan bu kurallar neticesinde araç sürücülerinin bu kusurlara daha çok dikkat etmesi ve önlemler

alması gerektiği ortaya çıkmaktadır. Ayrıca araç üreticileri ve servisler bu kusurlara bakım sırasında daha önem verip ek önlemler alabilirler. Birliktelik analizi sonucu birlikte görülen kusurlar ile TÜVTÜRK araç muayene istasyonlarında gerçekleştirilen muayene işleminde daha fazla dikkat edilecek hususlar belirtilmiştir. TÜVTÜRK araç muayene istasyonlarında muayene yapan personeller birliktelik analizi sonucunda çıkan hususlara dikkat edip bu bilgileri kullanarak araç muayenesini daha etkin ve güvenli bir şekilde yapabilirler.

Araç hata tespit sistemleri araç sağlığı, sürücü güvenliği ve trafik güvenliği açısından çok önemlidir. Bu inceleme ile araç bakımı ve güvenilirliği artacak, araç muayeneleri daha etkin bir şekilde yapılacaktır. Bu sayede iki saat olan araç muayene süresi azalacaktır. Araç muayene istasyonlarındaki çalışanlar ve araç sahipleri süreden tasarruf edecektir. Çalışmada yapılan birliktelik analizi ile muayenede birlikte çıkan kusurlar tespit edilip araç muayenesi daha etkili yapılabilir ve süreden tasarruf edilebilir. Araç kusurları üzerinde yapılan birliktelik analizi, araçlarda çıkabilecek hataların erken tespitinde yararlı olacaktır. Araç muayenesinde birlikte çıkan kusurlar saptanarak araç sahipleri araçların bakımında bu kusurlara daha çok dikkat edebilir. Bu tez çalışmasında yapılan sınıflandırma modeli ve birliktelik analizi ile kurum ve kuruluşların daha etkili kararlar almasına yönelik eğilimlerin ve davranış kalıplarının ortaya çıkarılması mümkün olmaktadır. Makine öğrenmesi modelleri ve derin sinir ağı modelleri kullanılarak verideki en önemli özellikler göz önüne alınarak araç muayene sonucunun tahmini ve birliktelik analizi ile ortaya çıkan kusurlar arasındaki temel örüntüler ile muayene işlem sürecine yardımcı olmak hedeflenmiştir.

KAYNAKLAR

- Agrawal R., Imielinski T. ve Swami, A. 1993. Mining association rules between sets of items in large databases, ACM SIGMOD International Conference on Management of Data, 207-216.
- Anonymous 2018. Web Sitesi: <https://medium.com/@tuncerergin/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir-97a0f5d34cad>, Eriřim Tarihi 22.08.2021
- Anonymous 2021a. Web Sitesi: <https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e>, Eriřim Tarihi 15.05.2021
- Anonymous 2021b. Web Sitesi: <https://www.kaggle.com/nareshbhat/outlier-the-silent-killer>, Eriřim Tarihi 02.04.2021
- Anonymous 2021c. Web Sitesi: <https://medium.com/kodcular/python-ve-anaconda-kurulumu-b8931bd80e64>, Eriřim Tarihi 11.03.2021
- Anonymous 2021d. Web Sitesi: <https://farukeryilmaz.com/jupyter-notebook-nedir-nasil-kullanilir-veri-bilimi/>, Eriřim Tarihi 12.03.2021
- Anonymous 2021e. Web Sitesi: <https://devnot.com/2019/tensorflow-nedir-nasil-kullanilir/>, Eriřim Tarihi 11.03.2021
- Anonymous 2021f. Web Sitesi: <https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/what-is-logistic-regression/>, Eriřim Tarihi 19.07.2021
- Anonymous 2021g. Web Sitesi: <https://medium.com/@k.ulgen90/makine-%C3%B6%C4%9Frenimi-b%C3%B6l%C3%BCm-2-6d6d120a18e1>, Eriřim Tarihi 20.07.2021
- Anonymous 2021h. Web Sitesi: <https://medium.com/deep-learning-turkiye/karar-a%C4%9Fa%C3%A7lar%C4%B1-makine-%C3%B6%C4%9Frenmesi-serisi-3-a03f3ff00ba5>, Eriřim Tarihi 20.07.2021
- Anonymous 2021ı. Web Sitesi: <https://medium.com/data-science-tr/makine-%C3%B6%C4%9Frenmesi-dersleri-5-bagging-ve-random-forest-2f803cf21e07>, Eriřim Tarihi 20.07.2021
- Anonymous 2021i. Web Sitesi: <https://www.veribilimiokulu.com/xgboost-nasil-calisir/>, Eriřim Tarihi 21.07.2021
- Anonymous 2021j. Web Sitesi: <https://towardsdatascience.com/https-medium-com-vishalorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d>, Eriřim Tarihi 22.07.2021

- Anonymous 2021k. Web Sitesi: <https://medium.com/@bernatas/roc-e%C4%9Frisi-ve-e%C4%9Fri-alt%C4%B1nda-kalan-alauc-97b058e8e0cf>, Eriřim Tarihi 23.07.2021
- Anonymous 2021l. Web Sitesi: <https://medium.com/@bernatas/roc-e%C4%9Frisi-ve-e%C4%9Fri-alt%C4%B1nda-kalan-alauc-97b058e8e0cf>, Eriřim Tarihi 26.07.2021
- Anonymous 2021m. Web Sitesi: <https://medium.com/deep-learning-turkiye/model-performans%C4%B1n%C4%B1-de%C4%9Ferlendirmek-metrikler-cb6568705b1>, Eriřim Tarihi 28.07.2021
- Anonymous 2021n. Web Sitesi: <https://en.wikipedia.org/wiki/AdaBoost>, Eriřim Tarihi 15.08.2021
- Anonymous 2021o. Web Sitesi: <https://ichi.pro/tr/python-da-adaboost-siniflandirici-155257329026137>, Eriřim Tarihi 16.08.2021
- Anonymous 2021p. Web Sitesi: http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/apriori/, Eriřim Tarihi 02.09.2021
- Atak, O.N. 2019. Derin Öğrenme Yöntemleri ile Sinema Filmi Başarım Tahmini. Yüksek Lisans Tezi. Eskişehir Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, 14, Eskişehir.
- Başar, M. D., Sarı, P., Kılıç, N. ve Akan, A. 2016. Detection of chronic kidney disease by using Adaboost ensemble learning approach,. 24th Signal Processing and Communication Application Conference (SIU),773-776.
- Bo, Q. 2010. Fault diagnosis method of automobile engine based on least squares support vector machine. 2010 2nd International Conference on Signal Processing Systems, V3-43-V3-46.
- Breiman, L. 2001. Random forests. Machine learning, 45(1), 5-32.
- Connelly, L. 2020. Medsurg Nursing; Pitman Vol. 29, Iss. 5, 353-354.
- Çelik, Ö., Altunaydın, S.S. 2018. A Research on Machine Learning Methods and Its Applications. Journal of Educational Technology and Online Learning, 1 (3), 25-40.
- Das, S., Geedipally, S. R., Dixon, K., Sun, X. ve Ma, C. 2019. Measuring the Effectiveness of Vehicle Inspection Regulations in Different States of the U.S., Transportation Research Record2019, Vol. 2673(5) 208-219.
- Dejun W., Tianliang, X., Chengdong, L. ve Lihua W., 2011. Fault diagnosis of automobile engine based on support vector machine. 2011 3rd International Conference on Advanced Computer Control, 320-324.

- Dener, M., Dörterler, M. ve Orman, A. 2009. Açık Kaynak Kodlu Veri Madenciliği Programları: WEKA’da Örnek Uygulama”, Akademik Bilişim’09 - XI. Akademik Bilişim Konferansı Bildirileri, 787-796.
- Elbiad, Z. 2013. Web Tabanlı Anket Sistemi İle Elde Edilen Verilerin Veri Madenciliği Yöntemi İle Analizi, Yüksek Lisans Tezi, İstanbul Aydın Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 18.
- Fayez, M. A. 2018. Diagnoses of Coronary Heart Disease (CHD) Using Data Mining Techniques Based on Classification. Yüksek Lisans Tezi. İstanbul Altınbaş Üniversitesi, Ankara, 11-49.
- Freund, Y. ve Schapire, RE. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119–139.
- Gang, H. 2016. Automobile Fault Diagnosis System Based on Improved Neural Network," 2016 International Conference on Smart City and Systems Engineering (ICSCSE), 494-497.
- Gao, X. Z., Ovaska, S. J. ve Dote, Y. 2000. Motor fault detection using Elman neural network with genetic algorithm-aided training. Smc 2000 conference proceedings. 2000 IEEE International Conference on Systems, Man and Cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions', vol.4., 2386-2392.
- Goodfellow, I., Bengio, Y. ve Courville, A. 2016. Deep Learning, The Mit Press, 163.
- Gislason, P. O., Benediktsson, J. A. ve Sveinsson, J. R. 2006. Random forests for land cover classification. Pattern Recognition Letters, 27(4), 294-300.
- Han, J. ve Kamber, M. 2000. Data Mining Concept and Techniques. Morgan Kaufmann Publishers, United States of America, 350-355.
- Hang, Q., Liu, Y. ve Liu, H. 2002. Fault diagnosis Study of the Inlet and Exhaust system Based on Vibration Signal” , Vehicle Engine, No.5 (Serial No.141), 34-37.
- He, J., Harris, J. R., Sawada, M. ve Behnia, P. 2015. A comparison of classification algorithms using Landsat-7 and Landsat-8 data for mapping lithology in Canada's Arctic. International Journal of Remote Sensing, 36(8), 2252-2276.
- Hu, Y., Yang, S., Liu, Y. ve Zhou, Y. 1998. Studies on Vibration Diagnosis Mechanism for Diesel Engine Tribological Faults. Transactions of CSICE, Vol.16 (1998) No.1, 50-61.
- Kalyani, P. 2012. Data Mining Techniques. International Journal of Advanced Research in Computer Science and Software Engineering, 2 (10), 439-442.

- Keleş, A. E. ve Kaya, M. 2014. Duvar İnşa Edilmesinde Verimliliği Etkileyen Faktörlerin Apriori Veri Madenciliği Yöntemi Kullanılarak Analizi, Akademik Bilişim Konferansı, 831-836.
- Kher, S. ve Chande, P. K. 1996. Intelligent diagnosis of transparent faults in automobiles," Proceedings of the 35th SICE Annual Conference. International Session Papers, 1415-1420.
- Kong, L. F., Zhu, S. S. ve Wang, Z. 2011. Gradient genetic algorithm-based oil fault diagnosis model for automobile engine. 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2127-2130.
- Kotsiantias, S. ve Kanellopoulos, D. 2006. Association Rules Mining: A Recent Overview. GESTS International Transactions on Computer Science and Engineering, Vol.32 (1), 71-82.
- Laribi, P. 2018. Genetik Algoritma ve K-en Yakın Komşu Kullanarak Metin Belgelerinin Sınıflandırılması. Yüksek Lisans Tezi (basılmamış). Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü, Van, 45.
- Lecun, Y., Bengio, Y. ve Hinton, G. 2015, Deep Learning, Nature, 521, 436-444.
- Lili, Z., Jiangwei, C. ve Bencun, Z. 2010. Study of Automobile Electric Controlled System Fault Diagnosis Based on Classification Pattern Recognition. The 2010 IEEE International Conference on Information and Automation, 2076-2081.
- Luckow, A., Cook, M., Ashcraft, Weill, N. E., Djerekarov, E. ve Vorster, B. 2016. Deep learning in the automotive industry: Applications and tools. 2016 IEEE International Conference on Big Data (Big Data), 3759-3768.
- Martín-delosReyes, L.M., Lardelli-Claret, P., García-Cuerva, L., Rivera-Izquierdo, M., Jiménez-Mejías, E. ve Martínez-Ruiz, V. 2021. Effect of Periodic Vehicle Inspection on Road Crashes and Injuries: A Systematic Review. Int. J. Environ. Res. Public Health, 18, 6476.
- Murphey, Y. L. Ve Chen Z. 2010. A Multi-Agent System for Complex Vehicle Fault Diagnostics and Health Monitoring. 15th IEEE International Conference on Engineering of Complex Computer Systems. 52(4), 1076-1098.
- Olgun, S. 2020. Yazılım Projelerinin Yönetiminde Maliyet Tahmini için Derin Öğrenme Tabanlı Yeni Bir Yaklaşım. Doktora Tezi, İstanbul Üniversitesi Lisansüstü Eğitim Enstitüsü / Endüstri Mühendisliği Ana Bilim Dalı, İstanbul, 25.
- Satoh , S., Yakuwa, F. ve Dote, Y. 2003. Combination of radial basis function (RBF) and time delayed neural networks (TDNN) for fault diagnosis of automobile transmission gears using general parameter learning and adaptation. SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance, vol.2, 1457-1462.

- Satoh, S., Shaikh, M. S. ve Dote, Y. 2001. Fast fuzzy neural network for fault diagnosis of rotational machine parts using general parameter learning and adaptation. SMCia/01. Proceedings of the 2001 IEEE Mountain Workshop on Soft Computing in Industrial Applications, 87-91.
- Taşdelen, D. A. 2019. Veri Madenciliğinin Tıp ve Sağlık Hizmetlerinde Uygulamaları. Yüksek Lisans Tezi. Ankara Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı. Ankara, 28.
- Talonen, J., Sirola, M. ve Sulkava, M. 2012. Network Visualization of Car Inspection Data using Graph Layout. The First International Conference on Data Analytics 39-42.
- Taniguchi, S., Akhmetov, D. ve Dote, Y. 1999. Fault detection of rotating machine parts using novel fuzzy neural network. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028), vol.1, 365-369.
- Taylor, J.G. 2005. The promise of neural networks, Springer-Verlag, Versionone, 1-131.
- Türkiye İstatistik Kurumu, Çevre, Enerji ve Ulaştırma İstatistikleri Daire Başkanlığı Enerji ve Ulaştırma İstatistikleri Grubu "Araç Kilometre Metodu"
- Tüysüzöğlü, G. 2021. Data Mining and Machine Learning in Enviromental Informatics. Yüksek Lisans Tezi. Dokuz Eylül Üniversitesi. İzmir, 35.
- United Nations Economic Commission for Europe Transport Division. Handbook on Statistics on Road Traffic, 2007.
- Weijie, W., Yuanfu, K., Xuezheng, Z. ve Wentao H. 2004. Study of automobile engine fault diagnosis based on wavelet neural networks. Fifth World, Congress on Intelligent Control and Automation, vol.2, 1766-1770.
- Winkler, J. ve Vogelsang, A. 2016. Automatic Classification of Requirements Based on Convolutional Neural Networks. 2016 IEEE 24th International Requirements Engineering Conference Workshops (REW), 39-45.
- Yao, Z. Ve Pan, H. 2010. Study on wavelet package analysis technology applied to fault diagnosis of vehicle transmission. The 2nd International Conference on Intelligent Computation Technology and Automation, Vol. 2, 182-185.
- Yaraş, N. 2020. Vehicle Type Classification with Deep Learning. Yüksek Lisans Tezi. İzmir Yüksek Teknoloji Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı, İzmir, 11.
- Yılmaz, A. 2019. Derin Öğrenme. Kodlab Yayın, İstanbul, 6.
- Youjun, Y., Xiang, L. ve Qun, Z. 2011. Development of automobile fault diagnosis expert system based on fault tree — Neural network ensemble. 2011 International Conference on Electronics, Communications and Control (ICECC), 2011, 2028-2031.

- Zafari, A., Kianmehr, M.H. ve Abdolazadeh, R. 2013. Modeling the effect of extrusion parameters on density of biomass pellet using artificial neural network. *Int J Recycl Org Waste Agricult* 2, 9.
- Zheng, S., Han, Z., Tang H. ve Zhang, Y.,2004. Application of support vector machines to sensor fault diagnosis in ESP system. *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, vol.6, 3334-3338.



EK 1

Araç muayene ve kusur tablolarını birleştiren sorgu

```
SELECT *
FROM (select c.MUAYENE_ID, c.kod, c.[MUAYENE_SONUC_KODU], count(*) sayi from (select b.MUAYENE_ID,
SUBSTRING(k.UBAK_KODU,1,CHARINDEX('.', k.UBAK_KODU)-1) kod
, b.[MUAYENE_SONUC_KODU]
from [dbo].[ARAC_2015_MART] b
inner join KUSUR_2015_MART k on b.MUAYENE_ID=k.MUAYENE_ID
left join [dbo].[KUSUR_ACIKLAMA] p on k.UBAK_KODU=p.KODU --and p.KUSUR_GRUBU='Ağır Kusur'
Union all
select b.MUAYENE_ID, SUBSTRING(k.UBAK_KODU,1,CHARINDEX('.', k.UBAK_KODU)-1) kod
, b.[MUAYENE_SONUC_KODU]
from [dbo].[ARAC_2015_OCAK] b
inner join [dbo].[KUSUR_2015_OCAK] k on b.MUAYENE_ID=k.MUAYENE_ID
left join [dbo].[KUSUR_ACIKLAMA] p on k.UBAK_KODU=p.KODU --and p.KUSUR_GRUBU='Ağır Kusur'
Union all
select b.MUAYENE_ID, SUBSTRING(k.UBAK_KODU,1,CHARINDEX('.', k.UBAK_KODU)-1) kod
, b.[MUAYENE_SONUC_KODU]
from [dbo].[ARAC_2015_SUBAT] b
inner join [dbo].[KUSUR_2015_SUBAT] k on b.MUAYENE_ID=k.MUAYENE_ID
left join [dbo].[KUSUR_ACIKLAMA] p on k.UBAK_KODU=p.KODU --and p.KUSUR_GRUBU='Ağır Kusur'
Union all
select b.MUAYENE_ID, SUBSTRING(k.UBAK_KODU,1,CHARINDEX('.', k.UBAK_KODU)-1) kod
, b.[MUAYENE_SONUC_KODU]
from [dbo].[ARAC_2016_MART] b
inner join [dbo].[KUSUR_2016_MART] k on b.MUAYENE_ID=k.MUAYENE_ID
left join [dbo].[KUSUR_ACIKLAMA] p on k.UBAK_KODU=p.KODU --and p.KUSUR_GRUBU='Ağır Kusur'
Union all
select b.MUAYENE_ID, SUBSTRING(k.UBAK_KODU,1,CHARINDEX('.', k.UBAK_KODU)-1) kod
, b.[MUAYENE_SONUC_KODU]
from [dbo].[ARAC_2016_OCAK] b
inner join [dbo].[KUSUR_2016_OCAK] k on b.MUAYENE_ID=k.MUAYENE_ID
left join [dbo].[KUSUR_ACIKLAMA] p on k.UBAK_KODU=p.KODU --and p.KUSUR_GRUBU='Ağır Kusur'
Union all
select b.MUAYENE_ID, SUBSTRING(k.UBAK_KODU,1,CHARINDEX('.', k.UBAK_KODU)-1) kod
, b.[MUAYENE_SONUC_KODU]
from [dbo].[ARAC_2016_SUBAT] b
inner join [dbo].[KUSUR_2016_SUBAT] k on b.MUAYENE_ID=k.MUAYENE_ID
left join [dbo].[KUSUR_ACIKLAMA] p on k.UBAK_KODU=p.KODU --and p.KUSUR_GRUBU='Ağır Kusur'
Union all
select b.MUAYENE_ID, SUBSTRING(k.UBAK_KODU,1,CHARINDEX('.', k.UBAK_KODU)-1) kod
, b.[MUAYENE_SONUC_KODU]
from [dbo].[ARAC_2017_MART] b
inner join [dbo].[KUSUR_2017_MART] k on b.MUAYENE_ID=k.MUAYENE_ID
left join [dbo].[KUSUR_ACIKLAMA] p on k.UBAK_KODU=p.KODU --and p.KUSUR_GRUBU='Ağır Kusur'
Union all
select b.MUAYENE_ID, SUBSTRING(k.UBAK_KODU,1,CHARINDEX('.', k.UBAK_KODU)-1) kod
, b.[MUAYENE_SONUC_KODU]
from [dbo].[ARAC_2017_OCAK] b
inner join [dbo].[KUSUR_2017_OCAK] k on b.MUAYENE_ID=k.MUAYENE_ID
left join [dbo].[KUSUR_ACIKLAMA] p on k.UBAK_KODU=p.KODU --and p.KUSUR_GRUBU='Ağır Kusur'
Union all
select b.MUAYENE_ID, SUBSTRING(k.UBAK_KODU,1,CHARINDEX('.', k.UBAK_KODU)-1) kod
, b.[MUAYENE_SONUC_KODU]
from [dbo].[ARAC_2017_SUBAT] b
inner join [dbo].[KUSUR_2017_SUBAT] k on b.MUAYENE_ID=k.MUAYENE_ID
left join [dbo].[KUSUR_ACIKLAMA] p on k.UBAK_KODU=p.KODU --and p.KUSUR_GRUBU='Ağır Kusur'
) c
group by c.MUAYENE_ID,c.kod,c.MUAYENE_SONUC_KODU
having count(*)>=1
--order by c.MUAYENE_ID
) as gTablo
PIVOT
(
count(sayi)
FOR kod IN ([1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12],[13],[14],[15])
)
AS p
```

EK 2

Makine öğrenmesi Python kodları

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

data = pd.read_csv("tum_veri.csv", sep = ";")
x_data = data.drop(["MUAYENE_ID", "DAHILI_ARAC_ID", "MUAYENE_SONUC_KODU"], axis=1)
y_data = data["MUAYENE_SONUC_KODU"]

data.MUAYENE_SONUC_KODU = [0 if each == "HK" else 1 if each == "AK" else 2 for each in
data.MUAYENE_SONUC_KODU]

from sklearn import preprocessing
le = preprocessing.LabelEncoder()
x_data['YAKIT_TIPI'] = le.fit_transform(x_data['YAKIT_TIPI'])
x_data['CINSI'] = le.fit_transform(x_data['CINSI'])
x_data['KULLANIM_AMACI'] = le.fit_transform(x_data['KULLANIM_AMACI'])
x_data['TESCIL_SEKLI'] = le.fit_transform(x_data['TESCIL_SEKLI'])

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size = 0.3, random_state=1)

#Logistic Regression
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(x_train, y_train)
print("test accuracy {}".format(lr.score(x_test, y_test)))
y_pred = lr.predict(x_test) #tahmin edilen değerler
y_true = y_test
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_true, y_pred)
# Roc curve
plt.plot_roc(y_true, predicted_proba)

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

EK 3

DNN Python kodları

```
import keras
from keras.models import Sequential
from keras.layers import Dense #nöron oluşturmak için

classifier = Sequential() #boş yapay sinir ağını oluşturur
classifier.add(Dense(11, init = 'uniform', activation = 'relu' , input_dim = 21))
classifier.add(Dense(11, init = 'uniform', activation = 'relu'))
classifier.add(Dense(3, init = 'uniform', activation = 'softmax'))
classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy' , metrics = ['accuracy']
)
classifier.fit(X_train, y_train, epochs=5)
y_pred = classifier.predict(X_test)
y_pred = (y_pred > 0.5)

pred_train= classifier.predict(X_train)
scores = classifier.evaluate(X_train, y_train, verbose=0)
print('Accuracy on training data: {}% \n Error on training data: {}'.format(scores[1], 1 -
scores[1]))

pred_test= classifier.predict(X_test)
scores2 = classifier.evaluate(X_test, y_test, verbose=0)
print('Accuracy on test data: {}% \n Error on test data: {}'.format(scores2[1], 1 - scores2[1]))

import scikitplot.metrics as splt
# confusion matrix
y_true = y_test.argmax(axis=1)
y_pred2 = y_pred.argmax(axis=1)
splt.plot_confusion_matrix(y_true,y_pred2)

from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, y_pred )
print("DNN Accuracy: %.2f%%" %(accuracy * 100.0))

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

EK 4

CNN Python kodları

```
import warnings
import tensorflow as tf
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau, EarlyStopping
from keras.layers import Conv1D, MaxPool1D
from keras.layers import Dense, Dropout, Flatten
from keras.models import Sequential
from keras.utils.np_utils import to_categorical
from sklearn import preprocessing
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

warnings.filterwarnings('ignore')
physical_devices = tf.config.list_physical_devices('GPU')
tf.config.experimental.set_memory_growth(physical_devices[0], True)

data = pd.read_csv("tum_veri.csv", sep=";")

model = Sequential()
model.add(Conv1D(32, 2, input_shape=(21, 1), activation='relu')) # convolution
model.add(MaxPool1D(pool_size=2)) # pooling
model.add(Conv1D(64, 2, activation='relu'))
model.add(MaxPool1D(pool_size=2)) # pooling
model.add(Flatten()) # flatten
model.add(Dense(128, activation='relu')) # fc
model.add(Dropout(0.3)) # dropout
model.add(Dense(num_classes, activation='softmax'))

# model compile
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.summary()

batch_size = 128
epochs = 10

checkpoint_callback = ModelCheckpoint(f'model_v2.h5', monitor='val_loss', verbose=1, save_best_only=True,
mode='min', save_weights_only=False)
learning_rate_reduction = ReduceLROnPlateau(monitor='val_loss', patience=2, verbose=1, factor=0.3,
min_lr=0.000001)
early_stop = EarlyStopping(monitor='val_loss', mode='min', patience=3)

callbacks = [checkpoint_callback, learning_rate_reduction, early_stop]

if train:
    model.fit(X_train, y_train,
            batch_size=batch_size,
            epochs=epochs,
            verbose=1,
            validation_data=(X_test, y_test)) #, callbacks=callbacks
    model.evaluate(X_test, y_test)
else:
    model.load_weights('model.h5')
```