T.C. ONDOKUZ MAYIS ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

ENDÜSTRİ MÜHENDİSLİĞİ BÖLÜMÜ

Makine Öğrenmesi Yöntemleri ile Müşteri Kayıp Analizi

Hazırlayanlar:

17060447 – **Ahmet Can KAHRAMAN**
16061637 – **Doğuhan CANKURT**
17060494 – **Furkan GÜRDAL**
17060472 – **Hakan KARABIYIK**

Danışman:

Dr. Öğretim Üyesi BARIŞ ÖZKAN

KASIM 2020

SAMSUN

T.C.

ONDOKUZ MAYIS ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

ENDÜSTRİ MÜHENDİSLİĞİ BÖLÜMÜ

Makine Öğrenmesi Yöntemleri ile Müşteri Kayıp Analizi

17060447 – **Ahmet Can KAHRAMAN**
16061637 – **Doğuhan CANKURT**
17060494 – **Furkan GÜRDAL**
17060472 – **Hakan KARABIYIK**

Danışman:

Dr. Öğretim Üyesi BARIŞ ÖZKAN

Samsun, 2020

İÇİNDEKİLER

Sayfa

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Endüstri Mühendisliği Tarım Dersi çalışması
Makine Öğrenmesi Yöntemleriyle Müşteri Kayıp Analizi
Ahmet Can KAHRAMAN
Doğuhan CANKURT
Furkan GÜRDAL
Hakan KARABIYIK
Ondokuz Mayıs Üniversitesi
Endüstri Mühendisliği Bölümü
2020

Telekomünikasyon sektöründe hizmet sağlayıcılar gelişmeye meyillidir. Firmalar hayatta kalma ihtiyacını karşılamak için mevcut müşterileri elde tutması büyük bir zorluktur. Yapılan anketlerde yeni müşteri edinmenin eldeki müşteriyi tutmaktan daha maliyetli olduğu görülmüştür. Bu nedenle telekom endüstrisinde müşteri verilerini toplanarak kaybedilebilecek müşterilerin bulunması ve gerekli önlemlerin alınması önemlidir. Bu çalışmada python kullanılarak kayıp analizi yapılmıştır. Makine öğrenmesi, fazla veri bulunan kümelerden anlamlı, işe yarar bilgileri ortaya çıkarabilir. Bu teknik son yıllarda sürekli gelişmektedir. Makine öğrenmesi ilk olarak veri temizleme işlemi ile başlar. Daha sonra hangi sınıflandırma algoritmaları uygulanacağı bulunur ve en iyi sonucu veren sınıflandırma algoritması belirlenir. Model üzerinden yorumlamalar yapılır.

Makine öğrenmesi firmalarda ayrılma gösterebilecek müşterilerin analizinde de kullanılabilmektedir. Yapılmış olan bu çalışmada telekomünikasyon firmasına ait müşterilerin, firmadan ayrılabilecek olanları üzerinde analizler yapılarak; ayrılabilecek müşteriler tahmin edilmeye çalışılmıştır. Firmadan vazgeçen müşteri analizinde sınıflandırma algoritmaları kullanıldı. Sınıflandırma algoritmaları karşılaştırılarak en yüksek doğruluğa sahip algoritma Random Forest algoritması olduğu bulunmuştur. Daha sonra müşteri kaybını en fazla etkileyen değişkenler ortaya çıkarılmış ve bunlar üzerinden nasıl müşteri kaybının azaltılabileceği üzerine öneriler sunulmuştur.

Anahtar Kelimeler: Ayrılan müşteri analizi, makine öğrenmesi, sınıflandırma, müşteri kaybı.

BÖLÜM BİR

GİRİŞ

1.1. Giriş

Teknolojinin sürekli olarak ilerlemesi ile şirketlerde müşteri rekabeti artmıştır. Şirketler pazar payını arttırmayı düşünse de pazar payını korumakta büyük bir önem kazanmıştır. Şirketlerde yeni müşteriler edinme, tavsiye edilebilirlik ve eldeki müşterileri kaçırmama ancak iyi müşteri ilişkileri yöntemi ile oluşturulabilir.

Çoğu şirketin yaptığı araştırmalara göre yeni müşteri elde etmenin maliyetinin, eldeki müşteriyi tutmanın maliyetinden daha fazla olduğu ortaya çıkmıştır. Bu yüzden eldeki müşterinin kaybedilmemesi değerli bir hale gelmiştir. Müşterilerin kaybedilmemesi eldeki müşterilerin ürün ve hizmet kullanımına devam etmesinden geçmektedir. Şirketler reklamları, kampanyaları, müşteri ilişkilerini iyi kullanarak müşterilerle aktif ilişki içinde olması gereklidir. Bu sayede müşterileri etkin hale getirmek, eldeki müşteri ile karlılık artırılmaya çalışılmalıdır.

Firmalar kazançlarını büyük bir bölümünü müşterilerin %20'sinden kazanırlar. Bu yüzden müşteriyi kaybetme olasılığını önceden tahmin etmek çok önemlidir. Eğer kaybedilen müşteriler kazancın büyük bir bölümünü sağlayan müşterilerden ise kaybedilen müşterinin önemi daha da artmaktadır.

Gelişen teknoloji ile daha çok telekomünikasyon, bankacılık, sigortacılık gibi işlerde milyonlarca müşterinin verisine ulaşılması kolaylaşmıştır. Elimizde bulunan bu veri setlerini düzenli bir hale getirerek onlardan yararlanabiliriz. Bu verileri veri madenciliği, makine öğrenmesi ile kullanışlı hale getirerek, aboneliği iptal edecek müşterileri, iptal etme potansiyeli olan müşterileri tespit etme mümkündür.

Telekomünikasyon sektörü hayatımızın her alanında var olmaya başlamıştır. Cep telefonlarıyla birlikte cebimize kadar girmiştir. Buna ek olarak bilgisayarlarda, televizyonlarda vb. kullanılmaktadır. Bununla birlikte telekomünikasyon şirketleri müşterilerinin rakip firmalara gitmesini önlemek için müşteri verilerini kullanarak, verileri analiz ederek müşterilerin ayrılma eğilimlerini önceden tespit edip müşteri kaybını yok etmek önemlidir.

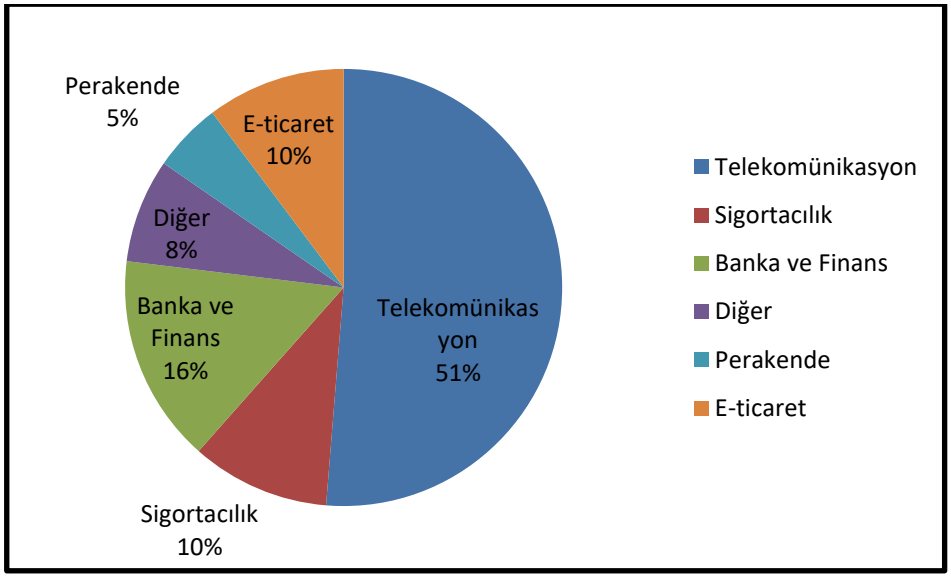
Telekomünikasyon, bankacılık, sigortacılık ve e-ticaret gibi sektörlerde şirketlerin çok yüksek sayılarda hizmet veya ürünlerini kullanan müşterileri vardır. Sayıları milyonlara ulaşan müşterilere ait eski verilerde düşünüldüğünde elimizde fazlaca bir veri kitlesi oluşmaktadır. Bu veriler kullanılarak müşteri kayıpları üzerinde çalışmalar yapılmaktadır. Bu çalışmamızda da son zamanlarda iyice gelişen makine öğrenmesi ile verilerin işlenmesi ve modellenmesi sağlanmıştır. Model, iptale gitmeyi düşünen müşterileri ortaya çıkaracak şekilde oluşturulmuş. Böylece iptale gitmeyi düşünen müşterileri önceden belirleyerek gerekli önlemlerin alınması sağlanabilir. Bu şekilde yeni müşteriler kazanmak maliyetli olduğu için eldeki müşteriyi kaybetmeyerek kar sağlamış olunur.

1.2. Literatür Araştırması

Müşteriler firmalar için en büyük katkıyı sağlayan sermaye olarak görülmeye başlanmıştır. Bu nedenle müşteri kayıp(ayrılma) analizleri önem kazanmıştır. Müşteri kaybı, müşterinin önce eğilim gösterip daha sonra rekabette olunan farklı firmalara yönelip onlardan hizmet almaya başlamasıdır. Bu bölümde müşteri kaybı analizleri ile ilgili çalışmalar incelenmiştir.

İnceleme yapılan 39 tez ve makale vb. yazılarına göre müşteri kaybı analizi çalışmalarının sektörlerde kullanımı oranları farklıdır. Aşağıdaki şekil 1.1.'deki pasta grafiğinde telekomünikasyon, sigortacılık, banka ve finans, perakende ve diğer sektörlerde kullanım oranı belirtilmiştir.

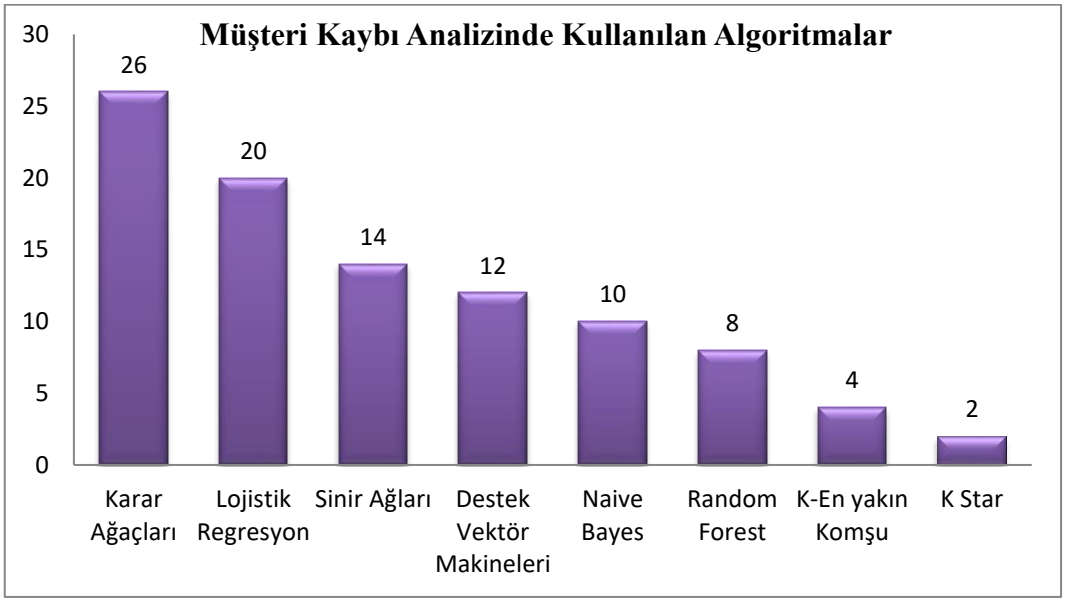
Şekil 1.1. Sektörlere göre müşteri kaybı analizi yapılma yüzdeleri



Müşteri kaybı analizi %51'lik oranla en fazla telekomünikasyon sektöründe yapılmıştır. Telekomünikasyon sektörünü %16'lık oranla banka ve finans sektörü takip ediyor. Daha sonra %10'luk oranlarla sigortacılık ve e-ticaret sektörleri geliyor. Geri kalan %13'lük kısmı ise %8'lik oranla diğer (sağlık, oyun, yazım vb.) sektörler ve %5'lik oranla perakende sektörü oluşturuyor

Yapılan literatür taramasında müşteri kaybı analizinde kullanılan sınıflandırma algoritmalarının sayıları çıkarılmış ve şekil 1.2'deki grafikte gösterilmiştir.

Şekil 1.2. Müşteri kaybı analizinde kullanılan algoritmalar



İncelenen tez ve makalelerin 26 tanesinde karar ağacı algoritması kullanılarak en fazla kullanılan sınıflandırma algoritmasıdır. Karar ağacı algoritmasını 20 çalışmada kullanılan lojistik regresyon takip etmektedir. Sinir ağları 14, destek vektör makineleri 12, Naive Bayes algoritması 10, random forest algoritması 8 çalışmada kullanılmıştır. Çalışmalarda en az kullanılan sınıflandırma algoritmaları 4 çalışmada kullanılan K-en yakın komşu algoritması ile 2 çalışmada kullanılan k star algoritmasıdır.

Aşağıdaki paragraflarda taraması yapılan bazı tez ve makalelerle ilgili bilgilere yer verilmiştir.

Dolgun, Özdemir ve Oğuz (2009), yayınladıkları 'Metin ve Web Madenciliği' isimli makalede telekomünikasyon şirketinin 2070 müşterisine ait 17 değişkenden oluşan verileri kullanılarak şirketten ayrılan müşterilerin analizi yapılmıştır. Bu analiz için karar ağacı algoritmalarından C5.0 kullanılmıştır. Telekomünikasyon şirketinden ayrılan müşterilerin ayrılma eğilimi göstermelerindeki sebebin yurt dışı dolanımı olduğu ortaya çıkmıştır.

Wu ve Zhang (2010), Çin'deki e-ticarette müşteri kaybı üzerine yaptıkları bir araştırmada sürekli müşteriler ile istikrarlı bir kar düzeyinin sürdürüldüğü belirtiliyor. Ayrıca müşteri kazanmanın zor ve maliyetli olduğu da vurgulanıyor. Bu yüzden kayıp tahmini giderek daha da önem bir hal aldığı ortaya çıkmaktadır. Yapılan bu çalışmada diğer kayıp analizlerinden farklı olarak veri madenciliği tekniklerine dayalı kayıp modeli müşterilerin anket verileri temelinde oluşturulmuş ve test edilmiştir.

Guo ve Qin (2015), Çin'de yayınlanan E-ticarette Müşteri Kaybetmelerinin Karar Ağacına Göre Analizi adlı makalede, e-ticarette rekabet gücünün artmasıyla müşterileri elde tutmak zor bir hal almıştır. Bu çalışmada müşterilerin kaybedildiğini ortaya çıkarmak için karar ağacı algoritması kullanılarak e-ticarette müşteri kaybetmenin analizi yapılmıştır.

Tosun (2006), Yapı Kredi Banka'sının müşterileri bilgileri ile yapılan bir tez çalışmasında kredi kartı müşterilerinin farklı özellikteki bilgileri veri madenciliği metotları ile incelenerek elde tutulamayan müşterilerin ortaya çıkarılması amaçlanmıştır. Bu çalışma 30000 adet müşteri verisi üzerinde yapılmıştır. Nedenleri ortaya çıkarmakta çok etkili olan karar ağacı yöntemi kullanılmıştır. Bu çalışmada oluşturulan C algoritması yeni eklemeler yapılacak şekilde tasarlanmıştır.

Lİao, Chen, Liu ve Chiu (2015), Tayvan da bir üniversitede yaptıkları çalışmada çevrimiçi oyunlarının sayısının artması müşteri sadakatsizliğini ortaya çıkardığı belirtilmektedir. Çalışmada müşteri kaybını tahminleme ve pazarlama stratejilerine dikkat çekilmiştir. Müşteri kaybı tahminlemesinde hibrit bir sınıflandırma modeli üzerinde duruldu. Uygulanan deneysel sonuçlarda hibrit modelinin bu çalışmaya çok uygun bir model olduğu ortaya çıktı.

Wanchai (2017), Tayland'da yazdığı bir makalede, Tayland Telekomünikasyon Endüstrisinde müşteri kaybı analizi yapılmıştır. Yapılan çalışmada öncelikle verilere ulaşıldı. Daha sonra Weka yazılımında karar ağacı algoritması, lojistik regresyon ve yapay sinir ağları algoritmaları kullanılarak analiz yapılmıştır. Modeller arasından en iyi sonucu karar ağacı algoritması verdiği ortaya çıkmıştır.

Odabaş (2017), yaptığı tez çalışmasında makine öğrenmesi ve veri madenciliği ile müşteri ayrılma analizi yapmıştır. Bu çalışmada sınıflandırma algoritması da kullanılmıştır. Sınıflandırma algoritması ile ortaya çıkan modellerin sonuçları çapraz geçerleme ve hold-out performans metotları ile değerlendirilmiştir. En iyi sonucu oluşturulan modellerden C4.5 karar ağacı algoritması vermiştir.

Spiteri ve Azzopardi (2018), Araç Sigortası için Müşteri Kaybı Analizi çalışmalarında mevcut müşterileri kaybetmenin çok fazla geliri kaybettirdiği vurgulanıyor ve yeni müşteri kazanmanın da maliyetli olacağından bahsediliyor. Şirketin kaybedebileceği müşterilerini tahmin edebilmek için Weka yazılımı ile müşteri kaybı analizi yapıyor. Analizde ise sinir ağları, lojistik regresyon ve karar ağacı algoritmaları kullanılıyor.

Kişioğlu (2009), yaptığı tez çalışmasında telekomünikasyon sektöründe aboneliğini iptal etmeye eğilimli müşterilerin özellikleri Bayes ağlarıyla model kurularak incelenmiştir. Veriler Türkiye içinden bir telekomünikasyon firmasından alınmıştır. Sürekli değişkenler CHAİD karar ağacı algoritması ile kesikli hale dönüştürülerek Bayes ağlarına uygun hale getirilmiştir. Yapılan analiz ile nedensel bir harita oluşturulmuştur. İptal nedenlerinin ortalama faturalar, konuşma süreleri, tarifelerin gibi farklı sebepler olduğu ortaya çıkmıştır. Bu çalışma sonunda abonelerin iptal sayılarını düşürmek için önerilerde bulunulmuştur. Ayrıca bu çalışmaya kadar müşteri kaybı analizinde Bayes ağları kullanılmamıştır.

Günay ve Ensari (2018), Türkiye'de yaptıkları çalışmada kaybedilecek müşterilerin tahmininde bulunmak için fazlaca bilinen makine öğrenmesi kullanılarak sınıflandırma algoritmalarının başarıları ölçülmüştür. Genel olarak bilinen sınıflandırma algoritmalarından karar ağacı, yapay sinir ağları, Naive Bayes algoritmalarına ek olarak hibrit bir yöntem tasarlanmıştır. Hibrit olarak geliştirilen yöntemde lojistik regresyon ve Naive Bayes algoritmaları kullanılmıştır. Bu iki algoritmanın ayrı ayrı kullanıldığı duruma göre hibrit yöntemin başarı oranının çok daha iyi olduğu ortaya çıkmıştır. Bu çalışmada en iyi sonucu yapay sinir ağları vermiştir.

Başkal (2019), yaptığı telekomünikasyon sektöründe müşteri segmentasyonu ve müşteri kaybı analizi çalışmasında, telekomünikasyon şirketindeki müşterilerin ayrılmayı düşündüğünde bunun farkına varıp bu müşteriyi elde tutmaya çalışmak amaçlanmıştır. Veri madenciliği, makine öğrenmesi teknikleri ile Weka, Python gibi programlar eğitilerek hangi müşterinin kaybedileceği ortaya çıkartılarak, müşteri kaybını azaltarak ortadan kaldırmayla ilgili bir çalışma yapılmıştır.

Kunt (2019), yaptığı çalışmada ise ürün ve hizmetten ayrılma ihtimali gözüken müşterilerin karar ağaçları, random forest, Xgboosting yöntemleri ve fazlaca kullanılan sınıflandırma yöntemlerinden olan Naive Bayes ve lojistik regresyon ile ne şekilde ortaya çıkarılacağı incelenmiştir. Bu yöntemler arasında en başarılı sonucu Xgboosting

yöntemi vermiştir. Uygulama olarak ta diğer çalışmalardan farklı KNİME uygulaması kullanılmıştır.

İncelen çalışmalarda Weka, Python, Knime, Microsoft Structured Query Language (Ms Sql), Statistical Package for the Social Sciences (SPSS) gibi teknolojilerin kullanıldığı görülmektedir.

1.3. Telekomünikasyon Sektörü

Teknolojinin gelişmesiyle birlikte günümüzde telekomünikasyon daha da önem kazanmaya başlamıştır. Telekomünikasyon temelde her tür metin, sinyal, video veya ses iletimini sağlar. Bu nedenle bilgi alışverişini gerçekleştirmek önemli bir teknolojidir. İletişim katılımcıları bu bilgi akışını telekomünikasyon veya elektromanyetik radyasyon yöntemleriyle uzanan kablolar aracılığıyla sağlar. Telekomünikasyon çok geniş bir kavramdır ve aynı anda birden fazla teknoloji anlamına gelir. Geçtiğimiz birkaç yıl içinde insanlar güvercinler, ıslıklar ve boynuzlarla iletişim kurmaya çalıştılar. Günümüzün modern dünyasında, elektromanyetik dalgalar önemli mesafelerde bile başarılıdır. Günümüzde sıklıkla kullandığımız telefon, internet ve fiber optik teknolojisi telekomünikasyonun en somut örnekleridir. Ekonomi ve toplum için önemli bire yere sahip olan iletişim sektörü son yıllarda büyük bir ilerleme göstererek insanlık için vazgeçilmez bir yere gelmiştir. Telekomünikasyon alanı için son dönemlerde hız kazanarak gelişen bu teknoloji, telekomünikasyon için var olabilmenin en büyük koşulu haline gelmiştir. Henüz süreçler tamamlanmamış olabilir ancak ileride her bireyin veya her nesnenin birbirine bağlantılı hale gelebileceğini şimdiden görebiliyoruz. Bu durum ise insan yaşamı üstünde tehlikeli yaratabilecek bir öneme sahiptir.

Türkiye'de telekomünikasyonun geçmişi Osmanlı döneminde, 23 Ekim 1840 (Türkiye'de Telekomünikasyon, 2013) tarihinde Sultan Abdülmecit tarafından Postahane-i Amirane'nin kurulması ile başlamaktadır. 1855 yılında Telgraf Teftiş Bürosu kurulduktan sonra posta ve telgraf teftişleri 1871'de birleştirilerek Posta ve Telgraf Nezaretine dönüştürüldü. Telefon hizmeti, Temmuz 1881'de Yeni Cami postanesi ile İstanbul Soğuk Çeşme'deki postane arasında tek hatlı bir telefonla sağlandı.1911'de Western Electric Power'a verilen 30 yıllık imtiyaz karşılığında, Türkiye'nin ilk telefon sistemi bu şirketler adına işletilen Dersaadet adlı bir telefon şirketi tarafından kuruldu. Birinci Dünya Savaşı sırasında şirket ülke tarafından işgal

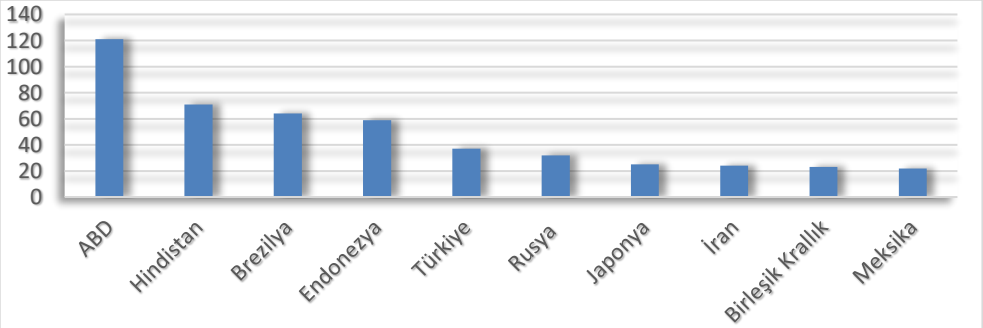
edilmiş ve 1935 yılına kadar İstanbul ve çevresinde telefon hizmeti vermeye devam etmiştir. Cumhuriyet kurulduktan sonra 1936'da 800.000 Türk lirasına satın alındı. Cumhuriyetin ilanından sonraki dönemde Osmanlı Devleti tarafından devralınan ve özel şirketler tarafından yapılan telgraf ve telefon görüşmeleri 1924 tarihli Telgraf ve Telefon Kanunu ile yeniden düzenlenmiştir.

Posta, Telgraf ve Telefon İdaresi (PTT) 1936 yılında kurulmuştur ve misyonu ülke çapında telekomünikasyon hizmetleri sunmaktır. PTT'nin idari departmanı, 1939'da Ulaştırma Bakanlığı'na bağlıydı ve 1953'te ekonomik bir kamu iktisadi teşebbüsü olarak yeniden düzenlendi. Hükümetin 1983 yılında iktidara geldiği ekonomik model ve ülke ekonomisindeki ağırlığını azaltma politikası çerçevesinde kamu sektörü imalat yatırımları azalmış, haberleşme, enerji ve ulaşımda altyapı yatırımları yoğunlaştırılmıştır. Bu dönemde özellikle telekomünikasyon sektöründe altyapı yatırımlarının hız kazanmasının bir parçası olarak Türkiye'de önemli değişiklikler yaşanırken, Turgut Özal döneminde Türkiye, ITU üyesi Avrupa ülkeleri arasında ana hat sayısı artışında en fazla gelişme oranını gösteren ülke olmuştur.

Telekomünikasyon sektöründeki gelişmeler dünyada değil Türkiye'de de yaygın bir hal alarak internet ve sosyal medya kullanımı önemli derecede artmıştır. Günümüzde nüfusun yaklaşık yüzde 48'i internet kullanımını aktif bir şekilde gerçekleştirmektedir. Yapılan araştırmalara bakarak internet kullanımının en büyük nedeninin sosyal medya olduğu görülmektedir. Büyük bir kullanıcı kitlesine sahip olan Instagram'ı kullanan ülkeler açısından sınıflandıracak olursak Türkiye bu sınıflandırmada beşinci sırada yer almaktadır. Instagram, 2018'de Avrupa'nın en hızlı büyüyen sosyal ağları arasındaydı. Alman istatistik portalı Statista tarafından yayınlanan verilere göre, Ekim 2018 itibarıyla dünyada en çok Instagram kullanıcısı olan ülkeler belirlendi.

Aşağıda şekil 1.3' de 2018 yılı sonlarında Instagram kullanıcılarının ülkelere göre gruplandırılmış sayısı gösterilmektedir.

Şekil 1.3. 2018 yılında Dünya'da ve Türkiye'de Instagram kullanım oranları



İnternet kullanıcıları dünyada olduğu gibi Türkiye'de de hızlı bir şekilde artmaktadır. Genel olarak bakacak olursak 82,4 milyon nüfusa sahip ülkemizde, nüfusun yüzde 72'sini oluşturan 59,36 milyon internet kullanıcısı, nüfusun yüzde 63'ünü oluşturan 52 milyon aktif sosyal medya kullanıcısı ve nüfusun yüzde 53'ünü oluşturan 44 milyon aktif mobil sosyal medya kullanıcısı bulunmaktadır.

Yıllık olarak Türkiye'de dijital değişim istatistiklerine baktığımızda, internet kullanımında kullanıcılarda yüzde 9 olarak yani 5 milyonluk bir artış görülüyor. Aktif sosyal medya kullanıcı sayısı da 2 milyon artmaktadır. Aktif mobil sosyal medya kullanıcı sayısı geçen yıl ile aynı.

2020'nin ikinci yarısından itibaren, küresel dijital manzara hızla gelişmeye devam ediyor ve devam eden korona virüs salgını, insanların günlük yaşamlarının tüm yönlerini etkilemeye ve yeniden şekillendirmeye devam ediyor. Pek çok ülke yaşamdaki kısıtlamaları kaldırmış olsa da insanların #evdekal önleminde benimsediği birçok yeni dijital davranış artış göstermiştir. Artık dünyanın yarısından fazlası sosyal medya kullanmakta ve #Evdekal zamanlarında oluşan dijital alışkanlıklar, kısıtlamalar kalksa dahi devam etti. Sosyal medya kullanıcıları geçen yıla göre%10'dan fazla arttı ve Temmuz 2020'nin başlarında küresel toplamları 3,96 milyara ulaştı. Bu, ilk defa dünya nüfusunun yarısından fazlasının sosyal medyayı kullanması anlamına gelmektedir. Büyüme trendi, son 12 ayda ortalama 1 milyondan fazla insanın her gün sosyal medyayı kullanmaya başladığını ve bu da saniyede neredeyse 12 yeni kullanıcı olduğunu gösteriyor. Son üç yıla baktığımızda, mobil ve İnternet kullanıcılarının büyümesi önemli bir büyüme gösterdi. Ancak sosyal medya kullanıcıları da artmaya devam ediyor. Aşağıda şekil 1.4'te de gösterilmektedir.

Şekil 1.4. Türkiye'de sosyal medya kullanıcı sayısı

Yaşanan tüm gelişme ve ilerlemeler, telekomünikasyon alanında yatırımın ne kadar gerekli ve önemli olduğunu göstermiştir. Telekomünikasyon sektörü bu kadar hız kazanırken sektörde müşteri kayıplarının da incelenip önlenmesi bu hıza katkı sağlayacaktır.

BÖLÜM İKİ

MATERYAL- METOD

2.1. Makine Öğrenmesi Nedir?

Makine Öğrenmesi tanımını yapmadan önce sıklıkla birbiri yerine kullanılan veri madenciliği kavramı ile arasındaki farkı açıklamak gerekir. Veri madenciliği, büyük miktarda veriden faydalı bilgi çıkarma sürecine denir. İnsanlar tarafından verilerdeki yeni, doğru ve yararlı kalıpları veya ihtiyaç duyanlar için anlamlı ve ilgili bilgileri keşfetmek için kullanılan bir araçtır.

Yapay zekâ ve bilgisayar oyunları alanında öncü olan Arthur Samuel, "Makine Öğrenimi" terimini icat etti. Arthur Samuel bunu "Bilgisayarlara açıkça programlanmadan öğrenme yeteneği veren çalışma alanı" olarak tanımladı.”

Basit olarak Makine Öğrenimi (ML), aslında programlanmadan, yani herhangi bir insan yardımı olmadan bilgisayarların deneyimlerine dayanarak öğrenme sürecini otomatikleştirmek ve iyileştirmek olarak açıklanabilir. Süreç, kaliteli verileri beslemek ve ardından verileri ve farklı algoritmaları kullanarak makine öğrenimi modelleri oluşturarak makinelerimizi (bilgisayarları) eğitmekle başlar. Algoritma seçimi, sahip olduğumuz verinin türüne ve ne tür bir göreve otomatikleştirmeye çalıştığımıza bağlıdır.

2.2. Makine Öğrenmesi Türleri

Makine öğrenmesi türleri denetimli, denetimiz, yarı denetimli ve pekiştirmeli öğrenme olarak dörde ayrılır.

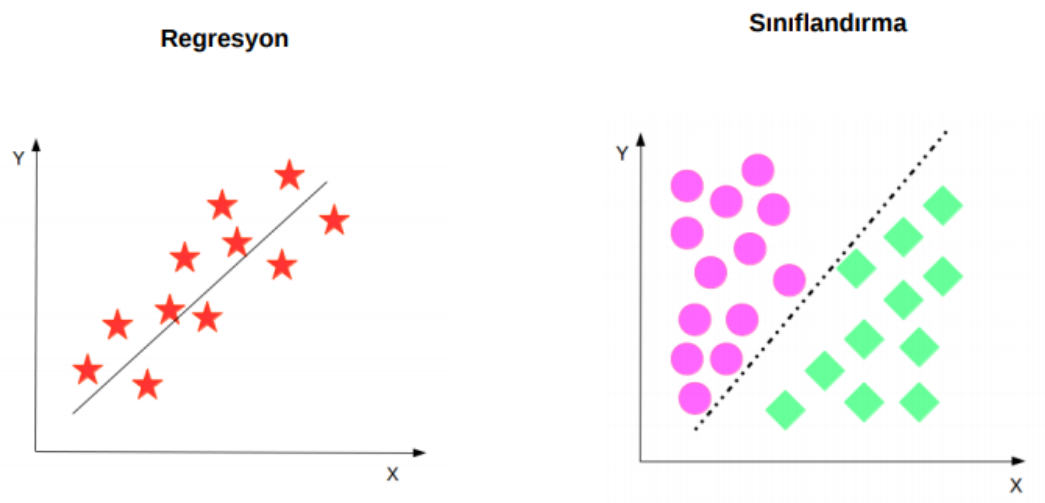
2.2.1. Denetimli Öğrenme

Denetimli öğrenmede veriler etiketlidir ve tahmin yapmak için model eğitilir. Girdi verisini ve doğru çıktı verisini kullanıcı sağlar. Bu süreç, eğitim verilerini ve o verilerden çıkan sonuçları sisteme vererek bu bilgilerle modelin eğitiminin sağlanması ve bir fonksiyon oluşturulmasını sağlamaktadır. Model, girdi verileri ile doğru çıktı arasındaki ilişkileri ve bağımlılıkları regresyon ve sınıflandırma problemlerini çözmek için belirlemeye çalışır. Regresyon modelleri, sayısal veya "gerçek" bir değeri tahmin etmek için kullanılır. Örneğin gayrimenkul piyasasında bir parametreye bağlı konut

satış fiyatları tahmininde, hisse senedi piyasası tahmini de regresyon problemidir. Sınıflandırma modelleri ise girdinin hangi kategoriye veya "sınıfa" ait olduğunu tahmin etmek için kullanılır. Örneğin, evin “belirli X fiyattan daha fazla veya daha düşük fiyatla satılıp satılmadığını” öğrenmek için bu örneği bir sınıflandırma problemine dönüştürebiliriz.

Firmaların verimliliğini ve rekabet gücünün tahmin edilmesi veya etki yüzdesi daha yüksek ilaçların geliştirilmesi problemleri regresyon problemi uygulama örneklerindendir. Hastanın gözlemlenen özelliklerine dayalı olarak belirli bir hastaya tanı atama, gelen e-postaları spam veya spam olmayan e-posta olarak ayırmak bir sınıflandırma problemi uygulama örneklerindendir. Şekil 2.1’de regresyon ve sınıflandırma örneği grafiksel olarak gösterilmiştir.

Şekil 2.1. Regresyon ve sınıflandırma örneği

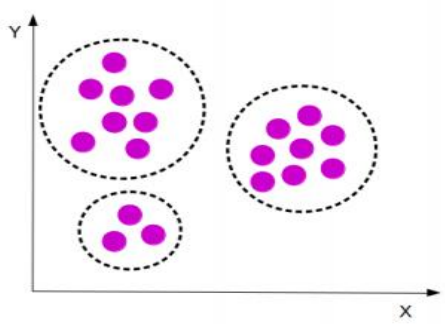


2.2.2. Denetimsiz Öğrenme

Denetimsiz öğrenmede verilerimiz etiketsizdir. Yani çıktı bir sınıf veya kategori ile ayrıştırılmadığı durumlarda kullanılan makine öğrenmesi türüdür. Eğitim ihtiyacı yoktur. Model, verileri analiz eder ve yalnızca kendi verisinin özelliklerine dayanarak verilerdeki örüntüleri ve yapıyı tanımlamaya çalışır. Denetimsiz öğrenmede tahmin etme amacı yoktur. Mevcut durumdaki benzerliğin analizi söz konusudur. Bu yaklaşımı problemlere göre kümeleme (clustering) ve ilişkilendirme (association) problemleri olarak ayırabiliriz. Kümeleme, popülasyonu veya veri noktalarını birkaç gruba ayırma yöntemidir. İlişkilendirme, verilerin parametrelerinin arasındaki yararlı ilişkileri bulan denetimsiz öğrenme yöntemidir. Şekil 2.2. de kümeleme ve ilişkilendirme örneği

gösterilmiştir. Başka bir örnek olarak da markette A ürününü satın alanların B ürününü de satın almaya eğilimli olması verilebilir.

Şekil 2.2. Kümeleme ve ilişkilendirme örneği



2.2.3. Yarı Denetimli Öğrenme

Bu tür öğrenmede algoritma, etiketli ve etiketlenmemiş verilerin bir kombinasyonu üzerine eğitilir. Bu kombinasyon çok az miktarda etiketli veri ve çok büyük miktarda etiketlenmemiş veri içerir. Temel proses, programcının önce denetimsiz bir öğrenme algoritması kullanarak benzer verileri kümelemesi ve ardından etiketlenmemiş verilerin geri kalanını etiketlemek için mevcut etiketli verileri kullanmasıdır.

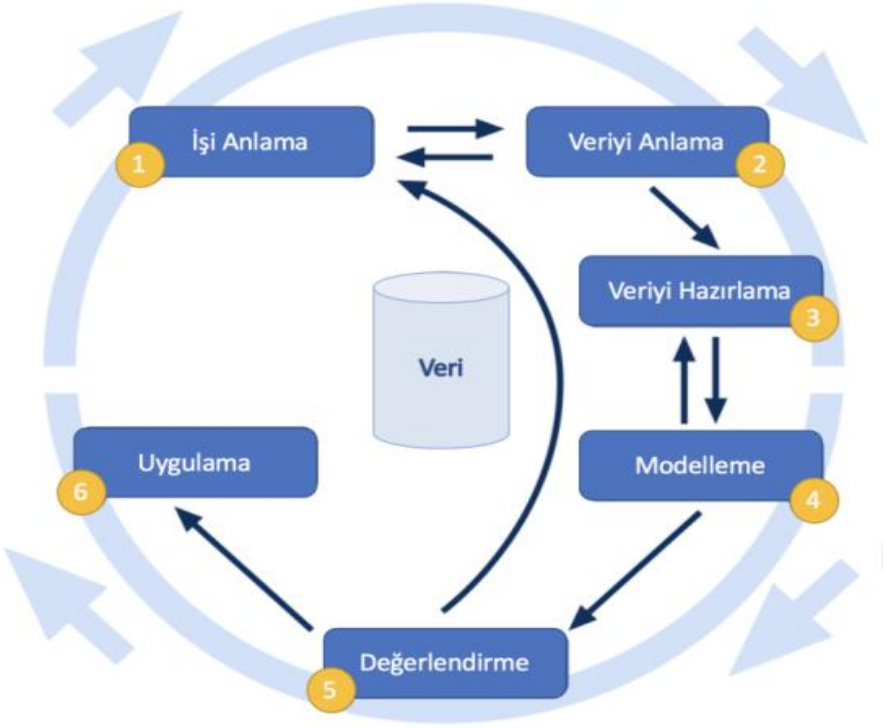
2.2.4. Pekiştirmeli Öğrenme

Bu yaklaşımda sistemin doğru kararlar sonucunda ödüllendirilip, yanlış kararlarda cezalandırılması söz konusudur. Amaç, uygun eylemleri uygulayarak ödülü en üst seviyeye çıkarmaktır.

2.3. CRISP-DM Metodolojisi

CRIPS-DM metodolojisi, makine öğrenmesi ile ilgilenen kişi, grup veya kuruluşlar tarafından en çok tercih edilen yöntemdir. Makine öğrenmesi sürecinde atılması gereken adımları detaylı bir şekilde açıklayan yöntem bilimidir. Metodolojinin 6 adımı şekil 2.3'te gösterilmiştir.

Şekil 2.3. CRISP-DM metodolojisinin altı adımı



Genel olarak 6 adımdan oluşmaktadır. 6 adım çalışmanın bu kısmında detaylı olarak anlatılmıştır.

2.3.1. İşi Anlama

CRISP-DM yönteminin ilk adımı olan işi anlama, projenin içeriğinin, gereklerinin ve hedeflerinin tanımlanmasıdır. Analist, bilgiyi veri madenciliği problemi olarak ifade eder ve ön plan geliştirir. Ön plan kısaca kim, neyi, ne zaman, nerede, niçin ve nasıl yapmalı sorularının cevaplarının bulunmasıdır.

İşi anlama adımı doğru şekilde yapılmadığı sürece, diğer adımlar ne kadar kusursuz yapılmış olsa bile sonuç güvenilmez olacaktır. Maksimum fayda sağlamak için işi anlama adımına gereken önem verilmelidir.

2.3.2. Veriyi Anlama

Veriyi anlama adımında ilk iş olarak, çalışma hedeflerine uygun veriler toplanır. Birden fazla kaynaktan ve değişik biçimlerde olabilen veriler, hiçbir şekilde değiştirilmeden incelenerek sonuçlarla birlikte raporlanır. Böylece eldeki verilerin makine öğrenmesi çalışması için yeterli olup olmadığı, eksik ya da yanlış veri içerip içermediği durumları incelenir ve bir sonraki adım olan veriyi hazırlama adımında, verinin nasıl hazırlanacağı veya hangi açıdan temizlenmesi gerektiğin ortaya çıkar.

2.3.3. Veriyi Hazırlama

Veriyi hazırlama adımında, bir sonraki adım olan modelleme adımı için her türlü veri hazırlığı işlemi yapılır. Makine öğrenmesi çalışmaları tek bir kaynaktan alınan veriler ile de yapılabilir, farklı kaynaklardan alınan veriler ile de yapılabilir. Farklı kaynaklardan alınan veriler birbirleriyle uyumsuzluk içinde olabilirler. Örneğin ölçü birimlerinin farklı ifade edilmesi. İşte bu tür uyumsuzluklar bu adımda giderilecektir.

Günümüzde hiçbir veri hazır şekilde bulunmaz. Bu nedenle veri hazırlama işine kesinlikle ihtiyaç duyulur. Bundan dolayı makine öğrenmesi çalışmalarında en çok çaba sarf edilen adımdır. Veri hazırlama aşamasındaki adımları şu şekilde inceleyebiliriz:

2.3.3.1. Veri Seçimi

Analiz için kullanılacak verilere karar verilen adımdır. Bu kararı vermek için kullanılacak kriterler arasında verinin makine öğrenmesi hedeflerimizle ilgisi, verinin kalitesi, veri hacmi veya veri türleri üzerinde sınırlamalar yer alır. Veri seçiminin, bir tablodaki kayıtların (satırların) seçiminin yanı sıra özniteliklerin (sütunların) seçimini de kapsadığını unutmayalım. Çok fazla öznitelik ile çalışmak zaman açısından ve sonuçların güvenilirliği açısından sorun çıkarabilir. Dolayısıyla sonuçlara doğrudan etki edeceği belirlenen veriler veya öznitelikler göz önüne alınmalıdır ve bu öznitelikler doğrultusunda çalışmaya devam edilmelidir.

2.3.3.2. Veri Temizleme

Seçtiğimiz veri arasında eksik, tutarsız veya yanlış girilmiş veriler olabilir. Bu aşamada bu tarz olumsuz durumlar ortadan kaldırılarak daha kaliteli veri setine dönüştürme işlemi yapılır. Örneğin araba satış sitelerinden alınan bir veri setinde aracın modeli kısmında 2050 yazıyorsa bu veri hatalıdır.

2.3.3.3. Veri Oluşturma

Makine öğrenmesi çalışması yapılırken, belirlediğimiz öznitelikler yetersiz olabilir. Bu gibi durumda yeni öznitelik üretilmesi ihtiyaç duyulabilir. Örneğin bir telekomünikasyon şirketinde müşteri bilgilerinin kayıtlı olduğu bir tablo düşünelim. Bu tabloda müşterinin kullandığı internetin boyutu ile ödediği ücret arasında sıkı bir ilişki olduğu görülür. İnternet boyutunun ücretine oranı ele alınarak yeni bir öznitelik oluşturulabilir.

2.3.3.4. Veri Birleştirme

Farklı kaynaklardan gelen aynı özniteliklere veya farklı özniteliklere sahip verileri tek bir tabloda birleştirme işlemidir.

2.3.3.5. Veri Biçimlendirme

Farklı kaynaklardan alınan veriler aynı özniteliği farklı semboller, kısaltmalar ile ifade edebilir. Örneğin cinsiyet özniteliğinin bazı verilerde E/K, bazı verilerde 0/1 olarak ifade edilmesi. Bu tür farklılıkların yok edildiği aşamadır.

2.3.4. Modelleme

Makine öğrenmesi çalışmalarının en önemli adımlarından biri olan modelleme adımında, elimizdeki veriye en uygun veri seçilmelidir. Verilerimize en uygun modelin seçimi ise çeşitli modelleme teknikleri kullanılarak oluşturulan modellerin denenmesi ile mümkün olacaktır. Dolayısıyla veri hazırlama ve modelleme adımlarına, en uygun modeli bulunca dönüşler yapılabilir ve işlemler tekrarlanabilir.

Model oluşturma süreci denetimli ve denetimsiz öğrenmenin kullanıldığı modellere göre farklılık göstermektedir. Denetimli öğrenmede kanıta dayalı tahminler yapan modeller oluşturulur. Denetimli öğrenme, verileri ve o verilerden çıkan sonuçları makineye tekrar baştan vererek bu bilgilerden bir fonksiyon çıkartılmasını sağlamaktır yani giriş verileri ile çıkış verileri arasında bir eşleşme oluşturulur. Böylece makine veriler arasında bağlantı kurar ve verileri birbirleriyle ilişkilendirir. Örneğin araba piyasasındaki arabaların özellikleri ile ilgili veriler verildiğinde, araç fiyatlarının önceden belirlenmesi. Denetimsiz öğrenmede ise modelin denetlenmeye ihtiyaç duyulmadığı bir makine öğrenme tekniğidir. Makine, veri kümesindeki verileri yorumlayarak ortak olanı bulmak ve bunları kümeleştirme işlemi yaparak anlamlı bir veri elde edebilmektedir. Örneğin alışveriş sitelerinde bir alışveriş sitesinde sepete eklenen bir ürünün yanında alınabilecek diğer ürünlerin tavsiye olarak müşteriye sunulması.

2.3.5. Değerlendirme

Değerlendirme adımında önceki adımlarda yapılan bütün işlemler kontrol edilir ve modelleme ile edilen sonucun çalışmanın en başında tanımlanan iş hedeflerine tam

olarak cevap verip vermediği değerlendirilir. Tekrarlanan değerlendirmeler sonucunda önceki adımlarda değişiklik ihtiyacı duyulabilir.

Bir önceki adımda kurmuş olduğumuz model veya modellerin kontrol edilip değerlendirilmesi çeşitli yöntemler vardır. Bu yöntemler arasından en basit ve en kullanışlı olanı basit geçerlilik testidir. Bu yöntemde verilerimizin %5 ile %33 arasındaki bölümü test verileri olarak ayrılır ve geriye kalan verilerimiz ile model öğrenimi çalışması yapılır. Daha sonra test olarak ayırdığımız veriler üzerinden testler gerçekleştirilir. Bir sınıflama modelinde yanlış olarak sınıflanan durum sayısının, tüm durum sayısına bölünmesi ile hata oranı; doğruluk olarak sınıflanan durum sayısının, tüm durum sayısına bölünmesi ile doğruluk oranı hesaplanır.

Sınırlı sayıda veriye sahip olunan durumlarda ise kullanılan yöntem çapraz geçerlilik testidir. Bu yöntemde verilerimiz rastgele iki eşit parçaya ayrılır. İlk olarak seçilen bir parça üzerinden model öğrenimi yapılır ve diğer parça ile test işlemleri yapılır. Daha sonra ise ikinci parça üzerinden model öğrenimi yapılır ve diğer parça ile test işlemleri yapılır. Yapılan iki değerlendirme sonucunda elde edilen hata oranlarının ortalaması kullanılır.

2.3.6. Uygulama

Çalışma sonucunda elde edilen bilgilerin, müşterilerin veya son kullanıcıların anlayabileceği şekilde düzenlenmesi ve sunulmasıdır. Buradaki önemli olan noktalar; dağıtım planının oluşturulması, planın izlenmesi, projenin gözden geçirilmesidir ve son raporun hazırlanmasıdır.

2.4. Makine Öğrenmesi Algoritmaları

2.4.1. Naive Bayes Algoritması

Bu sınıflandırma algoritması Bayes teoreminden ortaya çıkmıştır ve denetimli bir öğrenme algoritmasıdır. Bu algoritma ile sınıflandırma yaparken her bir değeri öteki değerlerden bağımsız bir şekilde sınıflandırılır. Örneğin bir ağacın çam ağacı olma olasılığı yapraklarının şekli ve kışın yaprak dökmesini birbirinden ayrı ele alarak tahminde bulunur. Yaprakları iğne şeklinde ise +1, kışın yaprak dökmüyorsa +1 şeklinde birbirinden bağımsız olarak değerlendirilir (Kunt, 2019).

Bu yöntemi uygulayarak ortaya çıkaracağımız her bir kategorinin gerçekleşme olasılığını birbirinden bağımsız olarak hesaplamamız gerekmektedir. Bu formül Bayes teoreminden türetilebilir. Formül aşağıdaki gibidir.

$$P(X\backslash C_i = \prod_{k=1}^{n} P(x_k\backslash C_i) = P(x_1\backslash C_i) * P(x_2\backslash C_i) * \dots * P(x_n\backslash C_i)$$

2.4.2. Lojistik Regresyon

Lojistik regresyon, istatiksel sınıflandırma algoritmasıdır ve denetimli öğrenme algoritmalarından da bir tanesidir. Sadece iki değer sahip neticelerin olasılığını bulabilir. Tahmin, tek veya daha fazla öngörücünün (numerik ve kategorik) uygulamasına dayanır. Lineer regresyon 0 ve 1 aralığından farklı tahminde bulunabilirken, lojistik regresyon ise sadece 0 ve 1 arasındaki değerler ile bir eğri ortaya çıkarır (Kazan vd., 2019).

2.4.3. Lineer Regresyon

Lineer regresyon sınıflandırıcısı denetimli bir öğrenme algoritmasıdır. Bu model, tek veya birden çok bağımsız değişkenle diğer bağımlı değişkenin bağlantısını modellemek için kullanılır. Lineer regresyonda amaç verilmiş olan a'ları ve b'leri kullanarak z değerlerini ortaya çıkarmaktadır. Z'leri elde ettikten sonra b sayıları bilinmese bile a değerleri verildiği için b'de hesaplanmaktadır (Peker vd., 2017). Aşağıdaki gibi formülüze edilebilir:

a_0=1 olmak şartıyla;

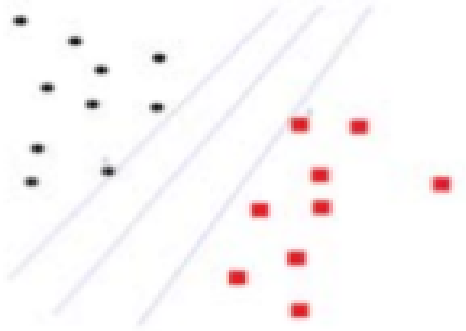
f(a) $\sum z_i a_i = za$

2.4.4. Destek Vektör Makineleri

Bu sınıflandırıcı denetimli öğrenme algoritmalarından biridir. İyi performans veren ve kolay bir düşünce ile ortaya çıkarılmış bir sınıflandırma algoritmasıdır. Destek vektör makineleri, kuram olarak iki sınıfa sahip modelleri doğrusal bir şekilde bölen destek vektörlerine dayanır. Şekil 2.4'te görüldüğü üzere iki farklı sınıf sonsuz sayıda doğru ayırmaktadır. Ancak iki sınıfı en iyi ayıran doğrular bunlar değildir. Şekil 2.5'te görüldüğü gibi iki sınıfı en iyi bölen doğru iki sınıfa da belirli uzaklıkta olan doğrudur. Bu düşünce ile oluşturulan bu sınıflandırma algoritması çok başarılıdır. Birçok yerde kullanılmaktadır (Özgür vd., 2012).

Şekil 2.4. Sınıfları ayıran sonsuz sayıda doğru örneği

Şekil 2.5. İki Sınıfı Ayıran Doğru Örneği (Destek Vektörleri)



2.4.5. Karar Ağacı

Sınıflandırmada kullanılan karar ağacı, denetimli öğrenme algoritmalarından bir tanesidir. Bu algoritma tüm eylem alternatiflerini, etkileyebilecek etkenleri ve bu etkenlere dayanan mümkün sonucu, verilere göre değerlendirebilen, yuvarlak, dikdörtgen, çizgi vb. simgeleri kullanarak kararı verecek kişiye problemi anlamada basitlik sağlayan grafiksel yöntemdir. Bu yöntem grafiksel olarak problemi ayrıntılarıyla ortaya çıkarmaktadır. Birden çok kararın peş peşe verileceği durumlarda oldukça kullanışlıdır (Kazan vd., 2019).

2.4.6. Random Forest Algoritması

Bu sınıflandırma yöntemi denetimli öğrenme algoritmasıdır. Sınıflandırma için kullanılabilen random forest algoritması, regresyon içinde tercih edilmektedir. Kullanılması kolay ve esnektir. Gelişigüzel ormanlardan gelişigüzel tercih edilmiş verilerden karar ağacı meydana getirir, her bir ağaçtan tahmin meydana çıkarır ve oylama yaparak en ideal çözümü tercih eder (Kazan vd., 2019).

2.4.7. K-Ortalamalar Algoritması

K-ortalamalar algoritması en çok kullanılan ve en basit makine öğrenme algoritmasıdır. K-ortalamalar algoritması, veriyi birbiriyle çakışmayan i adet farklı kümeye ayıran bir yöntemdir. İlk olarak verinin kaç adet kümeye (i) ayrılacağı belirlenmelidir, daha sonra K-ortalamalar algoritması gözlemlerden her birinin yalnızca bir kümenin elemanı olacak şekilde atamasını gerçekleştirir. Bu kümeleme metodu sezgisel ve basit bir matematiksel problemin bir sonucudur.
K-ortalamalar algoritmasının formül olarak gösterimi aşağıdaki gibidir:

$$J(V)= \sum_{i=1}^{c} \sum_{j=1}^{c_i} \left(\|x_i - v_j\| \right)^2$$

Burada; $\|x_i - v_j\|$, x ve y arasındaki öklid mesafesi, ci ,i kümesindeki veri noktalarının sayısı, c ise küme merkezlerinin sayısıdır.

K-Ortalamalar Algoritması adımları:

X = { x_1, x_2, ….. , x_n } kümesi veri noktalarının kümesi, V = { v_1, v_2, ….. , v_c } merkez noktalarının kümesi olsun.

1)Rastgele “c” küme merkezlerini seç.
2) Her veri ile küme merkezlerinin arasındaki mesafeyi hesapla.
3) Küme merkeziyle arasındaki mesafe, diğer küme merkezleri ile olan mesafeden daha az olan veriyi, yakın olan o küme merkezine ata.
4)Yeni küme merkezini aşağıdaki denklemle yeniden hesapla: $v_i= \left(\frac{1}{c_i}\right) \sum_{j=1}^{c_i} x_i$
5) Her verir noktasıyla, yeni küme merkezleri arasındaki mesafeyi yeniden hesapla.
6)Eğer hiçbir veri noktası atanmadıysa dur, diğer durumda üçüncü adıma dön ve tekrarla.

2.4.8. Hiyerarşik Kümeleme

Hiyerarşik kümeleme, dendrogam olarak adlandırılan ikili ağaç tabanlı veriyi meydana getirerek kümeleme yapmayı amaçlayan bir makine öğrenmesi algoritmasıdır. Dendrogam meydana getirildikten sonra, algoritmayı tekrar başlatmadan, aynı veri kümesi üzerinden değişik kümeleme çözümlerine ulaşmak için ağaç farklı seviyelerde

parçalanarak otomatik olarak doğru sayıda kümenin seçimi gerçekleştirilebilir. Hiyerarşik kümeleme algoritmasının başlangıcında verinin kaç adet kümeye ayrılacağının belirlenmesine gerek yoktur. Hiyerarşik kümeleme birleştirici kümeleme (Agglomerative Clustering) ve ayrıştırıcı kümeleme (Divisive Clustering) olmak üzere 2 alt grupta incelenebilir:

Birleştirici kümelemede öncelikle belirlenen bir yakınlık ölçüsüne göre benzerlik matrisi oluşturulur. Örnek sayısı kadar küme olduğu varsayılır. En yakın kümeler her seviyede birleştirilir ve benzerlik matrisi tekrardan oluşturulur. Bu adımlara tek bir küme kalana kadar devam edilir. Bu yöntem aşağıdan yukarıya kümeleme olarak da adlandırılmaktadır.

Ayrıştırıcı kümelemede işleme tek bir küme ile başlanır. Daha sonra birbirine en az benzeyen iki örnek bu tek küme içerisinden seçilir. Bu tek küme, seçilen iki örneğe göre bölünür. Bu şekilde devam ederek her alt küme için aynı işlem uygulanır. Bu yöntem yukarıdan aşağıya kümeleme olarak da adlandırılmaktadır.

2.5. Makine Öğrenmesinde Kullanılan Programlar

2.5.1. Knime

Konstanz Information Miner, Bu, Konstanz Üniversitesi'nin Veri Madenciliği Grubu tarafından geliştirilen bir veri madenciliği programıdır. KNIME, kullanıcılara bir yazılım geliştirme IDE'si sağlar. Kullanıcılar bu IDE'yi kullanarak kendi modüllerini geliştirebilirler. Bu program için kurulum gereksinimleri yoktur. Knime'ı veri almak üzere kullanmak için, verilerin .txt uzantısı veya. arff biçiminde olması gerekir.

2.5.2. Orange

Slovenya Ljubljana Üniversitesi Bilgisayar ve Bilgi Bilimi Bölümü yapay zeka ekibi tarafından geliştirilen bir programdır. Orange yazılımı C ++ dilinde geliştirilmiştir. Verileri yalnızca metin belgelerinden alınır.

2.5.3. Rapidminer

RapidMiner, Ralf Klinkenberg, Dortmund Teknoloji Üniversitesi Yapay Zeka Bölümü'nden Ingo Mierswa ve Simon Fischer tarafından tasarlanan bir programdır. Diğer programlardan en büyük farkı 22 dosya formatında veri alabilmesidir.

RapidMiner (veri madenciliği ve makine öğrenimi algoritmaları dahil), Weka gibi epeyce algoritmaya sahiptir. Veri analizi, ön işleme ve veri madenciliği yöntemleri gibi süreçleri içerir. Oracle, MS SQL Server, MySQL, IBM DB2 dahil birçok veri tabanını ve metin dosyasını destekler. Bu bakımdan en kapsamlı yazılımlardan biridir. Excel dosyası ile bağlantı kurabilir. MS Windows, Linux, Mac Os X işletim sistemlerinde çalışabilir.

2.5.4. Weka

Waikato Environment for Knowledge Analysis kelimelerinin kısaltılmasıdır. Waikato Üniversitesi'nin Java platformunda geliştirilen ve GNU Genel Kamu Lisansı'nı alan açık kaynaklı bir veri madenciliği programıdır. SQL veri tabanına erişmek için Java veri tabanı bağlantısını (JDBC) kullanır. Tüm veri madenciliği ve makine öğrenimi algoritmalarını içerir. Veri analizi, ön işleme ve veri madenciliği yöntemleri gibi süreçleri içerir. WEKA için özel olarak tasarlanmış. Arff (öznitelik ilişkisi dosya formatı) dosya formatı için uygundur.

2.5.5. Spss

Statistical Package for the Social Sciences, 1968'de yayınlanan istatistiksel analiz için kullanılan bilgisayar programıdır. SPSS, 2009'da IBM'e satıldı. Özellikle sosyal bilimler alanında istatistiksel analiz için kullanılır. Pazarlama şirketleri, sağlık araştırmacıları, anket şirketleri, devlet kurumları ve eğitim araştırmacıları gibi birçok alanda da aktif bir şekilde kullanılmaktadır.

2.5.6. Sas

1976 yılında Anthony Barr, James Goodnight, John Sall ve Jane Helwig'den oluşan dört kişi tarafından istatistiksel analiz sistemi adıyla kurulmuştur. Bugün SAS, dünyanın en büyük borsaya açık olmayan yazılım şirketlerinden biri haline geldi. SAS, IBM, Microsoft ve Oracle gibi şirketlerle şiddetli bir rekabet içindedir. SPSS'nin 2009 yılında IBM tarafından satın alınmasıyla SAS, IBM'in rakibi oldu.

2.5.7. R

R, istatistiksel hesaplamalar için geliştirilmiş bir bilgisayar programı ve aynı zamanda bir programlama dilidir. Binlerce yazılım paketi içerir. Bu yazılım paketlerini kullanarak veri madenciliği, verileri görselleştirmek için grafikler oluşturmak gibi

birçok işlem yapılabilir. Yeni Zelanda Auckland Üniversitesi'nden Ross Ihaka ve Robert Gentleman tarafından geliştirilmiştir ve R paketlerinin ihtiyaç sayısındaki artış nedeniyle sürekli olarak geliştirilmektedir. S yazılımının yerini alabilecek açık kaynak kod olarak geliştirilmiştir. İstatistikçiler için standart haline gelmekle beraber R, istatistiksel yazılım geliştirme ve veri analizi alanında kullanılır. Genel Kamu Lisansına (GNU) tabidir ve her işletim sisteminde kullanılabilir.

2.5.8. Python

Guido van Rossum, 1990 yılında Amsterdam'da geliştirme çalışmalarına başladı. 1991'den beri, Python programlama dili yalnızca gereksiz programlar için tamamlayıcı bir dil olarak görülüyordu. Bununla birlikte, son birkaç yılda Python, modern yazılım geliştirme, altyapı yönetimi ve veri analizinde birinci sınıf bir programlama dili olarak öne çıktı. Artık bir bilgisayar korsanının arka kapı oluşturucusu değil, web uygulaması oluşturma ve sistem yönetimi, veri analizi ve makine öğreniminde baş döndürücü bir ün kazandı. Diğer karmaşık programlama dillerini öğrenmek çok zaman alır ve yaygın kullanımları nedeniyle kullanımlarını öğrenmek zordur. Bununla birlikte, Python sözdizimi okunabilir ve ileriye dönüktür. İstikrarlı bir programlama dili sayesinde öğrenmeyi kolaylaştırır. Karmaşık veri analizi, günümüzde IT'nin en önemli konusu haline geldi. Bu durumlar için Python en uygun programlama dilidir. Python ara yüzündeki birçok kitaplık, makine öğrenimi ve veri bilimi için uygundur. Bu alanlardaki kitaplıklarda sağladığı yüksek kaliteli komutlar, makine öğrenimi kitaplıklarının ve diğer sayısal algoritma kitaplıklarının sürekli iyileştirilmesine büyük ölçüde yardımcı oldu.

2016 yılında makine öğrenmesinde ve veri madenciliği programlarında en sık kullanılan 10 programı ve kullanım oranlarını görebilirsiniz.

Makine Öğrenmesi programlarını tercih oranı tablo 2.1'de gösterilmiştir.

Tablo 2.1. Veri madenciliği programlarını tercih oranı tablosu

Veri Madenciliği Programları	Kullanım Oranı (%)
R	%49
Python	%45,8
SQL	%35,5
Excel	%33,6

RapidMiner	%32,6
Hadoop	%22,1
Spark	%21,6
Tableau	%18,5
KNIME	%18,0
Scikit-learn	%17,2

Python programı, veri analizi ve makine öğreniminde baş döndürücü bir ün kazanmış olduğu, kullanım ve öğrenim kolaylığından dolayı tercih oranı hızla artmaktadır. Bu durum yukarıdaki tabloda da kullanım oranının yüksek olmasından anlaşılmaktadır. Bizde kullanacağımız programı seçerken çok fazla tercih edilmesi, kullanım ve öğrenim kolaylığı olduğu için çalışmamızda Python programını kullanmayı tercih ettik.

BÖLÜM ÜÇ

ARAŞTIRMA BULGULARI VE İRDELEME

3.1. Problemin Tanımlanması

Sektörlerde en çok müşteri kaybı yapılma analizlerinin Telekomünikasyon sektöründe yapılıyor olması ve Telekomünikasyon sektörünün hızla gelişmesi ile bu sektörde artık müşteri kayıplarının da ön planda tutulmaya çalışılması ve yeni müşteri edinmenin eldeki müşteriyi firmada tutmaktan daha maliyetli olmasından dolayı Telekomünikasyon sektörü seçilmiş olup bu sektörde müşteri kayıp analizi yapılmaktadır. Pandemi dolayısı ile herhangi bir Telekom sektöründen veri toplama imkânı olmadığı için kullandığımız veri seti hazır verilerden elde edilmiştir. Telekom sektöründeki elde etmiş olduğumuz verilerle beraber firmadan ayrılma eğilimi gösteren müşteriler belirlenmektedir. Ayrılma eğilimi olan müşterilerin, neden sunulmuş olan hizmeti bırakıp, rakip firmalara yöneldiğini anlayabilmek için gerekli sorulara yanıt aranır. Elde edilen yanıtlara göre müşteri kaybını önlemek amacıyla uygun stratejiler belirlenir ve geliştirilir.

3.2. Veriyi Anlama ve Veriyi Hazırlama

Çalışma yapılan Orange Telekomünikasyon firmasına ait veri seti .csv formatında bulunmaktadır. Şekil 3.1' de veri setinin .csv formatındaki görüntüsü gösterilmiştir.

Şekil 3.1. Veri setinin .csv formatı

```
KS, 128, 415, 382-4657, no, yes, 25, 265.1, 110, 45.07, 197.4, 99, 16.78, 244.7, 91, 11.01, 10, 3, 2.7, 1, False.
OH, 107, 415, 371-7191, no, yes, 26, 161.6, 123, 27.47, 195.5, 103, 16.62, 254.4, 103, 11.45, 13.7, 3, 3.7, 1, False.
NJ, 137, 415, 358-1921, no, no, 0, 243.4, 114, 41.38, 121.2, 110, 10.3, 162.6, 104, 7.32, 12.2, 5, 3.29, 0, False.
OH, 84, 408, 375-9999, yes, no, 0, 299.4, 71, 50.9, 61.9, 88, 5.26, 196.9, 89, 8.86, 6.6, 7, 1.78, 2, False.
OK, 75, 415, 330-6626, yes, no, 0, 166.7, 113, 28.34, 148.3, 122, 12.61, 186.9, 121, 8.41, 10.1, 3, 2.73, 3, False.
AL, 118, 510, 391-8027, yes, no, 0, 223.4, 98, 37.98, 220.6, 101, 18.75, 203.9, 118, 9.18, 6.3, 6, 1.7, 0, False.
MA, 121, 510, 355-9993, no, yes, 24, 218.2, 88, 37.09, 348.5, 108, 29.62, 212.6, 118, 9.57, 7.5, 7, 2.03, 3, False.
MO, 147, 415, 329-9001, yes, no, 0, 157, 79, 26.69, 103.1, 94, 8.76, 211.8, 96, 9.53, 7.1, 6, 1.92, 0, False.
LA, 117, 408, 335-4719, no, no, 0, 184.5, 97, 31.37, 351.6, 80, 29.89, 215.8, 90, 9.71, 8.7, 4, 2.35, 1, False.
WV, 141, 415, 330-8173, yes, yes, 37, 258.6, 84, 43.96, 222, 111, 18.87, 326.4, 97, 14.69, 11.2, 5, 3.02, 0, False.
IN, 65, 415, 329-6603, no, no, 0, 129.1, 137, 21.95, 228.5, 83, 19.42, 208.8, 111, 9.4, 12.7, 6, 3.43, 4, True.
RI, 74, 415, 344-9403, no, no, 0, 187.7, 127, 31.91, 163.4, 148, 13.89, 196, 94, 8.82, 9.1, 5, 2.46, 0, False.
IA, 168, 408, 363-1107, no, no, 0, 128.8, 96, 21.9, 104.9, 71, 8.92, 141.1, 128, 6.35, 11.2, 2, 3.02, 1, False.
MT, 95, 510, 394-8006, no, no, 0, 156.6, 88, 26.62, 247.6, 75, 21.05, 192.3, 115, 8.65, 12.3, 5, 3.32, 3, False.
IA, 62, 415, 366-9238, no, no, 0, 120.7, 70, 20.52, 307.2, 76, 26.11, 203, 99, 9.14, 13.1, 6, 3.54, 4, False.
NY, 161, 415, 351-7269, no, no, 0, 332.9, 67, 56.59, 317.8, 97, 27.01, 160.6, 128, 7.23, 5.4, 9, 1.46, 4, True.
ID, 85, 408, 350-8884, no, yes, 27, 196.4, 139, 33.39, 280.9, 90, 23.88, 89.3, 75, 4.02, 13.8, 4, 3.73, 1, False.
VT, 93, 510, 386-2923, no, no, 0, 190.7, 114, 32.42, 218.2, 111, 18.55, 129.6, 121, 5.83, 8.1, 3, 2.19, 3, False.
VA, 76, 510, 356-2992, no, yes, 33, 189.7, 66, 32.25, 212.8, 65, 18.09, 165.7, 108, 7.46, 10, 5, 2.7, 1, False.
TX, 73, 415, 373-2782, no, no, 0, 224.4, 90, 38.15, 159.5, 88, 13.56, 192.8, 74, 8.68, 13, 2, 3.51, 1, False.
FL, 147, 415, 396-5800, no, no, 0, 155.1, 117, 26.37, 239.7, 93, 20.37, 208.8, 133, 9.4, 10.6, 4, 2.86, 0, False.
CO, 77, 408, 393-7984, no, no, 0, 62.4, 89, 10.61, 169.9, 121, 14.44, 209.6, 64, 9.43, 5.7, 6, 1.54, 5, True.
AZ, 130, 415, 358-1958, no, no, 0, 183, 112, 31.11, 72.9, 99, 6.2, 181.8, 78, 8.18, 9.5, 19, 2.57, 0, False.
SC, 111, 415, 350-2565, no, no, 0, 110.4, 103, 18.77, 137.3, 102, 11.67, 189.6, 105, 8.53, 7.7, 6, 2.08, 2, False.
VA, 132, 510, 343-4696, no, no, 0, 81.1, 86, 13.79, 245.2, 72, 20.84, 237, 115, 10.67, 10.3, 2, 2.78, 0, False.
NE, 174, 415, 331-3698, no, no, 0, 124.3, 76, 21.13, 277.1, 112, 23.55, 250.7, 115, 11.28, 15.5, 5, 4.19, 3, False.
WY, 57, 408, 357-3817, no, yes, 39, 213, 115, 36.21, 191.1, 112, 16.24, 182.7, 115, 8.22, 9.5, 3, 2.57, 0, False.
MT, 54, 408, 418-6412, no, no, 0, 134.3, 73, 22.83, 155.5, 100, 13.22, 102.1, 68, 4.59, 14.7, 4, 3.97, 3, False.
MO, 20, 415, 353-2630, no, no, 0, 190, 109, 32.3, 258.2, 84, 21.95, 181.5, 102, 8.17, 6.3, 6, 1.7, 0, False.
HI, 49, 510, 410-7789, no, no, 0, 119.3, 117, 20.28, 215.1, 109, 18.28, 178.7, 90, 8.04, 11.1, 1, 3, 1, False.
IL, 142, 415, 416-8428, no, no, 0, 84.8, 95, 14.42, 136.7, 63, 11.62, 250.5, 148, 11.27, 14.2, 6, 3.83, 2, False.
NH, 75, 510, 370-3359, no, no, 0, 226.1, 105, 38.44, 201.5, 107, 17.13, 246.2, 98, 11.08, 10.3, 5, 2.78, 1, False.
LA, 172, 408, 383-1121, no, no, 0, 212, 121, 36.04, 31.2, 115, 2.65, 293.3, 78, 13.2, 12.6, 10, 3.4, 3, False.
AZ, 12, 408, 360-1596, no, no, 0, 249.6, 118, 42.43, 252.4, 119, 21.45, 280.2, 90, 12.61, 11.8, 3, 3.19, 1, True.
OK, 57, 408, 395-2854, no, yes, 25, 176.8, 94, 30.06, 195, 75, 16.58, 213.5, 116, 9.61, 8.3, 4, 2.24, 0, False.
GA, 72, 415, 362-1407, no, yes, 37, 220, 80, 37.4, 217.3, 102, 18.47, 152.8, 71, 6.88, 14.7, 6, 3.97, 3, False.
AK, 36, 408, 341-9764, no, yes, 30, 146.3, 128, 24.87, 162.5, 80, 13.81, 129.3, 109, 5.82, 14.5, 6, 3.92, 0, False.
MA, 78, 415, 353-3305, no, no, 0, 130.8, 64, 22.24, 223.7, 116, 19.01, 227.8, 108, 10.25, 10, 5, 2.7, 1, False.
AK, 136, 415, 402-1381, yes, yes, 33, 203.9, 106, 34.66, 187.6, 99, 15.95, 101.7, 107, 4.58, 10.5, 6, 2.84, 3, False.
NJ, 149, 408, 332-9891, no, no, 0, 140.4, 94, 23.87, 271.8, 92, 23.1, 188.3, 108, 8.47, 11.1, 9, 3, 1, False.
```

Veriyi anlama ve verinin hazırlanması amacıyla veri, Python programlama dilinin de kullanılabildiği Jupyter Notebook IDE ‘ sine aktarıldı. Numpy, Pandas ve Scikit Learn kütüphanelerinden yararlanıldı.

Veri setinde 3333 gözlem birimi (satır) ve 21 değişken (sütun) bulunmaktadır. Şekil 3.2’ de görülen kod yardımıyla; verinin aktarımı, kütüphanelerin aktif edilmesi ve sütunlara başlık (header) bilgilerinin girilmesi sağlanmıştır.

Şekil 3.2. Verinin aktarımı, kütüphanelerin aktif edilmesi ve sütunlara başlıkların girilmesi kodu

```
import numpy as np
np.random.seed(1)  #yapılan işlemlerde sonuçların rastgele üretilmesini sağlar
from sklearn import cross_validation
from sklearn import preprocessing

import pandas as pd

columns = [
     'state',
     'account length',
     'area code',
     'phone number',
     'international plan',
     'voice mail plan',
     'number vmail messages',
     'total day minutes',
     'total day calls',
     'total day charge',
     'total eve minutes',
     'total eve calls',
     'total eve charge',
     'total night minutes',
     'total night calls',
     'total night charge',
     'total intl minutes',
     'total intl calls',
     'total intl charge',
     'number customer service calls',
     'churn']
data = pd.read_csv('ChurnDataset.txt', header = None, names = columns)
```

Veri setinde 6 kategorik değişken, 15 sayısal değişken bulunmaktadır. Bu değişkenlerin Türkçe tanımları ve değişken türleri Tablo 3.1’ de gösterilmiştir.

Tablo 3.1. Telekomünikasyon veri setine ait değişkenler, değişken türleri ve Türkçe tanımları

DEĞİŞKENLER			
	DEĞİŞKEN BAŞLIKLARI	AÇIKLAMASI	DEĞİŞKEN TÜRÜ
1	state	Eyalet	Kategorik
2	account length	Müşterililik süresi	Sayısal
3	area code	Alan kodu	Kategorik
4	phone number	Telefon numarası	Kategorik
5	international plan	Yurtdışı aramalarla ilgili paket vb.	Kategorik
6	voice mail plan	Sesli aramalarla ilgili paket vb.	Kategorik
7	number vmail messages	Sesli mesaj sayısı	Sayısal
8	total day minutes	Toplam gündüz konuşma süresi (dakika)	Sayısal
9	total day calls	Toplam gündüz arama sayısı	Sayısal
10	total day charge	Toplam gündüz ücretlendirme	Sayısal
11	total eve minutes	Toplam akşam konuşma süresi (dakika)	Sayısal
12	total eve calls	Toplam akşam arama sayısı	Sayısal
13	total eve charge	Toplam akşam ücretlendirme	Sayısal
14	total night minutes	Toplam gece konuşma süresi(dakika)	Sayısal
15	total night calls	Toplam gece arama sayısı	Sayısal
16	total night charge	Toplam gece ücretlendirme	Sayısal
17	total intl minutes	Toplam uluslararası konuşma süresi (dakika)	Sayısal
18	total intl calls	Toplam uluslararası konuşma sayısı	Sayısal
19	total intl charge	Toplam uluslararası ücretlendirme	Sayısal
20	number customer service calls	Müşteri hizmetleri aranma sayısı	Sayısal
21	churn	Kayıp	Kategorik

Tablo 3.1 ‘ de Arena code, phone number değişkenlerinin kategorik olmasına dikkat edilmelidir. Bu değişkenler sayısal değerler içermesine rağmen aralarında sayısal

büyüklüğün bir anlamı yoktur. Örneğin her bir “area code” (alan kodu) eyaletleri yani sayılamayan bir niteliği ifade etmektedir.

Veriyi anlama açısından betimleyici istatistiksel yöntemler kullanarak istatistiksel özeti Şekil 3.3’te gösterilmiştir. Betimsel istatistik aynı zamanda veriyi hazırlama aşamasında aykırı, eksik değerlerin tespiti açısından ön bilgi vermesi için kullanılabilir.

Verilere bakıldığında ortalama ve medyanın yakın olduğu değişken sütunlarında dağılımın homojen olduğunu söyleyebiliriz.

Şekil 3.3.Veriyi anlama açısından istatiksel özet

	count	mean	std	min	25%	50%	75%	max
account length	3333.0	101.064806	39.822106	1.00	74.00	101.00	127.00	243.00
area code	3333.0	437.182418	42.371290	408.00	408.00	415.00	510.00	510.00
number vmail messages	3333.0	8.099010	13.688365	0.00	0.00	0.00	20.00	51.00
total day minutes	3333.0	179.775098	54.467389	0.00	143.70	179.40	216.40	350.80
total day calls	3333.0	100.435644	20.069084	0.00	87.00	101.00	114.00	165.00
total day charge	3333.0	30.562307	9.259435	0.00	24.43	30.50	36.79	59.64
total eve minutes	3333.0	200.980348	50.713844	0.00	166.60	201.40	235.30	363.70
total eve calls	3333.0	100.114311	19.922625	0.00	87.00	100.00	114.00	170.00
total eve charge	3333.0	17.083540	4.310668	0.00	14.16	17.12	20.00	30.91
total night minutes	3333.0	200.872037	50.573847	23.20	167.00	201.20	235.30	395.00
total night calls	3333.0	100.107711	19.568609	33.00	87.00	100.00	113.00	175.00
total night charge	3333.0	9.039325	2.275873	1.04	7.52	9.05	10.59	17.77
total intl minutes	3333.0	10.237294	2.791840	0.00	8.50	10.30	12.10	20.00
total intl calls	3333.0	4.479448	2.461214	0.00	3.00	4.00	6.00	20.00
total intl charge	3333.0	2.764581	0.753773	0.00	2.30	2.78	3.27	5.40
number customer service calls	3333.0	1.562856	1.315491	0.00	1.00	1.00	2.00	9.00

Çalışmanın devamında yapılacak öznitelik seçme işlemlerinde yapılacak testler için parametrik testleri mi yoksa non parametrik testleri mi uygulanması gerektiğini bilmemiz gerekir. Bunun için SPSS ‘te her değişken için normallik testi yapıldı. Gözlem sayımız 50‘nin üstünde olduğu için burada Kolmogorov-Smirnov yaklaşımına bakmamız daha doğru olur. Burada “Kolmogorov-Smirnov” testinin “Sig.” değerleri 0.05‘den büyük olduğu gruplar için “veriler normal dağılım gösterir.”; “Sig.” değerleri 0.05‘den küçük olan gruplar için “veriler normal dağılıma uymamaktadır.” denilir. Şekil 3.4.’ te veri setinin tamamı ele alındığında normal dağılım göstermeyen gruplar

olduğu için, veri setimiz normal dağılıma uymamaktadır. Daha sonraki yapılacak işlemler için parametrik olmayan testler (non parametrik testler) seçilmelidir.

Şekil 3.4. Normallik testi

Tests of Normality

	Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Müşterililik_Suresi	,012	3333	,200*	,998	3333	,001
Sesli_mesaj_Sayisi	,446	3333	,000	,622	3333	,000
Toplam_Gündüz_Konuşma_Suresi	,009	3333	,200*	1,000	3333	,640
Toplam_Gündüz_Arama_Sayisi	,016	3333	,040	,998	3333	,000
Toplam_Gündüz_Ücretlendirme	,009	3333	,200*	1,000	3333	,640
Toplam_Akşam_Konuşma_Suresi_dk	,012	3333	,200*	1,000	3333	,712
Toplam_Akşam_Arama_Sayisi	,016	3333	,040	,999	3333	,009
Toplam_Akşam_Ücretlendirme	,012	3333	,200*	1,000	3333	,709
Toplam_Gece_Konuşma_Süresi_dk	,008	3333	,200*	1,000	3333	,627
Toplam_Gece_Arama_Sayısı	,019	3333	,008	,999	3333	,251
Toplam_Gece_Ücretlendirme	,009	3333	,200*	1,000	3333	,624
Toplam_Uluslararası_Konuşma_Suresi	,026	3333	,000	,994	3333	,000
Toplam_Uluslararası_Konuşma_Sayısı	,163	3333	,000	,906	3333	,000
Toplam_Uluslararası_Ücretlendirme	,029	3333	,000	,994	3333	,000
Müşteri_Hizmetleri_Arama_Sayisi	,229	3333	,000	,877	3333	,000

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

Verinin hazırlanma aşaması bir makine öğrenmesi projesinin en göz ardı edilebilen fakat en önemli aşamalarından biridir. Veride; eksik, aykırı, gürültülü değerler olabileceği için verinin hazırlanmadan algoritmanın eğitilmesi yanlış sonuçlar elde etmemize sebep olacaktır. Veri hazırlama aşamasını üç alt adımda incelemek mümkündür.

3.2.1. Veri Seçimi

Veri seçimi adımında, tüm mevcut veriden tanımladığımız problemimizi etkileyen bir alt küme oluşturulacaktır. Elimizdeki tüm veriyi kullanmamız modelimizin açıklanması ve yorumlanmasını daha karmaşık hale getireceği için daha büyük veri daha anlamlıdır yaklaşımı bizi yanlışa götürebilir. Bu doğrultuda öznitelik seçimi (feature selection) işlemi yapılacaktır. Öznitelik, veri seti içerisinde değişken olarak adlandırdığımız sütunlardır. Öznitelik seçimi, hedef değişken üzerinde en iyi açıklayıcılığa sahip değişkenleri bulma; açıklayıcılığı olmayan, hedef değişkenimizi en az etkileyen değişkenleri bulma sürecidir.
Öznitelik seçiminin yararları aşağıdaki gibi sıralanabilir:

Modelde oluşabilecek overfitting durumunun önüne geçebilir. Overfitting, modelin eğitim verisini ezberleyip yüksek doğruluk oranına sahip olması fakat test verisinde aynı doğruluk oranlarına ulaşılamamasıdır.

Gereksiz öznitelikler çıkarıldığı için makine öğrenmesi süreci içinde eğitim (train) aşaması daha hızlı gerçekleşmesi sağlanır.

Veri setinde bazı değişkenler arasında çok yüksek karşılıklı korelasyon olabilir. Bu durumda her iki değişkende hedef (target) değişkenin açıklayıcılığını aynı yönde etkiler. Bu durumda değişkenlerden birinin silinmesi model başarısını artırabilir.

Değişken seçiminde kullanılan yöntemler 3 ana başlığa ayrılabilir. Bu yöntemler ve kısaca çalışma prensipleri aşağıda açıklanmıştır.

Filtreleme Yöntemleri: Girdi değişkenlerinin ayrı ayrı hedef değişkeni ile arasındaki ilişkileri değerlendirir. Süreç, modelden bağımsız oluşturulur. Ayrı olarak değerlendirilen önemsiz değişkenlerin birlikte yüksek öneme sahip olabilmesi bu yöntemin dezavantajlarındandır.

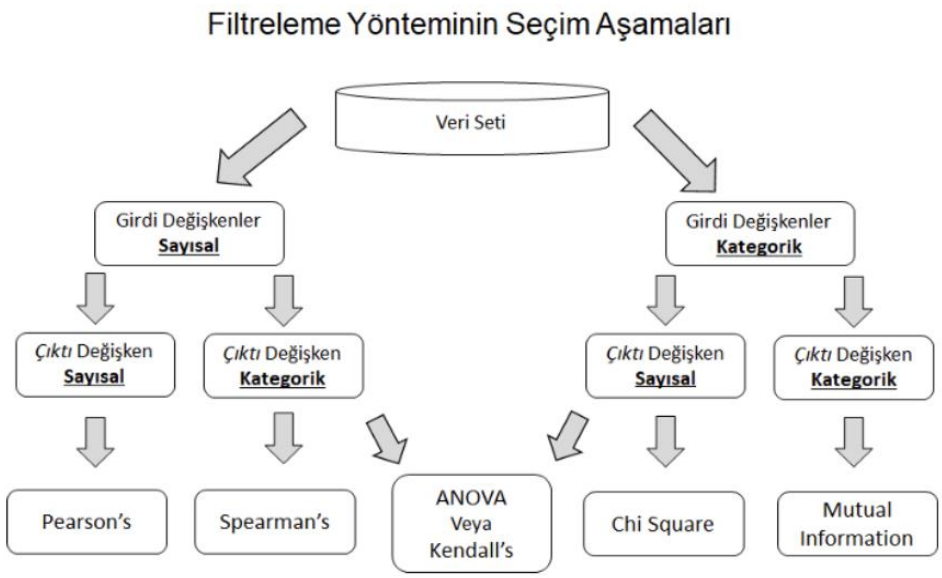
Sarmal (Wrapper) Yöntemleri: Veri setindeki değişkenlerden alt kümeler oluşturarak modele dahil eder ya da hariç bırakır. Deneme yanılma şeklinde ilerler. En iyi performansı veren modelin kullanıldığı değişkenleri seçer.

Gömülü Yöntemler: Bazı algoritmaların yapısı gereği kendisi değişken seçimi yapabilirler. Bunlar Lasso Regresyon ve Karar Ağacı algoritmalarıdır.

Modelimiz daha sonra seçileceği için filtreme yöntemiyle öznitelik seçimi yapılmasına karar verildi.

Filtre Yönteminin seçim aşamaları Şekil 3.5' de görülmektedir.

Şekil 3.5. Filtre yönteminin seçim aşamaları



Veri setimizde 5 adet kategorik girdi değişkeni, 15 adet sayısal girdi değişkeni bulunmaktadır. Çıktı değişkenimiz ise 1 adet kategorik (churn) değişkendir. Öncelikle Şekil 1.22'de görüldüğü gibi 15 adet sayısal girdi değişkenimizle çıktı değişkenimiz arasındaki korelasyonlar Spearman's Korelasyon Yöntemiyle bulundu. Spearman's Korelasyon sonuçları Şekil 3.6'da görülmektedir.

Hedef değişken ile arasında en düşük korelasyon değeri bulunan değişken "total night calls" değişkeni olduğu belirlendi.

Şekil 3.6. Korelasyon sonuçları

```
In [80]: corr_1  # account length ile churn arasındaki
Out[80]: 0.015583005906573652

In [81]: corr_2 # number v mail messages ile churn arasındaki
Out[81]: 0.0953581416079479

In [82]: corr_3 # total day minutes ile churn arasındaki
Out[82]: 0.1706773370176075

In [83]: corr_4 # total day calls ile churn arasındaki
Out[83]: 0.026311093290072682

In [84]: corr_5 # total day charge ile churn arasındaki
Out[84]: 0.1706773370176075

In [85]: corr_6 # total eve minutes ile churn arasındaki
Out[85]: 0.08859150772445837

In [86]: corr_7 #total eve calls ile churn arasındaki
Out[86]: 0.008578283319987489

In [87]: corr_8 #total eve charge ile churn arasındaki
Out[87]: 0.08858045464080103

In [88]: corr_9 #total night minutes ile churn arasındaki
Out[88]: 0.03434258037450492

In [89]: corr_10 #total night calls ile churn arasındaki
Out[89]: 0.004694244302659059

In [90]: corr_11 #total night charge ile churn arasındaki
Out[90]: 0.03435325208460943

In [91]: corr_12 #total intl minutes ile churn arasındaki
Out[91]: 0.060850355510749125

In [92]: corr_13 #total intl call ile churn arasındaki
Out[92]: 0.07475838522854272

In [93]: corr_14 #total intl charge ile churn arasındaki
Out[93]: 0.060850355510749125

In [94]: corr_15 #number customer service calls ile churn arasındaki
Out[94]: 0.13665664980779227
```

Kategorik değişkenlerden “phone number”, “state”, “area code” veri setinden çıkartıldı. Sebepleri ise gözle görülür şekilde hedef değişken ile ilgisi olmadığının görülmesidir. Aynı zamanda bu kategorik verilere Encoding işlemi yapılması gerekirdi. Örneğin; 3333 farklı “phone number” verisi olduğu düşünülürse ve encoding işleminin çalışma mekanizmasında her farklı değer için yeni değişken sütunları oluşturmak da olduğu için 3333 sütun oluşturması gerekirdi. İşlemin verimliliği çok düşük olacağı düşünüldüğünden yapılması uygun görülmemiştir.

Şekil 3.7 ‘da görülen kod ile veri setimizden daha önce seçtiğimiz değişkenler çıkarılmıştır. Güncel veri setinin 3333 satır ile 17 sütundan oluşmaktadır.

Şekil 3.7. Seçtiğimiz değişkenlerin çıkarılmasını sağlayan kod

```
data.drop('phone number', axis = 1, inplace = True)
data.drop('area code', axis = 1, inplace = True)
data.drop('state', axis = 1, inplace = True)
data.drop('total night calls',axis=1 , inplace=True)
print("Dataset preprocessing sonrasi: " + str(data.shape))
```

```
Dataset preprocessing sonrasi: (3333, 17)
```

Aşağıdaki şekil 3.8’de veri setinin son hali görülmektedir.

Şekil 3.8. Veri setinin son hali

	account length	international plan	voice mail plan	number vmail messages	total day minutes	total day calls	total day charge	total eve minutes	total eve calls	total eve charge	total night minutes	total night charge	total intl minutes	total intl calls	total intl charge	number customer service calls	churn
0	128	no	yes	25	265.1	110	45.07	197.4	99	16.78	244.7	11.01	10.0	3	2.70	1	False.
1	107	no	yes	26	161.6	123	27.47	195.5	103	16.62	254.4	11.45	13.7	3	3.70	1	False.
2	137	no	no	0	243.4	114	41.38	121.2	110	10.30	162.6	7.32	12.2	5	3.29	0	False.
3	84	yes	no	0	299.4	71	50.90	61.9	88	5.26	196.9	8.86	6.6	7	1.78	2	False.
4	75	yes	no	0	166.7	113	28.34	148.3	122	12.61	186.9	8.41	10.1	3	2.73	3	False.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3328	192	no	yes	36	156.2	77	26.55	215.5	126	18.32	279.1	12.56	9.9	6	2.67	2	False.
3329	68	no	no	0	231.1	57	39.29	153.4	55	13.04	191.3	8.61	9.6	4	2.59	3	False.
3330	28	no	no	0	180.8	109	30.74	288.8	58	24.55	191.9	8.64	14.1	6	3.81	2	False.
3331	184	yes	no	0	213.8	105	36.35	159.6	84	13.57	139.2	6.26	5.0	10	1.35	2	False.
3332	74	no	yes	25	234.4	113	39.85	265.9	82	22.60	241.4	10.86	13.7	4	3.70	0	False.

3333 rows × 17 columns

3.2.2. Veri Ön İşleme

3.2.2.1. Eksik Verilen Tespit Edilmesi ve Temizlenmesi

Bu aşamada eksik verilerin doldurulma veya çıkarılma işlemleri yapılacaktır. Eksik veriler; müşterinin belirtmek istemediği özelliklerden, bazı verilerin önemsiz görülüp veri girişi yapılmamasından veya güncellenmemiş, geçerliliğini kaybetmiş verilerin silinmesiyle oluşabilir. Algoritmanın yüksek verimlilikte çalışıp, kaliteli veri vermesi için kaliteli veri girdisi olmalıdır. Bunun için eksik verilerin olduğu satırın silinmesi veya eksik yerlerin ilgili sütunun ortalaması ile doldurulması gibi yöntemler kullanılabilir. Şekil 3.9' da görüldüğü gibi hiçbir değişkende eksik veri bulunmamaktadır.

Şekil 3.9. Eksik veri olmadığı göstergesi

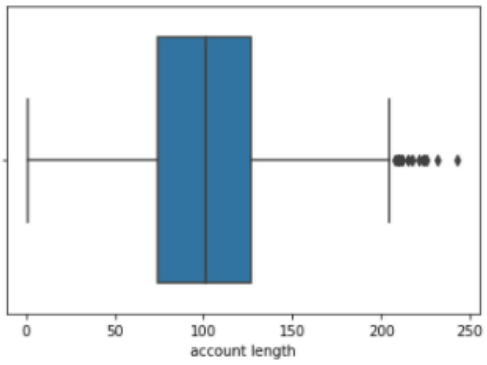
	Count
state	0
account length	0
area code	0
phone number	0
international plan	0
voice mail plan	0
number vmail messages	0
total day minutes	0
total day calls	0
total day charge	0
total eve minutes	0
total eve calls	0
total eve charge	0
total night minutes	0
total night calls	0
total night charge	0
total intl minutes	0
total intl calls	0
total intl charge	0
number customer service calls	0
churn	0

3.2.2.2. Aykırı Değerin Tespiti ve Temizlemesi

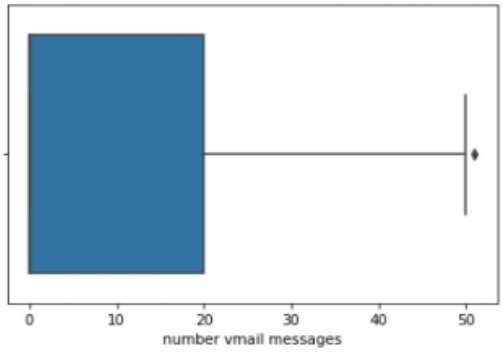
3.2.2.2.1. Box-Plot Yöntemiyle Aykırı Değerlerin Görselleştirilmesi

Aykırı veri; veri setimizdeki gözlemlerin genel eğiliminin dışında, aşırı yüksek veya aşırı düşük değerlere sahip gözlemlere denir. Aykırı değerleri görselleştirip tespit etmek için kutu grafiği (boxplot) kullanıldı. Birçok değişkende şekil 3.10, Şekil 3,11, şekil 3.12' de görüldüğü gibi aykırı değer mevcuttur.

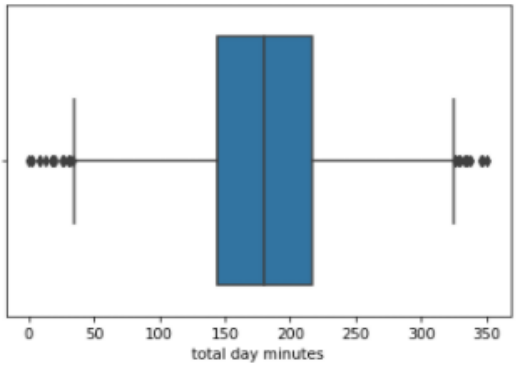
Şekil 3.10. Müşterililik süresi aykırı değer görseli



Şekil 3.11. Sesli mesaj aykırı değer görseli



Şekil 3.12. Toplam gündüz konuşma süresi aykırı değer görseli



3.2.2.2.2. Interquartile Range Yöntemiyle Tek Değişkenli Aykırı Değer Tespiti

Değişkenlerdeki aykırı değerlerin sayısını tespit edebilmek ve daha sonrasında işlem yapabilmek için alt ve üst sınır değerlerini tanımlamamız gerekir. Bunun için Interquartile Range (Çeyrekler Açıklığı) Yöntemi kullanılmıştır. Örnek olarak Şekil 7.7 de verilen kodlar ile "account length" değişkeninin alt sınır, üst sınır ve toplam aykırı gözlem sayısı bulunmuştur.

Şekil 3.13. Account length kodu

```
Q1=data_account_length.quantile(0.25)
Q3=data_account_length.quantile(0.75)
IQR=Q3-Q1
alt_sinir=Q1-1.5*IQR
ust_sinir=Q3+1.5*IQR
alt_aykirilar=(data_account_length < (alt_sinir))
ust_aykirilar=(data_account_length > (ust_sinir))
tum_aykirilar=(data_account_length < (alt_sinir)) | (data_account_length > (ust_sinir))

print("Toplam Gozlem Sayisi : ",data_account_length.shape[0])
print("Alt Sinir Degeri : " + "%.2f" % (alt_sinir))
print("Ust Sinir Degeri : " + "%.2f" % (ust_sinir))
print("Toplam Aykırı Gozlem:",data_account_length[alt_aykirilar].shape[0] + data_account_length[ust_aykirilar].shape[0])
```

```
Toplam Gozlem Sayisi :  3333
Alt Sinir Degeri : -5.50
Ust Sinir Degeri : 206.50
Toplam Aykırı Gozlem: 18
```

3.2.2.2.3. Local Outlier Factor Yöntemiyle Çok Değişkenli Aykırı Değer Tespiti

Çok değişkenli bir veri setinde aykırı değerleri bulmak için her bir değişkeni tek tek incelemek doğru bir yaklaşım değildir. Değişkenlerin; tek başınayken aykırı gözlem olarak algılanamayacak olanlar, çok değişkenli olarak eş zamanlı ele alındığında ortaya aykırı gözlemler çıkabilmektedir. Bu sebeple aykırı gözlemleri çok değişkenli şekilde incelemek gerekir. Bunun için Local Outlier Factor (LOF) yöntemi kullanılmıştır.

Local Outlier Factor, veri setindeki gözlemlerin bulundukları konumu yoğunluk tabanlı skorlayarak aykırı değer olabilecek değerleri tanımlayabilmemizi sağlıyor. Bir noktanın local yoğunluğu komşu noktalar ile karşılaştırılıyor. Eğer bir nokta komşularının yoğunluğundan anlamlı derecede düşük ise bu nokta diğerlerinden daha seyrek bir bölgede bulunduğu yorumu yapılıyor. Bu nokta (değer) aykırı değer olarak belirleniyor. Şekil 3.14'de görüldüğü gibi LOF yönteminin çalıştırılmasında, eşik değerin

girilmesinde ve temizlenmiş yeni veri setinin oluşturulmasında aşağıdaki kod satırları kullanılmıştır.

Şekil 3.14. LOF yönteminin çalıştırılmasında, eşik değerin girilmesinde ve temizlenmiş yeni veri setinin oluşturulması

```
import numpy as np
from sklearn.neighbors import LocalOutlierFactor
```

```
clf=LocalOutlierFactor(n_neighbors=20,contamination=0.1)
```

```
clf.fit_predict(df)
```

```
array([1, 1, 1, ..., 1, 1, 1])
```

```
df_scores=clf.negative_outlier_factor_
```

```
np.sort(df_scores)[0:50]
```

```
array([-2.08849881, -1.8776325 , -1.84833744, -1.73611832, -1.73084547,
       -1.67019112, -1.66795007, -1.605949  , -1.60160226, -1.5718816 ,
       -1.56663384, -1.55722365, -1.53828611, -1.52430958, -1.52384778,
       -1.51977413, -1.51006508, -1.50380212, -1.50001968, -1.49457878,
       -1.49455952, -1.4716013 , -1.46374379, -1.46288069, -1.45669655,
       -1.44935813, -1.44662757, -1.44126082, -1.43817628, -1.43475276,
       -1.43467896, -1.43247424, -1.42919394, -1.42671954, -1.41966077,
       -1.41834512, -1.41621184, -1.41472676, -1.40906387, -1.4090577 ,
       -1.40568672, -1.4049477 , -1.40305575, -1.40201579, -1.39866973,
       -1.39741425, -1.39488425, -1.39311729, -1.39157234, -1.39057257])
```

```
esik_deger=np.sort(df_scores)[12]
```

```
aykiri_tf=df_scores>esik_deger
```

```
aykiri_tf
```

```
array([ True,  True,  True, ...,  True,  True,  True])
```

```
yeni_df=df[df_scores>esik_deger]
```

Şekil 3.15’ de 12 adet aykırı değer içeren gözlem listelendi.

Şekil 3.15. Aykırı değer içeren gözlem listesi

	account length	number vmail messages	total day minutes	total day calls	total day charge	total eve minutes	total eve calls	total eve charge	total night minutes	total night charge	total intl minutes	total intl calls	total intl charge	number customer service calls
15	161	0	332.9	67	56.59	317.8	97	27.01	160.6	7.23	5.4	9	1.46	4
32	172	0	212.0	121	36.04	31.2	115	2.65	293.3	13.20	12.6	10	3.40	3
315	39	0	60.4	158	10.27	306.2	120	26.03	123.9	5.58	12.4	3	3.35	1
850	166	0	274.3	110	46.63	52.9	109	4.50	246.1	11.07	10.9	5	2.94	0
960	5	0	199.2	106	33.86	187.3	12	15.92	214.0	9.63	13.3	3	3.59	3
1345	98	0	0.0	0	0.00	159.6	130	13.57	167.1	7.52	6.8	1	1.84	4
1397	101	0	0.0	0	0.00	192.1	119	16.33	168.8	7.60	7.2	4	1.94	1
2663	172	0	169.8	123	28.87	183.1	94	15.56	395.0	17.77	12.7	7	3.43	2
2932	97	0	209.2	134	35.56	0.0	0	0.00	175.4	7.89	11.8	6	3.19	1
3075	181	40	105.2	61	17.88	341.3	79	29.01	165.7	7.46	6.3	3	1.70	2
3174	36	43	29.9	123	5.08	129.1	117	10.97	325.9	14.67	8.6	6	2.32	2
3219	150	35	139.6	72	23.73	332.8	170	28.29	213.8	9.62	8.8	2	2.38	2

Tek değişkendeki aykırı değerleri incelerken kullandığımız Box Plot ve Interquartile Range yöntemlerinde daha fazla sayıda aykırı değer gözlemlemiştik. Çok değişkenli incelendiğinde ise bu aykırı değer sayısında düşüş olduğu gözlemlenmiştir.

3.2.3. Veri Dönüştürme

Bu bölümde kategorik değişkenlerin verilerinin sayısal değerlere dönüştürme işlemi ve normalleştirme işlemleri yapıldı.

3.2.3.1. Kategorik Verileri Dönüştürme

Kategorik değişkenlerden; "international plan" , "voice mail plan" , "churn" string değerlere sahip olan (True/False ve Yes/No) 3 özniteliğin değerleri "0" ve "1 " ile değiştirildi. Şekil 3.16'da veri setinin önceki hali Şekil 3.17 'de son hali gösterilmektedir.

Şekil 3.16. Kategorik verinin önceki hali

	account length	number vmail messages	total day minutes	total day calls	total day charge	total eve minutes	total eve calls	total eve charge	total night minutes	total night charge	total intl minutes	total intl calls	total intl charge	number customer service calls
15	161	0	332.9	67	56.59	317.8	97	27.01	160.6	7.23	5.4	9	1.46	4
32	172	0	212.0	121	36.04	31.2	115	2.65	293.3	13.20	12.6	10	3.40	3
315	39	0	60.4	158	10.27	306.2	120	26.03	123.9	5.58	12.4	3	3.35	1
850	166	0	274.3	110	46.63	52.9	109	4.50	246.1	11.07	10.9	5	2.94	0
960	5	0	199.2	106	33.86	187.3	12	15.92	214.0	9.63	13.3	3	3.59	3
1345	98	0	0.0	0	0.00	159.6	130	13.57	167.1	7.52	6.8	1	1.84	4
1397	101	0	0.0	0	0.00	192.1	119	16.33	168.8	7.60	7.2	4	1.94	1
2663	172	0	169.8	123	28.87	183.1	94	15.56	395.0	17.77	12.7	7	3.43	2
2932	97	0	209.2	134	35.56	0.0	0	0.00	175.4	7.89	11.8	6	3.19	1
3075	181	40	105.2	61	17.88	341.3	79	29.01	165.7	7.46	6.3	3	1.70	2
3174	36	43	29.9	123	5.08	129.1	117	10.97	325.9	14.67	8.6	6	2.32	2
3219	150	35	139.6	72	23.73	332.8	170	28.29	213.8	9.62	8.8	2	2.38	2

Şekil 3.17. Kategorik verinin dönüşmüş hali

	account length	international plan	voice mail plan	number vmail messages	total day minutes	total day calls	total day charge	total eve minutes	total eve calls	total eve charge	total night minutes	total night charge	total intl minutes	total intl calls	total intl charge	number customer service calls	churn
0	128	0	1	25	265.1	110	45.07	197.4	99	16.78	244.7	11.01	10.0	3	2.70	1	0
1	107	0	1	26	161.6	123	27.47	195.5	103	16.62	254.4	11.45	13.7	3	3.70	1	0
2	137	0	0	0	243.4	114	41.38	121.2	110	10.30	162.6	7.32	12.2	5	3.29	0	0
3	84	1	0	0	299.4	71	50.90	61.9	88	5.26	196.9	8.86	6.6	7	1.78	2	0
4	75	1	0	0	166.7	113	28.34	148.3	122	12.61	186.9	8.41	10.1	3	2.73	3	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3328	192	0	1	36	156.2	77	26.55	215.5	126	18.32	279.1	12.56	9.9	6	2.67	2	0
3329	68	0	0	0	231.1	57	39.29	153.4	55	13.04	191.3	8.61	9.6	4	2.59	3	0
3330	28	0	0	0	180.8	109	30.74	288.8	58	24.55	191.9	8.64	14.1	6	3.81	2	0
3331	184	1	0	0	213.8	105	36.35	159.6	84	13.57	139.2	6.26	5.0	10	1.35	2	0
3332	74	0	1	25	234.4	113	39.85	265.9	82	22.60	241.4	10.86	13.7	4	3.70	0	0

3320 rows × 17 columns

3.2.3.2. Class Imbalance (Sınıf Dengesizliği) Çözümü ve Normalleştirme

3.2.3.2.1. Class Imbalance Çözümü

İlk olarak hedef değişkenimizin “churn olan “ve “ churn olmayan” örnek veri sayısı belirlendi. Şekil 3.18’deki kod ile veri seti ikiye ayrıldı. Churn olan veriler ve churn olmayan verilerin sayısı gösterildi. Churn olmayan 481 tane örnek veri ile churn olan 481 örnek veri birleştirildi (Şekil 3.19). Son veri setinde toplam 962 tane örnek veri bulunmaktadır. Burada amaç sınıf dengesizliğini önleyip modelin tahmin performansını düşürmemektir.

Şekil 3.18. Veri setini ikiye ayıran kod

```
data10 = data7[data7['churn']==1]
print("Churn olanlar-data10:"+ str(data10.shape))
data11 = data7[data7['churn']==0]
print("Churn olmayanlar-data11:"+ str(data11.shape))
```

```
Churn olanlar-data10:(481, 17)
Churn olmayanlar-data11:(2839, 17)
```

Şekil 3.19. Veri seti birleştirme kodu

```
data7 = data10.append(data11[:481])
print("Son veriseti :"+ str(data7.shape))
```

```
Son veriseti :(962, 17)
```

3.2.3.2.2. Normalleştirme

Modelin oluşturulması ve test aşamasına geçmeden önce elimizde veri setinin son halini "eğitim" ve "test" veri setleri olarak parçalandı (Şekil 3.20). Daha sonra veri seti -1 ile 1 arasında ölçeklendirilip normalleştirme yapıldı (Şekil 3.20).

Şekil 3.20. Verinin Normalleştirmesi

```
#Egitim  ve test verisini parcaliyoruz --> 80% / 20%
X = data5.loc[:, data5.columns != 'churn']
Y = data5['churn']
X_train, X_test, Y_train, Y_test =model_selection.train_test_split(X, Y, test_size=0.2, random_state=0)
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
#ölçeklendirme
scaler = preprocessing.MinMaxScaler((-1,1))
scaler.fit(X)
XX_train = scaler.transform(X_train.values)
XX_test  = scaler.transform(X_test.values)
YY_train = Y_train.values
YY_test  = Y_test.values
```

3.3. Modelleme

3.3.1. Makine Öğrenmesi Modellerinin Eğitilmesi ve Değerlendirilmesi

Bu aşamada eğitim veri seti olarak ayırdığımız verilerin sınıflandırmada kullanılan modeller üzerinde eğitilmesi ve test veri seti olarak ayırılan verilerin testi gerçekleştirildi. Model sonuçlarının değerlendirilmesi yapıldı. Kullanılan modeller aşağıda belirtilmiştir. Bunlar;

Temel Sınıflandırma Modelleri: Naive Bayes, Decision Tree (CART),K-NN, SVM, LDA, Logistic Regression

Kolektif Sınıflandırma Modelleri: BaggingClassifier, AdaBoostClassifier, RandomForestClassifier

Modellere ait scikit-learn kütüphanesinin paketleri import edildi. Daha sonra for döngüsüyle liste içindeki fonksiyonları yürütme amacıyla models adı altında liste oluşturuldu. Her bir model ismi ve metodun çağrılma ismiyle beraber listeye eklendi. Şekil 3.21' de bu kod gösterilmiştir.

Şekil 3.21. Metodu çağırma kodu

```
models = []
models.append(('Logistic Regression', LogisticRegression()))
models.append(('Naive Bayes', GaussianNB()))
models.append(('Decision Tree (CART)',DecisionTreeClassifier()))
models.append(('K-NN', KNeighborsClassifier()))
models.append(('SVM', SVC()))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('AdaBoostClassifier', AdaBoostClassifier()))
models.append(('BaggingClassifier', BaggingClassifier()))
models.append(('RandomForestClassifier', RandomForestClassifier()))
```

For döngüsü ile listedeki her model daha önce oluşturduğumuz eğitim ve test veri setleri üzerinden eğitildi ve her modelin ACC "Accuracy /Doğruluk " değeri şekil 3.22'deki kod ile yazdırıldı.

Şekil 3.22. Accuracy /Doğruluk kodu

```
for name, model in models:
    model = model.fit(X_train, Y_train)
    Y_pred = model.predict(X_test)
    from sklearn import metrics
    print("Model -> %s -> ACC: %%%.2f" % (name,metrics.accuracy_score(Y_test, Y_pred)*100))
```

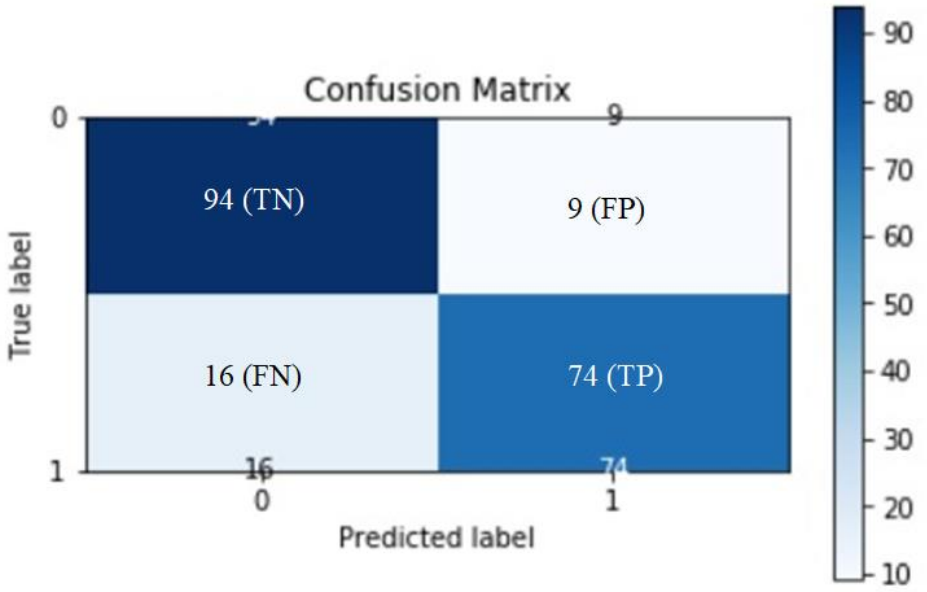
Çıkan ACC değerlerine göre, en yüksek yüzdeli model "en başarılı model"; en düşük yüzdeli modele "en başarısız model" şeklinde yorum yapabiliriz. Random Forest Classifier modeli %87.05 ACC oranı ile gerçeğe en yakın sonuçları veren model olmuştur. Oranları gösteren şekil 3.23'de verilmiştir.

Şekil 3.23. Sınıflandırma algoritmalarının doğruluk oranları

```
Model -> Logistic Regression -> ACC: %68.91
Model -> Naive Bayes -> ACC: %74.61
Model -> Decision Tree (CART) -> ACC: %81.87
Model -> K-NN -> ACC: %69.43
Model -> SVM -> ACC: %46.63
Model -> LDA -> ACC: %66.84
Model -> AdaBoostClassifier -> ACC: %77.72
Model -> BaggingClassifier -> ACC: %84.97
Model -> RandomForestClassifier -> ACC: %87.05
```

3.3.2. Model Çıktılarını Değerlendiren Metrikler (Confusions matrix, Accuracy)

Şekil 3.24. Confusion Matrix



Yukarıdaki şekil'3.24 aşağıda açıklanmıştır:

TN (True-negative): Gerçek verideki değerin negatif ve tahmin ettiğimiz değerinde negatif olduğunu gösterir. Kısaca yanlışa yanlış demektir. Çalışmamızdaki gerçek veride müşterinin ayrılmadığını ve test verisindeki tahminimizde de müşterinin ayrılmadığını gösterir.

TP (True-positive): Gerçek verideki değerin pozitif ve tahmin ettiğimiz değerinde pozitif olduğunu gösterir. Kısaca doğruya doğru demektir. Çalışmamızdaki gerçek veride müşterinin ayrıldığını ve test verisindeki tahminimizde de müşterinin ayrıldığını gösterir.

FP (False-positive): Gerçek verideki değerin negatif ve tahmin ettiğimiz değerinde pozitif olduğunu gösterir. Kısaca yanlışa doğru demektir. Çalışmamızdaki gerçek veride müşterinin ayrılmadığını ancak test verisindeki tahminimizde müşterinin ayrıldığını gösterir.

FN (False-negative): Gerçek verideki değerin pozitif ve tahmin ettiğimiz değerinde negatif olduğunu gösterir. Kısaca doğruya yanlış demektir. Çalışmamızdaki gerçek veride müşterinin ayrıldığını ancak test verisindeki tahminimizde müşterinin ayrılmadığını gösterir.

Şekil 3.24'e göre sınıflandırma algoritmaları ile elde edilen modelin doğruluk, hata oranı, hassasiyet, geri çağırma, f-skoru denklemleri aşağıda gösterilmiştir ve hesaplanmıştır:

Accuracy (Doğruluk oranı): Makine öğrenmesi sınıflandırma algoritmalarının testlerinde sıklıkla kullanılan accuracy, anlaşılması ve yorumlanması en basit ölçütlerden birisidir. 0-1 arasında değer alıp 1'e yaklaşan değerlerde model daha başarılı kabul edilir. Çalışmamızda accuracy değer 0,87 çıkmaktadır.

Accuracy= (TP+TN) / (TP+FP+TN+FN)

=(94+74) / (94+74+16+9)

=0,87

Hata oranı = 1- 0,87 = 0,13

Precision (Hassasiyet): Doğru pozitif edilenlerin, toplam pozitif tahminlere oranıdır. 0-1 arasında değer alır ve ne kadar 1'e yakınsak o kadar başarılı sayılır. Çalışmamızda precision değeri 0,89 çıkmaktadır.

Precision= (TP) / (FP+TP)

=(74) / (74+9)

=0,89

Recall (Geri çağırma): Pozitif durumların ne kadar başarılı tahmin edildiğini gösterir. 0-1 arasında değer alır. En iyi değer 1, en kötü değer 0'dır. Çalışmamızda recall değeri 0,82 çıkmaktadır.

Recall= (TP) / (TP+FN)

=(74) / (74+16)

=0,82

F-score: Test edilen verilerin doğruluğunun ölçümüdür. Hassasiyet ve geri çağırma metriklerinin harmonik ortalamasıdır. Çalışmamızda F-score değeri 0,86 çıkmaktadır.

F-score = (2*Pericison*Recall) / (Pericision + Recall)

=0,86

Aşağıdaki şekil 3.25'de görüldüğü üzere [classification_report] adlı fonksiyon sayesinde yukarıda hesaplanan değerleri bir arada görmekteyiz.

Şekil 3.25. Classification_report fonksiyon ile metrikler

```
from sklearn.metrics import classification_report
report = (classification_report(YY_test, Y_pred))
print(report)
```

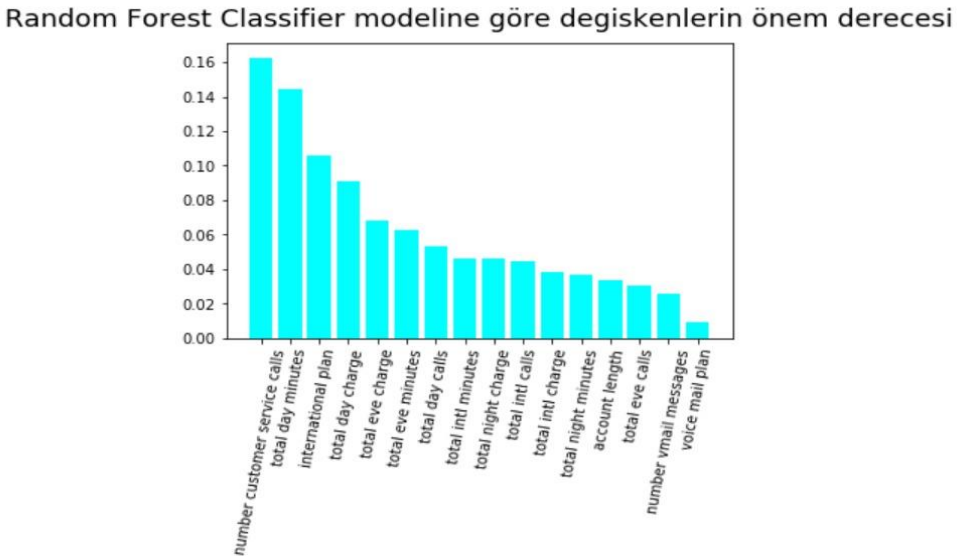
```
              precision    recall  f1-score   support

           0       0.85      0.91      0.88       103
           1       0.89      0.82      0.86        90

    accuracy                           0.87       193
   macro avg       0.87      0.87      0.87       193
weighted avg       0.87      0.87      0.87       193
```

3.3.3. Rondom Forest Classifier Modeline Göre Değişkenlerin Önem Derecesi

Şekil 3.26. Değişkenlerin önem derecesi



Şekil 3.26'da görüldüğü gibi müşteri kaybına en fazla etki eden değişkenler 0,16 (%16) müşteri hizmetlerini arama sayısıdır. Bu değişkeni toplam gündüz konuşma süresi, yurt dışını aramalarla ilgili paketler, toplam gündüz ücretlendirme ve toplam akşam ücretlendirme takip etmektedir. Müşteri kaybını en az etkileyen değişken ise sesli aramalarla ilgili paketlerdir. Burada belirtilen müşteri kayıplarına müşterilerin ekonomik durumu, telekomünikasyon şirketinde çalışanları müşteriye gerekli değeri vermemesi, paketlerin müşterileri tatmin edemeyecek seviyede olması etki etmiş olabilir. Sonuç beklediğimiz yakın çıkmıştır. Çünkü konuşma dakikaları, paketler ve müşteri hizmetleri gibi ana sebepler müşterileri kaybetmeye sebep olabilmektedir.

BÖLÜM DÖRT

SONUÇ VE DEĞERLENDİRME

Telekomünikasyon sektöründe büyük çalkalanmalar olabilmekte ve büyük müşteri kayıpları verilebilmektedir. Verilen bu kayıplar sonucu yeni müşteriler kazanmanın maliyeti eldeki müşteriyi kaybetmemenin maliyetinden çok daha fazladır. Bu nedenle müşteri kaybını tam anlamıyla bitirmek imkansız olsa bile müşteri kaybı kabul edilebilir bir seviyeye çekilebilir ve orada tutulabilir. Bu çalışmada makine öğrenmesi yöntemleri kullanılarak, telekomünikasyon firmasından ayrılıp başka firmaları tercih eden müşterileri ortaya çıkaran modeller geliştirilmiştir. Bu modeller ortaya konulurken sınıflandırma algoritmalarından yararlanılmıştır. Python uygulaması ile veri temizleme yapılmış daha sonra aynı uygulamada sınıflandırma algoritmalarının(Naive Bayes, Karar ağacı, lojistik regresyon, random forest vb.) doğruluk (accuracy) değerleri ortaya çıkarılmıştır. Sınıflandırma algoritmalarının doğruluk değerleri karşılaştırılarak en iyi algoritmanın Random Forest (rassal orman) olduğu belirlenmiştir. Random forest algoritmasının doğruluk oranının %87 olduğu belirlenmiştir.

Müşteri kaybına etki eden değişkenler belirlendi. En fazla etki eden değişkenler üzerinden müşteri kaybını azaltacak bazı iyileştirilmeler hayata geçirilebilir. Bunlar şu şekilde sıralanabilir:

Müşteri hizmetlerini aranma sayısı, müşteri kaybına en fazla etki eden değişkendir. Müşteri hizmetlerini arayan müşterilerin sorunları daha verimli bir şekilde giderilirse ayrılan müşteri oranı düşer. Telekomünikasyon firması müşteri hizmetleri üzerinde bir iyileştirme yapması gerekmektedir.

Müşteri kaybına etki eden diğer önemli değişken toplam gündüz konuşma süresidir. Toplam gündüz konuşma süresi üzerinde yapılacak kampanyalar ve artırılacak konuşma süresi, gündüz konuşma süresinden dolayı oluşan müşteri kaybını azaltabilir.

Yurtdışı aramalarla ilgili paketten dolayı ayrılan müşterileri kaybetmemek için bu müşterilere özel olarak daha uygun paketler müşteriye sunulabilir.

Toplam gündüz ücretlendirmelerinden dolayı ayrılan müşterileri elde tutabilmek için bu müşterilerin paketlerine ek olarak gündüz konuşma süresi eklenebilir veya paket ücreti makul bir seviyeye çekilebilir.

Toplam akşam ücretlendirmelerini fazla bulan ve bu durumdan dolayı ayrılabilecek müşterileri elde tutabilmek için çok sayıda alternatif paket sunulabilir.

Müşteri kaybına etki eden birçok faktör bulunmaktadır. Kısıtlı kaynaklardan dolayı etkisi en fazla olan değişkenlere yönelerek onlar üzerinde iyileştirmeler yapılmalıdır. İmkan varsa diğer değişkenler üzerinde de iyileştirmeler yapılabilir.

Belirlenen sonuçlarca telekomünikasyon şirketinden ayrılma eğilimine sahip müşteriler, çeşitli müşteri ilişkileri yöntemleri çerçevesinde ele alınıp ve de şirkete daha bağlı kalabilmeleri için çeşitli programlar, kampanyalar, reklamlar vs. uygulanması öneride bulunuldu. Böylece telekomünikasyon şirketi müşterileri kaybetmeyecek ve yeni müşteriler ile daha da büyüme gösterebilecektir. Telekomünikasyon şirketi bu sayede aidiyet duygusu yüksek müşteriler elde etmiş olacak ve süreklilik sağlayacaktır.

KAYNAKÇA

Atilla Özgür, Hamit Erdem, (2012). Saldırı Tespit Sistemlerinde Kullanılan Kolay Erişilen Makine Öğrenme Algoritmalarının Karşılaştırılması, Bilişim Teknolojileri Dergisi, (2), 41-48.

Başarslan, M. S. 2017. Telekomünikasyon Sektöründe Müşteri Kaybı Analizi, Yüksek Lisans Tezi, Düzce Üniversitesi, Fen Bilimleri Enstitüsü, Düzce

Başkal, R. 2019. Telekomünikasyon Sektöründe Müşteri Segmentasyonu ve Müşteri Kaybı Analizi, Yüksek Lisans Tezi, Haliç Üniversitesi, Lisansüstü Eğitim Enstitüsü, İstanbul.

Burez, J., Van den Poel, D.,2009. Handling class imbalance in customer churn prediction, Elsevier, (36), 4626-4636

Dolgun, M., Özdemir, T., Oğuz, D., (2009). Veri madenciliğinde Yapısal Olmayan Verinin Analizi: Metin ve Web Madenciliği. İstatikçiler Dergisi, (2), 48-58.

Feng Guo, Hui-Lin Qin,(2015). The Analaysis of Customers Churns in e-Commerce Based on Decision Tree, 2015 International Conference on Computer Science and Applications (CSA)

Gülsoy, N. 2019. Kredi Skorlama Süreçlerinde Veri Madenciliği ve Bankacılık Sektöründe Bir Uygulama, Doktora Tezi, Kayseri Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı, Kayseri.

Heng-liang Wu;Wei-wei Zhang;Yuan-yuan Zhang,(2010). An Empirical Study of Customer Churn in Ecommerce Based on Data Mining, 2010 International Conference on Management and Service Science

Hsiu-Yu Liao, Ōuan-Yu Chen, Duen-Ren Liu, Yi-Ling Chiu,(2015). Customer Churn Prediction in Virtual Worlds, 2015 IIAI 4th International Congress on Advanced Applied Informatics

Kıyak, E. 2006. CRISP-DM Yöntembilimini Kullanarak Deniz kuvvetleri Verisi Üzerinde Veri Madenciliği Sınıflandırma Tekniklerinin Karşılaştırılması, Yüksek Lisans Tezi, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli.

Kişioğlu, P. 2009. Telekomünikasyon Sektöründe İptal Analizi, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

Musa Peker, Osman Özkaraca, Betül Kesimal, (2017). Enerji Tasarruflu Bina Tasarımı için Isıtma ve Soğutma Yüklerini Regresyon Tabanlı Makine Öğrenmesi Algoritmaları ile Modelleme, Bilişim Teknolojileri Dergisi, (4), 443-449.

Kunt, M. S. 2019. Telekomünikasyon Sektöründe Müşteri Kaybı Analizi, Yüksek Lisans Tezi, Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.

Taşkın, E. 2020. Nicel Birikimin Nitel Değişime Etkisi: Kütüphane Yönetiminde Veri Madenciliği Uygulaması, Yüksek Lisans Tezi, Trakya Üniversitesi, Sosyal Bilimler Enstitüsü, İşletme Anabilim Dalı, Edirne.

Serap Kazan, Hakan Karakoca, (2019). Makine Öğrenmesi İle Ürün Kategorisi Sınıflandırma, Sakarya Üniversitesi Bilgisayar ve Bilişim Bilimler Dergisi, (1), 19-27.

Tosun, T. 2006. Veri Madenciliği Teknikleriyle Kredi Kartlarında Müşteri Kaybetme Analizi, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

Emrah YÜRÜKLÜ, Osman H. KOÇAL, 2012. Uludağ Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, Cilt 17, sayı 1, 1-16.

Mustafa TAKAOĞLU, Faruk TAKAOĞLU. K-means Ve Hiyerarşik Kümeleme Algoritmanın Weka Ve Matlab Platformlarında Karşılaştırılması, İstanbul Aydın Üniversitesi.

ŞAHAN, A. 2020. Stratejik Yönetim Perspektifinden Sigortacılık Sektöründe Makine Öğrenmesi Algoritmaları İle Anomali Tespiti, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İşletme Mühendisliği Anabilim Dalı, İstanbul.

Maria Spiteri, George Azzopardi,(2018). Customer Churn Prediction for a Motor Insurance Company, 2018 Thirteenth International Conference on Digital Information Management (ICDIM)

Melike GÜNAY, Tolga ENSARİ,(2018). Makine Öğrenmesi Yöntemleri ile Kayıp Müşteri Analizi, 2018 26. Sinyal İşleme ve İletişim Uygulamaları Konferansı

Paweena Wanchai,(2017). Customer Churn Analysis: A Case Study on the Telecommunication Industry of Thailand, 2017 12th International Conference for Internet Technology and Secured Transactions.

URL-1 Telekomünikasyon https://tr.wikipedia.org/wiki/Telekom%C3%BCnikasyon 20 KASIM 2020

Aksoy G. Destex Digital Blog https://www.destexdigital.com/blog/turkiyenin-2020-dijital-istatistikleri/ 20 KASIM 2020

URL-2 Medya Akademi https://medyaakademi.com.tr/2020/02/03/2020-sosyal-medya-kullanici-sayilari/ 21 KASIM 2020

URL-3 Branding Türkiye https://www.brandingturkiye.com/instagram-istatistikleri-guncel/ 22 KASIM 2020

URL-4 KPMG Sektörel Bakış 2019 – Telekomünikasyon https://home.kpmg/tr/tr/home/gorusler/2019/04/sektorel-bakis-2019-telekomunikasyon.html/ 22 KASIM 2020

URL-5 ML | Types of Learning – Supervised Learning https://www.geeksforgeeks.org/ml-types-learning-supervised-learning/?ref=lbp 20 KASIM 2020

URL-6 Makine Öğrenmesi Türkçe Kaynak https://github.com/SerayBeser/makine-ogrenmesi/ 20 KASIM 2020

ŞAHAK, İ. 2018. https://medium.com/@hibrahimsafak

KILINÇ, D. 2018. https://medium.com/deep-learning-turkiye/makine-öğrenimi-ve-derin-öğrenme-ile-müşteri-kayıp-churn-analizi-1-63a4513b8a6f

ŞİMŞEK, H. 2018. https://medium.com/data-science-tr/makine-öğrenmesi-dersleri-5-bagging-ve-random-forest-2f803cf21e07

ŞENER, Y. 2020. https://yigitsener.medium.com/makine-öğrenmesi-ile-müşteri-kaybı-churn-olasılık-tahminlemesi-bankacılık-sektöründen-örnek-f0f6cd37c00e

ŞENER, Y. 2020. https://yigitsener.medium.com/veri-bilimi-sınıflandırma-model-çıktılarını-değerlendiren-metrikler-confusion-matrix-accuracy-437f5633c82b

ŞENER, Y. 2020. https://yigitsener.medium.com/makine-öğrenmesinde-değişken-seçimi-feature-selection-yazı-serisi-genel-bakış-6ac5013d1ee

ŞENER, Y. 2020. https://yigitsener.medium.com/makine-öğrenmesinde-değişken-seçimi-feature-selection-yazı-serisi-filtreleme-yöntemleri-ve-415a894d5b93.

ÖĞÜNDÜR, G. 2019. https://medium.com/@gulcanogundur/öznitelik-seçimi-feature-selection-teknikleri-5cd8cbab7706

https://gelecegiyazanlar.turkcell.com.tr/