T.C.
İSTANBUL NİŞANTAŞI ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ İLE VERİ MADENCİLİĞİ TEKNİKLERİ KULLANILARAK ANAHTAR KELİME TAHMİNİ VE TEZLERDEKİ ANAHTAR KELİMELERİN DOĞRULUK ORANI TESPİTİ

YÜKSEK LİSANS TEZİ

Aynur GÜNAY

Enstitü Anabilim Dalı : Bilgisayar Mühendisliği
Enstitü Bilim Dalı : Bilgi Teknolojileri

Tez Danışmanı: Dr. Öğr. Üyesi Fatih ŞAHİN

MART 2023

T.C.
İSTANBUL NİŞANTAŞI ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ İLE VERİ MADENCİLİĞİ TEKNİKLERİ KULLANILARAK ANAHTAR KELİME TAHMİNİ VE TEZLERDEKİ ANAHTAR KELİMELERİN DOĞRULUK ORANI TESPİTİ

YÜKSEK LİSANS TEZİ

ORCID ID: 0000-0003-1550-8459

Aynur GÜNAY

Enstitü Anabilim Dalı : Bilgisayar Mühendisliği
Enstitü Bilim Dalı : Bilgi Teknolojileri

"Bu tez 08/03/2023 tarihinde aşağıdaki jüri tarafından Oybirliği / ~~Oyçokluğu~~ ile kabul edilmiştir."

JÜRİ ÜYESİ	KANAATİ	İMZA
Dr. Öğr. Üyesi Negar Sadat SOLEIMANI ZAKERI	BAŞARILI	
Dr. Öğr. Üyesi Mahsa TORKAMANIAN AFSHAR	BAŞARILI	
Dr. Öğr. Üyesi Fatih ŞAHİN	BAŞARILI	

Enstitü Müdürü
Onay

BEYAN

Bu tezin yazılmasında bilimsel ahlak kurallarına uyulduğunu, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, tezin herhangi bir kısmının bu üniversite veya başka bir üniversitedeki başka bir tez çalışması olarak sunulmadığını beyan ederim.

Aynur GÜNAY
08/03/2023

ÖNSÖZ

Tez sürecinde çalışmamı özenle takip eden değerli danışmanım Dr. Öğr. Üyesi Fatih ŞAHİN'e, görüş ve önerileriyle yardımlarını esirgemeyen değerli hocam Dr. Öğr. Üyesi Sajjad NEMATZADEH'e çok değerli bilgi ve tecrübelerini benimle paylaştıkları için teşekkürlerimi ve saygılarımı sunarım.

Yükseklisans eğitimim süresince büyük katkısı olan değerli hocam Doç. Dr. Mehmet ALTUNTAŞ'a yardım ve emekleri için teşekkürlerimi ve saygılarımı sunarım.

Tez çalışmamda bana destek veren başta Can Berk ÇELİK olmak üzere tüm arkadaşlarıma çok teşekkür ederim.

Son olarak bu günlere ulaşmamda emeklerini hiçbir zaman ödeyemeyeceğim anneme ve babama, herşeyde olduğu gibi tez çalışması boyunca da desteğini ve anlayışını esirgemeyen sevgili eşim Ferit'e, çocuklarım Rahmi Efe ve Meyra'ya sonsuz teşekkürlerimi sunarım.

Aynur GÜNAY
08/03/2023

İÇİNDEKİLER

KISALTMALAR

LSTM	:	Uzun Kısa Süreli Hafıza
HDFS	:	Hadoop Dağıtılmış Dosya Sistemi
AWS	:	Amazon Web Service
VM	:	Sanal Makine
NLP	:	Doğal Dil İşleme
POS	:	Metin Parçası
NER	:	İsimlendirilmiş Varlık Tanıma
İTÜ	:	İstanbul Teknik Üniversitesi
API	:	Uygulama Programlama Arabirimi
MAE	:	Ortalama Multak Hata
MSE	:	Ortalama Kare Hatası
KA	:	Karar Ağaçları
DVM	:	Destek Vektör Makinesi
KNN	:	K En Yakın Komşu
YSA	:	Yapay Sinir Ağları
CNN	:	Evrişimsel Sinir Ağları
RNN	:	Tekrarlayan Sinir Ağı
IDE	:	Entegre Geliştirme Ortamı
GPU	:	Grafik İşlemcisi

TABLO LİSTESİ

ŞEKİL LİSTESİ

İstanbul Nişantaşı Üniversitesi, Lisansüstü Eğitim Enstitüsü Yüksek Lisans Tez Özeti

Tezin Başlığı: Makine Öğrenmesi İle Veri Madenciliği Teknikleri Kullanılarak Anahtar Kelime Tahmini Ve Tezlerdeki Anahtar Kelimelerin Doğruluk Oranı Tespiti
Tezin Yazarı: Aynur GÜNAY **Danışman:** Dr. Öğr. Üyesi Fatih ŞAHİN
Kabul Tarihi: 08.03.2023 **Sayfa Sayısı:** vi (önkısım)+90(tez)
Anabilim Dalı: Bilgisayar Mühendisliği **Bilim Dalı:** Bilgi Teknolojileri
Günümüzde bilgisayar çağında istenilen veriye ulaşım kolaylaşmakla birlikte kirli verilerden dolayı doğru ve güvenilir veriye ulaşmak zorlaşmıştır. Anahtar kelimeler metnin konusunu anlamamızı kolaylaştırdığı için, metinlerde tanımı büyük önem arz etmektedir. Uzun metinleri ya da dokümanların tamamını okumaya gerek kalmadan metin ya da doküman hakkında genel bilgi sahibi olabilir, çıkarımlar yapılabilir. Arama işlemlerinde doğru anahtar kelimelerin seçimi ve veri ambarlarındaki verilerde belirtilen anahtar kelimelerin doğru bir şekilde tanımlanmış olması bizi doğru sonuçlara götürür. Böylelikle anahtar kelimelerle bir metnin ne ile alakalı olduğu konusunda hızlı bir şekilde tahminde bulunulabilir ve amacımızla alakalı olmayan metinleri okumakla zaman kaybedilmemiş olunur. Bu çalışmada metin madenciliğinde doğal dil işleme tekniği olan metin işleme ve analizi konusunda çalışma yapılmıştır. Tez dosyalarından elde edilen metin üzerinden, en çok kullanılan anahtar kelimeler bulunarak kullanıcılara öneri olarak sunulması amaçlanmıştır. Anahtar kelime çıkarımında, tez dosyaları üzerinden toplanan veri seti üzerinden hareketle çalışılmıştır. Bu işlem için yazım dili Türkçe olan 900 tez incelenmiştir. Öncelikle her bir tez dosyası için normalizasyon işlemleri yapılmış, tez özetinde belirtilen orjinal anahtar kelimeler de metinden alınarak txt dosyalarına ayrı ayrı kaydedilmiştir. Normalizasyon işleminden geçirilen tez içerikleri ve anahtar kelimeler tek bir txt dosyasına aktarılmıştır. Yinelemeli sinir ağı modeli olan derin öğrenme yöntemi Uzun-Kısa Süreli Bellek (LSTM) anahtar kelime çıkarımı için kullanılmıştır. LSTM'de eğitilen modelle, anahtar kelime önerilmiştir.
Anahtar Kelimeler: Yapay Zeka, Makine Öğrenmesi, Derin Öğrenme, Doğal Dil İşleme (NLP), Anahtar Kelime Çıkarımı

Istanbul Nişantaşı University, Graduate Education Institute Abstract of Master's Thesis

The Title of Thesis: Keyword Estimation and Accuracy of Keywords in Theses Using Machine Learning and Data Mining Techniques
Author: Aynur GÜNAY **Supervisor:** Assist. Prof. Dr. Fatih ŞAHİN
Date: 08.03.2023 **Nu. Of Pages:** vi (pre text)+90(main body)
Department Dalı: Computer Engineering **Subfield:** Information Technologies
Today, although it is easier to reach the desired data in the computer age, it has become difficult to reach accurate and reliable data due to dirty data. Since keywords make it easier for us to understand the subject of the text, its definition is of great importance in the texts. Without having to read long texts or entire documents, we can have general information about the text or document and make inferences. Choosing the right keywords in search operations and correctly defining the keywords specified in the data in the data warehouses will lead us to the right results. In this way, keywords can quickly predict what a text is related to, and time is not wasted on reading texts that are not relevant to our purpose. In this study, a study was conducted on text processing and analysis, which is a natural language processing technique in text mining. Based on the text obtained from the thesis files, it is aimed to present the most used keywords to the users as suggestions. In keyword inference, the data set collected through the thesis files was studied. For this process, 900 theses with Turkish written language were examined. First of all, normalization procedures were carried out for each thesis file, and the original keywords specified in the thesis abstract were taken from the text and recorded separately in the txt files. Thesis contents and keywords that were normalized were transferred to a single txt file. The deep learning method, an iterative neural network model, was used for Long-Short-Term Memory (LSTM) keyword inference. With the model trained in LSTM, the keyword is suggested.
Keywords: Artificial Intelligence, Machine Learning, Deep Learning, Natural Language Processing (NLP), Keyword Extraction

GİRİŞ

Son zamanlarda büyük veri üzerinde çalışmalar, hızla büyüyen veri kaynakları "veri madenciliği" çalışmalarının artmasına ve önem kazanmasına neden olmuştur.

Veri Madenciliği ve Bilgi Keşfi, verilerde daha önceden bilinmeyen, anlamlı ve değerli bilgiler elde etme işlemidir. Veri Madenciliği beş aşamadan oluşur (Yildirim ve diğerleri, 2008:2):

- Veri seçimi,
- Önişleme,
- Dönüştürme,
- Veri Madenciliği,
- Yorumlama

Veri madenciliği ile seçilen verinin ön işlemlerden geçirilerek işlenmesi gerekmektedir. Bunun için de metin madenciliği teknikleri kullanılarak veriden bilgi keşfi yapılabilmektedir. Elde edilen bilgiler ışığında ilgili kurumlar ya da uzmanlar tarafından önem taşıyan ve kullanılabilir sonuçlar elde edilerek tahminler yürütülebilmekte, bilimsel çıktılar üretilebilmektedir.

Veri madenciliği ve makine öğrenme teknikleri kullanılarak anahtar kelime belirleme için kullanılan bazı yöntemler aşağıdaki gibidir;

- TextRank: Bu yöntem, bir metin özetleme ve anahtar kelime seçme tekniğidir. Bu metod, metin içindeki kelime ve cümlelerin ağırlıklarını belirleyerek anahtar kelime çıkarımı yapar. Bu teknikte, graf teorisi kullanılır. Belge içindeki cümleler arasındaki bağlantıları analiz ederek belgenin anahtar konularını belirlemeye çalışır. TextRank, Google'ın PageRank algoritmasına benzer bir yapıya sahiptir. Belge içindeki cümleler arasındaki etkileşimi ve önemini hesaplar.
- TF-IDF: Bu yöntem, belirli bir metinde bir kelimenin görülme sıklığını (TF) ve belirli bir koleksiyondaki kelime görülme sıklığını (IDF) kullanarak anahtar kelime belirlenmesinde kullanılan bir ölçümdür. TF kelime frekansı anlamına gelir ve bir kelimenin belgedeki tekrar sayısını ölçer. IDF bir kelimenin tüm belgeler içindeki yaygınlığını ölçer. TF-IDF, bir kelimenin belgedeki önemini belirlemek için bu iki

değerin çarpımını kullanır. Yüksek TF-IDF değerlerine sahip kelimeler belgedeki önemli anahtar kelime olarak kabul edilir.

- Word2Vec: Bu yöntem, kelime veya n-gramlar için vektörler oluşturma yoluyla anahtar kelime belirlemeyi amaçlayan metinden öğrenme algoritmasıdır. Algoritma, her bir kelimeyi bir vektör olarak temsil eder ve benzer kelimelerin benzer vektörleri olduğunu öğrenir.

Yukarıda belirtilen algoritmalar ile bir metnin anahtar kelime çıkarımı yapılabilir. Literatürde sıklıkla bu yöntemler kullanılarak anahtar kelime çıkarımı yapıldığı görülmüştür. Bu tez çalışmasının ise farklılığı derin öğrenme algoritması olan LSTM ile anahtar kelime çıkarımı yapılmıştır.

Literatür taramasında, çalışma yapılan konuyla ilgili akademik yayınlar ve tezler incelenmiştir. Tezler için YÖK Tez Merkezi, akademik yayınlar için Google Akademik sitesindan yararlanılmıştır. Aramalarda özellikle son yıllarda hazırlanan tezler ve yayınlar kullanılmaya çalışılmıştır. Genellikle çalışmalarda anahtar kelime çıkarımı için kullanım sıklığına göre önerme yapılmış olup derin öğrenme yöntemlerinden LSTM modeli ile çalışmalara son yıllarda daha fazla karşılaşılmıştır. LSTM modeli ile yapılan çalışmalarda sıklıkla metinleri sınıflandırma ve özetleme çalışmaları yapıldığı gözlemlenmiştir.

Google Akademik üzerinden arama işlemi yapılırken arama ifadesi olarak "Metinlerden/dosyalardan anahtar kelime çıkarımı" kullanılmış ve arama ifadesine uygun olan akademik yayınlar incelenmiştir.

Literatür taraması kapsamında incelenen eserlerden bazıları aşağıdaki gibidir;
Çay, (2020) yaptığı çalışmada, derin öğrenme yöntemi ile akademik makalelerden otomatik olarak anahtar kelime çıkararak, elde edilen sonuçları daha önce var olan yöntemlerle kıyaslamıştır. Sonuç olarak makalenin özet ve kaynakçasına göre tam metni üzerinden yapılan anahtar kelime önerme ile sistemin başarısının daha yüksek olduğunu tespit etmiştir.

Dür, (2021) yaptığı çalışmada bilişim sektörüne ait özgeçmişler üzerinde anahtar kelime çıkarımı üzerine yaklaşımlarda bulunmuştur. Özgeçmiş yazarı tarafından belirtilen anahtar kelimelerle, Kelime Sayma, Word2Vec, BiGrams ve NN(LSTM) algoritması ile

elde edilen anahtar kelimeler karşılaştırılmıştır. Sonuç olarak LSTM modelle %72.2 başarı sağlanmış, önek ve sonek eklenmesi ile %76,7 başarı sağlanmış, konuşma bölümlerini NLTK kitaplığını kullanılarak modele tanıtarak başarı oranını %83,6'ya yükseltmiştir.

Yıldız, (2022) yaptığı çalışmada graf tabanlı yaklaşımlardan TextRank yöntemi ve istatistik tabanlı TFIDF yöntemi ile Türkçe makalelerden anahtar kelimeler çıkarmış ve orijinal anahtar kelimelerle karşılaştırma yapmıştır. Sonuç olarak TFIDF yönteminin TextRank yönteminden daha yüksek doğrulukta anahtar kelime çıkardığını belirtmiştir.

Makine öğrenmesi alanındaki çalışmalar günümüzde oldukça popülerdir. Veri miktarının artmasıyla birlikte bu verilerden anlamlı bilgiler çıkarmak ve ileriye dönük tahminlerde bulunabilmek için makine öğrenmesi yöntemleri kullanılmaktadır. Makine öğrenmesi uygulamaları geliştirebilmek için öncelikle uygun çalışma ortamının sağlanması gerekmektedir. Aksi takdirde eğitim işlemi çok uzun zamanlar alabilir, eğitim modelinin doğruluk oranı düşük çıkabilmektedir. Makine öğrenmesi uygulamaları, büyük miktarda veriyi işleyerek karmaşık modelleri eğitmek ve sonuçları tahmin etmek için yüksek hesaplama gücüne ihtiyaç duyarlar.

Makine öğrenmesi için kullanılan bilgisayarlar genellikle aşağıdaki özellikleri taşıması gerekmektedir;

İşlem gücü: Makine öğrenmesi uygulamaları yüksek miktarda hesaplama gücü gerektirir, bu nedenle çok çekirdekli işlemciler ve yüksek hıza sahip CPU'lar tercih edilir.

Bellek: Makine öğrenmesi uygulamaları genellikle büyük miktarda veri işlerler, bu nedenle yüksek kapasiteli RAM'lere ihtiyaç duyarlar. Veri yoğunluğu arttıkça daha fazla bellek gereksinimi ortaya çıkar.

Depolama: Makine öğrenmesi uygulamaları, büyük miktarda veri kullanırlar. Bu nedenle, yüksek kapasiteli depolama aygıtları gereklidir. SSD'ler veya HDD'ler gibi yüksek kapasiteli depolama aygıtları tercih edilir.

Grafik İşlemcisi (GPU): Makine öğrenmesi için GPU'lar, büyük miktarda matematiksel işlem gücü sağlarlar. Bu nedenle, makine öğrenmesi için kullanılan bilgisayarlar genellikle en az bir adet yüksek performanslı GPU içerir.

İşletim Sistemi: Makine öğrenmesi için kullanılan bilgisayarlar genellikle Linux veya macOS işletim sistemlerinde çalışırlar. Bu işletim sistemleri, makine öğrenmesi araçlarına daha iyi entegre olurlar ve daha iyi performans sunarlar.

Çalışmanın Konusu:

Tez çalışmasının konusu, Ulusal Tez Merkezinde bulunan Türkçe dili ile yazılmış tez dosyalarından derin öğrenme yöntemleri ile anahtar kelime tahmini yapmaktır.

Çalışmanın Amacı:

Tez çalışmasının amacı, Ulusal Tez Merkezinde bulunan Türkçe dili ile yazılmış tez dosyalarında belirtilen anahtar kelimelerle, derin öğrenme yöntemlerinden LSTM ile eğitilen modelin tahmin ettiği anahtar kelimelerin karşılaştırılmasıdır.

Çalışmanın Yöntemi:

Tez çalışmasında, Ulusal Tez Merkezinden elde edilen Türkçe dili ile hazırlanmış tezler metin madenciliği yöntemleri kullanılarak normalizasyon işlemlerinden geçirilmiştir. Derin öğrenme yöntemlerinden biri olan LSTM ile hazırlanan veri seti kullanılarak modelin eğitim süreci tamamlandıktan sonra başarı oranları elde edilmiştir. Son olarak da eğitilen model tahmin işlemlerinde kullanılmak üzere kaydedilmiştir.

Tez süreci üç aşamada tamamlanmıştır;

1. **Tez dosyalarının hazırlanması:**

Çalışma kapsamında öncelikle, Türkçe dilinde hazırlanmış olan tez dosyaları taranarak veri seti elde edilmiştir. Tez içeriklerinin doğrulanmış veriler olması çalışmada kolaylık sağlamıştır. Ancak her ne kadar tez içerikleri kontrol edilerek eklenmiş olsa da tezlerde birçok yazım hataları ile karşılaşılmıştır. Ayrıca tez yazım klavuzunda ortak bir yapı kullanılmamasından dolayı tez dosyalarında belirtilen özet içerisinde yer alan anahtar kelimelerin tespit edilmesi sırasında birçok sorunla karşılaşılmıştır. Yazar tarafından belirlenen orjinal anahtar kelimeleri metinden çıkarırken metin üzerinde sadece kelimeleri küçültme işlemi yapılmıştır. Çünkü bazı orijinal anahtar kelimeler rakam ya da işaret içerebilmektedir. Bu yüzden normalizasyon işlemlerinden geçirilmeden önce anahtar kelimeler tespit edilerek daha sonra tez içeriği normalizasyon işlemlerinden geçirilmiştir.

Çalışma sırasında karşılaşılan bazı sorunları belirtecek olursak;

- Anahtar kelime belirtilmemiş tezlerin olması,
- Tez özetlerinin olmaması,
- Anahtar kelime eklerken çeşitli ifadelerle eklenmiş olması, örn. “Anahtar Kelimeler.”, “Anahtar Kelimeler:”, “Anahtar Kelime:”,”Anahtar Sözcük:”,”Anahtar Sözcükler.” vb.
- Kaynakça ve sonrasındaki sayfaları içerikten çıkarmak isterken yine farklı ifadelerin olması, örn. “Yararlanılan Kaynaklar”, “Kaynaklar, “Kaynakça”, ”Bibliyografya” vb.

Normalizasyon işlemleri sırasında Türkçe karakterlerden kaynaklı birçok problemle karşılaşılmıştır. Hata alınan tezler tek tek incelenerek bu hatalar giderilmeye çalışılmıştır. Normalizasyon işlemleri kapamında yapılan işlemler aşağıdaki gibidir;

- Tüm kelimeleri küçük harfe çevirme
- Türkçe karakterler için pattern
- Romen rakamları kaldırma
- Rakamları kaldırma
- Boşlukları kaldırma
- Noktalama işaretlerini kaldırma
- İşaretleri kaldırma
- Kelimeleri köklerine ayırma işlemi (stemming)

Norrmalizasyon işlemleri tamamlandıktan sonra her bir tez için anahtar kelimeleri, tez içeriklerini ve tez numaraları text dosyalarına kaydedilmiştir.

Normalizasyon işlemi kapsamında içerikte yer alan stopwords’ler içerikten çıkartılmıştır. Stopwords, bir dildeki en sık kullanılan kelimelerden oluşan bir kelimeler listesidir. Bu kelimeler, metin işleme ve doğal dil işleme (NLP) uygulamalarında genellikle alınmaz, çünkü herhangi bir anlamsal bilgi taşımayan sadece dilbilgisi yapısına yardımcı olan kelimelerdir. Örneğin, ”*acaba, ama, ancak, belki, ben, beri, bile, bir, birçok, birçoğu, birkaç, birşey, birşeyi, böyle, bundan, böylece, bu, bu kadar, bu kadarı, bu şekilde, bu yüzden, çok, çünkü, da, daha, de, defa, diğer, diye, eğer, en, sonra, gibi, gün, sonra, hangi, her, herhangi, herkesin, hiç, iki vb…*” kelimeler içerikten çıkartılmıştır.

Tez içeriklerinde ve orijinal anahtar kelimelerde, kelimeleri köklerine ayırma (stemming) işlemi zemberek kütüphanesi kullanılarak yapılmıştır. Ön işleme sürecinin temel amacı anlamı bozmadan kelime sayısını azaltmaktır. Ancak dillerin morfolojik yapıları gereği her dilde bu adım doğru sonuçlara ulaşılamayabilir. Örneğin, İngilizce dilinin morfolojik yapısı Türkçe diline göre daha basit olduğundan dolayı İngilizce dilinde kök bulma işlemi daha kolay olmakla birlikte hata oranı daha azdır. Ancak sondan eklemeli bir dil olan Türkçe'de kök bulma işlemi oldukça zor ve anlam bozukluklarına neden olduğu için hata oranı da oldukça fazla olabilmektedir. Pythonda kullanılan Türkçe kök bulma kütüphaneleri kullanılmış ancak birçok hatayla karşılaşılmıştır. Kök bulma işleminde en başarılı olan kütüphane Zemberek olduğu görülmüş ve uygulamada zemberek kütüphanesi kullanılmıştır.

2. **Eğitim veri setinin hazırlanması:**

Normalizasyon işlemlerinden geçirilmiş olan her bir tezin tez numarası, anahtar kelimeleri ve tez içeriklerini içeren text dosyaları aralarında "/" olacak şekilde tek bir text dosyasında birleştirilmiştir. Yaklaşık 90 MB'lık bir veri dosyası oluşturulmuştur. Hazırlanan bu dosya eğitim veri seti olarak kullanılmak üzere kaydedilmiştir.

3. Modelin Eğitilmesi:

Hazırlanan veri seti eğitim veri seti olarak kullanılmıştır. Model, derin öğrenme yöntemlerinden LSTM kullanılarak eğitilmiştir.

Veri setinin bir kısmı test verisi, bir kısmı makinenin eğitim esnasında kendi kendini kontrol etmesi için kullanılan validasyon verisi, geriye kalanları da eğitim verisi olarak kullanılmıştır. Eğitim sırasında model LSTM katmanından son olarak da yoğun katmandan geçirilmiştir.

LSTM ile model eğitildikten sonra modelden doğruluk oranı elde edilmiştir. Eğitilen model kullanılarak test verisi üzerinden anahtar kelime tahmin işlemi yapılmıştır. Ayrıca eğitilen model daha sonra tezlerin anahtar kelime tahminlerinde kullanımak üzere kaydedilmiştir.

Çalışmanın Önemi:

Derin öğrenme yöntemi olan LSTM ile metinlerden anahtar kelime çıkarma işlemi için literatürde çok fazla çalışma yapılmadığı görülmüş, çoğunlukla LSTM'in verilen bir metinde cümlenin devamını tahmin etme ya da metinden özet çıkarma işlemlerinde kullanıldığı görülmüştür. LSTM son yıllarda birçok makine öğreniminde kullanılmaya başlanan bir modeldir. Popüler olan bu modelin tez çalışmasında kullanımının literatüre katkı sağlayacağı düşünülmektedir.

BÖLÜM 1: YAPAY ZEKA VE BÜYÜK VERİ TEKNOLOJİLERİ

1.1. Yapay Zeka

Günümüzde Yapay Zeka, bilgisayar dünyasının hızla büyüyen ve gelişen teknolojilerinden biridir. Bu teknoloji, bir bilgisayar programına kendi kendine düşünme ve öğrenme yeteneği sağlar. İnsanlar tarafından yapıldığında zeka gerektiren şeylerin makineler tarafından yaptırılmasıdır.

Yapay zekanın temel amacı:

- Makineleri daha akıllı ve insanlara daha yararlı hale getirmek
- İnsan zekasının yapabildiği işleri makinelere yaptırmak için programlar yazmak

Yapay zeka kurum ve kuruluşlar için çok önemlidir, çünkü işletmeler hakkında öngörülerde bulunulabilir, insan gücü ile yapılamayacak yapılsa bile doğruluğu hakkında emin olunamayacak süreçler yürütülebilir. Bugün her yerde hayatımızın her anında bulunmakta ve önemli, riskli, hayati görevleri yerine getirir duruma gelmiş durumdadır. Daha da hızlı bir şekilde gelişme göstereceği aşikardır. Örneğin alışveriş esnasında öneri şeklinde, telefonlarda ya da internet alışveriş sitelerinde sanal asistan olarak, trafikte görüntü işleme şekliyle araç takip ya da hız tesbit işlemlerinde, epostalarda ya da telefonlarda istenmeyen eposta ya da aramaların tesbit edilmesinde vb. küçük çaplı yapay zeka uygulamaları vardır. Bununla birlikte büyük çaplı yapay zeka uygulamaları da hizmet vermeye ve geliştirilmeye devam etmektedir. Sürücüsüz arabalar, uzayda otonom robotlar, konuşmayı yazıya çeviren ve diller arası çeviri programları, ameliyat robotları ve uzay teleskoplarından gelen terabyte büyüklüğündeki görüntü verisinde ilginç nesneleri tanımlayan programlar yapay zekanın en önemli ürünlerine örnek olarak verilebilir.

Yapay zeka, birçok alanda karşımıza çıkmakta ve hayatımızı büyük ölçüde kolaylaştırmakla birlikte zamandan ve insan gücünden tasarruf sağlamaktadır. Bu yüzden günümüzün en başarılı ve büyük kuruluşları, operasyonlarını iyileştirmek ve rakiplerine karşı avantaj elde etmek için yapay zekayı kullanmaktadır.

Günden güne artış gösteren devasa veri kaynakları insan gücü ile işlem yapmakta zorlanırken yapay zeka ile bu veriler işlenerek bilgi keşfi hızlı ve minimum hata ile gerçekleştirilebilmektedir.

1.1.1 Yapay Zeka Bileşenleri

- Veri toplama: Yapay zeka sistemleri için gerekli verilerin toplanması işlemini ifade eder.
- Veri hazırlama: Verilerin temizlenmesi, düzenlenmesi ve uygun şekilde hazırlanması yani veri hazırlık işlemini ifade eder.
- Öğrenme: Sistemin örnek veri kümesi üzerinden öğrenmesi ve model oluşturması işlemini ifade eder.
- Model seçimi: En uygun modelin seçilmesi ve uygulanması işlemini ifade eder.
- Eğitme: Modelin veriler üzerinde eğitilmesi ve ağın ağırlıklarının belirlenmesi.
- Model test etme: Modelin test veri kümesi üzerinde denenmesi ve performansının değerlendirilmesi işlemini ifade eder.
- Deployment: Modelin uygulamaya geçirilmesi ve sistemde kullanılması işlemini ifade eder.
- Üst yönetim: Yapay zeka sistemlerinin yönetimi ve performansının izlenmesi işlemini ifade eder.

Bu bileşenler, yapay zeka projelerinin tasarlama, planlanma, uygulanma ve bakım sürecini oluşturarak birbirine bağlı şekilde sistemin verimli çalışmasını sağlar.

1.1.2 Yapay Zekanın Getirdiği Avantajlar

Yapay zekanın getirdiği avantajlar aşağıda belirtildiği gibidir (Duggal, 2023);

- Yapay zeka sistemleri ile yapılan çalışmalarda hata oranı azdır.
- Yapay zeka sistemleri kullanılarak daha objektif ve doğru karar vermek için veriler analiz edilir.
- Risk içeren işlerde insanlar için tehlike oluşturabilecek işlerde yapay zeka sistemleri tercih edilebilir.
- Detay odaklı hassasiyet gerektiren işlerde insanlara göre daha performanslı ve doğru sonuçlar elde edilebilir,

- İnsanlar gibi yorulma, ara verme ya da günlük ihtiyaçlar olmadığı için yapay zeka sistemleri günlerce ara vermeden çalışılabilir,
- Veri ağırlıklı görevleri minimum sürede sonuçlandırabilir,
- Tutarlı ve başarılı sonuçlar elde edilebilir. Bu durum da güvenilirliği arttırır.
- İş sürekliliği sağlar.
- Yapay zeka ile hızlı sonuçlar alınabilir. Yapay zeka sistemleri, insanların yapamayacağı kadar hızlı ve verimli bir şekilde çalışabilir.
- Yapay zeka sistemleri, milyarlarca veriyi analiz ederek anlamlı sonuçlar üretebilir.

1.1.3 Yapay Zekanın Getirdiği Dezavantajları

- Yapay zeka sistemleri için çok büyük veri kümeleri gerekmektedir. Bu verilerin toplanması ve eğitilmesi zaman alıcı ve maliyetli olabilir. Bu yüzden yüksek maliyetli ve zaman gerektirir.
- İnsanların yapabileceği bazı işler yapay zeka ile makinelere yaptırılabilir. Bu da bazı mesleklerin yerini alarak işsizliğin artmasına sebep olur.
- İnsanlarda tembelliği ve hareketsizliği arttırır,
- Yapay zeka araçları oluşturmak için sınırlı kalifiye işçi kaynağı bulunur.
- İnsanlara özgü davranışlar öğretilemediği için sadece verilen görevi bilir, karşılaşılan sorunlar onun için anlamsızdır, çözüm üretemez.
- Yapay zeka sistemlerinin nasıl çalıştığını anlamak zordur ve açıklanması kolay değildir.

Bu belirtilen dezavantajların bir kısmını içermekte olup her yapay zeka sistemi için geçerli olmayabilir. Hatta yapay zeka teknolojisinin gelişmesiyle birlikte, dezavantajların azaltılması ve çözümlenmesi mümkündür.

1.1.4 Yapay Zeka Uygulama Alanları

Yapay zekanın başlıca uygulama alanları;

- Önerici sistemler: Geçmiş davranışlara göre yeni içerik önerilebilir. Örneğin mağazalarda ilişkili bir ürün, alışveriş alışkanlıklarına göre müşterilere düzenlenen kampanyalar, sosyal medya sitelerinde arkadaş önerileri vb.
- Ses ve görüntü işleme: Ses ve görüntünün işlenerek bilgi çıkarımı yapabilir. Örneğin ses, el yazısı ve yüz tanıma.

- Mekine çevirisi: Diller arası metin çevirileri yapabilir. Örneğin Google Translate, Seslisozluk Çeviri vb.
- Regresyon analizi: Geçmişte edinilen bilgilere dayanarak gelecekle ilgili tahminde bulunabilir.
- Nesne tanıma: Görüntü veya video verilerinde bulunan nesneleri tanıma, sınıflandırma, yerine bulma gibi işlemler için kullanılabilir.
- Robotik: Robotların yapması gereken görevleri tanımlamak için kullanılır.
- Otomatik makine öğrenme: Verileri analiz etmek ve kendini geliştirmek için kullanılabilir.

1.1.5 Yapay Zeka ve Makine Öğrenmesi Arasındaki İlişki

Yapay Zeka, makinelere insana benzeyen özellikler ve beceriler kazandırarak, insanların yapabildiği işleri yapmasını amaçlar. Yapay zeka, makine öğrenme, doğal dil işleme, görsel tanıma ve diğer benzer teknolojileri kullanır. Makine Öğrenmesi ise, yapay zeka sistemlerinin öğrenme ve gelişme kabiliyetini ifade eder. Makine öğrenmesi sistemleri verileri kullanarak kendilerini eğitir ve bu veriler üzerinden tahmin yapar. Makine öğrenmesi yapay zeka sistemlerinin temel teknolojik araçlarından biridir.

Yapay zeka, insanların bir durum karşısında verdiği tepkiyi taklit eden bir sistem geliştirmeyi hedefler. Makine Öğrenimi, kendi kendine öğrenen algoritmalar ile verileri işleyerek bir tahminde ya da öneride bulunabilir.

Yapay zekada hedef karmaşık problemi çözmek için doğal zekayı taklit etmek iken, makine öğrenmesinin hedefi, makinenin bu görevdeki performansını en üst düzeye çıkarmak için belirli bir görevle ilgili verilerden ders alarak öğrenme işlemini gerçekleştirmektir.

Kısaca, yapay zeka genel bir kavramdır ve makine öğrenmesi yapay zeka sistemlerinin bir bileşenidir.

1.1.6 Yapay Zeka ve Veri Madenciliği Arasındaki İlişki

Yapay Zeka ve Veri Madenciliği, birbirlerine yakın ve birbirlerini destekleyen teknolojik alanlar olarak ilişkilidir.

Veri Madenciliği, verilerin toplanması, temizlenmesi, ön işlemesi ve analiz edilmesi sürecini içeren bir teknolojidir. Bu veriler, Yapay Zeka ve Makine Öğrenmesi algoritmalarının eğitilmesi ve doğru sonuçlar üretmesi için gereklidir.

Sonuç olarak, Veri Madenciliği, Yapay Zeka ve Makine Öğrenmesi süreçlerinin başarısını etkileyen temel bir bileşendir.

1.2 Büyük Veri

Büyük veri, verinin işlenerek analiz edilerek anlamlı hale dönüştürülmüş şeklidir. Sıklıkla terabyte boyutlarında olan, çok büyük miktarda, hızlı ve sürekli olarak artan, çeşitli formatlarda bulunan verileri tanımlar. Büyük veri, sıradışı veri boyutlarından ve hızından dolayı, geleneksel veri işleme tekniklerinin ve yöntemlerinin işleme kapasitesinden çok daha büyük kalır.

Büyük veri teknolojileri, verilerin depolanması, işlenmesi ve anlamlı hale getirilerek yararlı sonuçlar çıkarılması için kullanılır.

Büyük verinin oluşumuyla birlikte verilerin depolanması, güvenliği ve kişisel verilerin işlenmesi, korunması ve analizi gibi problemler oluşmaktadır. Bu süreçler veri madenciliği, bilgisayar bilimi, makine öğrenmesi, istatistik ve matematiksel algoritmalar birlikte çalışmasını gerektirir.

Şekil 1: Büyük Veri *(https://medium.com, 2019)*

1.2.1 Büyük Veri Oluşumu Süreçleri

- Veri toplama: Veriler, internet, mobil cihazlar, sensörler, sosyal medya ve diğer kaynaklar tarafından toplanır.
- Veri depolama: Toplanan veriler, veritabanları, bulut depolama sistemleri ve diğer veri depolama yöntemleri ile saklanır.
- Veri işleme: Veriler, işlem ve analitik yöntemleri kullanılarak düzenlenir ve hazır hale getirilir.
- Veri analizi: Veriler, analitik ve görselleştirme yöntemleri kullanılarak anlamlı sonuçlar çıkarılır.
- Veri kullanımı: Çıkarılan veriler, çeşitli amaçlar için kullanılabilir, örneğin pazar araştırması, müşteri davranış analizi.

1.2.2 Büyük Veri Avantajları

- Büyük veride, verinin daha büyük ve çeşitli olmasından dolayı daha çok bilgi edinildiği için eksiksiz sonuçlar alınabilir.
- Eksiksiz sonuçlar, verilere daha fazla güven anlamına gelir. Bu da oluşabilecek problemlerin çözümünde sorun yaşanılmayacağını, çözüm üretiminde ve karar almada hızlı ve etkili sonuçlar elde edileceği anlamına gelir.
- Daha fazla veriye erişim sağlar ve bu verilerin anlamlı ve etkin kullanım imkânı verir.
- Verilerin analizi ve keşfedilmesi için güçlü araçlar sunar. Bu, daha fazla fikir ve bulgu keşfetmeye olanak tanır.
- Daha iyi kararlar vermek için daha fazla veri ve analiz sunar.
- Müşterilerin davranışlarını anlamaya ve daha etkili pazarlama kampanyaları oluşturmaya yardımcı olur.
- İşletmelerin verimliliğini artırmak ve maliyetlerini azaltmak için kullanılabilecek veriler sunar.
- Daha fazla fırsatı keşfetme ve değer yaratma imkânı sağlar.

1.2.3 Büyük Veri Dezavantajları

- Büyük veri, çok fazla kişisel ve hassas veri içerebilir, bu verilerin güvenliği ve gizliliği için endişe verici olabilir.

- Büyük veri setlerinde, verilerin doğruluğu ve güncelliği sık sık sorgulanabilir, bu da analizlerin doğruluğunu ve kalitesini etkileyebilir.
- Büyük veri setlerinin yönetimi ve depolanması, maliyetli ve zaman alıcı bir süreç olabilir. Veri depolamaya yönelik yeni teknolojiler geliştirilmesine rağmen, veri boyutu git gide artmakta. Büyük veri sahibi şirket/kurum ya da kuruluşlar verilerini yönetebilmek ve depolamak için mücadele ediyor. Bu da büyük maliyetler oluşturabiliyor.
- Büyük veri setleri, analiz etmek için güçlü bir bilgi ve teknik beceri gerektirebilir.
- Büyük veri setleri, güçlü bir işletim sistemi, depolama ve işlem gücü gerektirir, bu da ek maliyetlere neden olabilir.
- Büyük veri teknolojisi, verilerin kullanımı ve paylaşımı ile ilgili etik konuları da ortaya çıkarabilir.

1.2.4 Büyük Veri Depolama Alanları

Büyük veri depolama sistemleri, çok büyük veri setlerini depolamak için kullanılır. Bu sistemler, verilerin anlamlı şekilde analiz edilmesini ve kullanılmasını mümkün kılar.

Büyük veri depolama sistemlerine örnek verecek olursak;

- Hadoop Dağıtılmış Dosya Sistemi (HDFS): HDFS, verilerin dağıtılmış olarak saklanmasını ve veri işleme görevlerini yapmasını sağlar. Hadoop, büyük veri işleme görevleri için kullanılan bir platformdur ve verilerin analiz edilmesi, sınıflandırılması ve görselleştirilmesi gibi uygulamalarda kullanılabilir. Örnek olarak, veri madenciliği, sosyal medya analizleri, finansal analizler ve reklam analizleri gibi uygulamalarda Hadoop kullanılabilir.
- NoSQL veritabanları: NoSQL veritabanları, çok fazla veri içeren ve hızlı erişim gerektiren uygulamalar için tasarlandı. Örneğin MongoDB, Cassandra, Couchbase, Redis vb.
- Bulut depolama: Bulut depolama, büyük veri setlerinin dağıtılmış ve güvenli bir şekilde depolanmasını sağlar. Örnek olarak Amazon S3verilebilir. Amazon S3, Amazon Web Service (AWS)'nin bir veri depolama hizmetidir. Kullanıcılar, verilerini güvenli ve kolayca yönetebilecekleri bir ortamda saklamak için Amazon S3'ü kullanabilirler. Amazon S3, dosyaları, objeleri ve blokları depolamak için kullanılabilecek bir bulut tabanlı depolama hizmetidir ve dosyaların yedeklenmesi,

yönetimi ve paylaşılması gibi işlemleri destekler. Amazon S3, milyarlarca dosya için yüksek kapasite ve yüksek performans sunan bir bulut depolama platformudur.

- Sanal disk: Sanal disk, büyük veri setlerinin depolanması için kullanılan bir alternatiftir. Sanal disk, fiziksel diskler yerine kullanılan yazılımsal disklerdir. Sanal disk, kullanıcıların verilerini depolamak için kullanılabilecek bir disk görüntüsü oluşturur. Fiziksel disklere göre daha esnek ve kolay yönetilebilir bir yapıya sahiptir ve kullanıcıların verilerini depolamak için farklı seçenekler sunar. Sanal diskler, sanal makine (VM) veya sanal bir ortamda kullanılabilir ve fiziksel diskteki verilerin sanal disk üzerinde korunmasını sağlar. Sanal diskler, verilerin kolayca yedeklenmesi ve geri yüklenmesi için de kullanılabilir ve fiziksel disk arızaları durumunda verilerin korunmasını sağlar. Ayrıca, sanal diskler, fiziksel disklerin kapasitesinden daha fazla veri depolamak için de kullanılabilir.
- İlişkisel veri tabanları: Verilerin tablo olarak saklandığı ve tablo arasındaki ilişkilerin belirlendiği veri yönetim sistemidir. İlişkisel veri tabanları, verileri sütunlar ve satırlar şeklinde düzenler ve veriler arasındaki ilişkileri tanımlar. Her satır, bir veri nesnesi (örneğin, öğrenci, ders, notlar vb.) ile ilgili verileri içerir ve her sütun, veri nesnesi ile ilgili bir özelliği temsil eder. İlişkisel veri tabanları, verilerin güncellenmesi, sorgulanması ve analiz edilmesi gibi işlemleri kolaylaştırır ve verilerin güvenli bir şekilde saklanmasını sağlar. PostgreSQL, MySQL, Oracle, Microsoft SQL Server gibi veri tabanı yönetim sistemleri ilişkisel veri tabanı örnekleridir.
- Object Storage Sistemler: Object storage sistemleri verileri objeler olarak saklar. Bu objeler, verilerin metadataları ve verilerin kendisi gibi farklı veri parçalarını içerebilir. Her objenin benzersiz bir adresi vardır. Object storage sistemleri, verilerin yedeklenmesi ve paylaşılması gibi işlemler için daha uygun bir ortam sağlar.

BÖLÜM 2: VERİ MADENCİLİĞİ

2.1 Veri Madenciliği

Kurum ve kuruluşlarda toplanan büyük miktarlardaki verilerden değerli bilgiye erişmemizi sağlamak için veri üzerinde yapılan işlemlere veri madenciliği denilmektedir. Elde edilen bilgiler doğrultusunda birtakım kararlar alınabilmekte, sorunlar çözülebilmekte, mevcut riskler azaltılarak yeni fırsatlar elde edinilebilmekte ve ileriye yönelik tahminlerde bulunularak mevcut başarıyı katlayacak hedefler belirlenerek yönlendirmeler yapılabilmektedir.

Belli bir amaç doğrultusunda toplanan veriler işlenmedikçe anlamlı bir bilgiye ulaşılamaz.

Kurum ya da kuruluşlarda büyük veri üzerinde çalışmalar son zamanlarda önem taşımakta ve bu yöndeki çalışmalar ve destekler hızla artmaktadır. Bilgi teknolojilerinin gelişmesi ve veri madenciliği ile ilgili programların artması çalışmaları kolaylaştırmaktadır. Özellikle hizmet sektöründeki pandemiyle birlikte ön plana çıkan internet alışverişleri kullanıcıların alışveriş alışkanlıklarını öğrenerek doğru zamanda ve doğru yerde ilgili kişiler indirim ya da hatırmaltma olarak önlerine sunulabilmesi veri madenciliği sayesinde kolaylaşmıştır. Böylelikle satış işlemlerinde artış sağlanmıştır. Bu nedenle eldeki verilerin doğru kullanımı ve doğru işlenmesi büyük önem taşımaktadır. Günümüzde, sahip olunan bu bilgiler gücü temsil etmekte ve bu bilgiye erişim zorlaşmakta hatta belirli bir ücretle erişilebilmektedir.

Veri madenciliği oldukça geniş bir uygulama alanına sahip olmakla birlikte birçok disiplinle etkileşim halindedir. Veri madenciliğinin uygulama alanları; İstatistik, Makine Öğrenmesi, Veri Tabanı Teknolojileri, Görselleştirme ve Yapay Zeka gibi disiplinlerdir.

Şekil 2: Veri Madenciliği Disiplinleri*(Akgün ve Özek, 2020)*

2.2 Veri Madenciliği Avantajları

- Doğru ve güvenilir bilgi toplanmasına yardımcı olur.
- Diğer veri uygulamalarına kıyasla verimli, düşük maliyetlidir.
- İşletmelerin karlı üretim ve operasyonel düzenlemeler yapmasına yardımcı olur
- İşletmelerin bilinçli kararlar almasına yardımcı olur
- Riskleri ve dolandırıcılığı tespit etmeye yardımcı olur
- Veri bilimcilerin çok büyük miktardaki verileri hızla kolayca analiz etmesine yardımcı olur
- Veri bilimcilerin davranışların ve eğilimlerin otomatik tahminlerini hızlı bir şekilde başlatmasına yardımcı olur.

2.3 Veri Madenciliği Dezavantajları

- Çoğu veri analitiği aracı karmaşıktır ve kullanımı zordur. Veri bilimcilerin, araçları etkili bir şekilde kullanmak için doğru eğitime ihtiyacı var.
- Veri madenciliği teknikleri yanılmaz değildir, veri kümesi eksik ise bilgilerin tamamen doğru olmaması riski vardır.
- Şirketler, topladıkları kullanıcı verilerini diğer işletmelere satabilir. Bu durum kullanıcılarda gizlilik endişelerini artırır.
- Veri madenciliği, büyük veritabanları gerektirir ve bu da süreci yönetmeyi zorlaştırır. Alanında uzman personel ihtiyacı ortaya çıkar.

2.4 Veri Madenciliği Süreçleri

- Veri Filtreleme: Kullanılacak verilerin elde edilmesi.
- Veri Temizliği: Verinin içerisinden gereksiz, tutarsız ya da gürültülü olanların ayıklanması
- Veri Bütünleştirme: Farklı kaynaklardan elde edilen ilişkili verilerin birleştirilmesi.
- Veri Seçme: Temizlenen ve birleştirilen verilerden, analize uygun olanların seçilmesi.

- Ver Dönüştürme: Verilerin madencilik için uygun biçime dönüştürülmesi.
- Veri Analizi: Hazırlanmış olan verilerin uygun veri madenciliği algoritmaları ile analiz edilmesi.
- Yorumlama ve Sunma: Veri madenciliği uygulaması gerçekleştirildikten sonra, elde edilen sonuçlar yorumlanması ve sunulması.

2.5 Veri Madenciliğinde Karşılaşılan Problemler

Verinin boyutlarının büyümesi, beraberinde büyük sorunları da ortaya çıkarmaktadır. Çalışılan ortamdaki veri büyüklüğüne göre analiz sonuçları farklılık gösterebilmektedir. Bu nedenle veri madenciliği işlemlerinde öncelikle işlenmemiş veri üzerinde çalışmalar yapılması gerekmektedir.

Veri madenciliği uygulamalarında karşılaşabileceğimiz başlıca problemler;

Artık Veri: Veri kümesinde yer alan ancak mevcut probleme katkısı olmayan gereksiz alanlardır. Artık veri, veri madenciliği işlemlerini yavaşlatabilir ya da yanıltıcı sonuçlar oluşmasına neden olabilir. Artık veri, veri kaynaklarından elde edilen verilerin temizlenmesi ve seçilmesi işlemleriyle azaltılabilir.

Boş Veri: Veri madenciliğinde, veri kümesinde boş veya eksik değerlerin bulunmasıdır. Bu durum, veri analizi ve modelleme işlemlerinde sorunlar oluşturabilir ve doğru sonuçların elde edilmesini engelleyebilir. Boş veri, veri kaynaklarından elde edilen verilerin temizlenmesi ve doldurulması işlemleriyle çözülebilir.

Gürültülü Veri: Verinin toplandığı sırada oluşan sistem dışı hatalara gürültü denir. Ölçülen bir değerdeki hata veya yanlış nitelik değerleri gürültülü veri oluşumuna neden olur. Veri kümesinde birçok sınıfın değeri yanlış olabilir.

Gürültülü veri oluşumuna neden olan durumlar;

- Hatalı veri toplama araçları
- Veri girişi problemleri
- Veri iletimi problemleri
- Toplanan verinin kısıtların olmaması, min-max değer, numerik -string
- Nitelik isimlendirmelerde tuttarsızlıklar

Belirsizlik: Hatanın boyutu ve verideki gürültünün derecesi ile ilgilidir. Veri kümesinde bulunan bilginin tam olarak ne olduğu veya ne olacağının belirlenememesi durumudur. Belirsizlik, veri madenciliği işlemlerinde doğru sonuçların elde edilmesini engelleyebilir.

Örneğin, önceki davranışların gelecekteki davranışları tahmin etmek için kullanılması gibi veri madenciliği uygulamalarında belirsizlik, modelleme işlemlerinde ve sonuçların yorumlanmasında zorluklar yaratabilir. Belirsizliği azaltmak için veri kaynaklarından elde edilen verilerin temizlenmesi, filtrelemesi ve doğrulanması gibi işlemler yapılabilir.

Eksik Veri: Veri analizi için bazı niteliklerin değerleri bilinemeyebilir.

Eksik veri oluşumunun nedenleri;

- Verinin yanlışlıkla silinmesi,
- Diğer alanlarla tutarsızlığı nedeniyle silinmesi,
- Toplanan verinin hatalı olması durumundan dolayı silinmesi,
- Veri girişi sırasında önemsiz görülerek zorunlu alan tanımlanmaması

Eksik veriler için yapılan işlemler;

- Eksik verilerin bulunduğu satırların ya da niteliğin silinmesi,
  Bu işlem veriyi bozabilir ya da değerli verilerin kaybolmasına neden olur bu da istenmeyen bir durumdur. Çünkü bu durum sonucu değiştiebilir.
- Eksik alanlar için değişken atanması (null, boş, eksik vb),
- Eksik alanların geliştirici tarafından doldurulması,
- Aynı sınıftaki dolu kayıtların ortalama değeri ile eksik alanların doldurulması

Sınırlı Bilgi: Veri tabanları genellikle veri madenciliği dışındaki amaçlar için kullanılmaktadır. Veri madenciliğindeki gibi veri kümesine ait nitelikleri sunmak gibi bir hedefi yoktur. Sınırlı bilgi, veri kümesinde mevcut olan bilginin yetersiz veya eksik olması durumudur. Sınırlı bilgi, veri madenciliği işlemlerinde doğru sonuçların elde edilmesini zorlaştırabilir. Örneğin, veri kümesi sadece belirli bir zaman dilimi için verileri içeriyorsa, gelecekteki davranışların tahmin edilmesinde sınırlı bilgi nedeniyle zorluklara neden olabilir. Sınırlı bilgiye rağmen, veri madenciliği işlemleri yapmak için fikirler ve yöntemler vardır. Örneğin, veri kümesinin genişletilmesi, veri kaynaklarından ek verilerin elde edilmesi ya da veri analizi ve modelleme işlemlerinde yapay veri oluşturma gibi yöntemler kullanılabilir.

Veritabanı Boyutu ve Dinamik Veri: Veritabanlarının büyük boyutlu ve dinamik olma durumu veri madenciliği uygulamalarının uygulanabilmesine engel oluşturan önemli problemlerden biridir. Veritabanındaki veriler sürekli güncelleme, ekleme ve silme işlemlerine maruz kalmaktadır. Böyle bir durum, veri madenciliği çalışmalarında veri kümesine uygulanan kural setlerinin değişime uğrama problemine ve istikrar sorununa

neden olur. Çalışmalarda hatalı ve tutarsız sonuçlar elde edilmesine yol açar. Veritabanlarında verilerin sürekli eklenip güncellenmesi eş zamanlı olarak veri madenciliği algoritmalarının devreye alınması her iki taraf için de performans kaybına neden olur. Bu durum ciddi problemlere hatta veri kayıplarına bile neden olabilir. Veritabanında işlem yapılan verinin üzerinde başka bir işlem yapılamayacağı için bu durum da zaman kaybına neden olabilir. Ayrıca veritabanında değişen bir kaydın veri madenciliği veri kümelerinden hangisinde olduğunun tespitinin zorluğu da bir başka problemdir. Bu yüzden veri madenciliği çalışmalarında büyük boyutlu ve dinamik verilerle çalışmak üretilen sonuçların başarı oranını düşürecektir.

Farklı Tipteki Verileri Ele Alma: Kullanılan veri kümeleri makine öğreniminde olduğu gibi yalnızca kategorik veri türleri olmayabilir. Görsel veriler, coğrafi bilgi içeren veriler, ses veya video verileri, zamansal veriler gibi farklı tipteki veriler üzerinde de işlem yapılmasını gerektirir.

2.6 Veri Madenciliğinin Kullanıldığı Alanlar

Günümüzde büyüyen veriyle birlikte oldukça popüler olan veri madenciliği birçok alanda halihazırda kullanılmakta, farklı alanlarda da kullanılmaya başlanmıştır.

2.6.1 Pazarlama

- Pazarlama stratejilerinde
- Müşteri/Çalışan Memnuniyet analizlerinde
- Satınalma tercihlerinin belirlenmesinde
- Kampanya ürünlerinin tespitinde
- Müşteri İlişkilrinin iyileştirilmesinde
- Satış tahminlerinde
- Ödemelerini aksatan müşteri tespitinde
- Müşteri davranışlarını anlamada
- Müşteri profillerini oluşturmada
- Mağaza konum noktaları tespitinde

2.6.2 Bankacılık

- Kredi taleplerinin incelenmesinde
- Müşteri profilinin belirlenmesinde

- Portföy yönetininde
- Kredi kartı dolandırıcılığı tespitinde
- Risk yönetiminde
- Finanasal verileri analiz etmek

2.6.3 Bilim ve Sağlık

- Sağlık verileirni analiz etmede
- Tedavi sürecinin tespitinde
- Test sonuçlarının analizinde
- Yeni ürün geliştirmede
- Genetik hastalıkların tespitinde
- Yeni virus türlerinin keşfinde
- Hastalık tespitinde
- MR-Röntgen görüntülerinin incelemesinde
- İlaç türlerinin sınıflandırılmasında
- Yeni teknolojilerin keşfinde

2.6.4 Telekomünikasyon

- Hatların yoğunluk tespitinde
- Kalite ve iyileştirme analizlerinde
- Servis kalitesinin iyileştirilmesinde

2.6.5 Borsa

- Piyasa analizlerinde
- Hisse senedi tahminlerinde
- Alım-satım optimizasyonunda

2.6.6 Harita ve Coğrafi Bilgi Sistemleri

- Haritalama verilerinin analizlerinde
- Güzergah planlamasında
- Yolların ve bölgelerin optimizasyonunda
- Coğrafi bilgi sistemleri için veri toplamada

2.6.7 Eğitim

- Eğitim verilerinin analizlerinde
- Öğrenci başarısının takibinde
- Eğitim programlarını oprtimize etmek için
- Öğrenciye özgü öğrenme yolları oluşturmak için

2.6.8 Hukuk

- Hukuk verilerinin analizlerinde
- Davaların takibinde
- Hukuk davalarının tahmininde
- Hakaret içeren metinlerin analizinde
- Hukuk sürecini optimize etmek için

Veri madenciliği, birçok farklı alanda kullanılmakta ve her geçen gün yeni alanlara uygulanmaktadır.

BÖLÜM 3: METİN MADENCİLİĞİ VE DOĞAL DİL İŞLEME

3.1 Metin Madenciliği

Metin Madenciliği, veri madenciliği ve doğal dil işleme (NLP) tekniklerinin birleştirilmesiyle oluşan bir analitik süreçtir. Metin madenciliği, büyük metin veri setlerinden anlamlı ve kullanışlı bilgiler çıkarmaya çalışır. Bu bilgiler arasında, örneğin, belirli kelime ve cümlelerin frekansları, anlamlı konular, müşteri tercihleri/ihtiyçları vb. yer alabilir.

Metinlerde analiz işlemleri yapmak için veri madenciliğinin bir parçası olan metin madenciliği yöntemleri kullanılmaktadır.

Metin madenciliğinde veriler doğal dille yazılmış veriler olup veri madenciliğindeki veriler ise bilgi veritabanlarında toplanan verilerden elde edilmektedir.

Örneğin makaleler, haber sitelerindeki haber metinleri, köşe yazıları ve web sitelerindeki içerikler metin madenciliği teknikleri ile analiz edilebilmekte ve çıkarımlarda bulunulabilmektedir.

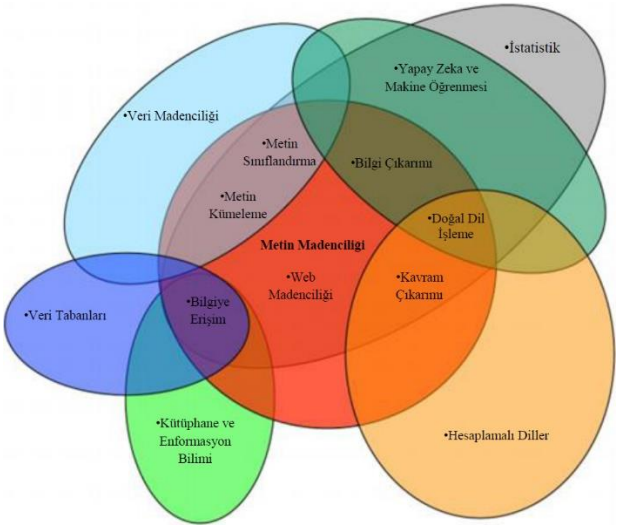
Metin madenciliği; insan gücüyle analiz edilemesi mümkün olmayan büyük verilerin daha güvenli ve hızlı bir şekilde hata oranını minimize ederek anlamlı hale getirmek için kullanılmaktadır. Metin madenciliği ile işlenmemiş birçok veri kaynağının işlenmesiyle anlamlı bilgiler elde edilebilmektedir.

Metin madenciliği; yapısal olmayan veriden ilginç, önceden bilinmeyen ve önemsiz olmayan bilgileri keşfeden, çok sayıda dokümanı analiz eden bir teknolojidir (Karadağ ve Takçı, 2010:1).

Metinlerde analiz işlemleri yapmak için veri madenciliğinin bir parçası olan metin madenciliği yöntemleri kullanılmaktadır. Metinler doğal dille yazılmış verilerdir. Örneğin makaleler, haber sitelerindeki haber metinleri, elektronik postalar, köşe yazıları, web sitelerindeki içerikler vb. metin madenciliği teknikleri ile kısa zamanda analiz edilebilmekte ve bu metinlerden anlamlı bilgiler elde edilerek etkili ve doğru karar alınabilmektedir.

Metin madenciliğindeki veriler doğal dille yazılmış veriler olup veri madenciliğindeki veriler ise bilgi veritabanlarında toplanan verilerdir.

Metin Madenciliğinin ilişkili olduğu disiplinler ve yöntemler aşağıdaki grafikte gösterilmiştir. Grafikte de görüldüğü üzere metin madenciliği diğer alanlarla da çalışma yapılması gereken bir veri işleme alanıdır.



Şekil 3: Metin madenciliğinin diğer disiplinler ile ilişkisi*(http://www.metinmadenciligi.com, 2021)*

3.2 Metin Madenciliği Kullanım Alanları

Metin Özetleme: Metin özetleme, metnin önemli kısımlarını belirleyerek, metnin anahtar kelimelerini ve cümlelerini kapsayan kısa bir özet çıkarma işlemidir.

Metin Çevirisi: Metinleri farklı dillere çevirmek için kullanılır.

Sesli Metin Dönüştürme: Sesli metinleri yazılı metinlere dönüştürme veya sesli metinleri anlamak için kullanılır.

Anahtar Kelime ve Cümle Çıkarımı: Metinlerin içeriğini hızlı bir şekilde anlamak ve metnin konusunu belirlemek için kullanılır.

Metin Özellik Çıkarımı: Metinlerden özellikleri çıkarmak için kullanılır. Bu özellikler, metinlerin yazım tarzı, dil kullanımı, ton vb.

Metin Öneri: Metinleri önermek için kullanılır. Öneri işlemi, metinlerle ilgili benzer içerikleri veya ilgili kelimeleri önerme şeklindedir. Örneğin, bir haber metninde okuyucunun ilgisini çeken başka haberleri veya ilgili konuları okuyucuya önerebilir.

Sentiment Analizi: Metinlerin duygu ve düşüncelerini analiz etmek için kullanılır. Metinlerin pozitif, negatif veya nötr olduğunu tahmin eder.

Metin Sınıflandırma: Metinleri sınıflandırma, metinleri belirli kategoriler veya etiketler altında sınıflandırmak için kullanılır. Örneğin, haber metinleri güncel haber, ekonomi, sağlık, teknoloji gibi kategoriler altında sınıflandırılabilir.

Metin Eşleştirme: Metinler arasındaki benzerlikleri belirlemek için kullanılır. Metinler arasındaki benzerlikleri ölçerek, metinlerin aynı konuda veya yazılış tarzında olup olmadığını tahmin eder.

Metin Katmanlama: Metinleri bölümlere ayırmak için kullanılır. Metinleri bölümler halinde ayırır ve her bölümün konusunu belirler.

Cümle Tamamlama: Cümleleri tamamlamak için kullanılır. Yazarların yazım hatalarını düzeltir veya kullanıcıların yazımını tamamlamak için kullanılır.

Metin Anlamı Çıkarımı: Metinlerin anlamını çıkarmak için kullanılır.

Metin Kavramsallaştırma: Metinleri kavramsallaştırmak için kullanılır.

Metin Sinonim Çıkarımı: Metinlerdeki kelimelerin sinonimlerini çıkarmak için kullanılır.

Metin Zenginleştirme: Metinleri zenginleştirmek için kullanılır. Zenginleştirme işlemi, metinleri daha anlamlı ve anlaşılır hale getirmek için yapılır.

Metin Üretimi: Yapay zeka ve makine öğrenmesi teknikleri ile, verilen bir kelime veya kelime öbeğine göre sıradaki kelimeyi tahmin ederek metinlerin otomatik olarak üretilmesi işlemidir. Metin üretimi, chatbotlar, metin tamamlama uygulamaları, hikaye ve roman yazarlığı vb. alanlarda kullanılabilir.

Metin madenciliği uygulamaları özellikle şirketler tarafından büyük önem taşımaktadır. Çoğunlukla müşteri yorumlarından, ürünlere ait yapılan yorumlardan, sosyal medyadan, şirkete gönderilen epostalardan, çağrı merkezine yapılan müşteri dönütlerinden gelen metinler incelenmektedir. Böylelikle müşterilerin şirket hakkındaki düşüncüleri öğrenilebilmekte, ürünlerdeki sorunlar bu dönütlere göre giderilmekte, yeni kampanyalar düzenlenebilmekte ve müşteri hizmetleri daha kaliteli hale getirilebilmektedir. Çünkü şirketler için müşteri memnuniyeti büyük önem arz etmekte, buna yönelik çalışmalar yapılmaktadır.

Metin madenciliğin diğer bir uygulama alanı da öznitelik çıkarımlarının yapılabilmesidir. Örneğin büyük şirketlere yapılan iş başvurularında özgeçmişlerin taranması, akademik araştırmalarda aranan nitelikteki bilgiye hızlı ve doğru şekilde ulaşmayı, spam epostaları engellemeyi sağlamaktadır.

Günümüzde büyük şirketler müşteri hizmetleri için chatbot kullanmayı tercih etmektedir. Chatbot, müşteriye hizmet veren, ürünlerle ilgili belirtilen sorunları otmatik yanıtlayan doğal dil anlama teknolojisini kullanır.

3.3 Metin Madenciliğinde Doğal Dil İşleme

Bilgi keşfi için hızlı, en az maliyetle, doğru ve anlamlı bilgi çıkarımı sağlayan metotlar tercih edilmektedir. Metin madenciliğinde kullanılan Doğal dil işleme (NLP), insan dilini otomatik olarak analiz etme ve anlama yeteneği nedeniyle araştırmalarda büyük önem taşımaktadır. Çalışmalar makine öğrenmesi yöntemlerinin artışı sonrası hız kazanmıştır. NLP çalışmaları çoğunlukla İngilizce dili üzerine yapılmaktadır. NLP çalışmaları kapsamında İngilizce dili üzerinde çalışan birçok etkili doğal dil işleme araçları geliştirilmiştir. Ancak Türkçe dili üzerine yapılan çalışmalar henüz yetersizdir. Özellikle dil verilerinin karmaşıklığının (lehçeler, kısaltmalar, Türkçe'nin sondan eklemeli dil özelliği vb.) ortaya çıkardığı zorluklar nedeniyle, NLP Türkçe metin analizlerinde henüz yüksek oranda başarı sağlanamamıştır. Oflazer (2012) yayınladığı bir makalede de Türkçe'nin sondan eklemeli yapısı, cümle öğelerinin dizilişlerinin çok değişken ve esnek olması, "sözcükleşmiş" birleşik isimlerin olması gibi unsurların Türkçe'de doğal dil işleme çalışmalarını ne kadar güçleştirdiğini ortaya koymuştur.

Metin madenciliği işlemleri sırasında birçok zorluklarla karşılaşılabilir. Çünkü eldeki veriler genellikle belirsiz, tutarsız olabilir. Sözdizimindeki farklılıkların yanı sıra metinde bölgesel lehçeler ve sosyal medya dilinde kullanılan kısaltmalar kullanılabilmekte, bu kullanımından kaynaklanan belirsizlikler metnin analiz işlemlerini karmaşık ve zor bir hale getirebilmektedir. Özellikle Türkçe dilinin sondan eklemeli dil olması, köklerine ayırma işlemi sırasında büyük zorluklara neden olmaktadır. Sonuç olarak, metin madenciliği algoritmaları kullanılırken, veri kümelerini sınıflandırırken, etiketlerken ve özetlerken bu tür durumlar için de eğitilmelidir

3.4 Türkçe'deki Doğal Dil İşleme Araçları

Türkçede, NLP araçları arasında birçok farklı araç bulunur. Ayrıca, bu araçlar genellikle birleştirilerek kullanılabilir. Örneğin metin özetleme ve anahtar kelime çıkarımı araçlarının birleştirilmesiyle metinlerin konusu ve önemli kısımları hızlı bir şekilde belirlenebilir.

Türkçe metinler için derin öğrenme yöntemleri ile geliştirilmiş bazı araçlar aşağıdaki gibidir;

- POS (part of speech) etiketleme: POS etiketleme, metin içindeki kelimelere belirli bir dilin gramer kurallarına göre bir özne, nesne, zarf, fiil, sıfat vb. olarak etiketlemek için kullanılan araçlar. Bu etiketleme, metnin anlamının ve yapısının daha iyi anlaşılmasına yardımcı olur.
- İsimlendirilmiş Varlık Tanıma (Named Entity Recognition-NER): Metin içerisinde belirli bir kategoriye (isim, yer, tarih, şirket, para birimi, vb.) ait kelime veya kelime gruplarının tanımlanması, kümelenmesi ve etiketlenmesi için kullanılan araçlar.
- Duygu Analizi: Metin veya ses içerikli verilerin makine tarafından duygusal durumunun anlaşılması için kullanılan araçlar. Duygu analizleri, birçok alanda kişilerin memnuniyetlerini, duygularını, düşüncelerini öğrenmek için kullanılır.
- Metin Özetleme: Bir metnin içeriğinin özet halinin oluşturulması için kullanılan araçlar. Metnin anahtar kelimeleri ve önemli bölümleri belirlenerek metnin özeti oluşturulur.

Türkçe için NLP araçlarının kullanımı, Türkçe dilinin yapısındaki zorluklardan dolayı daha zorlayıcı ve zaman alıcı olabiliyor. Türkçe dilinin çekim ekleri, yapım ekleri ve zaman ekleri gibi özellikleri, metinlerin doğru bir şekilde işlenmesi için daha fazla çalışma gerektiriyor. Bu nedenle, Türkçe için NLP araçlarının bu özelliklere ve Türkçe'nin yapısına göre geliştirilmesi ve kullanımı, özellikle Türkçe metinler üzerinde işlem yapan uygulamalar için oldukça önemlidir.

Türkçe için NLP araçlarının kullanımı, genellikle yerelleştirilmiş bir çözüme ihtiyaç duyar. Türkçe dilinde farklı lehçelerin olması farklı araçlar ve yöntemler kullanılmasını gerektirir.

Türkçe doğal dil işleme alanındaki çalışmalara örnek olarak Zemberek ve İTÜ Doğal Dil İşleme Yazılım Zinciri verilebilir. Bu çalışmalardan Zemberek açık kaynak kodlu olup en popüler doğal dil işleme kütüphanesidir. Zemberek ile yazım hataları düzeltilebilir, kelimeler kök ve eklerine ayrılabilir, kelimelerin sözcük türü belirlenebilir. Ancak zemberek bir yazılım kütüphanesi olup kullanıcılara bir arayüz sunmamaktadır.

İTÜ Doğal Dil İşleme Yazılımı, İstanbul Teknik Üniversitesi Doğal Dil İşleme grubu tarafından geliştirilmiştir. Türkçe NLP için bir çevrimiçi arayüz sunmaktadır. Normalizasyon, sözcük/cümle çözümleme, Türkçe karakter dönüştürücü, yazım denetleyici, belirsizlik giderici, varlık ismi tanıma gibi bileşenlerden oluşan bir platformdur. Ancak yazılım altyapısı olmayan araştırmacılar için çevrimiçi bir kullanıcı arayüzü sunmasına rağmen, açık kaynak kodlu değildir. Yazılım geliştiricilere, bu araçları kullanabilmeleri için API hizmeti vermektedir.

BÖLÜM 4: MAKİNE ÖĞRENMESİ

4.1 Makine Öğrenmesi

Makine Öğrenimi (Machine Learning), veri ve örneklerden öğrenebilen ve daha sonra bu bilgiyi kullanarak kendi tahminlerini yapabilen algoritmalar oluşturan yapay zekanın bir alt alanıdır. Amaç insan müdahalesi olmadan bilgisayarların öğrenme işlemini yapmasıdır. Bunun için makineye ne kadar fazla veriseti sağlanırsa öğrenme işleminin başarı oranı o kadar yüksek olur.

Şekil 4: Makine öğrenimi*(ALKAN, 2023)*

Makine öğrenimi, veri analizi, sınıflandırma, regresyon, kümeleme, öneri ve tahmin sistemleri gibi farklı alanları içermektedir. Makine öğrenimi algoritmaları, öğrenme işlemi sırasında girdi verileri kullanarak model oluşturur. Bu modelle veri setlerinden elde edilen bilgiye göre öneri veya karar verme işlemi yapar.

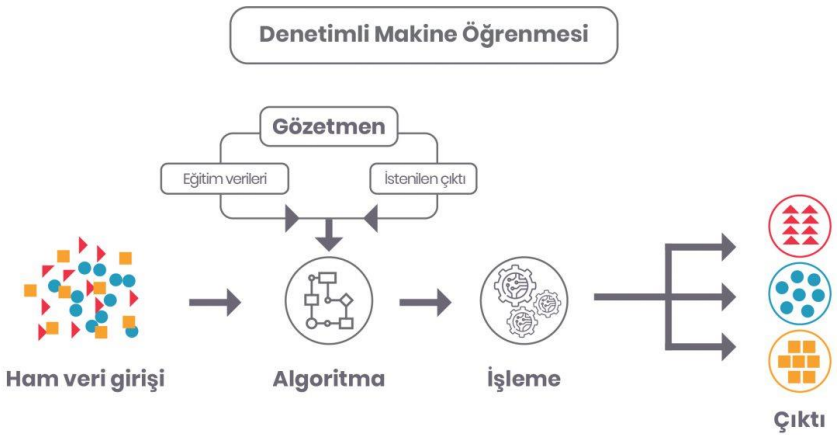
Makine öğrenimi, son zamanlarda çok hızlı bir şekilde gelişmekle birlikte birçok alanda kullanılmaktadır. Örneğin, ses tanıma, görüntü tanıma, sağlık teknolojisi, otomatik sürüş, sosyal medya analizi, bankacılık ve müzik öneri sistemleri gibi alanlarda kullanılmaktadır. Makine öğrenimi, veri analitik, sınıflandırma, regresyon, kümeleme, öneri sistemleri gibi farklı alanları içermektedir.

Makine öğrenimi, veri işleme ve veri analizi teknolojilerinin gelişmesiyle birlikte hızla büyüyen veri setlerinin işlenmesi ve anlamlandırılması için kullanılmaktadır. Örneğin, bir e-ticaret sitesinde milyarlarca satış verisi toplanabilir ve bunlar kullanıcıların davranışlarını ve tercihlerini anlamak için kullanılabilir. Makine öğrenimi, ses, görüntü

veya metin verilerinin analizi için birçok alanda otomatik ve doğru sonuçlar elde etmek için kullanılabilir.

4.2 Makine Öğrenmesi Algoritmaları

4.2.1 Tahmin Edici (Predictive) (Denetimli Algoritmalar)



Şekil 5: Denetimli Mekine Öğrenmesi *(https://blog.turhost.com/makine-ogrenmesi-machine-learning-nedir, 2021)*

Yeni verilerde tahminler oluşturmak için etiketli işlenmiş hazır veri seti kullanılarak sistemin eğitilmesi ile öğrenme sağlanır. Böylece algoritmalar, geçmişte öğrenilenleri yeni verilere uygulayabilir. Denetimli algoritmalarda eğitim verileri ve istenilen çıktı algoritmaya verilir. Öğrenme işlemi esnasında geliştiricinin denetimi, mudahalesi vardır. Algoritmanın sınırlarını kurallarını geliştirici belirler.

4.2.1.1 Regresyon

Regresyon analizi, aralarında neden-sonuç ilişkisi bulunan iki veya daha fazla değişken arasındaki ilişkiyi belirlemek ve bu ilişkiyi kullanarak o konu ile ilgili tahminler yapabilmek, bir bağımlı değişkenin bir veya daha fazla bağımsız değişkenle nasıl değiştiğini açıklamak için kullanılır. Analiz işleminde tek bir değişken kullanılıyorsa bu işleme tek değişkenli regresyon, birden çok değişken kullanılıyorsa çok değişkenli regresyon analizi denir.

Regresyon modellerinde, veri noktaları arasındaki ilişkiyi belirlemek için veri noktalarını en iyi şekilde açıklayan eşleşen bir doğru veya eğri çizmek için matematiksel bir fonksiyon kullanılır. Regresyon modelleri lineer ve non-lineer olmak üzere iki çeşittir.

Lineer regresyon, veri noktaları arasındaki ilişkiyi bir doğru çizgisi ile belirtmek için kullanılır. Veri noktaları arasındaki ilişkinin doğrusal olmaması durumunda non-lineer regresyon yöntemleri kullanılabilir. Non-lineer regresyon, veri noktaları arasındaki ilişkiyi daha karmaşık bir eğri ile belirtmek için kullanılır.

Regresyon analizinde, veri noktaları arasındaki ilişkiyi belirlemek için yaygın olarak kullanılan yöntemler arasında lineer regresyon, lojistik regresyon, polinom regresyon bulunmaktadır. Farklı veri setleri için en uygun olan yöntem seçilerek analiz yapılır.

Regresyon analizi sonucunda elde edilen modelin doğruluğunu değerlendirmek için R-kare, Ortalama Mutlak Hata (MAE) veya Ortalama Kare Hatası (MSE) gibi metrikler kullanılabilir. Bu metrikler, modelin veri setinde ne kadar iyi çalıştığını veya veri setinde ne kadar iyi tahmin yaptığını değerlendirir.

Regresyon analizinde sayısal veriler üzerinde çıkarımlar yapılır. Regresyon analizini denklem olarak düşünürsek bağımsız değişkenlerin bağımlı değişkene etkisini arayan değişkenler arasındaki ilişkiyi bulmak için kullanılan algoritmalardır. Çıktı değişkeni sürekli değerlerden oluşur.

Regresyon algoritmasının uygulama alanları; satış tahmini, deprem tahmini, enflasyon ve döviz kuru ilişkisi, ikinci el otomobil talebinin incelenmesi, hastalık risk faktörlerinin hastalık ile arasındaki ilişki, ekonomide istihdam ve ücret arasındaki ilişki vb. alanlarda da kullanılmaktadır.

4.2.1.2 Classification (Sınıflandırma)

Benzer özellikteki verilerin önceden belirlenmiş kategorilere ayrılması işlemine sınıflandırma denilir. Sürekli olmayan kategorik veriler (evet-hayır/iyi-orta-kötü gibi) üzerinde sınıflandırma yapılır. Sınıflandırma algoritmasında eğitim ve test verisi kullanılır. Eğitim verisi ile öğrenme işlemini gerçekleştirir, sınıfı belli olmayan test

verisini öğrenilen dağılım şekline göre sınıflandırma işlemini yapar. Amaç veriyi sadeleştirmek ve gelen veriler üzerinde tahmin yapmaktır.

Sınıflandırma algoritmalarına uygulama alanları;

- spam maillerin tespiti,
- müşteri yorumlarının analizi,
- hava durumu tahmini,
- cinsiyete göre boy-kilo tahmini,
- Yüz tanıma,
- Ses tanima,
- Hastalik teşhisi,
- Metinsel verileri sınıflandırma,
- El yazısı tanıma vb.

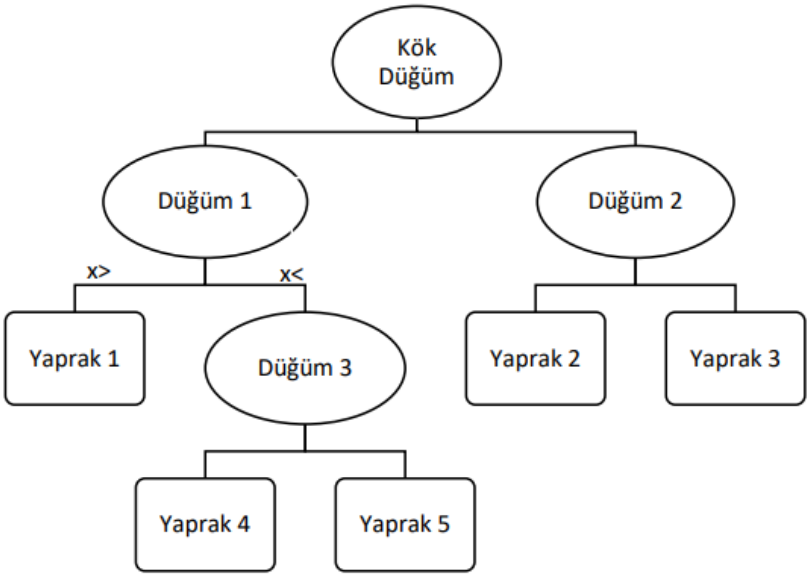
En çok kullanılan denetimli sınıflandırma algoritmaları;

4.2.1.2.1 Karar Ağaçları - KA

KA son zamanlarda yaygın kullanım alanına sahip bir sınıflandırma ve örüntü tanımlama algoritma türüdür. Kullanımın yaygın olmasının en önemli nedeni yorumlanmalarının kolay olması, veri tabanı sistemleri ile kolayca entegre edilebilmeleri ve güvenilirliklerinin iyi olmasıdır (Arslan, 2008:19).

Karar ağaçları, dalları, düğümleri ve yaprakları olan ve bir veri kümesini benzer değerlere sahip örnekleri içeren daha küçük alt kümelere bölen basit modellerdir.

- Kök düğüm: Tüm kararların başladığı ilk düğüm. Üst düğümü yok ve 2 çocuk düğümü vardır. En yüksek karar düğümüdür.
- Karar düğümleri: 1 ana düğümü olan ve alt düğümlere bölünmüş düğümler
- Yaprak düğümler: 1 ebeveyni olan ancak daha fazla bölünmeyen düğümler. Tahmini üreten düğümlerdir. Örnek veriler aynı sınıfa aitse, örnekleri bölecek nitelik kalmamışsa düğüm yaprak olarak sonlandırılır ve sınıf etiketini alır.
- Dallar: Tüm ağacın bir alt bölümü (alt ağaç olarak da tanımlanabilir)
- Maksimum derinlik: üst ve alt uç arasındaki maksimum dal sayısı



Şekil 6: Karar Ağacı Yapısı *(Topuz, 2021)*

Karar ağacında ağaç yapısını oluşturmak için 6 aşama bulunur (Erbudak, 2022:11-12)

1. Problemin tanımlanması,
2. Karar ağacının yapısının kurulması,
3. İşlemlerin yapılabilme olasılıklarının atanması,
4. Beklenen faydanın ilgili doğruluk noktası için hesaplanması,
5. En yüksek yararla ilgili karar noktasına taşınması,
6. Tahminin belirtilmesi

Karar ağacı temelli analizlerin yaygın olarak kullanıldığı sahalar;

- Belirli bir sınıfın muhtemel üyesi olacak elemanların belirlenmesi (Segmentation),
- Çeşitli vakaların yüksek, orta, düşük risk grupları gibi çeşitli kategorilere ayrılması (Stratification),
- Gelecekteki olayların tahmin edilebilmesi için kurallar oluşturulması,
- Parametrik modellerin kurulmasında kullanılmak üzere çok miktardaki değişken ve veri kümesinden faydalı olacakların seçilmesi,
- Sadece belirli alt gruplara özgü olan ilişkilerin tanımlanması,
- Kategorilerin birleştirilmesi ve sürekli değişkenlerin kesikliye dönüştürülmesidir.

Karar ağacı temelli tipik uygulamalar ise;

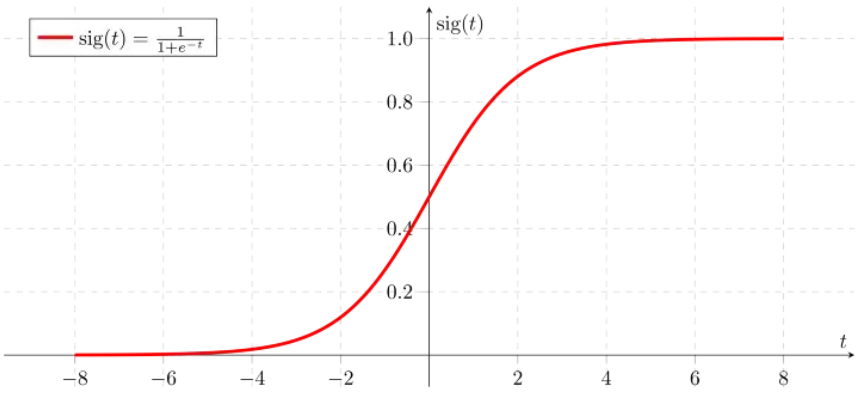
- Hangi demografik grupların mektupla yapılan pazarlama uygulamalarında yüksek cevaplama oranına sahip olduğunun belirlenmesi,
- Bireylerin kredi geçmişlerini kullanarak kredi kararlarının verilmesi,
- Geçmişte İşletmeye en faydalı olan bireylerin özelliklerini kullanarak işe alma süreçlerinin belirlenmesi,
- Tıbbî gözlem verilerinden yararlanarak en etkin kararların verilmesi,
- Hangi değişkenlerin satışları etkilediğinin belirlenmesi,
- Üretim verilerini inceleyerek ürün hatalarına yol açan değişkenlerin belirlenmesidir (Ekici, 2012:9).

Karar Ağacı Algoritma Türleri

- ID3 Algoritması
- C4.5 Algoritması
- C5.0 Algoritması
- CART Algoritması (Sınıflandırma ve Regresyon Ağacı)
- Ki-kare Algoritması (Ki-kare otomatik etkileşim tespiti)
- MARS Algoritması (Çok Değişkenli Uyarlanabilir Regresyon Çizgileri)
- CHAID Algoritması (Otomatik Ki-Kare Etkileşim Belirleme Analizi)
- QUEST Algoritması (Hızlı, Yansız, Etkili İstatistiksel Ağaç)

4.2.1.2.2 Lojistik Regresyon

Lojistik Regresyon Analizi bağımlı değişkenin tahmini değerlerini olasılık olarak hesaplayarak, olasılık kurallarına uygun sınıflama yapma imkanı veren bir yöntemdir.
Lojistik Regresyon, bir değişkenin sınıflandırılmasına yönelik bir tahmin yapma yöntemidir. Bu yöntem, binary (ikili) sınıflandırma problemlerinde yaygın olarak kullanılır.
Lojistik Regresyon sınıflandırma işlemi için Sigmoid Fonksiyonunu kullanır. Bu fonksiyonla, veriler 0-1 arasına yerleştirilir.



Şekil 7: Sigmoid fonksiyonu *(Akça, 2021)*

Resgresyon analizinde dikkat edilmesi gereken durumlar;

- Uygun tüm bağımsız değişkenler modele dahil edilmelidir.
- Uygun olmayan tüm bağımsız değişkenler dışlanmalıdır.
- Aynı birey üzerinde bir kez gözlem yapılmalı, tekrarlayan ölçümler olmamalıdır.
- Ölçüm hataları küçük olmalı, kayıp (eksik) veri olmamalıdır. Hatalar, katsayıların tahmininde yanlılığa ve modelin yetersizliğine neden olur.
- Bağımsız değişkenler arasında çoklu bağlantı (Multicollinearity) olmamalıdır.
- Aşırı değerler olmamalıdır.
- Örneklem büyüklüğü yeterli olmalıdır.
- Bağımlı değişkenin beklenen ve gözlenen varyansı arasında büyük bir fark varsa model yetersizdir ve yeniden tanımlanması gerekir.

Avantajları:

- Uygulanması ve yorumlanması basittir,
- Katı varsayımlar olmadığından bu durum uygulamada esneklik sağlar,
- Veri seti doğrusal olarak ayrılabiliyorsa iyi performans gösterir,
- Overfitting'e daha az meyillidir ama büyük veri kümelerinde overfit olabilir.

Dezavantajları:

- Gözlem sayısı özellik sayısından azsa, Lojistik Regresyon kullanılmamalıdır, aksi takdirde overfit olabilir.
- Lojistik regresyonun ayrım yapabilmesi için veri setinin doğrusal olarak ayrılabiliyor olması lazım.

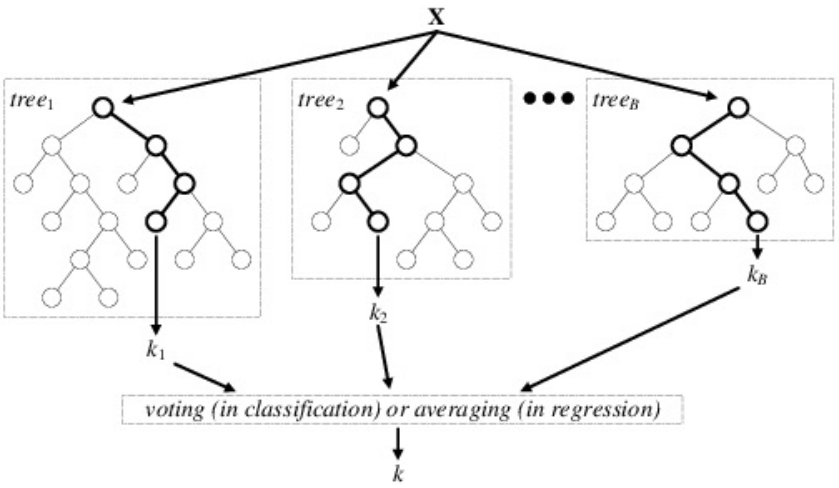
Lojistik regresyon kullanım alanları;

- Sağlık uygulamalarında kullanılır. Örneğin hastalıkların tanısı, kalp hastalıklarının risk analizi, kanser tanısı gibi uygulamalarda kullanılabilir.
- Bankacılık ve finans alanında kullanılabilir. Örneğin piyasa analizlerinde, kredi risk analizi, müşteri tercihleri, müşteri borçları, gibi konular için kullanılabilir.
- Pazarlama alanında kullanılabilir. Örneğin ürün tercihleri, müşteri satın alma davranışları, reklam tercihleri, kampanya gibi konular için kullanılabilir.
- İnsan kaynakları yönetiminde kullanılabilir. Personel tercihleri, pozisyon uygunluğu, eğitim ihtiyacı gibi konular için kullanılabilir.
- Sosyal bilimler alanında kullanılabilir. Örneğin kişilik özelliklerinin sınıflandırılması, iş başarısı, okul başarısı gibi konular için kullanılabilir.

4.2.1.2.3 Rasgele Orman

Rasgele Orman Algoritması, birden fazla karar ağacı kullanarak, sınıflandırma veya regresyon problemlerini çözmek için kullanılır. Her bir karar ağacı, rasgele seçilmiş özellikler kullanarak eğitilir ve sonuçların ortalaması alınarak bir tahmin yapılır. Bu yöntem, her bir karar ağacının daha az özellikle eğitilmiş olmasını ve dolayısıyla overfitting (aynı veriye çok iyi uyan ama farklı verilerde zayıf sonuç veren) riskinin azaltılmasını sağlar.

Rasgele Orman Algoritması, çoğunlukla veri setlerinde yüksek boyutlu veya karmaşık veriler için kullanılır. Ayrıca, sınıflandırma veya regresyon problemlerinde performansı, diğer popüler algoritmalarla karşılaştırıldığında yüksektir.



Şekil 8: Rasgele Orman Algoritması *(Yıldırım, 2018)*

Rasgele Orman Algoritmasının parametreleri arasında, karar ağaçlarının sayısı, alt veri kümelerinin boyutu ve öznitelik seçiminde kullanılan yöntemler yer alır. Bu parametreler, algoritmanın performansını etkileyebilir ve optimal performans için uygun olarak ayarlanması gerekir.

Rasgele Orman Algoritmasının avantajları:

- Her bir karar ağacı farklı veri noktaları ve öznitelikler kullanarak eğitilir, bu da aşırı öğrenme (overfitting) riskinin azaltılmasını sağlar.
- Öznitelik seçiminde daha iyi performans gösterir ve veri setindeki anlamlı öznitelikleri belirler.
- Çoklu sınıflandırma problemlerini birden fazla karar ağacı kullanarak çözebilir.
- Gürültü ve ayrık değerleri filtreleme yeteneği vardır.
- Yüksek boyutlu veya karmaşık veriler için uygundur

Rasgele Orman Algoritmasının dezavantajları:

- Rasgele Orman Algoritması, birden fazla karar ağacının eğitilmesi için büyük miktarda bellek ve işlem gücü gerektirebilir. Bu nedenle maliyeti yüksek olabilir.
- Algoritma tahminleri anlamsız olabilir.
- Özniteliklerin sayısı, örneklerin sayısından daha büyükse algoritma aşırı uyum yapabilir.
- Özniteliklerin önem derecesi hakkında bilgi sağlamaz

Rasgele Orman Algoritması kullanım alanları;

Sınıflandırma: Riskli / risksiz, sağlıklı / hasta gibi veri setindeki örnekleri belirli sınıflara sınıflandırmak için kullanılabilir.

Regresyon: Veri setindeki örnekler arasındaki ilişkileri tahmin etmek için kullanılabilir. Örneğin, maaş-yaş, satış-reklam bütçesi gibi.

Öznitelik seçimi: Veri setindeki öznitelikler arasındaki ilişkileri ve özniteliklerin önem derecelerini belirlemede kullanılabilir.

Gürültü filtreleme: Veri setindeki gürültü filtrelemek için kullanılabilir.

Anomali Tespiti: Veri setinde anormal olarak kabul edilen beklenmedik verileri bulmada kullanılabilir.

Bağımlılık analizi: Veri setindeki öznitelikler arasındaki bağımlılıkları belirlemede kullanılabilir.

Yüksek boyutlu veriler için veri madenciliği: Yüksek boyutlu verilerde veri madenciliği yapmak için kullanılabilir.

Görüntü tanıma: Nesnelerin tanımlanması ve sınıflandırılması, yüz tanıma gibi görüntüleri sınıflandırmak için kullanılabilir.

Ses tanıma: Konuşma tanıma, müzik tanıma gibi ses sinyallerini tanımak ve sınıflandırmak için kullanılabilir.

Finansal piyasalar: Hisse senedi fiyatlarının tahmini, kredi riski değerlendirmesi gibi finansal piyasalarda tahmin yapmak için kullanılabilir.

Sağlık: Hastalık belirtilerinin tanımlanması, tedavi seçeneklerinin tahmini gibi sağlık verileri analizi yapmak için kullanılabilir.

Ekolojik veriler: Ekolojik veriler analizi yapmak için kullanılabilir. Örneğin, bitki ve hayvan popülasyonlarının tahmini, ekosistemlerin durumunun değerlendirmesi gibi.

Makine öğrenimi: Makine öğrenimi projelerinde model seçiminde ve öznitelik seçiminde kullanılabilir.

Veri madenciliği: Veri madenciliği projelerinde öznitelik önem derecelerini belirlemek, sınıflandırma ve regresyon işlemleri yapmak için kullanılabilir.

NLP: Doğal Dil İşleme (NLP) projelerinde metin sınıflandırması, metin regresyonu ve öznitelik seçimi için kullanılabilir.

4.2.1.2.4 Naive Bayes Sınıflandırması

1812 yılında İngiliz matematikçi Thomas Bayes tarafından bulunan koşullu olasılık hesaplama formülüdür.

Bayes sınıflandırma algoritmasında çıktıyı tahmin etmek için çok sayıda nitelikten faydalanır. Diğer algoritmalarda etkisiz olarak görülerek göz ardı edilebilen nitelikler naïve bayes için küçük etkilere sahip olsa da tümüyle kullanıldığında etkileri büyük olabilir. Örnek veri setine göre tahmin edilmesi istenen durumun hangi sonucu vermeye yakın olduğunu bulan ve bu sonucu yüzdelik olarak ifade eden algoritmadır. Hesaplamada en yüksek olasılığa sahip sonucu seçmeyi hedefler ve olasılık değeri en yüksek olana göre sınıflar. Her bir özellik birbirinden bağımsız kabul edilir.

Tablo 1

Örnek Veri Kümesi

No	Hava	Sıcaklık	Rüzgar	Sonuç
1	Güneşli	Yüksek	Düşük	Hayır
2	Güneşli	Orta	Orta	Evet
3	Bulutlu	Yüksek	Orta	Evet
4	Bulutlu	Orta	Yüksek	Evet
5	Yağmurlu	Yüksek	Düşük	Hayır
6	Yağmurlu	Yüksek	Orta	Hayır
7	Güneşli	Düşük	Yüksek	Hayır

Tablo 1'de verilen örneği inceleyecek olursak;

Hava, sıcaklık, rüzgar, sonuç etkisine göre veri setimizi düşündüğümüzde Bayes algoritması bize aşağıda belirtilen durumların olasıklarını tahmin etmemizi sağlar;

HAVA'nın bulutlu olma durumunda, dışarı çıkma olasılığımız nedir?

Yani verilen bir input'un daha önce tanımlanmış veri setinde sonuca ulaşma oranı nedir?

Bu algoritmayı kullanabilmek için mutlaka sınıflandırılabilir, kategorilize edilebilir bir sonucunuzun olması gerekir. Burada çıktımız 2 kategoriden oluşmaktadır. EVET / HAYIR.

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

Şekil: Bayes sınıflandırma formülü

P(A): A olayının gerçekleşme olasılığı
P(B): B olayının gerçekleşme olasılığı
P(B|A): A olayı bilindiğinde B olayının gerçekleşme olasılığı
P(A|B): B olayı bilindiğinde A olayının gerçekleşme olasılığı

Naive Bayes Sınıflandırıcısının Avantajları

Her özellik birbirinden bağımsız kabul edildiği için performans yüksek.
Az veriyle başarılı çıktılar üretilir
Sürekli ve kesikli veriler ile kullanılabilir.
Yüksek boyutlu verilerde iyi çalışabilir.
Uygulaması hızlı ve kolay

Naive Bayes Sınıflandırıcısının Dezavantajları

Özellikler birbirinden bağımsız olması gerekir bu yüzden değişkenler arası ilişkiler modellenemez. Bu durum performansı engeller.

Naive Bayes Türleri

- Gaussian Naive Bayes: Özelliklerimiz sürekli verilerden oluşuyorsa ve ayrık değillerse bu verilerin bir Gauss dağılımından geldiği varsayılır.
- Multinomial Naive Bayes: Çok sınıflı kategorileri sınıflandırmak için kullanılır. Genellikle belge sınıflandırma (sağlık, spor, teknoloji vb.) problemleri için kullanılır. Bu işlemi belgede bulunan sözcüklerin sıklığına bakarak yapar.
- Bernoulli Naive Bayes: Tahminler sadece ikili değerlerden (iyi-kötü, var-yok gibi) oluşmaktadır.

Naive Bayes uygulama alanları;

- Spam mail filtreleme,
- Duygu analizi,

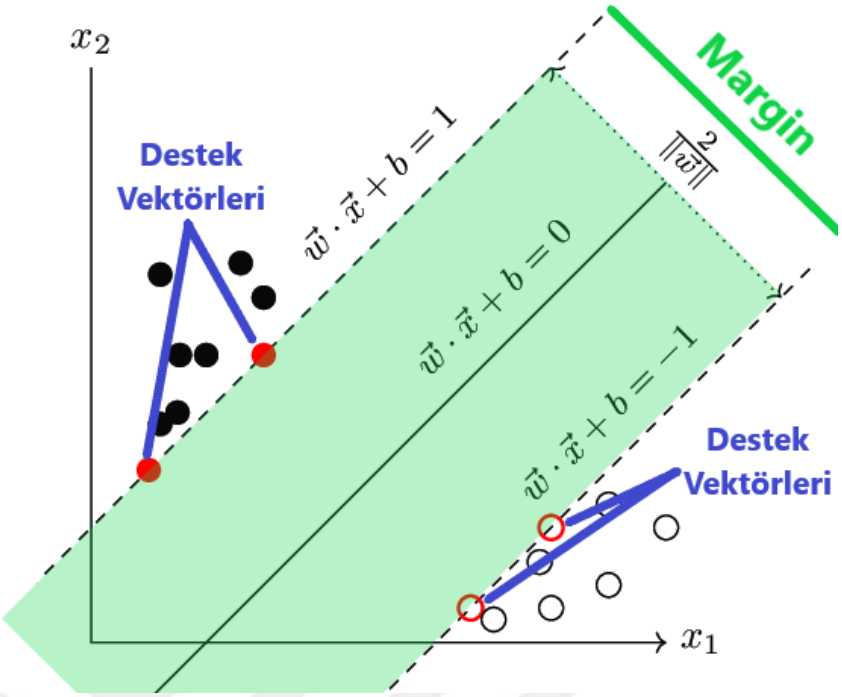
- Belge sınıflandırma,
- Özet çıkarma
- Öneri sistemleri
- Hastalık teşhisi

4.2.1.2.5 Destek Vektör Makinesi - DVM

Vapnik Chervonenkis tarafından geliştirilen DVM iki ayrı sınıftaki verileri birbirinden ayırmak için tasarlanmış istatistiksel öğrenme algoritmalarından biridir. DVM lineer olmayan örnek uzayını, örneklerin lineer olarak ayrılabileceği bir yüksek boyuta aktararak, farklı örnekler arasındaki maksimum sınırın bulunması esasına dayanır (Demirci, 2007:3).

DVM, regresyon ve sınıflandırma analizi için kullanılan uygulaması kolay ve oldukça etkili makine öğrenimi modelidir. Kategorisi bilinmeyen verileri özelliklerine göre veya eğitim verisinden öğrenilen örneklere göre ayrıştırma işlemini yapar. DVM ile çoğunlukla iki sınıflı modeller üzerinde çalışmalar yapılmaktadır. Ancak çok sınıflı sınıflandırma modellerine de destek vermektedir. Küçük ya da orta büyüklükteki veri setleri için uygun bir algoritmadır. Büyük verilerin eğitim süresi uzun olacağıdan pek tercih edilmez. Doğrusal olarak ayrılabilen veri kümelerinde olduğu gibi doğrusal olmayan veri kümelerinin ayrıştırılması için bu algoritma kullanılmaktadır.

DVM örnek uygulama alanları; Spam mail tespiti, yüz ve el yazısı tanıma vb.



Şekil 9: Destek Vektör Algoritması*(Akça, 2020)*

DVM, aynı düzlem üzerinde bulunan noktaları birbirinden ayırmak için iki ayrı sınıfa ait noktadalardan en uzak mesafede yer alacak şekilde bir doğru çizer. Şekilde siyah nokatalar bir sınıfı beyaz noktalar ise diğer bir sınıfı ifade eder.

- wx + b > 0 (1 sınıfındaki veri noktaları için)
- wx + b < 0 (0 sınıfındaki veri noktaları için)

Formülde x bir girdiyi, w düzleme dik olan normal vektörü, b ise kayma miktarıdır.

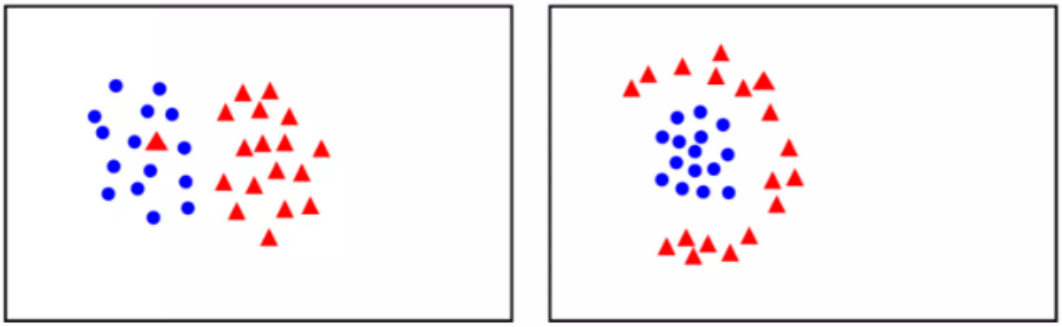
Destek Vektörleri (Support Vector): Bir sınıfa ait noktaların hiper düzeleme en yakın olan veri grubu noktaları olarak ifade edilebilir. Bu noktalar hiper düzlemle birbirinden ayrılmış sınıfların birbirine olan sınırlarını ifade eder.

Hiper Düzlem (Hyper Plane): İki sınıfa sahip bir veri kümesini doğrusal olarak ayıran ve sınıflandıran çizgidir. Veri noktalarının hiper düzleme olan mesafeleri ne kadar fazla ise sınıflandırmamızın başarı oranı artar.

Marjin (Margin): Destek vektörleriyle sınırlanan ve bu sayede Hiper Düzlemi belirlememize yardımcı olan alandır.

Genel olarak Destek Vektör Makinaları (DVM) ikiye ayrılır;

Doğrusal Olmayan Destek Vektör Makinaları

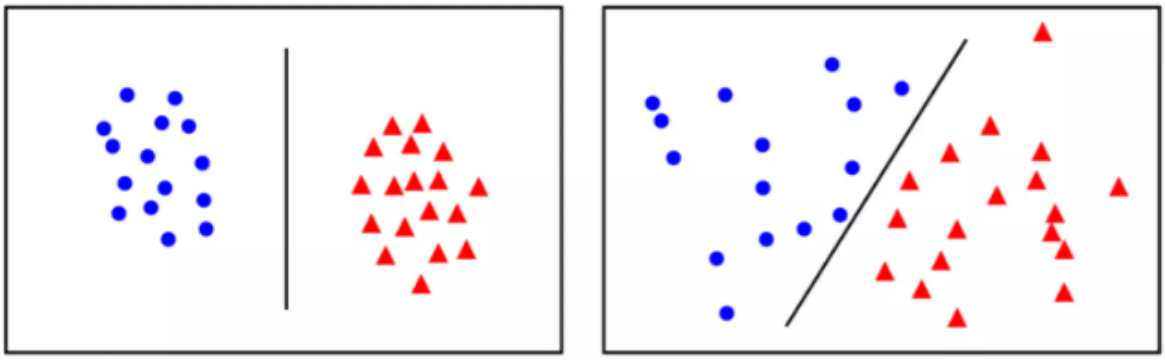


Şekil 10: Doğrusal Olmayan Destek Vektör Kümesi*(Taş, 2016)*

İki sınıfa ait noktaların iç içe geçmiş şekilde yer alması tamamen ayrıştırılamadığı durumudur. Ayrıştırma işlemi için veriler çok boyutlu uzaya taşınarak veri setinin doğrusal ayrımı sağlanır.

Doğrusal Destek Vektör Makinaları

Veri kümelerinin bir doğruyla ayrıştırılabildiği durumdur.



Şekil 11: Doğrusal Destek Vektör Kümesi *(Taş, 2016)*

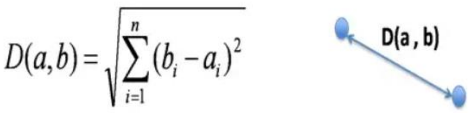
4.2.1.2.6 K En Yakın Komşu -KNN

KNN, sınıflandırılacak örnek veri noktasının bulunduğu sınıfın ve en yakın komşunun, k değerine göre belirlendiği bir sınıflandırma yöntemidir.

Her bir sınıfın özelliklerinin önceden belirlenmiş olması, veri kümesinin büyük olması ve belirlenen k değerinin uygun seçilmesi çok önemlidir.

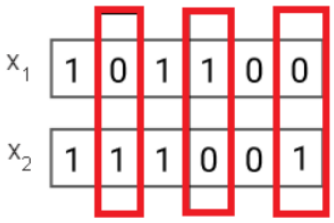
KNN algoritması çıktıyı tahmin etmek için gözlemler arasındaki mesafeyi hesaplar. Uzaklıkları hesaplamak için kullanılan başlıca uzaklık metrikleri;

- **Öklit Uzaklığı**: Öklit, iki nokta arasındaki doğrusal uzaklıktır. Algoritmalarda yakınlığın ölçümlenmesi için en fazla kullanılan uzaklık ölçütüdür.



Şekil 12: Öklit Formülü *(Çetinyamaç, 2022)*

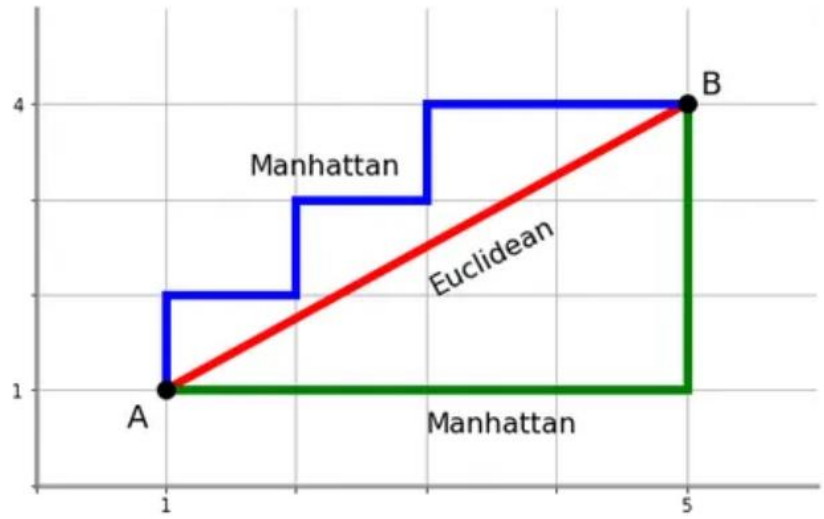
- **Hamming Uzaklığı:** İki binary veri dizisini karşılaştırmak için bir uzaklık ölçütüdür. Eşit uzunluktaki iki binary diziyi karşılaştırırken Hamming mesafesi, iki bitin farklı olduğu bit konumlarının sayısıdır.



Şekil 13: Hamming Uzaklığı

Hamming mesafesi, d(101100, 111001) = 010101 = 3

- **Manhattan Uzaklığı:** İki nokta arasındaki uzaklık, bu iki noktadan geçen ve eksenler boyunca dik kesişen doğru parçalarının ölçülen uzaklık hesaplaması olarak ifade edilmektedir.



Şekil 14: Manhattan mesafesi ve noktalar arasındaki Öklid mesafesi*(Wet, 2022)*

Şekil 14'te belirtildiği gibi bir A(1,1) ve B(5, 4) noktaları arasındaki mesafeyi manhattan ile iki ayrı yolla bulabiliriz. Ancak manhatten mesafesi şekilden de anlaşılacağı üzere daha uzundur.

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|$$

Şekil 15: Manhattan Formülü

- **Minkowski Uzaklığı:** Öklid ve Manhattan mesafesinin genelleştirilmiş bir metrik formudur.

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

Şekil 16: Minkowski Formülü*(Çetinyamaç, 2022)*

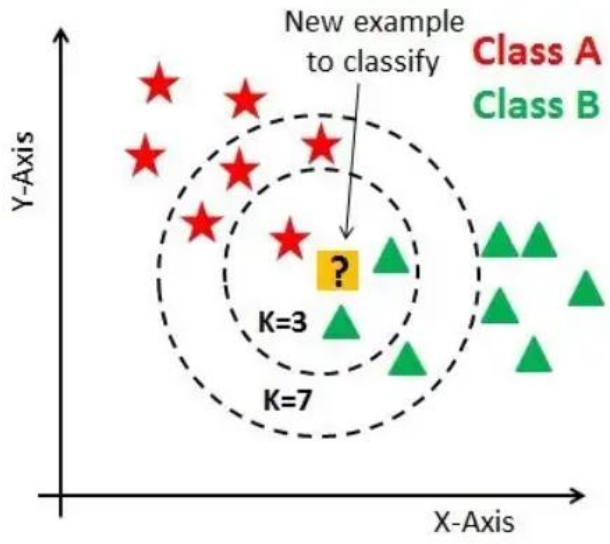
KNN algoritmasının avantajları;

- Uygulaması ve yorumlaması kolay,
- Doğrusal olmayan veri kümeleri için kullanışlı,
- Oldukça yüksek doğruluğa sahip,
- Gürültülü eğitim verilerine karşı dirençli,

- Eğitim çok hızlı
- Büyük eğitim verilerinde etkili

KNN algoritması dezavantajları

- Yüksek hesaplama maliyeti
- Büyük veride yüksek bellek harcaması
- Değişken sayısı fazla olması durumunda iyi sınıflandırma yapılamaz
- Hızlı olmasına rağmen tembel bir algoritmadır
- Parametrelere (komşu sayısı, uzaklık ölçütü vb.) ve ilgisiz niteliklere duyarlı



Şekil 17: KNN sınıflandırma örneği*(Çetinyamaç, 2022)*

KNN algortitması iki değer üzerinden hesaplama yapar;

K (Komşuluk Sayısı): Hesaplama işleminden önce en çok kaç komşuya göre hesaplama yapılacağı belirtilir.

Uzaklık: Tahmin edilecek noktanın diğer noktalara mesafesi hesaplanır.

KNN algoritması aşamaları;

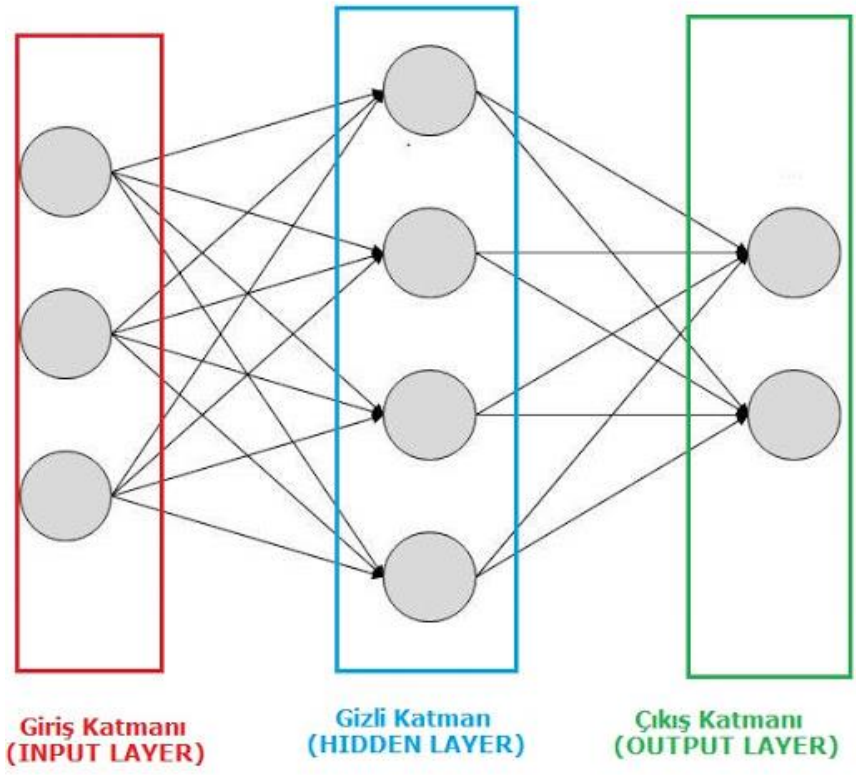
- İlk olarak k değeri belirlenir
- Örnek veri setine eklenecek gözlem değerinin mevcut verilere olan öklit uzaklıkları hesaplanır.

- Hesaplanan uzaklıklara göre sıralama işlemi yapılır, komşular arasında en yakın k tane komşu bulunur.
- Seçilen en yakın komşuların hangi sınıfta oldukları saptanır,
- En uygun komşu sınıfı seçilerek gözlem değeri o sınıfa atanır.

4.2.1.2.7 Yapay Sinir Ağları -YSA

Yapay sinir ağları (YSA), insan beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri, herhangi bir yardım almadan otomatik olarak gerçekleştirebilmek amacı ile geliştirilen bilgisayar sistemleridir (Yıldırım, 2020).

YSA, karar ağaçları gibi son zamanlarda yaygın olarak kullanılan veri madenciliği tekniklerindendir.



Şekil 18: YSA Katmanları*(Yıldırım, 2020)*

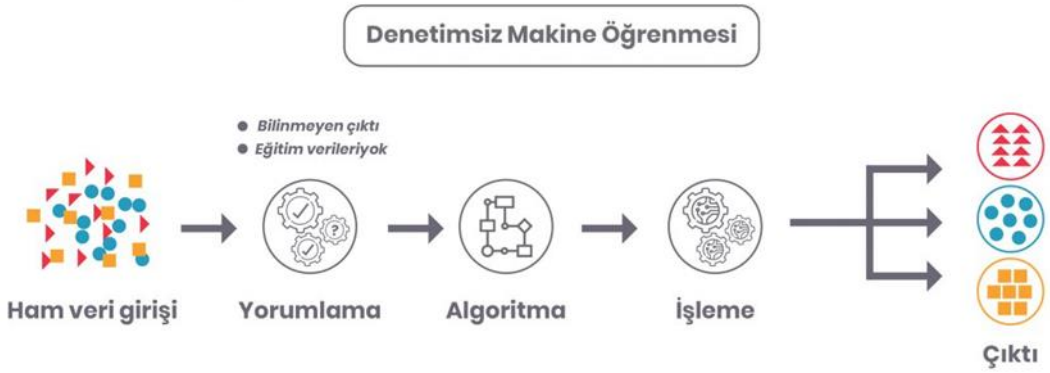
YSA insan beyni örnek alınarak geliştirilmiş bir modeldir. Beyindeki biyolojik sinir ağlarının işleyişine benzer şekilde geliştirilmiştir. Karmaşık problemler YSA ile kolaylıkla çözülebilir.

YSA üç ayrı katmandan oluşur; Giriş Katmanı, Gizli Katman, Çıkış Katmanı
Yapay sinir ağları, kaynak (girdi), çıktı ve iç (gizli) düğümlerle yönetilen bir grafik olarak görülebilir. Girdi düğümü giriş katmanında, çıktı düğümü ise çıkış katmanında bulunur. Gizli düğümler gizli katmanda bulunur. KA yorumlama ve uygulama yönünden kolay olmakla birlikte YSA ise zor ve daha karmaşıktır.

Yapay sinir ağları aşağıdaki temel özelliklere sahiptir (Öztürk ve Şahin, 2018:29):

- Doğrusal Olmama
- Paralel Çalışma
- Öğrenme kabiliyeti
- Genelleme
- Hata Toleransı ve Esneklik
- Eksik Verilerle Çalışma
- Çok Sayıda Değişken ve Parametre Kullanma
- Uyarlanabilirlik

4.2.2 Tanımlayıcı (Descriptive) (Denetimsiz Algoritmalar)



Şekil 19: Denetimsiz Makine Öğrenmesi*(https://blog.turhost.com/makine-ogrenmesi-machine-learning-nedir, 2021)*

Denetimsiz Algoritmalar: Etiketlenmemiş verilerle ilgilenir. Veri seti etiketlenmediği için belirli bir çıktısı da yoktur. Öğrenme işlemi belirli bir veri setine ya da kurala göre yapılamamaktadır. Bu yüzden denetimsiz algoritma olarak adlandırılmıştır. Genellikle öznitelik çıkarımı, kümeleme, benzerlik analizleri için kullanılır.

Denetimsiz algoritmalar, veri setinin sınıflandırılmadığı durumlarda kullanılabilir. Örneğin, müşterilerin satın alma durumlarını analiz etmek istenirse, sınıflandırma etiketleri olmadan yani sınıflandırılmadan, müşteriler arasındaki benzerlikleri veya farklılıkları bulmak için kullanılabilecek algoritmalara örnektir.

4.2.2.1 Clustering (Kümeleme)

Dağınık halde bulunan verilerin benzerliklerine veya davranışlarına göre gruplandırılması işlemidir. Sınıflandırma işleminden farkı kümelerin belirli olmamasıdır. Eğiticisiz öğrenme metotlarıdır. Verileri en iyi temsil edecek vektörleri bularak tüm verileri bulunan bu vektörlerle kodlar. Çoğu sıkıştırma algortimalarının temelinde kümeleme algoritmaları kullanılır. Kümeleme yönteminde kullanılan algoritma K-Means algoritmasıdır.

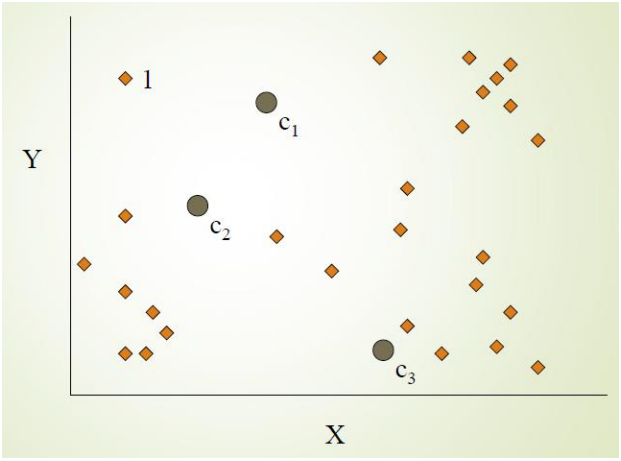
K-Means Algoritması: Popüler ve hızlı bir algoritma olan K-Means, veri noktalarını benzerliklere göre gruplandırır. Kümeleme problemini çözen en basit algoritmadır. Örneğin belirli bir ürün üzerinden müşterilere beğenileri ve alışverişleri doğrultusunda indirimleri hızlı bir şekilde sunabilir. K-means algorimasında amaç, gruplama işlemi sonunda elde edilen kümelerin, küme içi benzerlik oranlarının maksimum, kümeler arası ise minimum düzeyde olmasını sağlamaktır. Sadece sayısal değerlerle çalışır.

K-Means Algoritması Adımları

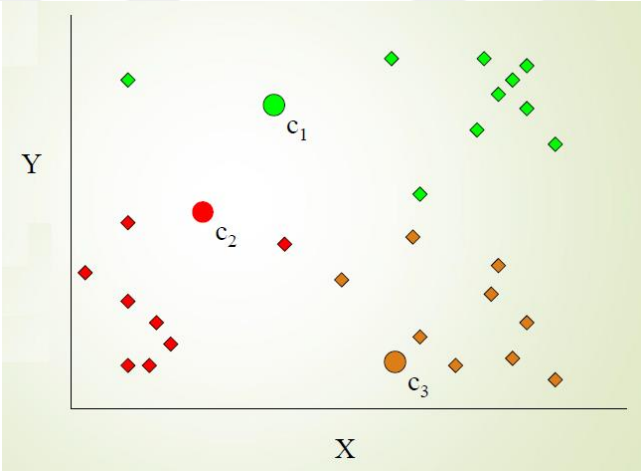
Bilinen bir k değeri için k-means demetleme algoritmasının aşamaları aşağıdaki gibidir;

- Veri kümeleri k gruplarına ayrılır.
- Küme merkezi olarak rastlantısal k noktaları seçilir.
- Öklid mesafe işlevine göre her nesne en yakın Merkez noktanın olduğu kümeye atanır.
- Tüm nesnelerin ortalaması hesaplanır.
- Her kümeye aynı noktalar ardışık olarak atanana kadar adım 2'ye geri dönülür işlemler tekrarlanır.

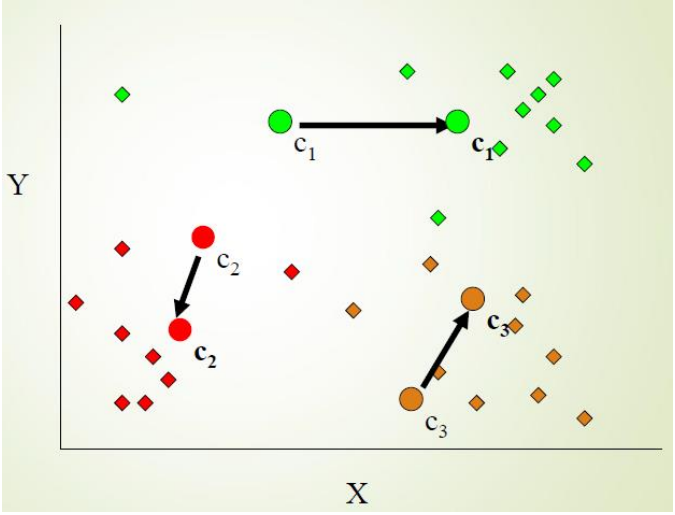
Örnek üzerinden anlatmak gerekirse;



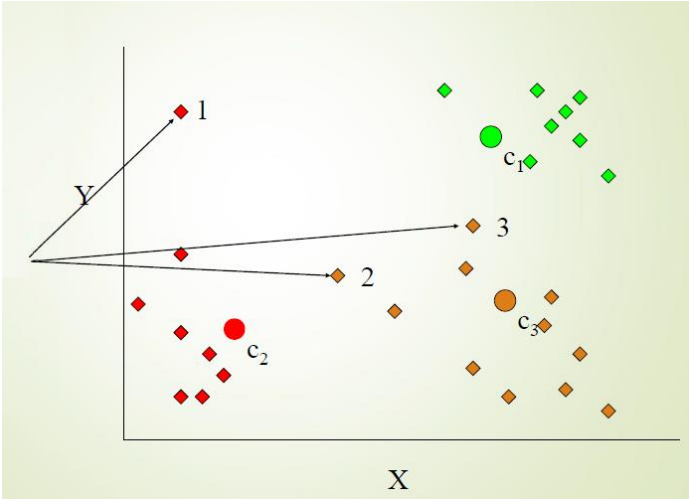
Rastgele c1, c2, c3 şeklinde küme merkezleri atanır.



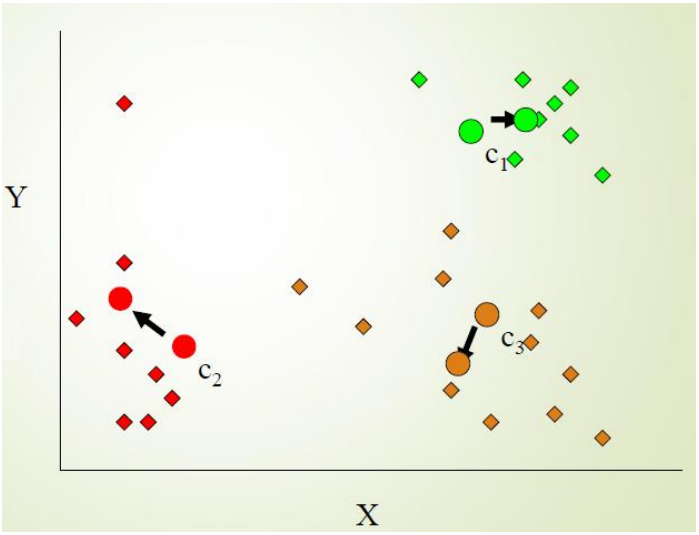
Her örnek en yakınındaki Merkez kümesine atanır.



Başlangıçta belirtilen merkezler kendi kümelerinin merkezine alınır.



Her örnek en yakınındaki merkezin kümesine atanır.



Merkezler yeniden kendi kümelerinin merkezine alınarak kümeleme işlemi tamamlanır

4.2.2.1 Birliktelik Kuralı

Veri kümeleri arasındaki birliktelik ilişkilerini ortaya çıkaran veri madenciliği yöntemlerinden biridir. Yani, bir öğenin görülmesi durumunda diğer öğenin de görülme olasılığını ölçer. Özellikle büyük veri kümelerinde birliktelik kurallarının belirlenmesi gerekir.

Örneğin, bir mağazada satılan ürünler arasındaki ilişkileri ortaya çıkarmak için kullanılabilir, eğer bir müşteri çocuk bezi aldıysa, o müşteri aynı zamanda mama da alma olasılığı yüksektir" gibi.

Birliktelik kuralı, veri setinde yer alan öğeler arasındaki ilişkileri belirlemek için kullanılan Apriori algoritması veya FP-growth algoritması gibi algoritmalar kullanılarak uygulanır. Apriori algoritması, veri setinde yer alan tüm öğeleri tek tek incelemeyi ve öğeler arasındaki ilişkileri belirlemek için kullanılır. FP-growth algoritması ise, öğeler arasındaki ilişkileri hızlı bir şekilde belirlemek için kullanılır.

Birliktelik kuralı, öznitelikler arasındaki ilişkileri veya zaman serilerini analiz etmek için de kullanılabilir.

Birliktelik kuralı, veri madenciliği, sağlık, pazarlama, finans, bankacılık vb. alanlarda kullanılabilir. Bir bankada müşteri harcama durumu analiz edilerek kredi riskini azaltmak, bir mağazada satılan ürünler arasındaki ilişkileri çıkararak satış stratejileri oluşturmak veya sağlık verilerini analiz ederek hastalığı tahmin etmek gibi durumlar örnek olarak verilebilir.

Birliktelik kuralının avantajları:

- Basit ve anlaşılırdır
- Hızlı ve etkili sonuçlar alınır
- Çoklu uygulama alanları vardır
- Öznitelikler arasındaki ilişkileri ortaya çıkarır

Birliktelik kuralının dezavantajları:

- Birliktelik kurallarının uygulanması, veri kümelerinin çok büyük olması durumunda zordur ve zaman alabilir.
- Gürültülü verilerden etkilenebilir ve yanıltıcı sonuçlar verir.
- Birliktelik kuralları, veriler arasındaki bağımlılığı açıklamaz. Bunun sonucu, öğeler arasındaki bağımlılığın nedenini anlamak için ek araştırma gerekebilir.

4.3 Makine Öğrenmesi Avantaj ve Dezavantajları

- Makine öğrenimi algoritmaları, veri setlerinden elde edilen bilgiye göre bir model oluşturur. Veri setlerinden elde edilen bilgiye dayanarak modele tahmin veya karar verme işlemi yapar. Böylelikle, veri setlerinin işlenmesi ve anlamlandırılması için insan gücüne ihtiyaç kalmaz.

- Makine öğrenimi algoritmaları, veri setlerinin büyüklüğüne veya karmaşıklığına bağlı olarak işlem yapabilir.
- Makine öğrenimi algoritmaları kullanılarak ses ve görüntü verilerinin analizi yapılabilir.
- Makine öğrenimi algoritmaları, kullanıcıların tercihlerine ve davranışlarına göre öneri sistemleri oluşturabilir.
- Makine öğrenimi algoritmaları kullanarak, yapılan analizler daha anlamlı ve doğru sonuçlar elde edilir.
- Makine öğrenimi algoritmaları, gerçek zamanlı verileri işleyerek veri setlerini sürekli olarak güncelleştirir.
- Makine öğrenimi algoritmaları, veri setlerinin büyüklüğüne veya karmaşıklığına göre işlem yapabilir.
- Makine öğrenimi algoritmaları ile verilerin sınıflandırılması ve tahmin edilmesi mümkündür. Örneğin, müşteri satın alma davranışlarının analizi, sağlık verilerinin analizi gibi alanlarda önemlidir.

4.4 Makine Öğrenimi Araçları

- Python ve R programlama dilleri
- NumPy, Pandas, Matplotlib gibi veri analitik kütüphaneleri
- TensorFlow, PyTorch, scikit-learn gibi makine öğrenme kütüphaneleri
- Jupyter Notebook gibi düzenli ve interaktif programlama ortamları
- Weka, RapidMiner, KNIME gibi veri madenciliği ve makine öğrenme platformları bulunur.

BÖLÜM 5: DERİN ÖĞRENME

5.1 Derin Öğrenme

Derin öğrenme makine öğrenmesinin önemli bir alanıdır. Ağırlıklı olarak yapay sinir ağları kullanır. Son zamanlarda, derin öğrenme algoritmaları kullanılarak birçok çalışmalar yapılamakta, elde edilen veri setleri ile tahminlerde bulunularak en doğru tahminleme algoritmalar kullanılmaktadır. Derin öğrenme, öznitelikleri verilen veriler üzerinden makine öğrenimi yapılmakta, bu öğrenme işlemi sonucunda öneriler, tahminler, öngörüler sunulabilmektedir. Derin öğrenme algoritmaları ile doğal dil işleme NLP alanında birçok çalışmalar yapılmakta olup başarılı sonuçlar elde edilmektedir.
Derin öğrenme süreçleri;

- Veri toplama ve hazırlama: Eğitim veri seti için gerekli verilerin toplanması ve hazırlanması.
- Model tanımlama: Derin öğrenme modelinin tanımlanma, yapılandırma ve optimizasyon işlemleri.
- Eğitim: Modelin veri setleriyle eğitilme işlemi.
- Model değerlendirme: Eğitilmiş bir modelin performansının ve doğruluğunun izlenmesi ve gerektiğinde güncellenmesi işlemi.
- Modelin periyodik olarak güncellenmesi: Verilerin güncellendiği veya modelin performansının optimize edilmesi gerektiği durumlarda modelin yeni verilerle güncellenmesi işlemi.
- Kullanım: Eğitilmiş modelin gerçek dünya uygulamalarında kullanılması işlemi.

Bu belirtilen süreçler, her derin öğrenme uygulaması için değişebilir ve bazı adımların atlanması veya yeni adımların eklenmesi de mümkündür. Önemli olan nokta, verilerin doğru bir şekilde hazırlanması, modelin doğru bir şekilde tanımlanması ve eğitilmesi ve modelin performansının sürekli olarak izlenmesidir.

Ensemble learning: Birden fazla modelin birleştirilmesi ve ortak bir sonuç üretilmesi.

Derin öğrenme yöntemlerinin uygulama alanları arasında aşağıdakiler yer almaktadır (Küçük ve Arıcı, 2018:77);

- Dil modelleme ve doğal dil işleme

- Konuşma ve ses işleme
- Bilgi erişimi
- Nesne tanıma ve bilgisayarlı görü
- Çok modlu ve çok görevli öğrenme

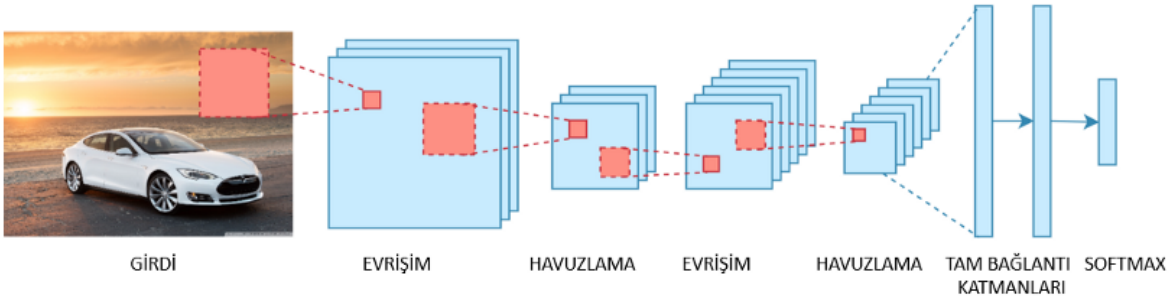
5.2 Derin Öğrenme Algoritmaları

Derin öğrenme algoritmaları, genellikle sinir ağlarının bir türüdür ve aşağıdaki alt türleri içerebilir:

- Evrişimsel Sinir Ağları
- Tekrarlayan Sinir Ağı
- Uzun Kısa Süreli Hafıza Algoritması

5.2.1 Evrişimsel Sinir Ağları (CNN)

Girdi verisi olarak görüntüleri kullanan popüler bir derin öğrenme algoritmasıdır. CNN, yapay sinir ağlarından esinlenerek oluşturulmuş ve toplanmış bilgileri uçtan uca öğrenebilen mimaridir. CNN, yeterli kapasitesi ve akıllı model yapısı sayesinde büyük ölçekli verilerin de üstesinden gelebilmektedir (Arı ve Hanbay, 2019:1398).



Şekil 20: Evrişimli Sinir Ağları Yapısı*(Sayraci, 2021)*

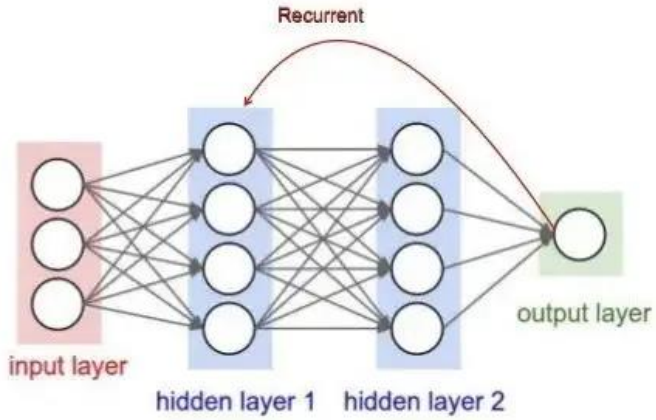
Evrişimsel sinir ağları katmanları aşağıdaki gibidir;

- Giriş katmanı: Verilerin gösterildiği bölüm
- Evrişim katmanı: Görüntü verilerinin özelliklerini çıkarmak için kullanılan bir katmandır. Bu katmanda verilere evrişim filtreleri uygulanır.
- Havuzlama katmanı: Verilerin boyutunun azaltılması için kullanılan bir katmandır. Bu katman, verinin boyutunun düşürülmesi ile birlikte verilerin özelliklerinin korunmasını sağlar.

- Tam Bağlantı Katmanı: CNN'in son adımı bu katmanda gerçekleşir. Verilerin sınıflandırılması veya tahmin yapılması için kullanılan bir katmandır.

5.2.2 Tekrarlayan Sinir Ağı (RNN)

Zaman serisi verileri için kullanılan bir derin öğrenme algoritmasıdır. RNN, girdi verilerini işlerken kendi hafızalarından faydalanır. RNN'in tekrarlayan olarak isimlendirilmesinin nedeni, bir dizinin her öğesi için aynı görevi yerine getirmesinin, çıktının geçmiş verilerin anımsanmasına bağlı olmasıdır.



Şekil 21: RNN Mimarisi*(Gürbüz, 2020)*

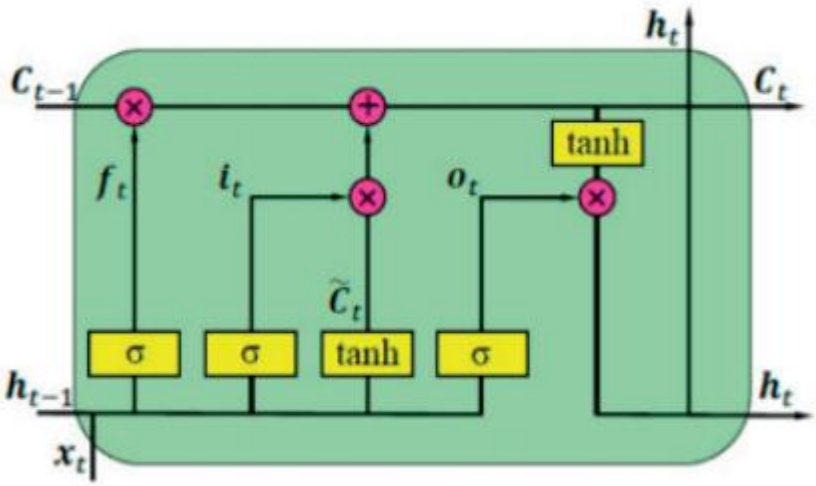
RNN mimarisi; giriş katmanından, gizli katmandan ve çıkış katmanından oluşmaktadır. Tüm katmanlar birbirinden bağımsızdır. Bir önceki giriş durumu depolanarak yeni giriş değeriyle birleştirilir. Böylelikle yeni giriş ile bir önceki girişin ilişkisi sağlanır.

RNN'ler, metin çevirisi, metin yazma, ses tanıma, müzik yapma gibi uygulamalarda kullanılabilir.

5.2.3 Uzun Kısa Süreli Hafıza Algoritması LSTM

Uzun Kısa Süreli Hafıza (LSTM) algoritması, zaman serisi verileri için kullanılan, eski verilerin etkilerinin unutulmamasını sağlayan bir öğrenme algoritmasıdır. Özellikle uzun süreli bağımlılıkları takip etmekte zorluk çeken problemlerde tercih edilir. LSTM algoritması sıralı verilerin modellenmesinde kullanılır. RNN algoritmasının daha gelişmiş bir türüdür ve RNN'lerin zayıf yanı olan uzun süreli hafıza tutma becerisini çözmek için tasarlanmış bir algoritmadır.

LSTM, çok katmanlı sıralı bir yapıya sahiptir. Temel olarak üç katmandan oluşur. Bu katmanlar; girdi katmanı, unut katmanı ve çıktı katmanıdır. Her bir katman içerisinde geçmişteki bilgiyi saklamakta ve gelecekteki bilgiyi tahmin etmekte kullanılan hücreler bulunur. Bu hücrelere "hafıza hücresi" denilir. İlk katman olan unut katmanı hangi bilginin unutulup unutulmayacağına, girdi katmanı hangi bilginin depolanacağına, çıktı katmanı ise hangi bilginin çıktı olacağına karar verir.



Şekil 22: LSTM Mimari Yapısı*(Kara, 2019)*

Kara (2019:885) yapmış olduğu çalışmada belirttiği gibi, LSTM algoritmasında öncelikle girdi olarak Xt ve $ht-1$ bilgileri kullanılarak nelerin silineceğine karar verilir. Bu işlemler unut katmanında (ft) Eşitlik (1) kullanılarak yapılır, aktivasyon fonksiyonu olarak sigmoid kullanılır.

$$f_t = \sigma(W_{f,x} * X_t + W_{f,h} * h_{t-1} + b_f) \quad 1$$

İkinci adımda, yeni bilgilerin belirleneceği girdi katmanı devreye girer ve öncelikle (i_t) Eşitlik (2) kullanılarak sigmoid fonksiyonu ile bilgiler güncellenir. Ardından Eşitlik (3) ile yeni bilgiyi oluşturacak aday bilgiler tanh fonksiyonu tarafından belirlenir.

$$i_t = \sigma(W_{i,x} * X_t + W_{i,h} * h_{t-1} + b_i) \quad 2$$

$$\tilde{C}_t = tanh(W_{c,x} * X_t + W_{c,h} * h_{t-1} + b_c) \quad 3$$

Eşitlik (4) tarafından yeni bilgiler oluşturulur.

$$C_t = C_{t-1} * f_t + i_t * \tilde{C}_t \quad (4)$$

Son olarak çıktı katmanında Eşitlik (5) ve (6) kullanılarak çıktı verileri elde edilir.

$$o_t = \sigma(W_{o,x} * X_t + W_{o,h} * h_{t-1} + b_o) \quad (5)$$

$$h_t = o_t * tanh(C_t) \quad (6)$$

Yukarıda ifade edilen süreç tekrarlanarak devam eder. Ağırlık parametreleri (W) ve bias parametreleri (b) gerçek eğitim değerleri ile LSTM çıktı değerleri arasındaki farkı minimize edecek şekilde model tarafından öğrenilmektedir (Kara, 2019:885).

LSTM, ses ve görüntü gibi zaman serisi verilerinin analizinde, metin üretiminde, ses tanıma, el yazısı tanıma, müzik besteleme gibi uygulamalarda kullanılmaktadır. Ayrıca, finansal tahminler, makine öğrenimi veya nörolojik uygulamalar gibi alanlarda da LSTM kullanılabilir. LSTM, Google Translate, Siri gibi çok kullanılan araçların geliştirilmesine de katkıda bulunmuştur.

LSTM, RNN'ler gibi geriye doğru yayılım yapar ve veri setinde geçmişteki bilgiyi kullanır. Uzun süreli bağımlılıkları takip etmekte daha iyi bir performans gösterir ve daha kompleks veri setlerinde kullanılabilir.

LSTM, tekrarlayan sinir ağlarının (RNN) belleğini genişleten bir modeldir. RNN'ler mevcut sinir ağında kullanılmak üzere kalıcı önceki bilgileri kullanmaları nedeniyle "kısa süreli hafızaya" sahiptir. Mevcut görevde önceki bilgiler kullanılır (Alpay, 2020:453).

LSTM algoritmasının eğitim süreci zaman alıcı olabilir bu nedenle, bazı uygulamalarda LSTM algoritması yerine hızlı öğrenme algoritmaları kullanılabilir. Örneğin, klasik yapay sinir ağları, sınıflandırma algoritmaları, LSTM algoritmasına göre daha hızlı öğrenme sağlar.

LSTM (Uzun Kısa Süreli Hafıza) algoritmasının avantajları;

- Uzun süreli bağımlılıkları takip edebilir. Örneğin, metindeki bir kelimeyle ilgili anlamın birkaç cümle sonra anlaşılması gerektiğinde, LSTM algoritması bu bağımlılığı takip edebilir.
- Gürültüye dayanıklıdır. Gürültülü verilerde dahi iyi performans gösterir.

- Çoklu çıktı oluşturabilir.

Dezavantajları;

- Yüksek miktarda veri gerektirir. Küçük veri setinde performansı düşebilir.
- Yüksek miktarda bellek ve işlem gücü gerektirir
- Eğitim süreci zaman alıcı olabilir.
- Veri setinin kalitesi ve özellikleri ile doğrudan ilişkilidir. Bu nedenle, veri seti özellikleri ve kalitesi düşükse, LSTM algoritmasının performansı düşebilir.

BÖLÜM 6: BULGULAR

6.1 Geliştirmede Kullanılan Araçlar

Geliştirme aşamasında Python dili ile yazılım gerçekleştirilmiş olup JET BRAINS ürünü olan PyCharm Community IDE kullanılmıştır.

Şekil 23: PyCharm Uygulaması

6.2 Kullanılan Kütüphaneler

import os

İşletim sistemi ile ilgili işlemleri yapmak için kullanılır. Bu kütüphane ile dosya işlemleri yapılabilir.

from tkinter import *

Kullanıcı arayüzü elemanları (buton, label, menü, resim vb.) gibi işlemleri yapmak için kullanılır.

import wx

Kullanıcı arayüzü elemanları (buton, label, menü, resim vb.) gibi işlemleri yapmak için kullanılır.

from flask import Flask, render_template

“flask” web uygulamaları geliştirmek için kullanılır. "render_template" ise HTML sayfalarının dinamik olarak oluşturulmasını sağlar.

import glob

Belirli bir dizin ya da klasör altındaki dosyaları elde etmek için kullanılır.

import re

Düzenli ifadeler (regular expressions) kullanarak metin işleme işlemlerini gerçekleştirmek için kullanılır.

import pandas as pd

Tablo verilerini (dataframes) okumak, yazmak ve işlemek için kullanışlı fonksiyonlar ve araçlar sağlar.

import fitz

PDF dosyaları üzerinde işlem yapmak için kullanılır.

import numpy as np

Çok boyutlu diziler ve matrisler oluşturmak ve işlemek için kullanılır.

from keras.preprocessing.text import Tokenizer

"keras.preprocessing.text" modülü, metin verileri için önişleme işlemlerini yapmak için kullanılır. "Tokenizer" ise doğal dil işleme (NLP) problemleri kullanılır. "Tokenizer", verilen bir metin kümesindeki kelimeleri ve sembolleri makine öğrenmesi algoritmalarında kullanmak için sayısal bir diziye dönüştürür.

from keras_preprocessing.sequence import pad_sequences

"keras_preprocessing.sequence" modülü, makine öğrenmesi uygulamalarında kullanılan dizi verilerini işlemek için kullanılır. "pad_sequences" ise bir dizi veriyi belirli bir boyuta göre yeniden şekillendirmek için kullanılır. Farklı uzunluktaki dizileri aynı boyuta getirmek için kullanılır.

import nltk

Metin verilerinin işlenmesi, sınıflandırılması ve etiketlenmesi gibi birçok farklı NLP işlemleri için kullanılabilir.

from keras.models import Sequential

"keras.models" modülü, yapay sinir ağı modelleri oluşturmak için kullanılır. "Sequential", katmanları (layers) sırayla ekleyerek bir model oluşturulmasını sağlar.

from keras.layers import Embedding, LSTM, Dense

"keras.layers" modülü, yapay sinir ağı modelleri oluşturmak için kullanılır. "Embedding" sınıfı, bir sözcük gömme (embedding) katmanı oluşturmak için kullanılır. "LSTM" sınıfı, bir uzun-kısa süreli bellek katmanı oluşturmak için kullanılır. LSTM katmanı, metin sınıflandırma, dil tahmini ve çeviri gibi NLP görevlerinde yaygın olarak kullanılır."Dense" sınıfı, bir yoğun (dense) katmanı oluşturmak için kullanılır. Bu katman, girdideki her bir düğüm çıkıştaki her bir düğüm ile bağlıdır. Dense katmanı, metin sınıflandırma, resim tanıma ve öngörü gibi birçok makine öğrenmesi uygulamasında kullanılır.

import warnings

Uygulamadaki uyarıları kapatmak için kullanılır.

import jpype

JPype, Python dilinde yazılan programların Java kodlarına erişmesine ve Java sanal makinesinde (JVM) çalışan Java sınıflarını kullanmasına izin veren bir Python-Java entegrasyon kütüphanesidir.

6.3 Uygulama Geliştirme Aşamaları

Uygulama için Ulusal Tez Merkezinden, yazım dili Türkçe olan, çoğunlukla 2020 yılı ve sonrasına ait Fen, Sosyal ve Tıp Enstitü grubunda yer alan 900 tez indirilmiştir. İndirilen bu tezler Python dili kullanılarak taranmak üzere klasöre aktarılmıştır.

Şekil 24: PyCharm Uygulaması

6.3.1 Tez Dosyalarının Okunması

Ulusal Tez Merkezinden indirilen tezlerin içeriklerinin uygulama yapısına uygunluğunu tespit etmek için tez dosyaları uygulama üzerinden taranarak kontrol edilmiştir.

Bu kapsamda yapılan kontroller aşağıdaki gibidir;

- Tez içeriğinde özet sayfası var mı,
- Özet sayfasında anahtar kelime var mı,
- Tez özeti anahtar kelime ile bitmiş mi,
- Anahtar kelime bulunamayan tezlerde anahtar kelime nasıl belirtilmiş, anahtar kelime belirtilmemişse dosya kapsam dışı bırakılır, anahtar kelime farklı ifadeyle eklenmişse bu ifade kontrole eklenir.
- Tez içeriğinde kaynakça var mı, kaynakça bulunamadıysa bu tezler incelenir. Eğer farklı ifadeyle eklenmişse bu ifade kontrole eklenir, kaynakça içermiyorsa işlemlere devam edilir.

- Kaynakça bilgisi içeren tezlerde, kaynakça ve sonrasındaki sayfalar veri kümseine dahil edilmez.
- İçerik bilgisi alınabilmiş mi, alınamadıysa tez dosyası kapsam dışı bırakılır.

Belirtilen kurallara uygun olan tezlerde veri kümesi oluşturma işlemleri yapılır.

Tez dosya tarama uygulaması Pycharm üzerinden ayağa kaldırılarak web ortamında çalıştırılır. Uygulamaya http://localhost:8085/ adresinden ulaşılır. Tarayıcıda ilgili link girildiğinde Şekil 24'te belirtilen ekran açılır. Ekranda bulunan "Tezleri Tara" butonu kullanılarak daha önceden klasöre eklenen 900 tez tek tek taranarak içerik, tez numaraları ve orjinal anahtar kelimelerini içerecek şekilde ayrı ayrı txt dosyalarına aktarılır.

Şekil 25: Uygulama Tez Tarama Sayfası

Tüm tez dosyaları tek tek taranarak içerik bilgisi "tezNo_context.txt" dosyasına, tez numarası "tezNo_title.txt" dosyasına, tez özetinde yer alan orjinal anahtar kelimeler ise "tezNo_keywords.txt" dosyasına aktarılır.

Tez dosya tarama aşamasında her tezin içerik metni için normalizasyon işlemleri yapılmıştır. Yapılan normalizasyon işlemleri aşağıdaki gibidir;

- Tüm kelimeleri küçük harfe çevirme
- Türkçe karakterler için pattern
- Romen rakamları kaldırma
- Rakamları kaldırma
- Boşlukları kaldırma

- Noktalama işaretlerini kaldırma
- İşaretleri kaldırma
- Kelimeleri köklerine ayırma işlemi (stemming)

Ayrıca Ersoy'un github sitesinden sunduğu projesinden elde edilen stopWords_tr.txt dosyası uygulamada kullanılmıştır (Ersoy, 2021). Geliştirme süresince saptanan yeni kelimeler de bu dosyaya eklenmiştir. İçerik bilgisi öncelikle tokenize edilerek her bir kelime birbirinden ayrılmıştır. Daha sonra bu kelimeler eğer Stopwords dosyasında yer alan kelimeler ise bu kelime içerikten kaldırılmıştır.

Tez içeriğindeki ve anahtar kelimelerdeki tüm kelimeler Zemberek kütüphanesi kullanılarak köklerine ayırma işlemi yapılmıştır. Böylelikle benzersiz kelime sayısı azaltılmış olup bu durum eğitim sürecinin de kısalmasını sağlamıştır.

Özet sayfasından orjinal anahtar kelime çıkarımı esnasında belirli bir standartla dosyadan okuma işlemi yapılmış olup, anahtar kelimeden sonra başka bilgiler (danışman bilgisi, sayfa sayısı, enstitü bilgisi vb.) eklenen tez dosyaları kapsam dışı bırakılmıştır.

Belirlenen standart tez özet sayfasında en alt satırda anahtar kelime yer alacak şekilde aşağıdaki ifadeleri içermesi durumunda okuma işlemi gerçekleşmektedir. Tezlerde anahtar kelime belirtilirken farklı yazımlarla karşılaşılmış olup anahtar kelimesi alınamayan her bir tez dosyası uygulama üzerinden tek tek saptanarak kapsam dışı bırakılmıştır. Yazım şekillerine göre kontroller sağlanarak her bir tezin orjinal anahtar kelimeleri içerikten alınarak tezNo_keywords.txt dosyasına eklenmiştir.

Tezlerde anahtar kelime bilgisi alınırken aşağıdaki ifadeler kullanılmıştır;

- aKelime1 = 'anahtar kelimeler:'
- aKelime2 = 'anahtar sözcükler:'
- aKelime3 = 'anahtar kelimeler;'
- aKelime4 = 'anahtar kelime:'
- aKelime5 = 'anahtar kelimeler.'

Tez içeriğinde sonuç bölümünden sonra kaynakça ve bazı tezlerde özgeçmiş yer aldığı gözlemlenmiştir. Bu sayfaların tez içeriğinden anahtar kelime çıkarımında kullanımının uygun olmadığı düşünüldüğü için kapsam dışı bırakılmıştır. Ancak bu çalışma esnasında bazı tezlerde kaynakça belirtilirken farklı ifadelerin kullanıldığı görülmüştür. Belirlenen bu ifadelerin içerikte de geçebileceği düşünülmüş olup sayfanın en başında yer alması durumunda bu sayfalar içerikte kapsam dışı bırakılmıştır.

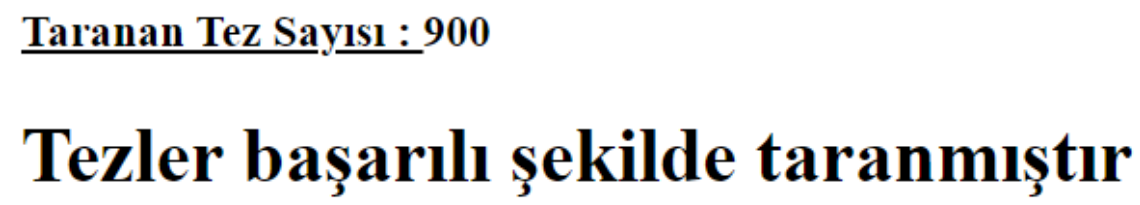
Tezlerde kaynakça belirtilirken aşağıdaki ifadeler kullanıldığı görülmüştür;

- kaynaklar
- yararlanılan kaynaklar
- bibliyografya
- kaynakçalar
- kaynakça

```
Reading file C:\Users\aynur.gunay\PycharmProjects\tezOkuma\readPdf\pdfFile\615922.pdf
 geleneksel, tamamlayıcı tıp, bilgi, tutum
C:\Users\aynur.gunay\PycharmProjects\tezOkuma\output\TOPLU\615922_title.txt
The file C:\Users\aynur.gunay\PycharmProjects\tezOkuma\output\TOPLU\615922_title.txt was written.


Reading file C:\Users\aynur.gunay\PycharmProjects\tezOkuma\readPdf\pdfFile\615990.pdf
 cerrahi, cinsiyet, memnuniyet.
C:\Users\aynur.gunay\PycharmProjects\tezOkuma\output\TOPLU\615990_title.txt
The file C:\Users\aynur.gunay\PycharmProjects\tezOkuma\output\TOPLU\615990_title.txt was written.


Reading file C:\Users\aynur.gunay\PycharmProjects\tezOkuma\readPdf\pdfFile\616494.pdf
 kanalopatiler, nöbet, kalp kapak patolojisi
C:\Users\aynur.gunay\PycharmProjects\tezOkuma\output\TOPLU\616494_title.txt
The file C:\Users\aynur.gunay\PycharmProjects\tezOkuma\output\TOPLU\616494_title.txt was written.
```

Şekil 26: Uygulama Tez Tarama Çıktı

Tarama işlemi tamamlandıktan sonra Şekil 27'deki ekran gelir.



Şekil 27: Uygulama Tez Sonuç Ekran

6.3.2 Veri Kümesinin Oluşturulması

Çalışmanın ikinci adımı olarak her bir tez için icerik, tez numarasi ve anahtar kelimeleri içeren txt dosyalarının tek bir txt dosyasında birleştirme işlemi yapılmıştır. Bu işlem için

toplam 900*3=2700 dosya birleştirilmiştir. Birleştirilen text dosyasının ismi "ThesisData.txt" olarak belirtilerek klasöre kaydedilmiştir.

```
./output/TOPLU_yedek\747961_context.txt
./output/TOPLU_yedek\747961_keywords.txt
./output/TOPLU_yedek\747961_title.txt

./output/TOPLU_yedek\748076_context.txt
./output/TOPLU_yedek\748076_keywords.txt
./output/TOPLU_yedek\748076_title.txt

./output/TOPLU_yedek\748096_context.txt
./output/TOPLU_yedek\748096_keywords.txt
./output/TOPLU_yedek\748096_title.txt

./output/TOPLU_yedek\749742_context.txt
./output/TOPLU_yedek\749742_keywords.txt
./output/TOPLU_yedek\749742_title.txt
```

Şekil 28: Uygulama Veri Birleştirme Çıktı

6.3.3 Modelin Eğitilmesi

Tez bilgilerini içeren ve ön işlemlerden geçirilen veri kümesi LSTM modelinde kullanılmak üzere hazırlanmıştır. Tez dosyalarından elde edilen bu veriler ilgili kişilerce belirlenmiş olup doğrulanmış verilerdir. Modeli eğitirken amaçlanan, eğitim verisini doğru seçmek ve doğru algoritmalarla minimum hatada doğru sonuçlar elde edebilmektir. Hatanın minimum olması modelin anahtar kelimeleri doğru tahmin etmesini sağlayacaktır. O yüzden amaç hatayı minimum seviyede tutacak şekilde modeli eğitmek ve uygun parametreleri kullanmaktır.

Elimizdeki veri setindeki tez içerikleri model eğitilirken training veri seti olarak kullanılacaktır. Anahtar kelimeler ise target veri seti olarak modelde belirtilir. Training veri setinin bir kısmı ayrılarak test verisi olarak kullanılmaktadır. Test verisi model tarafından bilinmemektedir. O yüzden tahminde bulunulurken training veri setine göre bir tahmin yürütülmektedir. Test verisi kullanımındaki amaç modelin verilen bilgiler doğrultusunda başarı oranını görebilmektir.

Training verisinin bir kısmı da validation veri olarak ayrılmaktadır. Validation veri, modelin eğitimi esnasında her epoch işleminden sonra model doğru öğrendi mi diye

kontrol etmek için kullanılmaktadır. Train accuracy ile validation accuracy arasında doğru orantı vardır. Yani training accuracy değeri ne kadar yüksek olursa validation accuracy değeri de o kadar yüksek olmaktadır. Modelde asıl amaçlanan da budur. Bu yüzden veri setine en uygun modeli belirlemek bizi doğru sonuçlara götürecektir.

Veri seti normalizasyon işlemlerinden geçirilerek hazırlanmış olmasına rağmen tarining işleminden önce yapılması gereken işlemler bulunmaktadır. Öncelikle veri seti dataframe'e alınır. DataFrame, iki boyutlu veriler içeren bir yapıdır. Dataframe'e txt içerisinde yer alan tez numarası, içerik ve anahtar kelime bilgileri aktarılır. Dataframe'e aktarılan verilere kolon isimleri tanımlanır. Dataframe içerisindeki içerik bilgileri eğitim verisi için tokenize edilir. Tüm textlerimizin içerisindeki kelimelerin hepsini öğrenmek için tokenizer'ın fit_on_text fonksiyonu kullanılır. Bu işlemle veri kümesinde herhangi bir değişiklik yapılmamaktadır. Fakat tokenizer, giriş verisiyle ilgili bilmesi gereken şeyleri öğrenmektedir. Yani eğitim verisinde kullanılacak kelimeleri, kelimelerin frekanslarını vb. Tokenizer her bir kelimeye bir index atamaktadır. O kelime 1. satırda da olsa en son satırda da olsa aynı numarayla indekslenir. Böylelikle kelimeler tekilleştirilmiş olunur ve toplam tekil kelime sayısına ulaşılmış olunur. Daha sonra giriş verilerinin sayısal ifadelerle temsil edilmesi için tokenizer'ın texts_to_sequences metodu kullanılır. Sonuç olarak her bir kelime sayısal ifadelere dönüştürülür. Aynı işlem anahtar kelimeler için de uygulanır.

Tokenizer işleminde model eğitilmeden önce veri setlerinin aynı boyuta getirilmesi gerekmektedir. Bu işlem için tokenizer'ın padding metodu kullanılır. Elimizdeki veri setinde de farklı uzunlukta tez içerikleri ve aynı şekilde farklı uzunlukta anahtar kelimeler bulunmaktadır. Öncelikle maksimum uzunluktaki içerik ve maksimum uzunluktaki anahtar kelime sayıları bulunur. Daha sonra diğer içerikler de maksimum içeriğe göre aynı uzuluğa getirilir. Aynı işlem anahtar kelimeler için de yapılır.

Veri kümesi, eğitim ve test verisi olacak şekilde belirtilerek iki bölüme ayrılmaktadır. Artık veri setimiz eğitim için hazır durumdadır. Sıradaki işlem, modeli ve katmanları belirlemektir.

Öncelikle tensorflow kütüphanesinin içindeki Keras'ı kullanarak yeni bir Sequential model nesnesi oluşturulur ve modelin katmanları eklenir. İlk eklenen katman input

katmanıdır. Modeldeki kelimelerin tek tek daha az boyutlu gerçek değerli vektörler olarak ifade edilmesi için modele Embedding katmanı eklenir. Embedding katmanı, her kelimeyi bir sayısal vektöre dönüştürmek için kullanılan bir Keras katmanıdır. Bu, makine öğrenimi modelinin kelimelerin gömülmesini öğrenmesine ve daha sonra bu gömülmeleri kullanarak metnin anlamını anlamasına yardımcı olur. Bu işlem aynı zamanda kelime kodlaması olarak da adlandırılır. Embedding katmanının girdisi, kelime dizinleri şeklindedir ve çıktısı, kelime gömülmeleri şeklindedir. Kelime dizinleri, her kelimeyi bir benzersiz sayısal değere dönüştüren bir kelime sözlüğünden alınır. Sözlükteki her kelime birbirinden farklı bir sayısal değerle eşleştirilir. Kelimeler arasındaki mesafeler bu şekilde temsil edilir.

Modelde LSTM algoritması kullanılmaktadır. LSTM(100) ifadesinde, LSTM katmanının 100 gizli hücresi olduğu belirtilmektedir. Gizli hücre sayısını yüksek tutmak, modelin daha fazla bilgiyi işlemesini sağlar ancak bu durum daha fazla hesaplama gücü gerektirir ve aşırı öğrenmeye neden olabilir. Bu yüzden bu parametreye uygun bir değer atamak önemlidir.

Keras Dropout, bir yapay sinir ağı modelinin aşırı öğrenmesini azaltmak için kullanılan bir yöntemdir. Aşırı öğrenme, modelin eğitim verilerine çok fazla uyum sağlaması ve genelleştirme yapamaması durumudur. Dropout, ağın farklı bölümlerini rastgele olarak devre dışı bırakarak, her eğitim örneği için farklı bir ağ oluşturur. Bu, ağın farklı özellikleri öğrenmesine ve daha iyi genelleştirme yapmasına yardımcı olur. Keras'ta Dropout yöntemi, katmanlar arasında eklenerek kullanılır. Bu yüzden modelde katmanlar arasında droput yöntemi kullanılmıştır.

Keras BatchNormalization, bir yapay sinir ağı modelindeki katmanların çıkışlarını normalleştirmek için kullanılan bir yöntemdir. Bu yöntem, eğitim sürecinde her minibatch'teki, yani verilerin küçük boyutlu halindeki verilerde, girdilerin ortalaması ve varyansı kullanılarak çıkışların standartlaştırılmasını sağlar. Böylece, ağdaki diğer katmanların girdileri daha tutarlı bir şekilde işleyebilir ve eğitim sürecinde daha hızlı bir şekilde ilerleyebilir. BatchNormalization, aşırı öğrenmeyi önlemeye yardımcı olur ve daha büyük ölçekli bir yapay sinir ağı oluşturmanıza izin verir. Ayrıca, eğitim sürecindeki diğer yöntemlerle birleştirilerek daha iyi sonuçlar elde edilebilir. Keras'ta BatchNormalization katmanı, bir modelde diğer katmanlarla aynı şekilde tanımlanabilir. Bu katman, özellikle

çok katmanlı modellerde performansı artırır ve ağın daha iyi genelleştirilmesine yardımcı olur. Bu nedenle modelin performansını arttırmak için katmanlar arasında BatchNormalization yöntemi eklenir.

Dense(max_keywords, activation='sigmoid') ifadesinde Dense katmanı, tam bağlantılı bir katmandır ve hücrelerin tamamı önceki katmandan gelen her bir girdiye bağlıdır. Bu katmanın çıkışı, verilen sınıfların olasılık dağılımını hesaplamak için bir aktivasyon fonksiyonuna verilir. Sigmoid aktivasyon fonksiyonu, çıktı değerlerini 0 ile 1 arasına sıkıştıran bir fonksiyondur. Bu nedenle, sigmoid aktivasyon fonksiyonu, ikili sınıflandırma problemleri için son katmanda sıklıkla kullanılır.

Model eğitimine başlamadan önce son adım olarak modelin derleme işlemi yapılır.

model.compile(optimizer=’adam’,loss="binary_crossentropy",
metrics=[metrics.binary_accuracy])

loss=’binary_crossentropy’ ifadesi, sınıflandırma problemleri için kullanılan bir kayıp fonksiyonudur.

optimizer='adam' ifadesi, modelin eğitiminde kullanılan optimizasyon algoritmasını ifade eder. Adam algoritması daha hızlı ve daha güvenilir bir şekilde optimum çözüme ulaşılmasını sağlar. Bu nedenle, derin öğrenme uygulamalarında sıklıkla tercih edilen bir optimizasyon algoritmasıdır.

Eğitim ve test sırasında model tarafından değerlendirilecek metriklerin listesi “metrics” parametresinde belirtilir.

Model eğitim işlemi için aşağıdaki ifade kullanılmaktadır;

model.fit(x=X_train, y=y_train, batch_size=2, epochs=4, validation_split=0.1, steps_per_epoch=5, verbose=1)

X değeri tez içerik bilgisini, y değeri anahtar kelime bilgisini ifade eder.

‘batch_size’ ifadesi, bir eğitim döngüsünde kullanılan küme boyutunu belirleyen bir parametredir. Eğitim verileri, genellikle büyük boyutlu veri setleri olduğu için, tüm veri setini tek seferde modele vermek mümkün değildir. Bu nedenle, veri seti daha küçük

parçalara ayrılır ve her bir parça model tarafından ayrı ayrı işlenir. ‘batch_size’ parametresi bu toplu işlemin boyutunu belirler. ‘batch_size=2’ parametresi modelin her bir adımda 2 örnekten oluşan bir veri kümesiyle eğitildiğini belirtir. Bu süreç, tüm veri seti tamamen işlenene kadar devam eder.

‘validation_split’, modelin eğitimi sırasında veri kümesinin bir bölümünü doğrulama (validation) verisi olarak ayırmak için kullanılan bir parametredir. ‘validation_split=0.1’ ifadesi, eğitim veri setinin %10'unun doğrulama veri seti olarak ayrıldığı anlamına gelir. Böylelikle, modelin eğitiminde %90'lık bir veri seti kullanılırken, %10'luk bir veri seti doğrulama amaçlı olarak kullanılmaktadır.

‘epochs’, modelin eğitimi sırasında tüm eğitim veri kümesinin kaç kez işleneceğini belirleyen bir parametredir. ‘epochs=4’ ifadesi, modelin eğitimi sırasında veri kümesinin 4 kez işleneceği anlamına gelir.

‘steps_per_epoch’ parametresi, her bir eğitim döngüsünde ne kadar adım alınacağını belirler. ‘steps_per_epoch=5’ ifadesi, bir epoch boyunca modelin 5 adım işleyeceği anlamına gelir. Böylelikle, eğitim veri seti, her bir epoch'ta 5 eşit parçaya bölünecektir.

Keras'ta eğitilen LSTM modelini model.save() fonksiyonu kullanılarak dosyaya kaydedilir.

Son olarak; modelin performansını ölçmek için ‘model.evaluate()’ fonksiyonu kullanılır. Belirtilen test veri kümesindeki doğruluk (accuracy) veya kayıp (loss) değerlerini almak için bu fonksiyona test olarak belirtilen veri kümeleri parametre olarak eklenir. Bu test verilerine göre sonuçlar alınır.

SONUÇ

Makine öğrenmesi tekniklerinden derin öğrenme algoritması kullanarak tez içeriklerinden anahtar kelime çıkarımı yapılmıştır. Veri setinin doğrulanmış verilerden oluşması ve içeriğin hazırlanma aşamalarına hâkim olunması karşılaşılan sorunların daha çabuk çözülmesini sağlamıştır. Derin öğrenme yöntemlerinden biri olan LSTM ile model eğitim süreci tamamlanmıştır.

Eğitimde kullanılan **900** tezden oluşturulan veri seti yaklaşık **90 MB** boyutundadır. Veri setinde yer alan benzersiz kelime sayısı **282.883**'tür. Ayrıca en fazla kelimeye sahip tez dosyasının kelime sayısı **128.851** olarak bulunmuştur. En fazla anahtar kelime sayısı ise **32**'dir. Bu sayılara göre x_train ve y_train veri kümeleri oluşturulmuştur.

```
max_context: 128851
max_keywords: 32
vocab_len 282883
```

Modeldeki kelimelerin tek tek daha az boyutlu gerçek değerli vektörler olarak ifade edilmesi için modele Embedding katmanı eklenir. Daha sonra sırası ile LSTM katmanı ve yoğun katman eklenir.

Modelin performansını ölçmek için 'model.evaluate()' fonksiyonu kullanılır. Belirtilen test veri kümesindeki doğruluk (accuracy) veya kayıp (loss) değerlerini almak için bu fonksiyona test olarak belirtilen veri kümeleri parametre olarak eklenir. Bu test verilerine göre sonuçlar alınır.

score = model.evaluate(X_test, y_test, verbose=1)

Sonuç olarak elde edilen doğruluk (accuracy) değeri **0.7122**'dir. Kayıp (loss) değeri ise **0.1117** olarak alınmıştır.

Eğitilen modelden anahtar kelime tahmin edebilmek için normalizasyon işlemlerinden geçirilmiş bir tez dosyası seçilmiş ve bu tezin anahtar kelime tahmini yapılmıştır. Tahmin edilen kelimelerin en az 2 tanesinin orijinal anahtar kelimeler içerisinde yer aldığı görülmüştür.

Uygulama üzerinden alınan sonuçlar aşağıdaki gibidir,

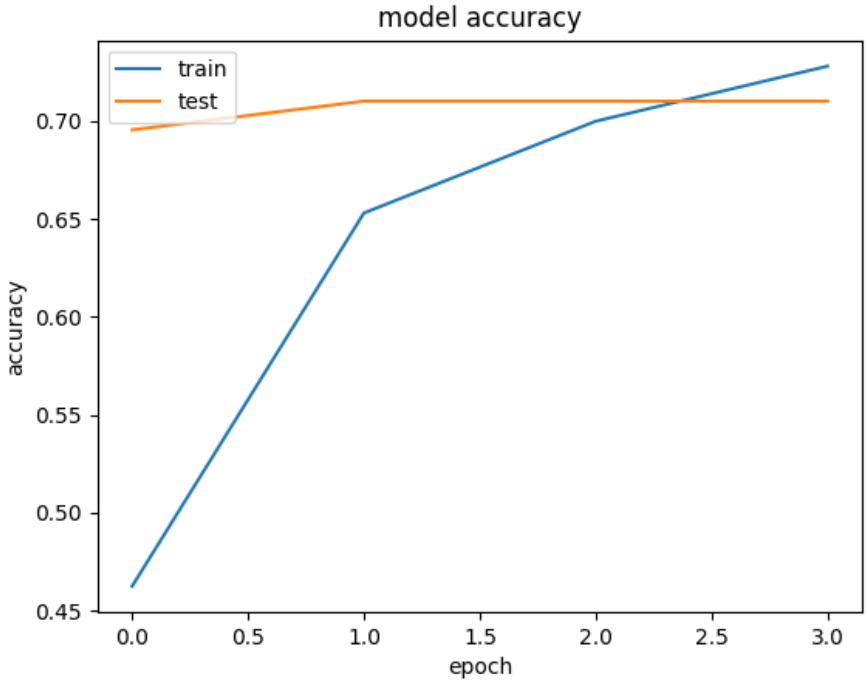
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, None, 100)         28288400

 lstm (LSTM)                 (None, 100)               80400

 dropout (Dropout)           (None, 100)               0

 batch_normalization (BatchN  (None, 100)              400
 ormalization)

 dense (Dense)               (None, 32)                3232

=================================================================
Total params: 28,372,432
Trainable params: 28,372,232
Non-trainable params: 200
```

Şekil 29: Uygulama Çıktı-1

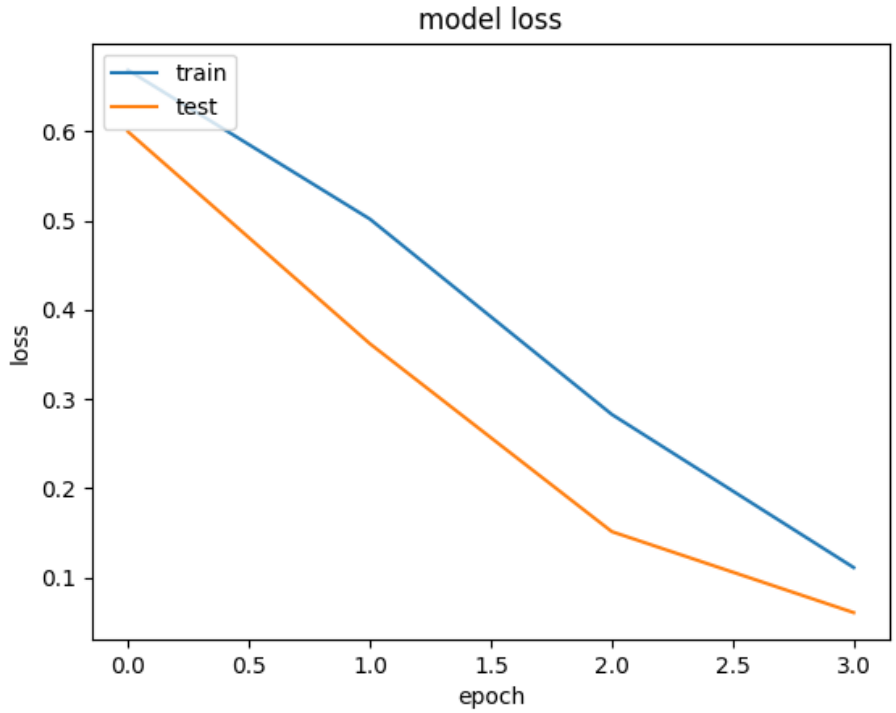
```
=================================================================
Total params: 28,372,432
Trainable params: 28,372,232
Non-trainable params: 200
_________________________________________________________________
Epoch 1/4
5/5 [==============================] - 9217s 1947s/step - loss: 0.6862 - binary_accuracy: 0.4281
Epoch 2/4
5/5 [==============================] - 11602s 2522s/step - loss: 0.5464 - binary_accuracy: 0.6812
Epoch 3/4
5/5 [==============================] - 11845s 2563s/step - loss: 0.3551 - binary_accuracy: 0.7625
Epoch 4/4
5/5 [==============================] - 11932s 2612s/step - loss: 0.1595 - binary_accuracy: 0.7156
3/3 [==============================] - 652s 219s/step - loss: 0.1117 - binary_accuracy: 0.7122
**********
0.7121527791023254
```

Şekil 30: Uygulama Çıktı-2

Modelin doğruluk ve kayıp değerlerine ilişkin grafikleri aşağıdaki gibi alınmıştır;



Şekil 31: Uygulama Grafik-1



Şekil 32: Uygulama Grafik-2

KAYNAKLAR

Ahmet, K. (2019). Uzun-Kısa Süreli Bellek Ağı Kullanarak Global Güneş Işınımı Zaman Serileri Tahmini. *Gazi University Journal of Science Part C: Design and Technology*, 7(4), 882-892.

Akca, M. F. (2020). *Nedir Bu Destek Vektör Makineleri?*, 20 Aralık 2023 tarihinde https://medium.com/deep-learning-turkiye/nedir-bu-destek-vekt%C3%B6r-makineleri-makine-%C3%B6%C4%9Frenmesi-serisi-2-94e576e4223e adresinden alındı.

Akgün, K. ve M. B. Özek. (2020). Eğitsel veri madenciliği yöntemi ile ilgili yapılmış çalışmaların incelenmesi: İçerik analizi. *Uluslararası Eğitim Bilim ve Teknoloji Dergisi*, 6(3), 200.

Akça, M. F. (2021). *Lojistik Regresyon Nedir? Nasıl Çalışır?*, 25 Aralık 2022 tarihinde https://mfakca.medium.com/lojistik-regresyon-nedir-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-4e1d2951c5c1 adresinden alındı.

Alkan, M. A. *Makine Öğrenimi Nedir?*, 19 Aralık 2023 tarihinde https://www.endustri40.com/makine-ogrenimi-nedir adresinden alındı.

Alpay, Ö. (2020). LSTM mimarisi kullanarak USD/TRY fiyat tahmini. *Avrupa Bilim ve Teknoloji Dergisi*, 452-456

Arı, A. ve D. Hanbay. (2019). Tumor detection in MR images of regional convolutional neural networks. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 34(3), 1395-1408

Arslan, H. (2008). *Sakarya Üniversitesi Web Sitesi Erişim Kayitlarinin Web Madenciliği İle Analizi*. Yüksek Lisans Tezi. Sakarya: Sakarya Üniversitesi, Fen Bilimleri Enstitüsü.

Çay, G. (2020). *Derin Öğrenme Yöntemiyle Akademik Makaleler İçin Anahtar Kelime Çikarimi*. Yüksek Lisans Tezi. Elazığ: Firat Üniversitesi, Fen Bilimleri Enstitüsü.

Çetinyamaç, E. (2021). *KNN Algoritması ve KNN Sınıflandırma Modeli Kullanılarak Kredi Onay Değerlendirilmesi*, 25 Aralık 2023 tarihinde https://erkancetinyamac.medium.com/knn-algoritmas%C4%B1-ve-knn-

s%C4%B1n%C4%B1fland%C4%B1rma-modeli-kullan%C4%B1larak-kredi-onay-de%C4%9Ferlendirilmesi-da5b06c85dda adresinden alındı.

Demirci, D. A. (2007). *Destek Vektör Makineleri İle Karakter Tanima*. Yüksek Lisans. İstanbul: Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü.

Duggal, N. (2023). *Advantages and Disadvantages of Artificial Intelligence,* 20 Ocak 2023 tarihinde simplilearn: https://www.simplilearn.com/advantages-and-disadvantages-of-artificial-intelligence-article adresinden alındı.

Dür, M. B. (2021). *Specific Keyword Extraction From Unstructured Curriculum Vitae Using Deep Learning Methods*. Yüksek Lisans. Ankara: Çankaya Üniversitesi, Fen Bilimleri Enstitüsü.

Ekici, E. (2012). *Farkli Siniflandirma Yöntemlerinin Karşilaştirilmasi Ve Bir Uygulama*. Yüksek Lisans. Elazığ: Fırat Üniversitesi, Fen Bilimleri Enstitüsü.

Erbudak, A. E. (2022). *Veri Madenciliği Ve Makine Öğrenimi İle Döviz Kuru Tahmini Uygulamasi*. Yüksek Lisans. İstanbul: Altınbaş Üniversitesi, Lisansüstü Eğitim Enstitüsü.

Ersoy, M. (2021). *TextRank-ve-RAKE-Algoritmasi-ile-Metinlerden-Anahtar-Kelime-Cikarma-Projesi*. 10 Ocak 2023 tarihinde https://github.com/NtwrkX/TextRank-ve-RAKE-Algoritmasi-ile-Metinlerden-Anahtar-Kelime-Cikarma-Projesi adresinden alındı.

Gürbüz, B. (2020). *Tekrarlayan Sinir Ağı -Recurrent Neural Networks (RNN),* 22 Ocak 2023 tarihinde https://medium.com/@batincangurbuz/tekrarlayan-sinir-a%C4%9F%C4%B1-recurrent-neural-networks-rnn-17b517dd0b3e adresinden alındı.

Karadağ, A. ve Hidayet T. (2010). *Metin Madenciliği ile Benzer Haber Tespiti*. Akademik Bilişim 2010, Muğla.

Küçük, D. ve Nursal A. (2018). Doğal Dil İşlemede Derin Öğrenme Uygulamalari Üzerine Bir Literatür Çalişmasi. *Uluslararası Yönetim Bilişim Sistemleri ve Bilgisayar Bilimleri Dergisi*, 2(2), 76-86.

Oflazer, K. (2016). Türkçe ve Doğal Dil İşleme. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, *5*(2).

Öztürk, K. ve Mustafa Ergin Ş. (2018). Yapay sinir ağları ve yapay zekâ'ya genel bir bakış. *Takvim-i Vekayi*, 6.2: 25-36.

Sayraci, B. (2021). *Derin Öğrenme Ve Evrişimsel Sinir Ağlari*, 2 Şubat 2023 tarihinde https://medium.com/deep-learning-from-deepest/deri%CC%87n-%C3%B6%C4%9Frenme-ve-evri%CC%87%C5%9Fi%CC%87msel-si%CC%87ni%CC%87r-a%C4%9Flari-446dc8a8d2f adresinden alındı.

Taş, O. (2016). *Destek Vektör Makineleri - Support Vector Machine,* 10 Ocak 2023 tarihinde https://www.slideshare.net/oguzhantas/destek-vektr-makineleri-support-vector-machine adresinden alındı.

Topuz, S. (2021). *Eğitsel Verilerde Weka Ve Orange Veri Madenciliği Yazilimlarindan Elde Edilen Analiz Sonuçlarinin Karşilaştirilması.* Yüksek Lisans Tezi. Ankara: Hacettepe Üniversitesi, Eğitim Bilimleri Enstitüsü.

Uzun, Y., Fatma Nur U. ve Esra Ç. (2021). Veri Madenciliği ve Kullanım Alanları.

Wet, Rijk. (2022). *Manhattan Distance Calculator*, 15 Şubat 2023 tarihinde https://www.omnicalculator.com/math/manhattan-distance adresinden alındı.

Yıldırım, E. (2020). *Yapay Sinir Ağı (Artificial Neural Network) Nedir?*, 18 Şubat 2023 tarihinde https://www.veribilimiokulu.com/yapay-sinir-agiartificial-neural-network-nedir/ adresinden alındı.

Yildirim, P., Mahmut U. ve Abdulkadir G. (2008). Hastane Bilgi Sistemlerinde Veri Madenciliği. *Çanakkale On Sekiz Mart Üniversitesi Akademik Bilişim*.

Yıldırım, Y. (2018). *Rassal Orman Algoritması,* 23 Aralık 2022 tarihinde https://yavuz.github.io/rassal-orman/ adresinden alındı.

Yıldız, A. M. (2022). Keyword Extractıon Wıth Textrank And Tfıdf From Turkısh Articles. *International Informatics Congress*.

(2019). *Big Data (Büyük Veri) Nedir?*, 15 Aralık 2022 tarihinde https://medium.com/dusunenbeyinler/big-data-b%C3%BCy%C3%BCk-veri-analizi-d53d8f8ab52b adresinden alındı.

(2021). *Makine Öğrenmesi (Machine Learning) Nedir*, 15 Aralık 2023 tarihinde https://blog.turhost.com/makine-ogrenmesi-machine-learning-nedir adresinden alındı.

(2021). *Metin Madenciliği Nedir*, 18 Aralık 2022 tarihinde http://www.metinmadenciligi.com adresinden alındı.

(2021). *Makine Öğrenmesi (Machine Learning) Nedir?*, 19 Aralık 2023 tarihinde https://blog.turhost.com/makine-ogrenmesi-machine-learning-nedir adresinden alındı.