
GAZİ ÜNİVERSİTESİ
Gazi
1926

SAĞLIK BİLİMLERİ ARAŞTIRMALARINDA KULLANILAN İSTATİSTİKSEL YÖNTEMLERİN METİN MADENCİLİĞİ İLE İNCELENMESİ

Özen TAŞTAN

YÜKSEK LİSANS TEZİ
İSTATİSTİK ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ŞUBAT 2024

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Özen TAŞTAN
07/02/2024

SAĞLIK BİLİMLERİ ARAŞTIRMALARINDA KULLANILAN İSTATİSTİKSEL YÖNTEMLERİN METİN MADENCİLİĞİ İLE İNCELENMESİ

(Yüksek Lisans Tezi)

Özen TAŞTAN

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Şubat 2024

ÖZET

Sağlık bilimlerinde yapılan bilimsel araştırmalar diğer birçok alanda olduğu gibi genellikle örneklem üzerinde yürütülmektedir ve örneklemden elde edilen istatistikler kullanılarak popülasyonun parametrelerine ilişkin çıkarımlar yapılmaktadır. Bu çıkarımlar istatistiksel yöntemler aracılığı ile yapılmaktadır. Araştırmacılar çalışmalarında, kullandıkları istatistiksel yöntemleri "İstatistiksel Analizler" başlığı altında sunmaktadırlar. Çalışmada metin madenciliği yöntemi kullanılarak sağlık bilimleri alanında seçilen ulusal ve uluslararası dergilerde yayınlanmış araştırma makalelerinde kullanılan istatistiksel yöntemler tespit edilmiştir. Sağlık bilimleri alanındaki ulusal dergilerde yayınlanan makaleler için DergiPark platformu ve sağlık bilimleri alanındaki uluslararası dergilerde yayınlanan makaleler için ise PubMed platformu kullanılmıştır. Makale veri setlerine ulaşmak amacıyla DergiPark ve PubMed üzerinden bir robotik süreç otomasyon yazılımı olan *UiPath* kullanılarak makalelere ait *PDF* dosyalarına erişilmiş ve kaydedilmiştir. Robota DergiPark'tan "istatistiksel analiz" içeren, 2023 yılına ait sağlık bilimleri dergilerinde yayınlanan makaleler indirtilmiştir. PubMed platformundan ise "statistical analyses" içeren, makale tipi "clinical trial" olan ve 2023 yılına aile sağlık bilimleri dergilerinde yayınlanan makaleler indirtilmiştir. Robot yardımıyla indirilen makalelere ait *PDF* dosyaları *Python* kodu yazılarak açılmıştır ve metin koleksiyonu oluşturulmuştur. Kodlama, yazım dili Türkçe ve İngilizce olan metin verileri için ayrı ayrı yapılmıştır. Açılan her makale dosyasındaki harfler küçük harfe dönüştürülmüştür. Kod içerisinde aratılacak olan 72 adet istatistiksel yöntem ve 12 adet programı içeren, yazım dili Türkçe ve İngilizce olan, iki adet *PDF* dosyası oluşturulmuştur. Oluşturulan dosyalardaki yöntemler ve programlar tüm makalelerde aratılmıştır ve bulunup bulunmamasına göre "1" ve "0" olarak kodlanmıştır. Türkçe ve İngilizce metinlerde kullanılan istatistiksel yöntemlerin sayıları ayrı ayrı toplanarak bir sonuç verisi oluşturulmuştur. Bu sonuca göre ulusal ve uluslararası makalelerde kullanılan istatistiksel yöntemlerin ve programların kullanılma sıklıkları kıyaslanmıştır. Hem ulusal hem uluslararası dergilerde yayınlanan makalelerdeki istatistiksel yöntemler benzerlik göstermiştir. Tek değişkenli analizlerin çok değişkenli analizlere kıyasla oldukça fazla olduğu görülmüştür.

Bilim Kodu	:	20514
Anahtar Kelimeler	:	Metin madenciliği, bilgi çıkarımı, istatistiksel yöntemler
Sayfa Adedi	:	81
Danışman	:	Prof. Dr. Bülent ÇELİK

INVESTIGATION OF STATISTICAL METHODS USED IN HEALTH SCIENCES RESEARCH USING TEXT MINING

(M. Sc. Thesis)

Özen TAŞTAN

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

February 2024

ABSTRACT

Scientific research in health sciences, as in many other fields, is generally conducted on samples, and inferences are made regarding the parameters of the population using statistics obtained from the sample. These inferences are made through statistical methods. Researchers present the statistical methods they use in their studies under the title "Statistical Analyses". In the study, statistical methods used in research articles published in selected national and international journals in the field of health sciences were determined by using the text mining method. The DergiPark platform was used for articles published in national journals in the field of health sciences, and the PubMed platform was used for articles published in international journals in the field of health sciences. In order to access the article data sets, *PDF* files of the articles were accessed and saved using *UiPath*, a robotic process automation software, via DergiPark and PubMed. Articles containing "statistical analysis" and published in health sciences journals dating back to 2023 were downloaded to the robot from DergiPark. Articles containing "statistical analyses", article type "clinical trial" and published in family health sciences journals in 2023 were downloaded from the PubMed platform. *PDF* files of the articles downloaded with the help of the robot were opened by writing *Python* code and a text collection was created. Coding was done separately for text data written in Turkish and English. Letters in each opened article file have been converted to lowercase. Two *PDF* files, written in Turkish and English, were created, containing 72 statistical methods and 12 programs to be searched in the code. The methods and programs in the created files were searched in all articles and coded as "1" and "0" depending on whether they were found or not. A result data was created by adding the numbers of statistical methods used in Turkish and English texts separately. According to this result, the frequency of use of statistical methods and programs used in national and international articles was compared. Statistical methods in articles published in both national and international journals were similar. It has been observed that univariate analyzes are quite common compared to multivariate analyses.

Science Code	:	20514
Key Words	:	Text mining, information extraction, statistical methods
Page Number	:	81
Supervisor	:	Prof. Dr. Bülent ÇELİK

TEŞEKKÜR

Bu çalışmanın belirlenmesi, araştırılması, hazırlanması ve sonuçlanması sürecinde bilgisiyle bana yol gösteren, her zaman destek veren değerli danışman hocam Prof. Dr. Bülent ÇELİK'e sonsuz saygılarımı ve teşekkürlerimi sunarım. Ayrıca bu süreçte her açıdan desteğini eksik etmeyen aileme ve arkadaşım Yasemin ÇİFÇİ'ye verdikleri motivasyon ve desteklerinden dolayı teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

RESİMLERİN LİSTESİ

Resim **Sayfa**

Resim **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
GLMM	Generalized Linear Mixed Model
IDF	Inverse Document Frequency
PDF	Portable Document Format
RPA	Robotic Process Automation
regex	Regular Expressions
TF	Term Frequency
URL	Uniform Resource Loader

1. GİRİŞ

Bilgi keşfi (knowledge discovery), bilgi madenciliği (knowledge mining) ve makine öğrenmesi (machine learning) gibi adlarla da anılan veri madenciliği (data mining) büyük veri yapılarından değerli bilgilerin çıkarılması süreci/yöntemleri olarak tanımlanabilir (Altunkaynak, 2019: 13). Veri madenciliği bilgi teknolojisinin doğal gelişiminin bir sonucu olarak görülebilir. Son 50 yıllık süreçte verinin dijital ortamda depolanma kapasitesi muazzam bir hızla artmıştır. Bunun bir sonucu olarak 1970'li yıllardan başlayarak veri tabanı yönetim sistemleri geliştirilmiş ve bunun devamında elde edilen verilerin işlenmesi ve bilgiye dönüştürülmesine yönelik yaklaşımların ortaya çıkmasına neden olmuştur (Han, Kamber ve Pei, 2011). Veri madenciliğine ilişkin yöntemler veri tabanı yönetim sistemlerinin oluşturulmasından sonra ortaya çıkmıştır. Veri tabanı yönetim sistemleri sayesinde çok sayıda veri bir araya getirilerek veri tabanları oluşturulmuştur. Veri tabanlarında istenilen kayıtların ortaya çıkarılabilmesi için yapısal sorgulama dili olan *SQL* geliştirilmiştir.

Veriden yararlı bilgileri çıkarabilmek için 1980'li yılların sonlarından itibaren veri madenciliği yöntemleri uygulanmaya başlanmıştır. Veri madenciliği yöntemleri aktif bilgi girişi ve çıkışının yapıldığı, gerekli gereksiz her türlü bilginin bulunduğu veri tabanlarına doğrudan uygulanamamaktadır. İlk olarak yararlı bilgilerin çıkarılacağı (madenciliğin uygulanacağı) alt veri kümesi veri tabanından süzülür ve statik hale (veri giriş çıkışının olmaması, gerekli dönüşümlerin yapılmış olması vb.) getirilmektedir. Veri madenciliği yöntemlerinin uygulanacağı bu alt veri kümesine "veri ambarı" denir. Veri madenciliği yöntemleri veri ambarı üzerinden uygulanmaktadır (Altunkaynak, 2019: 14).

Veri madenciliği yöntemleri arasında sınıflandırma, kümeleme, birliktelik kuralları, regresyon gibi klasik yöntemlerin dışında metin madenciliği, web madenciliği ve sosyal ağ analizleri gibi ileri seviye veri madenciliği yöntemleri de bulunmaktadır.

Veri madenciliği yöntemleri saha araştırmaları, klinik araştırmalar ve tanı koyma, müşteri ilişkileri yönetimi, gen analizleri, sahtecilik ve dolandırıcılık tespiti gibi birçok alanda kullanılmaktadır. Metin madenciliği ise dokümanlar arası benzerlik ve karşılaştırma, klinik metinlerden bilgi çıkarılarak teşhis koyma, bir ürünün yorumlarının değerlendirilmesi,

kitaplarda geçen tüm kelimeleri veri seti kabul ederek yazar tanımlama, metin içerikli tüm verilerin sınıflandırılması, duygu analizi, metin özetleme gibi birçok alanda kullanılmaktadır.

Metin madenciliğinde veri kaynağı metinlerden oluşmaktadır ve bu metinlerden yapılandırılmış veri oluşturmak amaçlanmaktadır. Günümüzde tüm bilgiler (klinik, hukuk vb.) metin dokümanları olarak tutulmaktadır. İnternet üzerinden dijital kütüphaneler, depolar ve bloglar, sosyal medya ağı ve e-postalar gibi büyük miktarda metin verisi bulunmaktadır (Sagayam, 2012). Bu sebeple metin madenciliğinin varlığı metin olarak tutulmuş tüm büyük hacimli verilerin analizini önemli ölçüde kolaylaştırmıştır. Bu büyük hacimli verilerden değerli ve faydalı bilgiler elde etmek için uygun modelleri belirlemek ise oldukça zordur (Padhy, 2012). Klasik veri madenciliği yöntemleri ile bilgi toplamak fazla zaman ve çaba gerektirdiğinden, bu araçlar metinsel verileri işlemede yetersiz kalmaktadır (Talib, 2016). Bu nedenle metin madenciliği, belirli bir formatı olmayan metinsel verileri işlemek için geliştirilen genellikle doğal dil işleme teknikleriyle birlikte uygulanmaktadır. Doğal dil işleme çalışmaları daha çok yapay zekâ altındaki dilbilim bilgisine dayalı çalışmaları kapsamaktadır. Metin madenciliği ise daha çok istatiksel olarak metin üzerinden sonuçlara ulaşmayı hedeflemektedir. Metin madenciliği çalışmaları sırasında çoğu zaman doğal dil işleme kullanılarak özellik çıkarımı da yapılmaktadır (Şeker, 2015).

Bu çalışmada Türkçe ve İngilizce metinlerde kullanılan istatistiksel analiz yöntemleri tespit edilip, makalelerde kullanılma sıklıklarına göre karşılaştırılmıştır.

Tez 8 bölümden oluşmaktadır. Giriş bölümünden sonra ikinci bölümde metin madenciliği ile ilgili literatür araştırmasına yer verilmiştir. Üçüncü bölümde, veri tabanı, veri ambarı ve veri madenciliği tanımı, uygulama alanları, yöntemleri açıklanmıştır. Dördüncü bölümde çalışmada kullanılan veri madenciliği yöntemlerinden birisi olan metin madenciliği kavramına, uygulama alanlarına, yöntemlerine ve uygulanabileceği yazılım programlarına değinilmiştir. Beşinci bölümde robotik süreç otomasyonu süreçlerine değinilmiştir. Altıncı bölüm olan uygulama bölümünde, çalışmada kullanılan *Python* programı ve kütüphaneleri, verinin elde edilmesi, veri koleksiyonunun oluşturulması ve kodlama aşamaları belirtilmiştir. Yedinci bölümde, uygulama sürecinin açıklandığı bulgular bölümü yer almaktadır. Sekizinci bölümde ise uygulama sonucu karşılaştırmalara ve diğer çalışmalara önerilere sonuç ve öneriler bölümünde değinilmiştir.

Çalışmanın konusu ve amacı

Bu çalışmada metin madenciliği yöntemi ile sağlık bilimleri alanında seçilen ulusal ve uluslararası dergilerde Ocak 2023-Kasım 2023 tarihleri arasında yayınlanmış araştırma makalelerinde kullanılan istatistiksel yöntemler tespit edilmiş ve karşılaştırmalar yapılmıştır. Sağlık bilimleri alanındaki ulusal dergilerde yayınlanan makaleler için DergiPark platformu ve sağlık bilimleri alanındaki uluslararası dergilerde yayınlanan makaleler için ise PubMed platformu kullanılmıştır. Makale veri setlerine ulaşmak amacıyla DergiPark ve PubMed üzerinden bir robotik süreç otomasyon yazılımı olan *UiPath* kullanılarak makalelere ait *PDF* dosyalarına erişilmiş ve kaydedilmiştir. Robota DergiPark platformundan "istatistiksel analiz" içeren, 2023 yılına ait sağlık bilimleri dergilerinde yayınlanan makaleler indirtilmiştir. PubMed platformundan ise "statistical analyses" içeren, makale tipi "clinical trial" olan ve 2023 yılına aile sağlık bilimleri dergilerinde yayınlanan makaleler indirtilmiştir. Kodlama içerisinde kullanılacak olan istatistiksel yöntemler dosyaları, yazım dili Türkçe ve İngilizce olan metin veri setleri için ayrı ayrı oluşturulmuş ve kod blokları ayrı ayrı çalıştırılmıştır. Robot yardımıyla indirilen makalelere ait *PDF* dosyaları *Python* kodu yazılarak açılmıştır ve metin koleksiyonu oluşturulmuştur. Açılan her makale *PDF* dosyasına metin madenciliği ön işleme adımları uygulanmıştır. Oluşturulan yöntemler dosyasındaki tüm istatistiksel yöntemler makalelerde aratılmıştır ve yöntemin olup olmamasına göre "0" ve "1" olarak kodlanmıştır. Türkçe ve İngilizce metinlerde kullanılan istatistiksel yöntemlerin sayıları ayrı ayrı toplanarak bir sonuç verisi oluşturulmuştur. Bu sonuca göre ulusal ve uluslararası makalelerde kullanılan istatistiksel yöntemlerin kullanılma sıklıkları karşılaştırılmıştır.

2. LİTERATÜR

Gün geçtikçe verinin toplanma şekline bağlı olarak veri madenciliğine daha çok ihtiyaç duyulmaktadır ve veri madenciliği yöntemlerine ilişkin birçok çalışmalar yapılmaktadır. Veri madenciliği alt alanı olan metin madenciliği ile ilgili de birçok çalışma yapılmıştır ve hala devam etmektedir.

Anadolu Ajansı haberlerinden metin veri setini oluşturmuş ve bu haberleri türlerine göre kategorilerine ayıran bir program yazılmıştır. Program *Visual Studio.Net* ile yazılmış ve veri tabanı olarak *Sql Server* kullanılmıştır. Program arayüzünde sözlük, eğitim dokümanları, sınıflandırılmış dokümanlar ve kategoriler gibi bölümler yer almaktadır. Metin veri setine; metin ön işleme adımlarından noktalama işaretlerinin kaldırılması, küçük harf dönüşümü, kategorilerin belirlenmesi, sözlük oluşturulması, joker (wild card) olarak kullanılabilecek kelimelerin bulunması, kelime değerlerinin bulunması adımları uygulanmıştır. Çalışmada sınıflandırma için naive bayes ve k-en yakın komşuluk algoritmaları kullanılmış ve bu iki algoritmanın performansları karşılaştırılmıştır. Sınıflandırma sonuçlarına göre doğruluk oranı en yüksek olan algoritma naive bayes olarak bulunmuştur (Pilavcılar, 2017).

Dokümanlar arasındaki benzerlikleri metin madenciliğini *.Net* platformu üzerinde bir masaüstü uygulaması geliştirerek yapılmıştır. Uygulama birden fazla doküman girdi olarak verilerek, aralarındaki benzerlikleri bulacak şekilde geliştirilmiştir. Uygulamaya yüklenen doküman içerisinde bulunan tüm cümleler diğer dokümanlar içerisinde aratılarak kartezyen çarpımı ile benzerlikleri hesaplanmıştır. Dokümanlar üzerinde metin madenciliği ön işleme adımlarından olan noktalama işaretlerinin ve anlamsız kelimelerin (ama, ile, çünkü, ve, veya, …) kaldırılması işlemleri yapılmıştır. Sonuç kısmında karşılaştırma yapılan cümle sayısı ve uygulamanın çalışmanın toplam süresi gösterilmiştir. Uygulama için metin madenciliği benzerlik hesaplama algoritmaları olan kosinüs ve jaccard algoritmaları kullanılıp başarıları test edilmiştir. Uygulama için Namık Kemal'in hayat hikayesi araştırılmış ve 10 adet doküman *PDF* ve *txt* formatında kaydedilmiştir. Farklı sitelerden alınan bilgiler uygulamaya girdi olarak verilerek dokümanlar arasındaki benzerlik incelenmiştir. Yapılan uygulama sonucunda cümlelerin karşılaştırılmasında jaccard algoritmasının daha başarılı olduğu sonucuna varılmıştır (Döven, 2013).

E-ticaret sistemlerinde bulunan ürünlere yapılan yorumları metin madenciliği ile türlerine göre sınıflanmıştır. Metin madenciliği için *Monkey Learn* web sitesi kullanılmıştır. Sınıflandırma algoritması olarak bayes teoremi kullanılmış ve olasılığın sıfır çıkma ihtimalinin önlenmesi için ise laplace dönüşümü yapılmıştır. Sınıflandırma sonucunda büyük veriler daha anlamlı ve küçük verilere dönüştürülmüş, problem yaşayan kullanıcıların belirlenmesinde yöntem olarak kullanılmıştır (Yılmaz, 2021).

İnternet sitelerinin içeriğini inceleyerek metin madenciliği ile bu internet sitelerinin e-ticaret sitesi olup olmadığının tespit edileceği bir *Java* uygulaması geliştirilmiştir. Bir kısmı e-ticaret sitelerinden oluşan bir eğitim kümesi oluşturulmuştur. Bu eğitim veri setine metin madenciliği ön işleme adımlarından olan anahtar kelime sözlüğü ve vektör uzay modeli oluşturularak veri seti eğitilmeye hazır hale getirilmiştir. Eğitim verisi denetimli öğrenme tekniklerinden olan k-en yakın komşu ve naive bayes algoritmalarıyla eğitilmiştir. Bu iki algoritmanın eğitim verisindeki başarı sonuçları karşılaştırılmış ve model oluşturulmuştur. Kullanıcıların sitelerin e-ticaret sitesi olup olmadığını test edebileceği *Java* tabanlı bir arayüz programı yazılmıştır. Sonuçlara göre kullanılan metin sınıflandırma algoritmalarından naive bayes algoritmasının daha iyi sonuç verdiği tespit edilmiştir (Kaşıkçı ve Gökçen, 2014).

Yazarı belli olmayan, anonim İngilizce metinlerin yazarının tahmin edilmesi üzerine çalışılmıştır. Farklı yazarların toplam 24 metninden oluşan metin veri seti oluşturmuştur. Oluşturulan bu veri setini metin madenciliği ile inceleyerek metinler hakkında bilgi çıkarımı, yazarların özelliklerinin çıkarılması ve sınıflandırma yapmış ve sınıflandırma yöntemlerinin başarıları karşılaştırılmıştır. Metin madenciliği metin ön işleme adımlarından olan noktalama işaretlerinin kaldırılması, anlamsız kelimelerin silinmesi ve tüm harflerin küçük harfe dönüştürülmesi adımları *R Studio* paket programı ile yapılmıştır. Sık tekrar eden kelimeler, zarflar ve sıfatlar bulunmuştur. Yazarlara ait özellikler tespit edilip, ward kümeleme algoritması ve k-en yakın komşu algoritmasıyla sınıflandırılmıştır. Çalışma sonucu k-en yakın komşu algoritmasının doğruluk oranı %85 ile bu çalışma için en iyi algoritma olarak bulunmuştur (Çam, 2019).

Türkçe edebi ve sanat metinlerindeki kelimelerin birlikte kullanılma sıklıklarını birliktelik analizi ile incelenmiştir. Metinlerden elde edilen 78112 kelimeden oluşan bir veri tabanı oluşturulmuştur. Uygulamada *Java* programlama dili ve *Visual Basic* platformu

kullanılmıştır. Veri tabanının oluşturulması için ise *Microsoft SQL Server* ve *Microsoft Office Access* programları kullanılmıştır. Kelimelerin köklerinin bulunması için "zemberek" algoritması kullanılmıştır. Kelimelerin birlikte kullanılmalarına bakılarak kelimelerin gerçek anlam bilgisi tahmin edilmiştir. Çalışmada 30 adet kelimenin birliktelik analizi sonuçlarına göre, uygulamanın kelimelerin anlamlarını doğru tahmin etme oranı %86 olarak belirlenmiştir (Bayer, 2011).

Metin madenciliği ile yazılım geliştirme taleplerinin sınıflandırılmasını incelemiş, sınıflandırma algoritmaları karşılaştırılmıştır. Veri setinde özel bir firmanın müşterilerinin taleplerinin başlığı, açıklaması ve sınıfı olmak üzere 3 kolon bulunmaktadır. Metin veri setinde ön işleme adımları *C#* ile geliştirilen uygulama ile yapılmıştır. *PRETO* uygulaması ile ön işlemlerden elde edilmiş veriden doküman terim matrisi hazırlanmış ve *C#* programlama dilinde *WEKA* uygulamasında kullanılacak hale getirilmiştir. Son olarak veriler, *WEKA* uygulamasında makine öğrenmesi algoritmaları ile test edilmiştir. Naive bayes, naive bayes multinomial ve rasgele orman sınıflandırma algoritmaları karşılaştırılmıştır. Çalışma sonucunda en iyi sınıflandırma algoritması rasgele orman olduğu gözlemlenmiştir (Tekin, 2018).

Metin madenciliği kullanarak sağlık alanındaki dokümanlardan bilgi çıkarmayı amaçlanmıştır. PubMed veri tabanından insanlardaki kanser vakaları ile ilgili yapılan çalışmalar ve fareler üzerinde yapılan kanser çalışmalarını içeren araştırmalara ilişkin dokümanlar elde edilmiştir. Elde edilen yapısal olmayan dokümanlar *Knime* paket programı kullanılarak metin madenciliği teknikleriyle yapısal hale dönüştürülmüştür. Metin ön işleme adımlarında noktalama işaretlerinin kaldırılması, ondalık ayraçların kaldırılması, harflerin küçük harfe dönüştürülmesi, kelimelerin köklerinin bulunması işlemleri yapılmıştır. Terim frekansları hesaplanarak en sık kullanılan kelimeler tespit edilmiştir. Eğitim ve test veri setleri oluşturulmuştur. K-en yakın komşu algoritması ile sınıflandırma yapılmıştır. Sonuç olarak k-en yakın komşu algoritmasının doğruluk oranı %59,8 olarak belirlenerek kısmen başarılı bir sınıflama yapılmıştır (Toplu, 2019).

Metin madenciliği yöntemleri ile 25 divan şairinin eserleri veri seti kabul edilerek yazarı bilinmeyen divan edebiyatı şiirleri için yazar tanıma uygulaması geliştirmiş ve uygulamanın doğruluğunu test edilmiştir. Çalışmada *WEKA*, *Java* ve *Rapidminer* programları kullanılmıştır. Metin ön işleme adımında tüm harflerin küçük harfe dönüştürülmesi, kelime

uzunluğu 3'ten az olan kelimelerin silinmesi işlemleri yapılmıştır. Naive bayes, k-en yakın komşu, karar ağacı, destek vektör makinesi, kosinüs benzerlik ölçütü, çok terimli naive bayes sınıflandırma yöntemleri kullanılmıştır. En yüksek doğruluk oranına sahip çok terimli naive bayes algoritması olarak belirlenmiştir (Bilgin, 2018).

Ülkemizdeki spor bilimleri alanında yayınlanmış olan makalelerde sıklıkla kullanılan istatistiksel yöntemleri belirlemiş ve bu yöntemlerin doğru ya da yanlış kullanımı incelenmiştir. DergiPark platformunda "fiziksel aktivite", "fiziksel uygunluk", "egzersiz", "vücut komposizyonu" ve "antrenman" içeren, 2008-2017 yılları arasındaki makaleleri seçip SPSS programı yardımıyla, 221 makalede kullanılan yöntemlere ilişkin istatistiksel analizler gerçekleştirilmiştir. Analiz sonucu bağımsız örneklem *t*-testinin en sık kullanılan istatistiksel yöntem olduğu sonucuna ulaşılmıştır (Kamuk, 2020).

İran Tıp Arşivi'nde yayınlanan 423 makale için istatistiksel yöntemlerin incelenmesi ve makale kabul süresine olan etkisini araştırılmıştır. Analiz sonucunda, istatistiksel yöntemler arasında en sık tanımlayıcı istatistiklerin olduğu ve söz konusu değişkenlerin kabul süresi üzerinde anlamlı bir etkisinin olmadığı sonucuna ulaşılmıştır (Madadizadeh, Bahariniya, 2022).

1631 doküman için *Scopus* programı ile bibliyometrik analiz yapılmıştır. Analiz sonucu makale başına ortalama alıntı sayısı 14,78 ve medyanı ise 3 olarak bulunmuştur. En sık atıf yapılan makale, *Journal of Urban Technology* dergisinde yayınlanan, 1276 kez atıf yapılan ve alan ağırlıklı atıf etkisi 13,22 olan "*Smart cities in Europe*" makalesi olarak tespit edilmiştir (Sgambati, Gargiulo, 2022).

Lojistik ve tedarik dergilerinde büyük veri yöntemlerinin kullanıldığı yaklaşık 2,000 makalenin olduğu, bu makalelerden 43 adedi bilimsel araştırma çalışmalarının konusunu anlattığı sonucuna varılmıştır. Büyük veri yöntemlerinin ve analizlerinin, lojistik ve tedarik dergilerinde seyrek olarak kullanıldığı gözlemlenmiştir. Yapılan trend analizi sonucuna göre ise lojistik ve tedarik dergilerinde büyük veri yöntemlerinin kullanılmasının her yıl arttığı gözlemlenmiştir (Yudhistyra, Risal, Raungratanaamporn, Ratanavaraha, 2020).

Covid-19 salgınıyla birlikte trakeostomi kullanımına ilişkin bilimsel makalelerin özetlenmesini amaçlanmıştır. *Web of Science* veri tabanından elde edilen, 1980-2021 yıları

arasındaki trakeostomi ile ilgili yayınlanmış 3573 makale bibliyometrik ve istatistiksel yöntemler ile analiz edilmiştir. Sonuçlara göre ilk üç ülke arasında ABD, İngiltere ve Almanya, ilk üç kurum arasında Harvard Üniversitesi, Michigan Üniversitesi ve Pennsylvania Üniversitesi, en fazla yayına sahip ilk üç dergi arasında *Laryngoscope, International Journal of Pediatric Otorhinolaryngology* ve *Otolaryngology-Head and Neck Surgery* yer almaktadır. Makale başına ortalama atıf sayısına göre en etkili üç dergi arasında ise *Chest, Critical Care Medicine* ve *Journal of Trauma-Injury Infection and Critical Care* dergileri bulunmaktadır (Ülger ve Baldemir, 2022).

R paket programından *JATSdecoder* kullanılarak metin madenciliği yöntemini bilimsel raporlara uygulanmıştır. "Get.stats()" fonksiyonunun istatistiksel sonuç çıkarma, yeniden hesaplama ve kontrol etme işlemleri "statcheck" ile yapılmıştır. 49 makaledeki istatistiksel olarak anlamlı sonuçları içeren manuel veri seti oluşturulmuştur. "Statcheck" ve "get.stats()" fonksiyonları karşılaştırılmıştır ve sonuç olarak "get.stats()" fonksiyonunun, "statcheck" fonksiyonundan daha iyi performans gösterdiği belirlenmiştir (Böschen, 2021).

Türkiye'de eğitimde ölçme ve değerlendirme alanında yapılan bilimsel yayınlar bibliyometrik yöntemlerle incelenmiş, yayın üretiminin ve yazar/atıf ilişkilerinin niceliğini ve niteliğini artırabilecek bulguların tespit edilmesi amaçlanmıştır. 2006-2015 yıları arasında eğitimde ölçme ve değerlendirme alanında yayımlanmış *Web of Science* veri tabanında taranan Türkiye adresli 205 makale ve 51 tam metin üzerinde çalışılmıştır. Yayınlar sosyal ağ analizi yöntemi kullanılarak, bilgi yapısı, ilişkiler ve eğilimler görselleştirilmiş ve ortaya çıkan ağın yapısı ile ilgili istatistiksel bilgiler (frekans, yüzde vb.) verilmiştir. Ortak atıf ve ortak kelime analizleri yapılarak eğilimler tespit edilmiştir (Ukşul, 2016).

Arap literatüründe yer alan yaygın olarak kullanılan istatistiksel yöntemleri incelenmiştir. 2014-2018 yıları arasında 8 adet Arap dergisinde yayınlanan makalelerin incelenmesi sonucunda en sık kullanılan yöntemler arasında tanımlayıcı yöntemler ve tahmine dayalı yöntemler yer almaktadır. En fazla parametrik yöntem *t*-testi ve anova iken, parametrik olmayan yöntem ise ki-kare yöntemi olarak tespit edilmiştir (Ibrahim, 2021).

Bu çalışmada ise metin madenciliği kullanılarak sağlık bilimleri alanında ulusal ve uluslararası seçilen dergilerde yayınlanmış araştırma makalelerinde kullanılan istatistiksel yöntemler tespit edilmiş ve karşılaştırma yapılmıştır. Sağlık bilimleri alanındaki ulusal

dergilerde yayınlanan makaleler için DergiPark platformu ve uluslararası sağlık bilimleri dergilerinde yayınlanan makaleler için ise PubMed platformu kullanılmıştır. Bu çalışma DergiPark ve PubMed platformlarından çok sayıda makale *PDF*'lerine *RPA* robotu yardımıyla kolayca erişim sağlanabilmesi, *PDF*'lerin kaydedilmesi ve *PDF* içerisinden bilgi çıkarım aşamalarının bulunduğu, bu alanda çalışma yapmak isteyen araştırmacıların yararlanabileceği bir çalışmadır. Yararlanılabilecek konular aşağıdaki gibidir:

-DergiPark platformundaki makalelerin sadece başlık, yazar, özet, yıl, anahtar kelimeleri gibi bilgilerine *.Net, Python* gibi programlama dillerinden isteklerin yapılabilmesi için "request" ile erişim sağlanabilmektedir. Fakat makale *PDF*'lerine erişim direkt olarak sağlanamamaktadır. Bu çalışmada DergiPark makale *PDF*'lerine erişim *RPA* robotu ile otomatize edilmiştir.

-PubMed platformundaki makalelerin *PDF*'lerine "request" ile direkt erişim sağlanamamaktadır. Bu çalışmada PubMed platformundaki makale *PDF*'lerine yine *RPA* robotu sayesinde erişilip otomatize edilmiştir.

-Çalışma ihtiyacına göre *UiPath* süreci içerisinde Browser, *URL*, filtreler değiştirilip süreç farklı çalışmalara uyarlanabilmektedir.

-Yazılan *Python* kodu ile dosya içerisindeki birden fazla *PDF* açılabilir, *PDF*'ler içerisinde arama yapılabilir, aratılan bilginin olup olmamasına göre "1" ve "0" yazdırılabilir ve sonuç verisi oluşturulabilmektedir.

3. VERİ MADENCİLİĞİ METODOLOJİSİ

Veri madenciliği bilgi teknolojisinin doğal gelişiminin bir sonucu olarak görülebilir. Son 50 yıllık süreçte verinin dijital ortamda depolanma hızı gittikçe artmıştır. Bunun bir sonucu olarak 1970'li yıllarından başlayarak veri tabanı yönetim sistemleri geliştirilmiş ve bunun devamında elde edilen verilerin işlenmesi ve bilgiye dönüştürülmesine yönelik yaklaşımların ortaya çıkmasına neden olmuştur (Han, Kamber ve Pei, 2011).

Veriden yararlı bilgileri çıkartabilmek için 1980'li yılların sonlarından itibaren veri madenciliği yöntemleri uygulanmaya başlanmıştır. Veri madenciliğinde ilk iş olarak madenciliğin uygulanacağı alt veri seti veri tabanından süzülür ve statik hale (veri giriş çıkışının olmaması, gürültüden arındırılmış olması, veriye uygun dönüşüm işlemlerinin yapılması gibi) getirilir. Veri madenciliğinin uygulanacağı bu alt veri setine veri ambarı denir (Altunkaynak, 2019: 14).

3.1. Veri Tabanları

Veri tabanı verilerin depolanabilir, yönetilebilir, güncellenebilir ve taşınabilir olduğu bir bilgi kümesidir. Veriyi belirli şekilde kategorize etmeyi ve verilerle ilgili ilişkiler oluşturmayı amaçlayan sistemlerdir. Amaç veriye hızlı ve doğru erişmek, verinin güvenlik kurallarını belirlemek, veriyi en az yer kaplayacak şekilde depolamaktır. Verinin işlendiği tüm alanlarda (bankalar, havayolları, üretim, pazarlama, satış vb.) kullanılmaktadır.

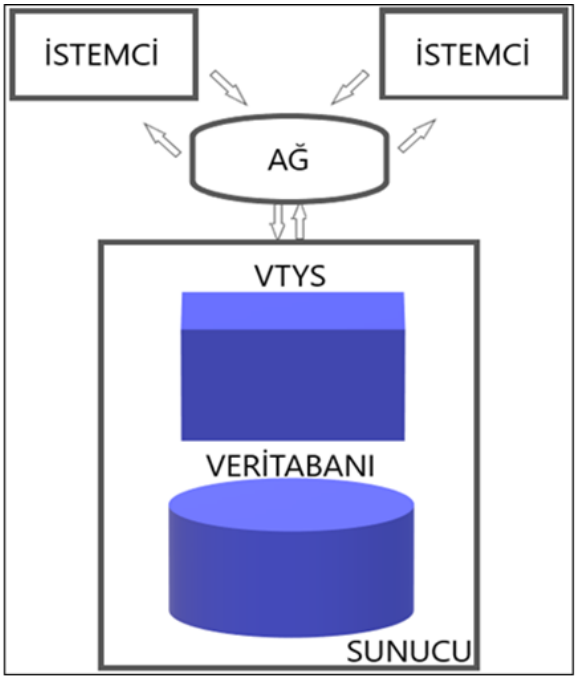
Veri tabanlarında istenilen kayıtların elde edilmesi için yapısal sorgulama dilleri (*SQL* gibi) geliştirilmiştir. Gerekli ve gereksiz her veri tutulmamaktadır. Benzer özelliklere sahip veriler gruplanarak tablolar üzerinde tutulur ve bu tablolar birbirleriyle ilişkilendirilerek gerekli bilgiye ulaşılmaktadır.

Veri tabanı türlerinden en sık kullanılan ilişkisel veri tabanlarıdır. İlişkisel veri tabanları büyük verinin daha az alanda depolanmasını ve veriye hızlı ve doğru erişilmesini sağlamaktadır.

3.2. Veri Tabanı Yönetim Sistemleri

Veri tabanı yönetim sistemleri, veri tabanının tanımlanması, oluşturulması, sürdürülmesi, saklanması ve yönetilmesi için kullanılan bir yazılım sistemidir. Günümüzde *Microsoft Access, SQL Server, Oracle, MySQL, PostgreSQL, Sybase* gibi birçok veri tabanı yönetim sistemleri bulunmaktadır.

Veri tabanı mimarileri ilk olarak 1960'lı yıllarda "ağ modeli" ile geliştirilmiştir. Bu mimari, veriler arasındaki bağıntıyı bir ağ yapısı ile kurarak verilere bu ağ yapısı aracılığıyla bağlantılı bir erişim sağlamaktaydı. Ağ modelinde temel veri yapısı küme kurgusuna dayanmaktaydı. Aynı yıllarda *IBM* tarafından "sıradüzen model" olarak adlandırılan alternatif bir model geliştirilmiştir. Bu model ile veriler arasındaki bağıntılar bir sıradüzen içerisinde oluşturulmuştur. Veriler içerisinde üst (parent) ve ast (child) veri parçacıkları olan bir ağaç yapısı içinde organize edilmektedir. Bu yapıya göre ast veri gruplarındaki bir kayıt, tekrarlı bilgiye sahip olabilir. 1970'li yıllarda bu modellerin üzerine "ilişkisel model" geliştirilmiştir ve ticari ürün olarak piyasaya sürülmüştür. Günümüzde "nesneye yönelik veri modeli" ve birçok model geliştirilmiş olmasına rağmen ilişkisel veri modeli halen yaygın olarak kullanılmaktadır (Çağıltay ve Tokdemir, 2010: 14, 16).

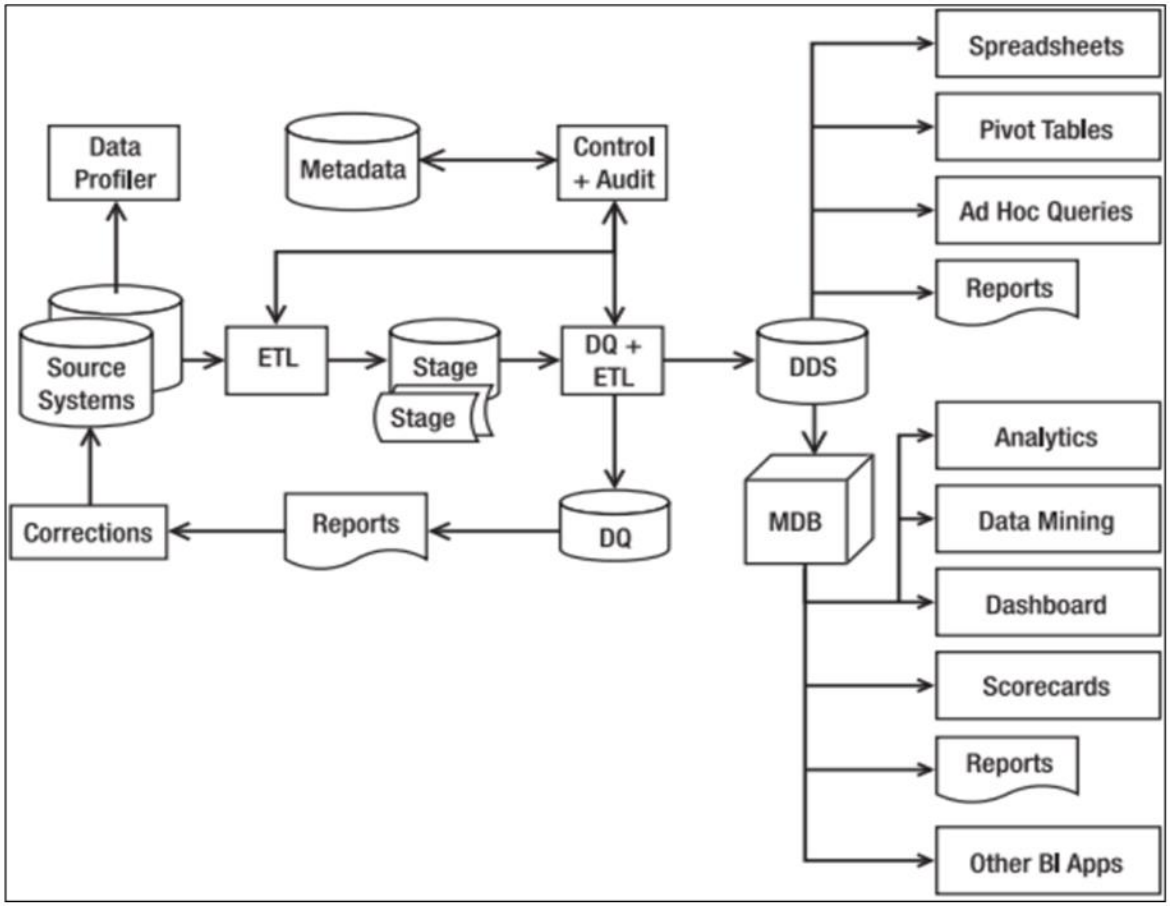


Şekil 3.1. Basit veri tabanı sistemleri mimarisi

Şekil 3.1'de basit veri tabanı sistemleri mimarisi yer almaktadır. Birinci katmanın görevi, kullanıcı arayüzlerini kullanıcıya sunmak ve temel veri işlemlerini gerçekleştirmek olan istemci sistemleridir. İkinci katmanın görevi ise, geçerlilik işlemlerinin yapılması ve veri tabanına erişimin sağlanması olan veri tabanı sunucusudur. Görüldüğü üzere veri tabanı ve Veri tabanı yönetim sistemleri aynı sunucuda bulunmaktadır. Kullanıcılar farklı bir istemci üzerinden iletişim ağı ile veri tabanı sistemine erişmektedir (Çağıltay ve Tokdemir, 2010: 14, 16).

3.3. Veri Ambarları

Veri ambarları, veri tabanlarından daha hızlı çalışan, konu bakımından özelleştirilmiş ve dolayısıyla daha az veri saklayan bir sistemdir. Veri tabanlarında bulunan ve belirli bir süre kullanılacak olan veriler işlenerek veri ambarlarına kaydedilmektedir. Veri tabanından farkı, farklı veri kaynaklarından besleniyor olmasıdır. Veri ambarları üzerinde güncelleme ve silme işlemleri yapılamamaktadır. Sadece sorgulamalar yapılabilmektedir.



Şekil 3.2. Veri ambarı sistem diyagramı

Şekil 3.2'de veri ambarı sistem diyagramı yer almaktadır. Veriler, veri tabanları gibi kaynak sistemlerden ayıklama, dönüştürme ve yükleme sistemi ile yüklenmektedir. Dönüştürme ve yükleme sistemi, kaynak sistemlere bağlanabilen, verileri okuyabilen, dönüştürebilen ve boyutlu veri kaynağı olan veri açıklama özellikleri gibi hedef bir sisteme yükleyebilen bir sistemdir. Veri ambarı sisteminde veri açıklama özelliklerinin kullanılmasının nedeni ise verilerin ihtiyaç duyulan analizlere göre değiştirilebilmesidir (Kaydım, 2021).

3.4. Veri Madenciliği

Veri madenciliği büyük veri yapılarının bulunduğu veri tabanlarından yararlı bilgilerin elde edilmesi için kullanılmaktadır. Veri madenciliği yöntemleri ise, çok sayıda verinin bir araya gelmesiyle meydana gelen veri tabanlarının oluşturulmasına bağlı olarak ortaya çıkmıştır. Zamanla verinin elde edileceği kaynaklara göre veri madenciliğinin alt dalları oluşmuştur. Bu alt dallara metin madenciliği, web madenciliği, görsel madencilik örnek olarak gösterilebilmektedir.

Veri madenciliği yöntemlerinin bir kısmı istatistiksel yöntemlere dayalıdır. Fakat veri madenciliği yöntemlerini klasik istatistiksel yöntemlerden ayıran birkaç özellik söz konusudur. Bunlardan bazıları,

- Veri madenciliği yöntemlerinin nispeten daha büyük veri yapılarında kullanılıyor olması,
- Klasik istatistiksel yöntemlerin ihtiyaç duyduğu birçok varsayımın (dağılım varsayımları, varyans homojenliği varsayımı, dağılımlar arasındaki yakınsamalar için gerekli olan varsayımlar vb.) çoğu zaman gerekli olmaması,
- Hipotezler oluşturup bu hipotezi test etmeye yönelik verileri toplamak yerine elde bulunan verilerden yararlı sonuçlara ulaşmaya çalışması,
- Yeni veriler elde ettikçe elde edilen modellerin veya kuralların eğitilmesine devam edilmesi

şeklinde verilebilir (Hosking, Pednault ve Sudan, 1997).

Veri madenciliği yöntemlerinin uygulanması amacıyla birçok programlama dilinin araçları bulunmaktadır: *SPSS, Rapidminer, Python, WEKA, Statistica Data Miner, DBMiner, Oracle Darwin, R Studio* gibi yazılımlarda veri madenciliği yapılabilmektedir.

3.4.1. Veri madenciliğinin uygulama alanları

Veri madenciliği yöntemleri büyük verinin olduğu her alanda, Pazar analizleri, sahtekarlık analizleri, bankacılık ve finansal analizler, genetik analizler, sağlık bilimlerinde klinik tanı koyma gibi birçok alanda kullanılmaktadır. Aşağıda veri madenciliğine ilişkin alt dalların kullanım alanlarına değinilmiştir.

Veri Madenciliği:

- Müşterilerin demografik özellikleri arasındaki ilişkinin kurulması
- Müşterinin satın alma alışkanlıklarına göre yeni bir ürünü alıp almayacağının tahmini
- Hastalığın ilerleyip ilerlemeyeceğinin tahmini
- Hastalık tanısı koyma
- Demografik ve aile öyküsü incelenerek potansiyel hastalıkların belirlenmesi
- Hastaneye gelen bireylerin doktorlar ve hastane hakkındaki memnuniyet düzeylerinin belirlenmesi
- Dolandırıcılık ve sahtekarlık analizi
- Müşterinin önceki kredi taleplerine göre yeni bir kredi önerisinin yapılması
- Yapılan harcamalara göre müşterilerin sınıflandırılması
- Kart harcamalarındaki değişikliklerin tahmin edilmesi
- Belirli bir hastalığa sebep olan genlerin genetik analizi ile belirlenmesi

Metin Madenciliği:

- Arama motorları
- Dokümanlar arası benzerliğin bulunması
- Yazar tanıma
- İnternet sitesindeki yorumların sınıflandırılması
- Dokümanların konularının tahmin edilmesi

Görsel Madencilik:

- Spor alanında müsabakaların analizi
- Hasta bireylerde kanserli kitlenin belirlenmesi

- Sensör sistemleri ile kişilerin tanınması (e-sınav merkezlerinde sensörlü ekranların kişiyi tanıması, savunma sanayide sensörler ile araçların geliştirilmesi)
- Bir maçta hangi oyuncuların birlikte daha iyi oynadığının belirlenmesi
- Uzaydan çekilen fotoğrafların görsel madencilik ile kuraklığın belirlenmesi ve tahminlemesinin yapılması
- Araç takip sistemleri, park sensörü, kaza önleme sistemleri

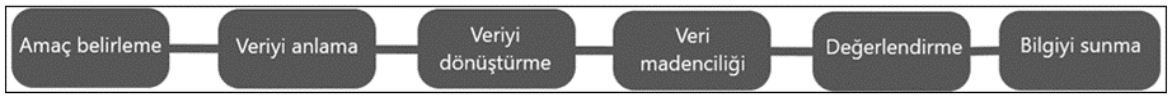
Web Madenciliği:

- Web sayfalarının içeriğinden bilgi çıkarımı
- Arama motorları
- Forum sayfalarındaki kullanıcıların davranışlarının incelenmesi
- Facebook, Twitter gibi sosyal ağlardaki kişilerin birbirleriyle olan etkileşimlerinin incelenmesi
- Suç örgütlerinin tespit edilmesi

3.4.2. Veri madenciliği türleri

Doğrudan veri madenciliği, varsayım deneme ve varsayımı daha iyi duruma getirme ve saf veri madenciliği olmak üzere 3 türe ayrılmaktadır. Doğrudan veri madenciliği, belirlenmiş bilgilerin elde edilmesini içermektedir. Varsayım deneme ve varsayımı daha iyi duruma getirme, varsayımlara dayalı çalışmaların sistemler tarafından doğrulanması, değiştirilmesi ve önerilerin yapılmasıyla ilgilidir. Saf veri madenciliği, hiçbir kısıtlama olmadan, amacın ve hedef bilginin bilinmediği durumlarda uygulanmaktadır (Farboudi, 2009).

3.4.3. Veri madenciliği süreci



Şekil 3.3. Veri madenciliği aşamaları

Şekil 3.3'te veri madenciliği sürecine ilişkin bileşenler yer almaktadır. Veri madenciliğinin ilk adımı ve en önemli adımı olan amacın belirlenmesi ve iş tanımının yapılmasıdır. Bu

adımda amaca uygun verilere karar verilmesi gerekmektedir. İkinci adım olan veriyi anlama, verinin bulunduğu veri tabanı ya da veri ambarı incelenerek, veriye ilişkin hangi tabloların bulunduğu, verideki değişkenlerin tiplerinin ne olduğu ve nasıl işlenmesi gerektiğinin karar verildiği aşamadır. Veriyi dönüştürme adımı, verinin analize uygun hale getirilerek işlenmesi, kullanılacak verideki anlamlı ya da anlamsız değişkenlerin belirlenmesi, gürültülü verilerin silinmesi aşamasıdır. Veri, belirlenen amaca uygun model için kullanılabilir hale getirilmektedir. Hangi modelin belirlenen amaç için daha iyi olduğunun belirlenmesi için veri seti eğitim ve test veri seti olmak üzere ikiye ayrılır. Test veri setine genellikle %20 yüzdesel ayrım ile karar verilmektedir. Eğitim veri seti kullanılarak özelliklerden sınıf değerini tahmin edecek bir model oluşturulur. Eğitilen bu veride kullanılan yöntemlerin hangisinin daha iyi odluğuna doğruluk oranı hesaplanarak karar verilmektedir. Öğretilen bu yöntem test verisine uygulanmaktadır. Değerlendirme aşamasında elde edilen verilerin ve kurulan modellerin belirlenen amaca uygun olup olmadığı değerlendirilmektedir. Sürecin son aşaması olan bilgiyi sunma aşaması, gerçekleştirilmiş tüm adımların doğrulandığı, çalışmaya ilişkin bulgu ve sonuç bilgisinin sunulduğu aşamadır.

3.4.4. Veri madenciliğinde kullanılan yöntemler

Veri madenciliği yöntemleri özellik çıkarımı, sınıflandırma, kümeleme ve birliktelik analizi olmak üzere dört başlıkta incelenebilmektedir (Altunkaynak, 2019: 17, 18).

Özellik seçme yöntemleri

Gözlem sayısının ve değişken sayısının fazla olduğu durumlarda veri madenciliği işlemleri yapılmadan önce yararlı değişkenlerin belirlenip, veri boyutu küçültülerek verideki en etkili özelliklerin seçilmesi gerekmektedir. Böylece kullanılan algoritmaların daha etkin sonuçlar vermesi sağlanmaktadır (Gorunescu, 2011).

Özellik seçim yöntemleri sarmal ve filtreleme yöntemleri olmak üzere iki başlık altında incelenebilmektedir. Sarmal yöntemler, modelin doğrulama gücünü maksimum yapacak özelliklerin bulunduğu yöntemlerdir. Filtre yöntemlerinde ise özellikler tek tek incelenmektedir (John,Kohavi ve Pfleger, 1994).

Relief algoritması, ki-kare algoritması, ergs algoritması ve ifser algoritması olmak üzere 4 adet özellik seçme yöntemi bulunmaktadır. Relief algoritması her bir özelliğe [-1,1] arasında ağırlık değeri atamaktadır. Ağırlık değerinin hesaplanmasında her bir birimin bulunduğu sınıftaki en yakın birime ve diğer sınıftaki en yakın birime uzaklığı dikkate alınır. Ki-kare algoritmasında, sınıf ve özellikler arasındaki ki-kare değerleri hesaplanmaktadır. Sınıf ile özellik arasında çapraz tablo oluşturulmaktadır. En büyük ki-kare değerine sahip özellik en ayırt edici özelliktir. Ergs algoritması, chebyshev eşitsizliğine dayalı aralıkların hesaplanarak, bu aralıkların örtüşme değerlerinin bulunmasına dayanmaktadır. Ifser algoritması, bir sınıfa ait aralığın başka bir sınıfa ait aralığı kapsaması durumunu içermektedir (Altunkaynak, 2019: 17).

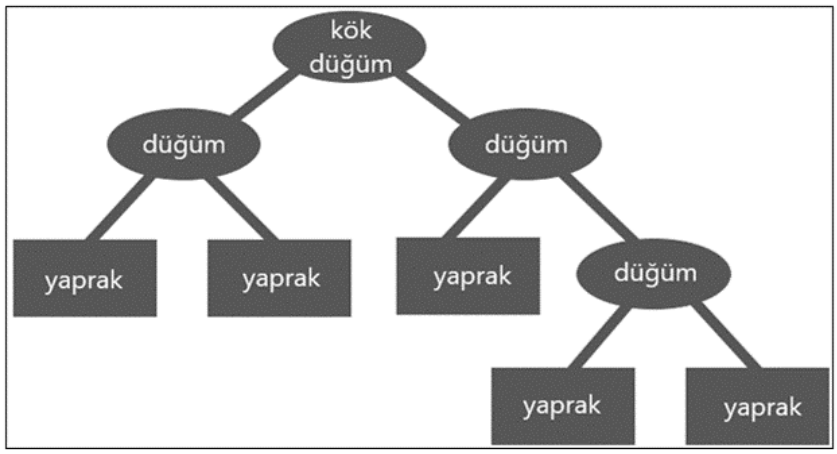
Sınıflandırma yöntemleri

Sınıflandırma yöntemi, mevcut verilerin belirli kriterlere göre sınıflara ayırılması ve yeni eklenecek olan verilerin hangi sınıfa dahil olacağının tahmin edilmesidir.

Klasik istatistikte yaygın olarak kullanılan regresyon analizi bir çeşit sınıflandırma yöntemidir. Ancak regresyon analizi için birçok varsayım (gözlemler arası bağımsızlık, değişkenler arası bağımsızlık, varyansın sabit olması gibi) vardır. Veri madenciliğinde kullanılan sınıflandırma algoritmalarında ise bu tarz varsayımlar gerekli değildir (Altunkaynak, 2019: 18). Sınıflandırma yöntemleri arasında karar ağacı algoritmaları, kural temelli algoritmalar, Naive bayes sınıflandırma, k-en yakın komşu yöntemi yer almaktadır.

Karar ağacı algoritmaları

Karar ağaçları, ağacın oluşturulması ve sınıflandırmanın yapılması olmak üzere iki adımda oluşmaktadır. Daha kolay oluşturulması, yorumlanması, veri tabanları ile entegre olmasından dolayı en yaygın kullanılan sınıflandırma yöntemleri olarak sayılmaktadır. Karar ağaçları, uzay görüntülerinin sınıflandırılması, gelecek olayların tahmin edilmesi, müşterilerin kredi geçmişine bakılarak krediye uygun olup olmadığının karar verilmesi gibi birçok alana uygulanabilmektedir (Esen, 2009). Karar ağacı yöntemleri ID3 algoritması, J48 algoritması, CART algoritması, CHAID algoritması, karar ağacının budanması yöntemlerinden oluşmaktadır.



Şekil 3.4. Basit karar ağacı yapısı

Şekil 3.4'te basit bir karar ağacı yapısına yer verilmiştir. Şekilden görüldüğü üzere karar ağacı yapısı, ilk dallandırmanın başladığı kök düğüm, dallandırmanın sonlandığı yapraklar, düğümler ve düğümler veya yapraklar arası bağlantı olan dallardan oluşmaktadır. Dallandırmalar verinin daha küçük sınıflara ayrılmasını sağlamaktadır.

Kural temelli algoritmalar

Kural temelli algoritmalar, bağımlı değişkenin frekans dağılımından yararlanarak sınıflandırma kurallarını oluşturmaktadır. Sınıflandırmayı bağımsız değişkenleri dikkate almadan uygulayan ZeroR yöntemi ve sadece bir bağımsız değişkeni dikkate alarak uygulayan OneR yöntemi, adlarını sınıflandırma için seçtikleri bağımsız değişken sayısından almaktadır (Altunkaynak, 2019: 110).

Naive bayes sınıflandırma

Bayes olasılığına bağlı olarak sınıflandırma yapılmaktadır. Koşullu olasılığın *n* tane ayrık olay için genelleştirilmiş halidir. Amaç, bağımsız değişken vektörünün değeri biliniyorken bağımlı değişken değerinin tahmin edilmesidir. Eşitlik (3.1)'de *n* tane olay için bayes olasılık tanımlanmıştır. Eşitlik (3.2)'de ise bağımsızlık koşulu altında bayes olasılık tanımlanmıştır (Altunkaynak, 2019: 119).

$$P(X)=\sum_{i=1}^{n} P(X/Ci)P(Ci) \quad (3.1)$$

$$P(X/C_j)=\prod_{i=1}^{p} P(Xi/Cj) \quad (3.2)$$

Bağımlı değişken değerinin tahmini için eşitlik (.2)'de yer alan $P(C_j/X)$ bayes olasılıkları hesaplanarak en büyük olasılık değerine ait sınıf seçilmektedir. $P(C_j/X)$ değerinin maksimize edilmesi $P(X/C_j).P(C_j)$ değerinin maksimize edilmesiyle aynı olduğundan bu ifadeyi maksimum yapacak C_j sınıfının belirlenmesi gerekmektedir. $P(X/C_j).P(C_j)$ ifadesi sonsal olasılığı içerdiğinden buna "en büyük sonsal" denir ve $\text{argmax}\{P(X/C_j).P(C_j)\}$ olarak tanımlanmaktadır. $\text{argmax}\{P(X/C_j).P(C_j)\}$ değerinin büyüklüğü önemsenmektedir (Altunkaynak, 2019: 119).

K-en yakın komşu yöntemi

K-en yakın komşu yöntemi, sayısal bağımsız değişkenlere ilişkin gözlemler arasındaki uzaklıklara dayanmaktadır. İlk olarak komşu sayısı olan k değerinin belirlenmesi gerekmektedir. Büyük verilerde k değerinin büyük bir sayı alınması tavsiye edilmektedir (Steinbach ve Tan, 2005). Her gözlemin bütün komşuları için gözlem ve komşusu arasındaki uzaklık hesaplanmaktadır. En düşük değere sahip olan sınıflar birleştirilmektedir (Şentürk, 2011).

Sınıflandırma performansının ölçümü

Veri seti eğitim ve test verisi olarak ikiye ayrıldıktan sonra yapılan sınıflandırmaya ilişkin performans ölçüleri hesaplanmaktadır. Çizelge 3.1'de yaygın olarak kullanılan performans ölçüleri yer almaktadır.

Çizelge 3.1. Yaygın olarak kullanılan performans ölçüleri (Altunkaynak, 2019: 136)

Performans Ölçüsü	Formül
Doğruluk (Accuracy)	ACC = (TP + TN) / N
F1 Ölçüsü	F1= 2TP / (2TP+FP+FN)
Sensitivity	TPR= TP / (TP+FN)
Specificity	TNR= TN / (FP+TN)
Pozitif Kestirim Değeri	PPV= TP / (TP+FP)
Matthews Korelasyon Katsayısı (MCC)	(TP*TN – FP*FN) / $\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}$

Birliktelik kuralları yöntemleri

Birliktelik analizi verilerin birbirleriyle birlikte ne sıklıkla kullanıldığını, birbirleriyle olan bağlantısının belirlenmesi için uygulanmaktadır. Birliktelik kurallarının en belirgin örneği sepet analizidir. Örneğin müşterinin hangi ürünleri aynı anda aldığı ya da satın alma eğilimde olduğunun belirlenmesidir.

Apriori algoritması en yaygın kullanılan birliktelik kuralları algoritmasıdır. Birliktelikler incelenirken destek ve güven değerleri dikkate alınmaktadır. Destek değeri, bir özelliğin tüm gözlemlerdeki oranını göstermektedir. Güven değeri ise, X verilmişken Y'nin koşullu olasılığını ifade etmektedir (Altunkaynak, 2019: 141). Kaldıraç değeri ise destek değerinin güven değerine bölümü ile bulunmaktadır.

Kümeleme analizi yöntemleri

Kümeleme analizinde, önceden sınıflara ayırmadan verinin homojenliğine bakılarak küme oluşturulmaktadır. Kümeleme işlemindeki asıl amaç sınıflar içi benzerliği maksimum, sınıflar arası benzerliği minimum tutmaktır. Kümeleme yöntemleri bölünmeli, hiyerarşik, yoğunluğa dayalı, grid tabanlı, model tabanlı, örüntü tabanlı ve kısıt tabanlı olmak üzere 7 şekilde sınıflandırılmaktadır (Farboudi, 2009). En sık kullanılan kümeleme yöntemleri arasında k-ortalama, k-medoids, DIANA, AGNES ve DBSCAN yöntemleri yer almaktadır.

4. METİN MADENCİLİĞİ

Günümüzde bilgi teknolojilerinin gelişmesi ve büyük veri kullanımının hızla artmasıyla veri madenciliğine bağlı olarak metin madenciliği de gelişmektedir. Kullanılan veri kaynaklarından birisi de metin madenciliğinde kullanılan metin veri kaynaklarıdır. Büyük metin veri kaynaklarından metin verilerinin elde edilmesi, analiz edilmesi, yararlı bilgilerin çıkarılması gibi birçok işlemin uygulanması yazılım programlarında yer alan metotlar, fonksiyonlar ve bu fonksiyonların zamanla geliştirilmesiyle kolayca sağlanmaktadır.

Metin madenciliği uygulaması için kullanılan birçok programlama dilleri bulunmaktadır. *Python, Rapidminer, R Studio, Knime* ve *WEKA* programları örnek olarak verilebilmektedir.

4.1. Metin Madenciliği Çalışma Alanları

Metin madenciliğine ilişkin çalışma alanlarına, klinik araştırma ve tanı koyma, risk yönetimi, müşteri hizmetleri, iş zekâsı, yazar tanıma, içerik sınıflandırma, dolandırıcılık ve sahtekarlık analizleri, Pazar ve sepet analizleri, metinler arası karşılaştırma, metin özetleme gibi birçok örnek verilebilmektedir.

Metin madenciliği yöntemlerinin kullanıldığı enformasyon getirimi, doğal dil işleme aşaması, adlandırılmış varlık tanıma, örüntüsü tanımlı varlıkların bulunması, eş atıf, duygu analizi ve ilişki, kural ve olay çıkarımları gibi yöntemler örnek olarak verilebilmektedir. Enformasyon getirimi, ilgili metne ilişkin ön bilginin toplandığı çalışmalardır. Doğal dil işleme çalışmaları, özellik çıkarımı ve metinden anlamsal bilgilerin bulunması amacıyla kullanılmaktadır. Adlandırılmış varlık tanıma, metin işleme aşamasında birkaç istatistiksel özelliklerin çıkarılması için kullanılmaktadır. Metin içerisindeki kişi isimleri, semboller, kısaltmalar gibi bilgilerin çıkarılması örnek olarak verilebilmektedir. Hedeflenen kelime gruplarının metin içerisinden çıkarılması, sayılması, yoğunluğunun bulunması gibi işlemler bu çalışmalarda yapılabilmektedir. Örüntüsü tanımlı varlıkların bulunması çalışmalarında, metnin içerisinden istenilen bilgiler genellikle regex (düzenli ifadeler) ile çalışılmaktadır (Şeker, 2012). Eş atıf çalışmalarında, bir varlığa atfeden isim kelime gruplarını ve diğer terimlerin bulunmasını ve ayrıştırılmasını hedeflemektedir. Duygu analizi çalışmaları, metinlerde geçen duygusal ifadelerin çıkarılmasını amaçlamaktadır. En sık kullanılan

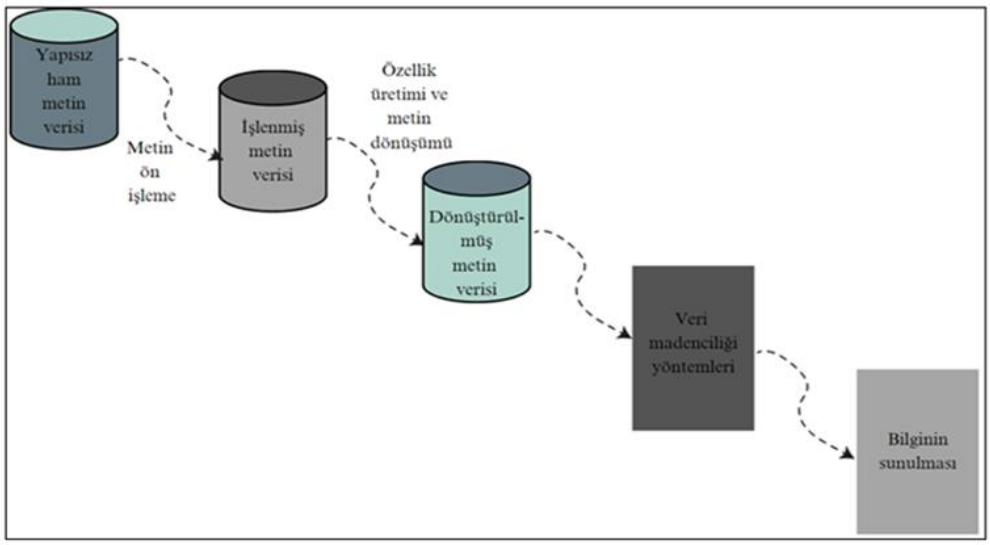
duygusal kutupsallığa göre, bir konu hakkında geçen yorumların olumlu/olumsuz olmak üzere iki gruba ayrılması gerekmektedir (Şeker, 2013). İlişki, kural ve olay çıkarımları çalışmaları, çeşitli amaçlarla metin içerisindeki olayların çıkarılması ve bu olaylar arasındaki olası ilişkiler belirlenebilmektedir (Şeker, 2010).

4.2. Metin Madenciliği İle Veri Madenciliği Arasındaki Farklar

Metin madenciliği yapılandırılmamış metin verilerinden doğrudan ya da verinin yapılandırılması ile yararlı bilgiye ulaşılma yöntemidir. Veri madenciliğinde ise, belirli bir yapıya uydurulmuş ve analiz edilmeye hazır veriler kullanılmaktadır. Her ikisinde de yararlı bilgiye ulaşmak amaçtır. Yararlı bilgiye ulaşmak amacıyla ise, istatistiksel yöntemler ve makine öğrenmesi yöntemleri kullanılabilmektedir.

4.3. Metin Madenciliği Yöntemi

Metin madenciliği veri kaynağını metin verileri olarak kabul eden ve metin verilerinden yararlı bilgilerin çıkarılmasını amaçlayan bir veri madenciliği alt dalıdır. Veri tabanındaki veriler yapısallaştırılmış ve belirli tablolara eklenmiş verilerdir fakat metin içerisindeki veriler veri tabanı verilerindeki gibi bir yapıya sahip değildir. Yapısız veriler işlenebilir bir yapıya getirilerek yapılandırılmış veri elde edilebilmektedir veya yapılandırılmadan yapısız verinin üzerinde işlemler yapılabilmektedir.



Şekil 4.1. Metin madenciliği süreci

4.3.1. Metin madenciliği adımları

Metin madenciliği süreci metin koleksiyonunun oluşturulması, ön işleme işlemlerinin yapılması, özellik çıkarımı, veri madenciliği ve bilginin sunumu gibi aşamalardan oluşmaktadır.

Metin koleksiyonun oluşturulması

Metin koleksiyonu oluşturmak amacıyla, çalışılacak veri kaynağı (web siteleri, bilgisayar ortamındaki metinsel belgeler, veri tabanları, veri ambarları vb.) belirlenip, belirlenen veri kaynağından yapılandırılmamış ham metin verisi elde edilmektedir. Elde edilen veriler doğal dil ile yazılmış, belirli bir yapısı olmayan metin verileridir.

Ön işleme adımlarının uygulanması

Yapılandırılmamış metin verilerinin, üzerinde çalışılabilir hale getirilmesi amacıyla metin ön işleme işlemleri yapılmaktadır. Bu aşamada verilerin temizlenmesinin yanı sıra, metin içerisindeki noktalama işaretlerinin ve sayıların silinmesi, durak kelimelerin (ve, veya, fakat vb.) silinmesi, metin içerisindeki tüm harflerin küçük harfe dönüştürülmesi, kelimelerin köklerinin bulunması, sıfat zarf gibi dilbilimsel kelimelerin bulunması gibi birçok işlemi içermektedir.

Belirtkeleme

Belirtkeleme (tokenization), metin içerisindeki karakter dizisinin belirli simgelerle ayrılması, sayılar ve noktalama işaretleri gibi istenmeyen karakterlerin atılması işlemidir (Allahyari, Safaei, Pouriyeh, Trippe, Kochut, Assefi ve Gutierrez, 2017).

Durak kelimelerinin silinmesi

Durak sözcükleri (stopwords), metinde yer alan edat, bağlaç gibi kelimelerdir. Metin içerisinde oldukça fazla sayıda bulunan bu kelimeler, metin madenciliği süreci için yararlı bilgi olmamasından dolayı genellikle silinmektedir.

Metnin küçük harfe dönüştürülmesi

Metin içerisinde arama yapılması gerektiğinde küçük harfe duyarlı bir yapı bulunmamaktadır. Bu nedenle metindeki tüm harfler küçük harfe dönüştürülerek daha sağlıklı sonuçlar elde edilebilmektedir.

Kelimelerin köklerinin bulunması (lemmatization,stemming)

Kelimelerin morfolojik analizinin yapılması, yani bir kelimenin birçok çekimli türlerini tek bir kelime altında toplayarak analiz edilebilecek hale getirilmesidir (Allahyari, Safaei, Pouriyeh, Trippe, Kochut, Assefi ve Gutierrez, 2017).

Özellik seçme

Metin madenciliği çalışmalarında metnin analize uygun hale getirilmesi gerekmektedir. Bu sebeple metin içerisindeki ilgilenilen kelimelerin tespit edilmesi gerekmektedir. Özellik belirleme, ön işlemden geçen metnin içerisinden önemli kelimelerin tespiti ve ilgisiz kelimelerin çıkarılması işlemlerinin yapıldığı adımdır (Oğuz, 2009). Metinlerin içerisinden bilgisayarın işleyebileceği verilerin çıkarılmasıdır. Terim ağırlıklandırma yöntemleri, *n*-gram ve kelime torbası modelleri ile özellik seçilebilmektedir.

Terim ağırlıklandırma

Terim ağırlıklandırma yöntemleri arasında terim frekansı (TF), ters doküman frekansı (IDF), TF-IDF indekslerinin birlikte kullanıldığı yöntem, logaritmik normalize TF(LTF), genişletilmiş normalize terim frekansı (ANTF) ve ölçeklenmiş terim frekansı (STF) gibi yöntemler yer almaktadır.

Terim frekansı (TF)

Bir terimin dokümandaki sıklığını ifade etmektedir. Bir terimin yani kelimenin bir metin içeriğinde geçme sıklığının fazlalığına göre terim frekans değeri de o kadar fazladır. Kelimenin metin içerisindeki kullanılma sayısının toplam kelime sayına bölümü ile bulunmaktadır.

Ters doküman frekansı (IDF)

Bir terimin birden fazla dokümanda bulunma sıklığıdır. Bir terim ne kadar az dokümanda bulunursa o kadar önemli bir terim olarak kabul edilmektedir. Örnek olarak durak kelimeleri, birçok dokümanda bulunmaktadır ve doküman içeriği hakkında bilgi vermemektedir. Bu nedenle bu terimlerin önemli terimler olmadığı kabul edilmektedir (Uçkan, Hark, Seyyarer ve Karcı 2019). Toplam doküman sayısının kelimeyi içeren doküman sayısına bölünmesiyle elde edilen sayısının logaritması ile hesaplanmaktadır.

Terim frekansı ve ters doküman frekansı (TF-IDF)

Bir terimin metin içeriğindeki sıklığı fazla ve diğer dokümanlardaki sıklığı az ise, ilgili terimin iyi bir özellik ve iyi bir sınıflandırma değişkeni olduğu anlamına gelmektedir. TF ve IDF değerinin çarpımı ile elde edilmektedir.

N-gram modeli

N-gram modeli, olasılık ve istatistiksel doğal dil işleme gibi dizilimlerin, sözcük ya da cümlelerin olasılıklarını inceleyip modelleyen alanlarda kullanılmaktadır. En genel kullanım amacı (n-1) birim kullanılarak n'inci birimin tahmin edilmesidir (Eroğlu, 2005). Basit *n*-gram modeline göre her birimin diğer birimlerin peşinden gelme olasılığı eşittir (Jurafsky ve Martin, 2023). Bir kelimenin olasılığının yalnızca kendinden önce gelen kelimeye bağlı olması varsayımına Markov varsayımı denmektedir. Markov modellerinde bir birimin olasılığı kendinden *n* önceki birimlerin olasılıkları kullanılarak bulunmaktadır. Kelime tabanlı bigram model her bir kelimeye bir durum atanmış olan bir çeşit basit Markov zinciri olarak düşünülmektedir. Bu şekilde *n*-gram modeller için genelleştirme yapılacak olunursa n-gram modeller için: (Eroğlu, 2005)

Bigram model: sadece bir önceki kelimeye bakılır
Trigram model: önceki iki kelimeye bakılır
n-gram model: önceki (n-1) kelimeye bakılır.

Böylece bigram modele 1.derece Markov modeli ve n-gram modele ise (n-1). Derece Markov modeli denmektedir (Eroğlu, 2005).

Kelime çantası modeli

Metinlerin sayısal türe dönüştürülmesi amacı için kullanılan bir yöntemdir. Kelime çantası modeli ile sınıflandırma çalışması yapılan metinler kelimelere ayrılmaktadır. Oluşturulan her kelime bir özellik olarak tanımlanabilmektedir. Her bir özelliğin ilgili metinde geçme sıklığı hesaplanarak metinsel bir veri sayısal hale dönüştürülebilmektedir (Aksu, Karaman 2020).

Veri madenciliği yöntemlerinin uygulanması

Yapılandırılmamış verilerin yapılandırılmış hale getirilip, analize uygun hale getirilmesiyle veri madenciliği yöntemleri kullanılabilir duruma gelmektedir. Bu aşamada veri madenciliği yöntemlerinden olan sınıflandırma, kümeleme ve birliktelik analizi bulunmaktadır. Sınıflandırma yöntemlerinden karar ağaçları (ID3, C4.5, CART, CHAID algoritmaları), kural temelli algoritmalar (ZeroR, OneR yöntemleri), Naive bayes sınıflandırma, k-en yakın komşuluk yöntemi kullanılabilmektedir. Birliktelik kurallarından apriori ve ayırma algoritması kullanılabilmektedir. Kümeleme yöntemlerinden ise, uzaklık ölçüleri (Öklit, Manhattan, Minkowski vb.), bölünmeli kümeleme yöntemleri (k-ortalama, k-medoid vb.), hiyerarşik kümeleme yöntemleri (AGNES, DIANA vb.), yoğunluk tabanlı kümeleme yöntemleri (DBSCAN vb.) kullanılabilmektedir.

Bilginin sunulması

Veri madenciliği uygulamalarından sonra tüm işlemlerin ve algoritmaların doğru bir şekilde çalıştığının testinin yapılması gerekmektedir. Başarılı işlemlerin ardından analiz sonucu elde edilen bilgiler sunulmaktadır.

5. ROBOTİK SÜREÇ OTOMASYONU

Robotik süreç otomasyonu (*RPA*), dijital sistemler ve yazılımlarla etkileşime giren insan eylemlerini taklit eden yazılım robotlarının oluşturulmasını, dağıtılmasını ve yönetilmesini kolaylaştıran bir yazılım teknolojisidir. Tıpkı insanlar gibi, yazılım robotları da ekranda ne olduğunu anlamak, doğru tuş vuruşlarını tamamlamak, sistemlerde gezinmek, verileri tanımlayıp çıkarmak ve çok çeşitli tanımlanmış eylemleri gerçekleştirmek gibi işlemleri yapabilmektedir (UiPath, 2022). Ancak yazılım robotları, bu işlemleri insanlardan daha hızlı, tutarlı ve aksatmadan yapabilmektedir.

RPA robotları, uygulamalara giriş yapmak, dosyaları taşımak, veri tabanına bağlanıp verilere erişip bu verileri kopyalamak ve güncellemek, formları doldurmak, belirli periyotlarda gerçekleştirilen analizleri yapmak, raporları oluşturmak ve bu raporları e-posta ile göndermek, karmaşık hesaplamaları yapmak, metni analiz etmek, yapılandırılmamış verileri tespit etmek gibi birçok görevi yapmaktadır.

Şekil 5.1. *RPA* yazılımlarının tercih edilmesinin nedenleri (UiPath, 2022)

Şekil 5.1'de *RPA* yazılımlarının tercih edilmesinin sebeplerinden bazıları yer almaktadır. *RPA*, kuruluşların iş akışlarını otomatize ederek hem maliyetten hem de zamandan daha karlı bir hale getirmektedir. Sıradan ve çalışanların zamanını alan görevlerin yazılım robotlarına yaptırılmasıyla çalışanların memnuniyetini ve üretkenliğini artırmaktadır.

Yapılan araştırmalara göre RPA, manuel hataların %57'sini azaltmaktadır (Kirkwood, 2019). Çalışanların %68'i otomasyonun kendilerini daha üretken hale getirdiğine inanmaktadır (Wu, 2020). Deloitte global *RPA* anketi, katılımcıların %92'si *RPA*'nın daha iyi uyumluluk açısından beklentilerini karşıladığını ya da beklentilerini aştığını göstermektedir (Deloittle, 2018). Yöneticiler, %60'ı *RPA*'nın çalışanların daha stratejik çalışmalara odaklanmasını sağladığını belirtmektedir (Kirkwood, 2019).

5.1. RPA Kullanım Alanları

RPA, belirli periyotlarda yapılan rutin işlerin bulunduğu tüm alanlarda kullanılmaktadır. Finans, sağlık, üretim, kamu, hukuk, müşteri hizmetleri, bilgi teknolojileri gibi birçok alanda kullanılmaktadır. Hemen hemen tüm alanlarda uygulanabilir olduğu için gün geçtikçe kullanım oranı artmaktadır. İş kurallarına dayalı büyük veriler içeren ve belirli periyotlarla tekrarlanan görevler otomasyon ile daha kolay bir duruma gelebilmektedir.

5.2. RPA Yazılımları

RPA yazılımları arasında *UiPath, Aisera, Automation Anywhere, Laiye Technology, Blue Prism, Zapier, IBM RPA, Microsoft Power Automate, SAP RPA* gibi birçok yazılım bulunmaktadır. Tüm bu yazılımların avantaj ve dezavantajları bulunmaktadır. Çizelge 5.1'de bazı *RPA* yazılımların avantajları ve dezavantajlarına değinilmiştir.

Çizelge 5.1. *RPA* yazılımlarının avantajları ve dezavantajları

RPA YAZILIMI	AVANTAJLARI	DEZAVANTAJLARI
UiPath	En iyi *RPA* yazılımı ve kurulum kolaylığı Kullanıcı dostu arayüz ve doküman çokluğu Microsoft ürünleriyle entegre	Pahalı lisans Kodlama bilgisinin zorunluluğu Kısmen performans yavaşlığı Hata tespitinin eksikliği
Automation Anywhere	Kullanıcı dostu arayüz Kodlama bilgisinin gerekli olmaması Raporlama araçlarının olması	Hataların çok sık oluşması Yeni sürümlerinin olumsuz özellikleri
Blue Prism	Kullanıcı dostu arayüz Yüksek güvenlik	Lisanslarının pahalı olması Doküman eksikliği Yeni sürümlerinin olumsuz özellikleri
Microsoft Power Automate	Kullanıcı dostu arayüz Doküman çokluğu Performans hızı Microsoft ürünleriyle entegre	Hata tespitinin eksikliği Kurulum zorluğu
IBM RPA	Kullanıcı dostu arayüz ve kolay kurulum	Doküman eksikliği
SAP RPA	Kullanıcı dostu arayüz Chatbot uyumluluğu	Lisanslarının pahalı olması Doküman eksikliği ve hata çokluğu

5.2.1. UiPath

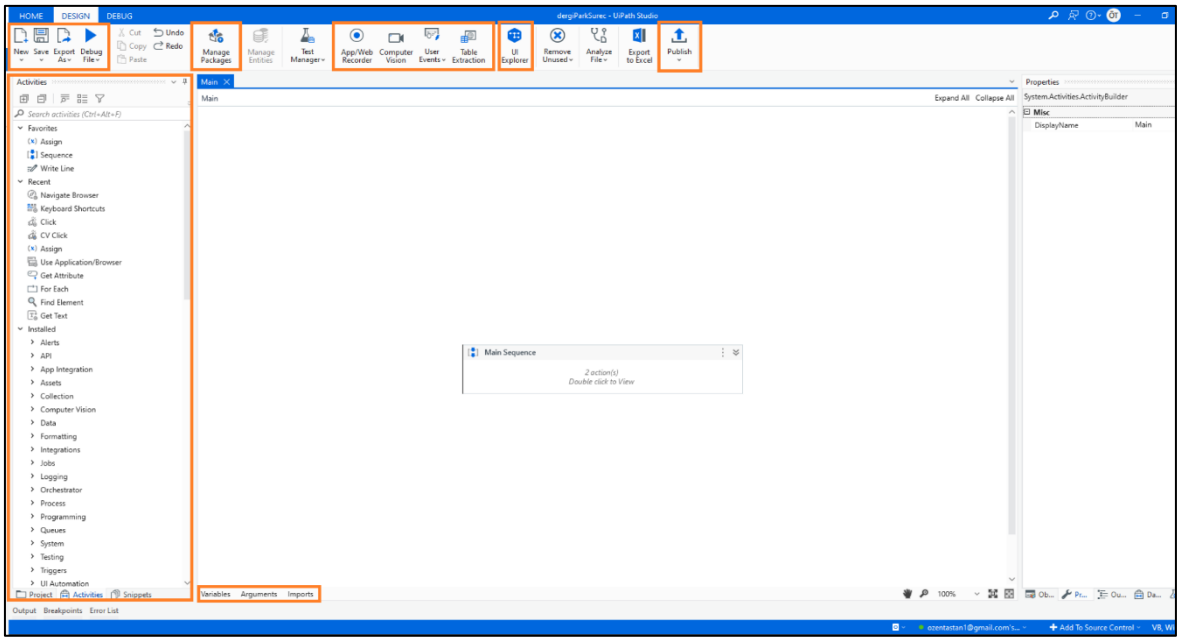
UiPath, 2005 yılında kurulmuş Romanya merkezli *RPA* ve yapay zekâ odaklı bir yazılım firmasıdır. *UiPath Academy* sayfası açık ve her seviyeye uygun eğitim platformudur. *UiPath*, *RPA* ile başladığı sürece ihtiyaçların artması sebebiyle farklı ürünlerde geliştirmeye başlamıştır. Geliştirdiği ürünler arasında, kurum içerisindeki işlerin otomasyona uygun olup olmadığına ilgili kişilerce karar verilen platform olan *Automation Hub*, mevcut olan verilerle yapılan aksayan görevlerin ve bu görevlerde hangi noktalarda otomasyonun olacağını belirten ürün olan *Process Mining*, kişisel makine üzerinde çalışan ve yapılmış projenin adımlarını tespit eden ürün olan *Task Mining* ve yapılmış işlerin çalışma süresinde önemli yerlerin ekran görüntüsünü alan ürün olan *Task Capture* bulunmaktadır.

Görevlerin otomasyona çevrilebilmesi için geliştiriciler tarafından kullanılan ve görevlerin robotlara yaptırılma için sürecin inşa edilmesi *UiPath Studio* ve *UiPath StudioX* ürünleri ile gerçekleşmektedir. Daha sonra robotlara yaptırılmış görevlerin yönetilmesi ve çizelgelenmesi ise *Orchestrator* ve *Cloud Platform* üzerinden yapılmaktadır. Bütün bu süreçler sonucunda, robotlar belirli periyotlarda görevleri gerçekleştirmektedir. Bazı durumlarda robotlar görevini gerçekleştirme sırasında insanların müdahalesine ihtiyaç duymaktadırlar. *Action Center* ürünü üzerinden insanlar göreve müdahale edilebilmektedir. Bu ürün insanların ve robotların birlikte çalıştığı adım olarak bahsetmek mümkündür. Tüm adımlar sonrasında yapılan görevlerin raporlanması için *Insights* ürünü kullanılmaktadır.

Bu çalışmada, DergiPark ve PubMed platformları üzerinden *PDF* dosyalarının indirilmesi amacıyla *UiPath Studio* ürünü kullanılmıştır.

UiPath Studio

Robotik süreç otomasyonu projelerinin geliştirildiği, testlerinin yapıldığı bir platformdur. Belirli periyotlarda yapılan ve belirli kuralları olan tüm karmaşık görevler bile otomatize edilebilmektedir. Görevlerin karmaşıklığına göre yazılım hakkında teknik bilginin ve programlama bilgisinin olması gerekmektedir. Resim 5.1'de *UiPath Studio* platformunun geliştirme arayüzü yer almaktadır.



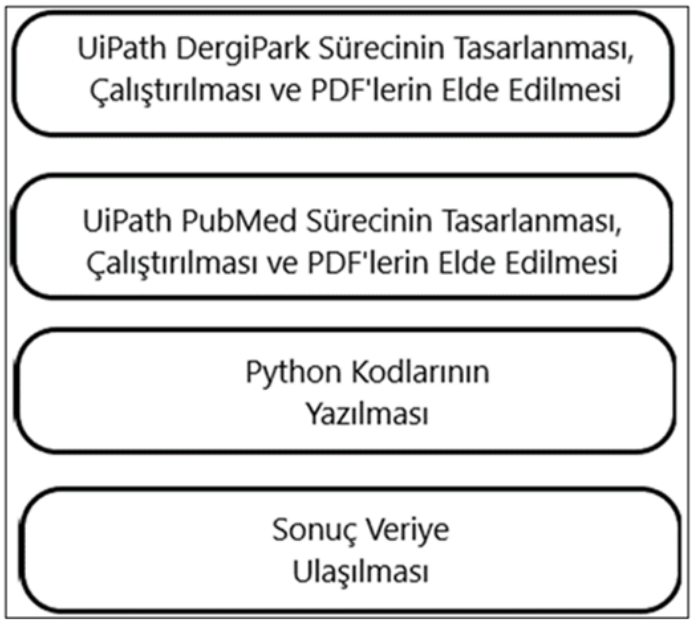
Resim 5.1. *UiPath Studio* geliştirme arayüzü

Resim 5.1'de görüldüğü üzere "Design" sekmesinde geliştirme yapılmaktadır. "New" seçeneği ile yeni iş akışları oluşturulabilmektedir. "Save" seçeneği ile oluşturulan iş akışları, "Export As" seçeneği ile şablon proje olarak kaydedilmektedir. "Debug File" ile tüm projenin baştan başlatılabilir ya da "debug" modda çalıştırılabilmektedir. "Manage Package" seçeneği ile proje tasarımı boyunca gerekli olan paketler yüklenebilmektedir. "App/Web Recorder", "Computer Vision" ve "Table Extraction" özellikleri çalışılan arayüz üzerinde yapılacak olan işlemlerin daha kolay yapılmasını sağlamaktadır. "Ui Explorer" özelliği ile arayüzdeki elementlere erişebilmek için ihtiyaç duyulan selektörlere ulaşabilmek amacıyla kullanılmaktadır. "Publish" özelliği ile tamamlanmış görevlerin ya da üzerinde güncelleme yapılan projelerin yayınlanması gerçekleştirilmektedir. Sol menü çubuğunda bulunan "Activities" sekmesinde robota yaptırılacak işler için komutlar verilmektedir. "Project" sekmesinde tasarlanan projeye ait dosyalar ve yüklü olan paketler gözükmektedir. "Snippets" sekmesinde ise varsayılan olarak oluşturulmuş iş akışları yer almaktadır. Arayüzün alt kısmında "Variables", "Arguments" ve "Imports" sekmeleri bulunmaktadır. Buralarda projede kullanılacak değişkenlerin ve argümanların oluşturulması ve projeye kod kütüphanelerinin eklenmesi işlemleri yapılmaktadır.

6. UYGULAMA

Bu çalışma kapsamında, *UiPath RPA* robotu ile indirilen ulusal ve uluslararası dergilerde yayınlanan makalelere metin madenciliği uygulanarak istatistiksel analiz yöntemlerinin kullanılma sıklığının tespit edilmesi ve ulusal-uluslararası makalelerdeki yöntemlerin kullanılma sıklıkları karşılaştırılmasının yapılması amaçlanmıştır. Bu bölümde çalışmanın amacı doğrultusunda yapılan uygulamalar açıklanmıştır.

Şekil 6.1'de uygulamada yapılan işlemlere ilişkin adımlar yer almaktadır. İlk olarak DergiPark platformundan robota *PDF*'lerin indirilmesi amacıyla *UiPath* süreci tasarlanıp çalıştırılmıştır ve *PDF* verilerine erişilmiştir. İkinci olarak PubMed platformundan robota *PDF*'lerin indirtilmesi amacıyla *UiPath* süreci tasarlanıp çalıştırılmıştır ve *PDF* verileri elde edilmiştir. Daha sonra elde edilen makaleler üzerinde işlemlerin yapılması amacıyla Python programında gerekli kodlar yazılmıştır ve kod blokları her bir makale için çalıştırılmıştır. Tüm makaleler için kodların çalıştırılması sonucunda ise sonuç veriye ulaşılmıştır.



Şekil 6.1. Uygulamada yapılan işlemlere ilişkin adımlar

6.1. Metin Verilerinin Elde Edilmesi

Çalışmada kullanılan metin veri seti, *UiPath RPA* robotu kullanılarak DergiPark ve PubMed sitelerinden indirilen makalelerin *PDF* dosyalarından oluşmaktadır.

DergiPark akademik internet sitesindeki 654.674 makale arasından, 2023 yılına ait sağlık bilimleri alanındaki dergilerde yayınlanan, "İstatistiksel Analiz" başlığını içeren araştırma makaleleri filtrelenmiştir. Filtreleme sonucu DergiPark sitesinden toplam 410 adet makalenin *PDF* dosyaları indirilmiştir.

PubMed platformu tıp kütüphanesi olduğundan sadece sağlık bilimleri alanında yayınlanan makaleleri içermektedir. PubMed internet sitesinden ise "Statistical Analyses" konu başlığını içeren, 2023 yılına ait, metin ulaşılabilirliği "free full text" olan ve makale tipi "clinical trial" olan makaleler filtrelenmiştir. Filtreleme sonucu *PDF* dosyalarına ulaşılabilen 305 adet makale indirilmiştir.

6.1.1. DergiPark platformundan makaleleri kaydetme otomasyonu

DergiPark Akademik internet sitesinden, 2023 yılına ait sağlık bilimleri alanındaki dergilerde yayınlanan, "İstatistiksel Analiz" içeren araştırma makaleleri filtrelenmiştir. Resim 6.1'de görüldüğü gibi filtreleme sonucu 410 adet makale bulunmaktadır. Bu aşamadan sonra *UiPath* sürecine geçilmiştir.

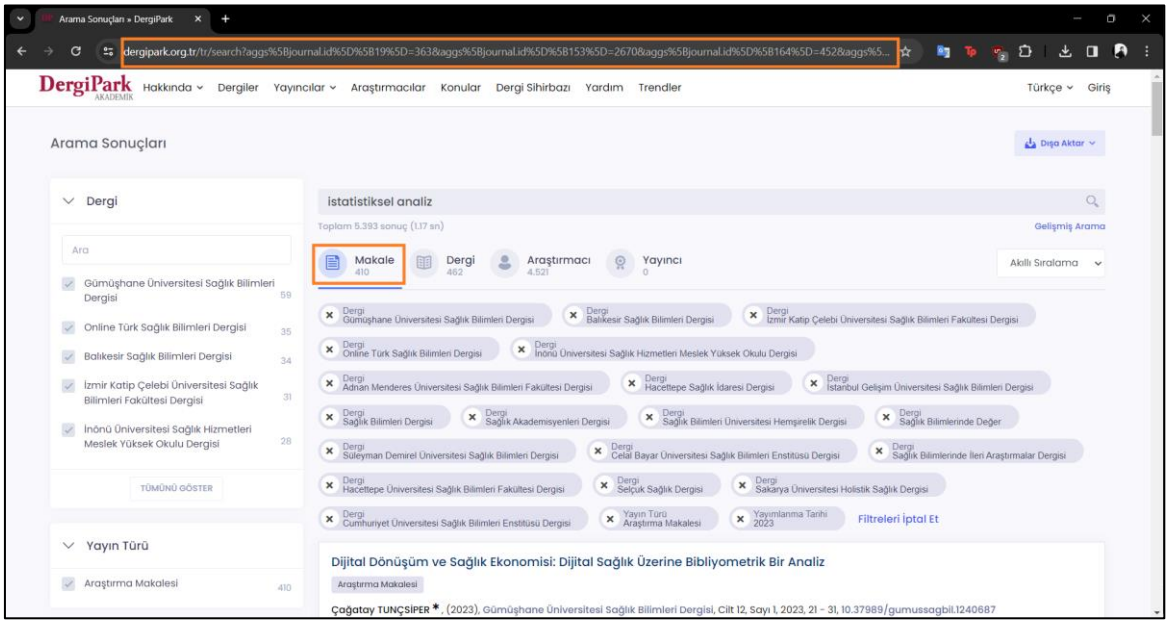
DergiPark platformundan filtrelenerek seçilen sağlık bilimleri alanındaki dergiler ve makale sayıları Çizelge 6.1'de verilmiştir.

Çizelge 6.1. DergiPark platformundan filtrelenen dergiler ve makale sayıları

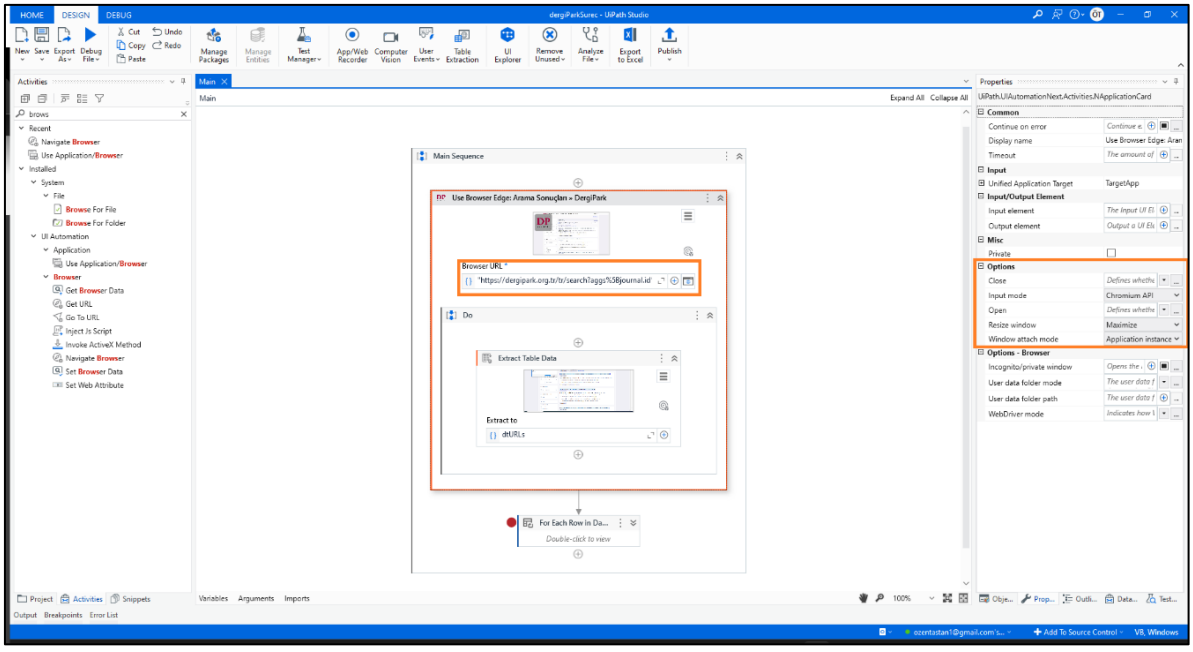
Dergi Adı	Makale Sayısı
Gümüşhane Üniversitesi Sağlık Bilimleri Dergisi	59
Online Türk Sağlık Bilimleri Dergisi	35
Balıkesir Sağlık Bilimleri Dergisi	34
İzmir Katip Çelebi Üniversitesi Sağlık Bilimleri Fakültesi Dergisi	31
İnönü Üniversitesi Sağlık Hizmetleri Meslek Yüksek Okulu Dergisi	28
Adnan Menderes Üniversitesi Sağlık Bilimleri Fakültesi Dergisi	24
Hacettepe Sağlık İdaresi Dergisi	23
İstanbul Gelişim Üniversitesi Sağlık Bilimleri Dergisi	22
Sağlık Bilimleri Dergisi	19
Sağlık Akademisyenleri Dergisi	18
Sağlık Bilimleri Üniversitesi Hemşirelik Dergisi	17
Sağlık Bilimlerinde Değer	16
Süleyman Demirel Üniversitesi Sağlık Bilimleri Dergisi	15
Celal Bayar Üniversitesi Sağlık Bilimleri Enstitüsü Dergisi	13
Cumhuriyet Üniversitesi Sağlık Bilimleri Enstitüsü Dergisi	12
Sağlık Bilimlerinde İleri Araştırmalar Dergisi	12
Hacettepe Üniversitesi Sağlık Bilimleri Fakültesi Dergisi	11
Selçuk Sağlık Dergisi	11
Sakarya Üniversitesi Holistik Sağlık Dergisi	10

Resim 6.1'de işaretlenen *URL* kısmında, robotun gideceği adres yer almaktadır. Bu sebeple *URL* kopyalanıp, Resim 6.2'de "Use Application Browser" aktivitesi içerisindeki "Browser

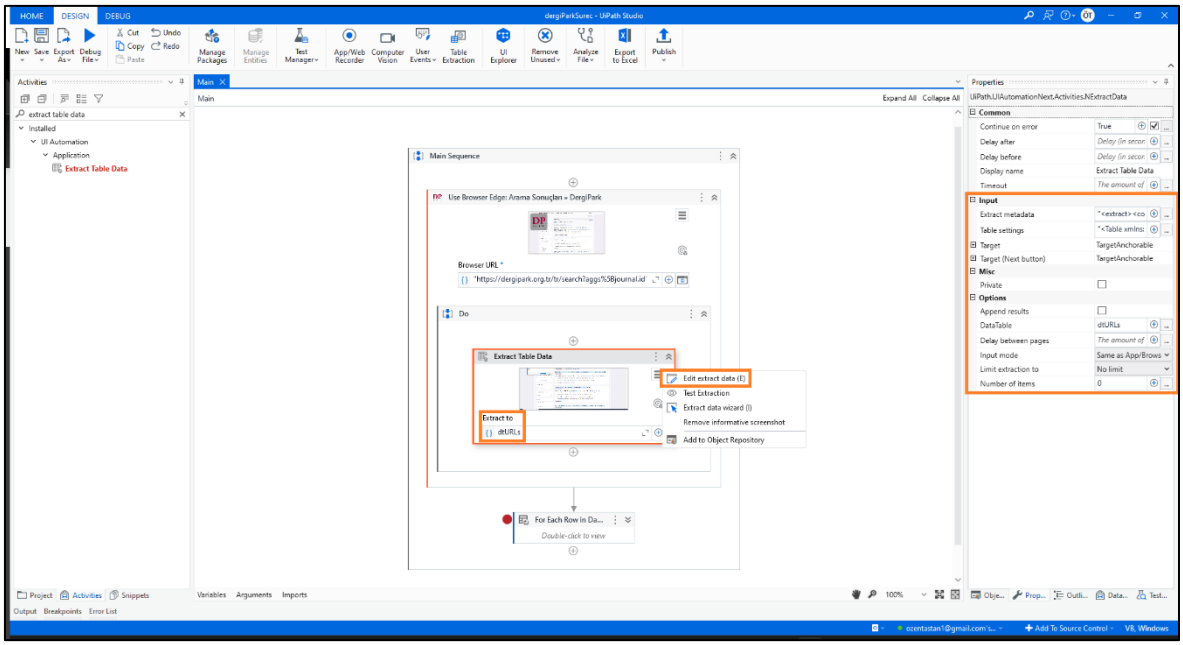
URL” bölümüne çift tırnak içerisinde yazılmaktadır. Böylece süreç çalıştırıldığında robot belirtilen *URL*’e gitmektedir.



Resim 6.1. DergiPark platformu filtreleme aşaması

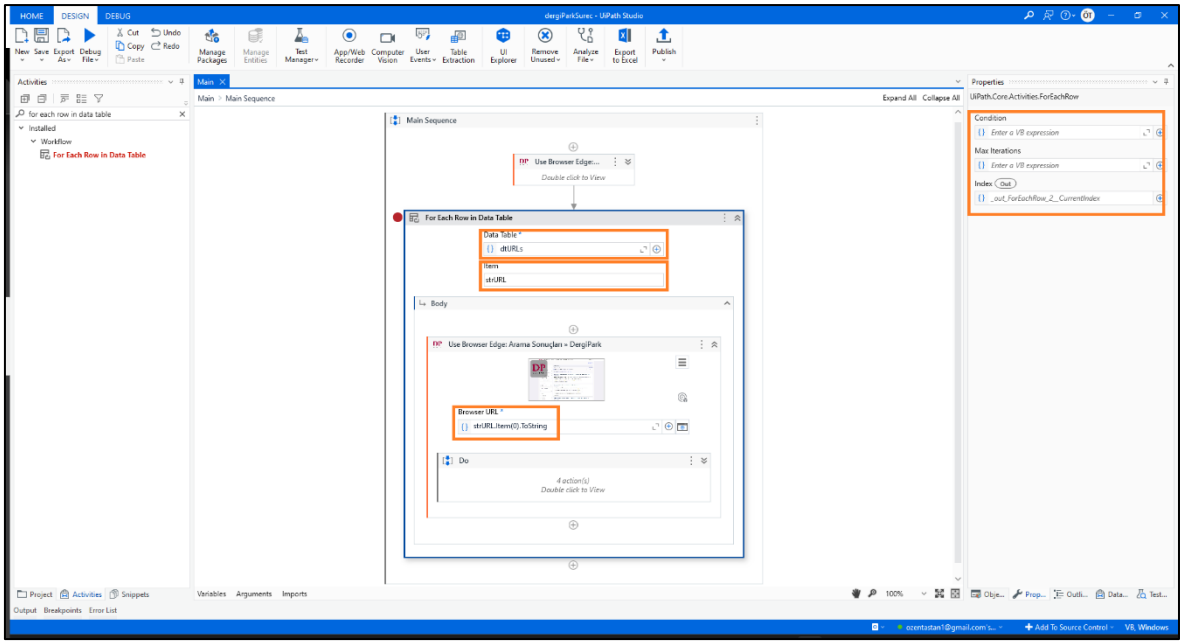


Resim 6.2. DergiPark süreci *URL* adresine gitme aşaması



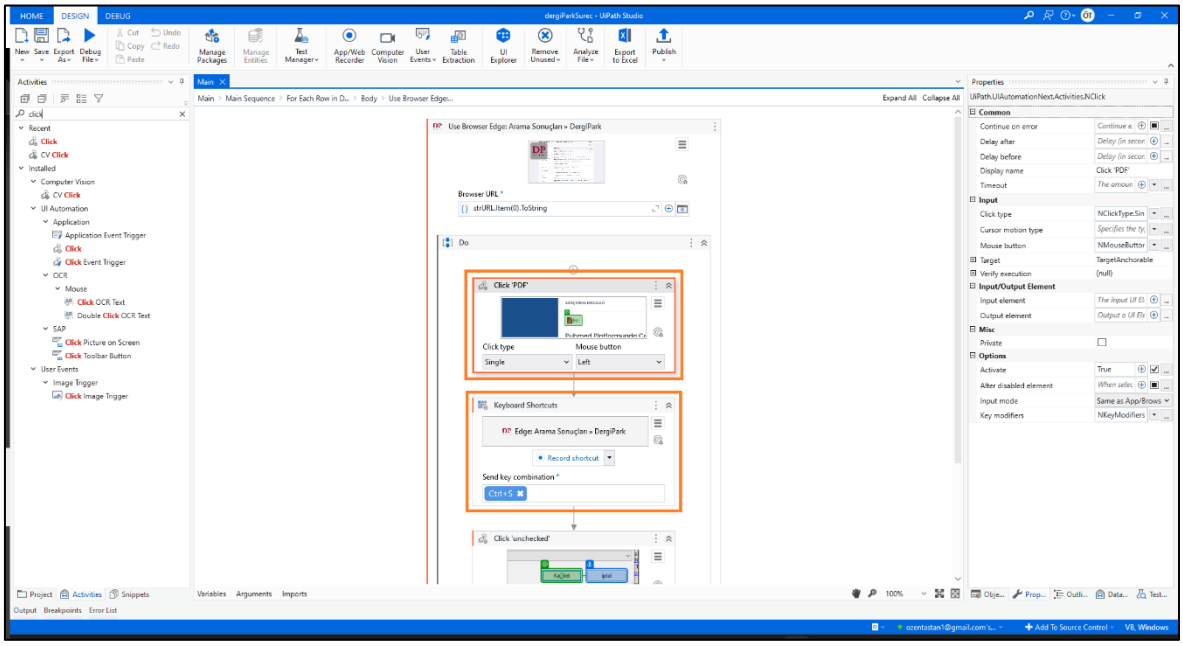
Resim 6.3. DergiPark süreci makalelerin *URL*'lerini alma aşaması

Resim 6.3'te robot, belirtilen *URL*'e gitmiştir ve "Extract Table Data" aktivitesi ile tüm makalelerin *URL*'leri alınarak "dtURLs" adındaki bir "DataTable" üzerine kaydedilmektedir.



Resim 6.4. DergiPark süreci *URL*'den *PDF* indirme birinci aşama

Resim 6.4'te tüm makalelerin *URL*'lerinden oluşan "dtURLs" tablosunda yer alan her bir makalenin, *URL* adresine giderek *PDF* indirmek için ilk adım tamamlanmaktadır. Bu adımda "Data Table" kısmına oluşturulan "dtURLs", "Item" kısmına ise kaydedilen bu *URL*'lerin sırasıyla açılması ve işlemlerin yapılabilmesi için oluşturulmuş "strURL" değişkeni girilmektedir. *URL*'lerin tekrar açılması için "Use Application Browser" aktivitesi kullanılmıştır. "Browser *URL*" bölümüne girilen "strURL.Item(0).ToString" kodu her bir makalenin *URL* adresinin karakter tipine çevrilerek ilgili adrese gidilmesi için yazılmıştır.

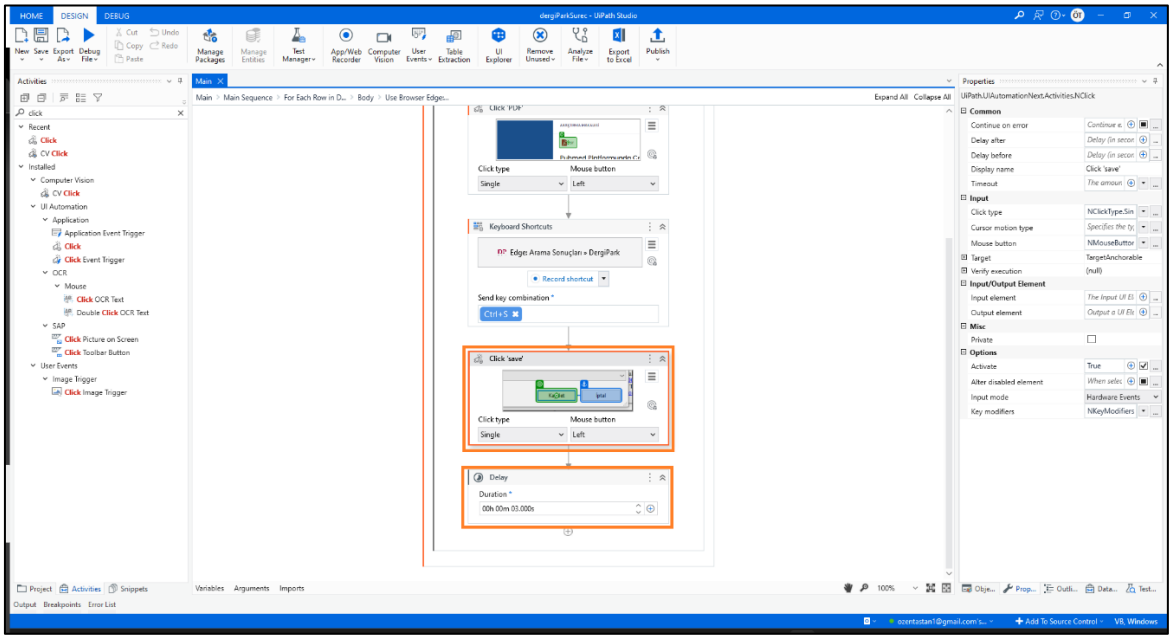


Resim 6.5. DergiPark süreci *PDF* indirme ikinci aşama

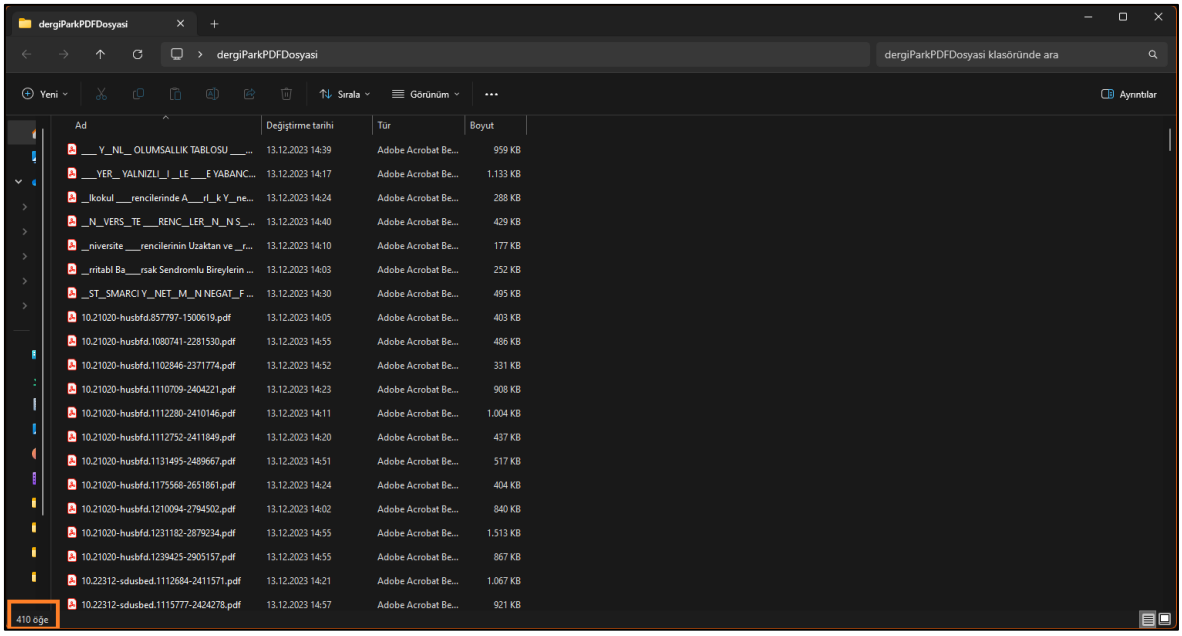
Resim 6.5'te her bir açılan "strURL" değişkeni için sayfadaki *PDF* logosuna tıklanması için "Click" aktivitesi kullanılmaktadır. Robot *PDF* logosuna sadece 1 kere farenin sol butonunu kullanacağından, "Click type" bölümü "Single" ve "Mouse button" bölümü ise "Left" seçilmektedir. Bu işlem sonrasında robot *PDF* sayfasına ulaşmaktadır. *PDF* sayfası açıldıktan sonra "Keyboard Shortcuts" aktivitesi ve "Ctrl+S" kısayolu ile *PDF* dosyası oluşturulan klasör içerisine kaydedilmiştir.

Resim 6.6'da robotun *PDF*'i kaydetmesi için "Click" aktivitesi kullanılmaktadır. "Kaydet" butonuna 1 kere farenin sol butonuna tıklaması gerekeceğinden, "Click type" bölümü "Single" ve "Mouse button" bölümü ise "Left" seçilmektedir. Bu işlem sonrasında robot PDF dosyasını seçilen klasöre kaydetmektedir. Tüm bu aşamalardan sonra DergiPark

platformundan makalelere ait *PDF*'lerin indirilme aşaması tamamlanmış olmaktadır. Resim 6.7'de robotun *PDF* dosyalarını klasöre kaydettiğine ilişkin görsel bulunmaktadır.



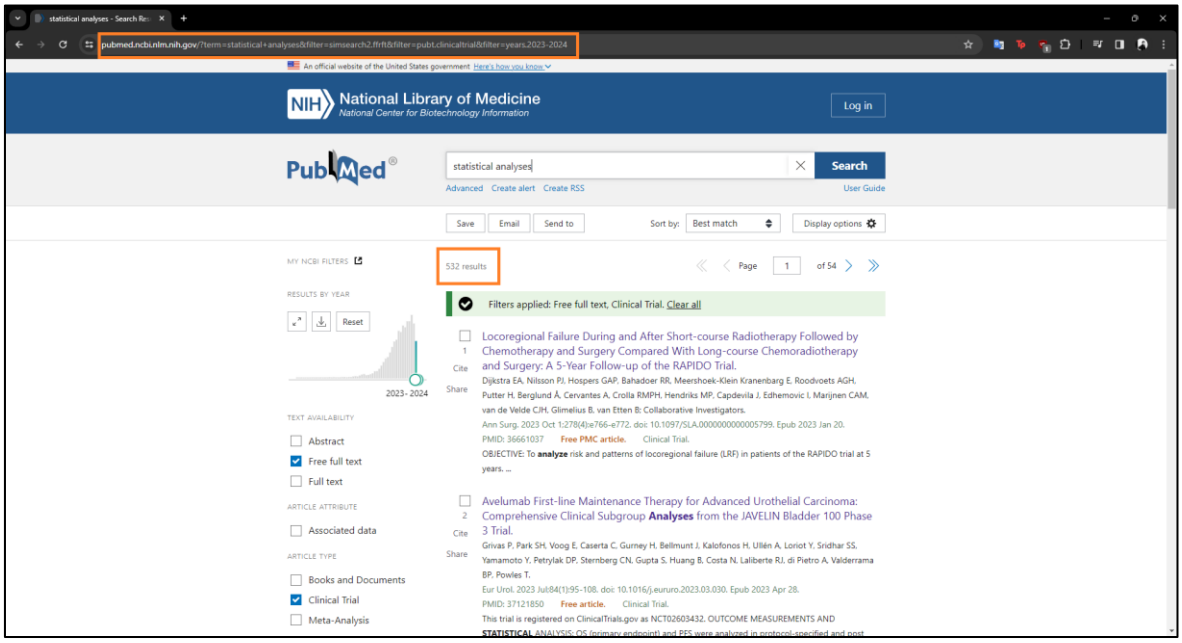
Resim 6.6. DergiPark süreci *PDF* kaydetme aşaması



Resim 6.7. DergiPark süreci *PDF*'lerin kaydedildiği klasör

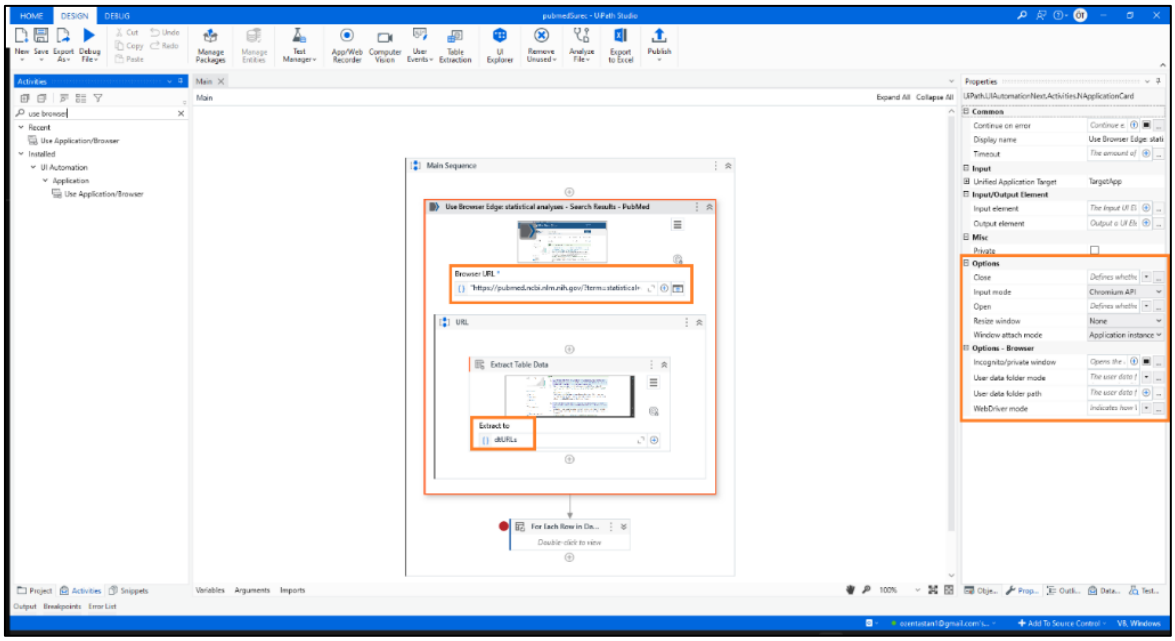
6.1.2. PubMed platformundan makaleleri kaydetme otomasyonu

PubMed uluslararası tıp kütüphanesinden, “statistical analyses” içeren, 2023 yılına ait, metin ulaşılabilirliği “free full text” olan ve makale tipi “clinical trial” olan makaleler filtrelenmiştir. Resim 6.8’de görüldüğü gibi filtreme sonucu 532 adet makale bulunmaktadır. Fakat *RPA* robotunun indirme aşamasında tüm linklere tıklayamamasından dolayı 305 adet makaleye erişilebilmiştir. Bu aşamadan sonra PubMed *Uipath* sürecine geçilmiştir.



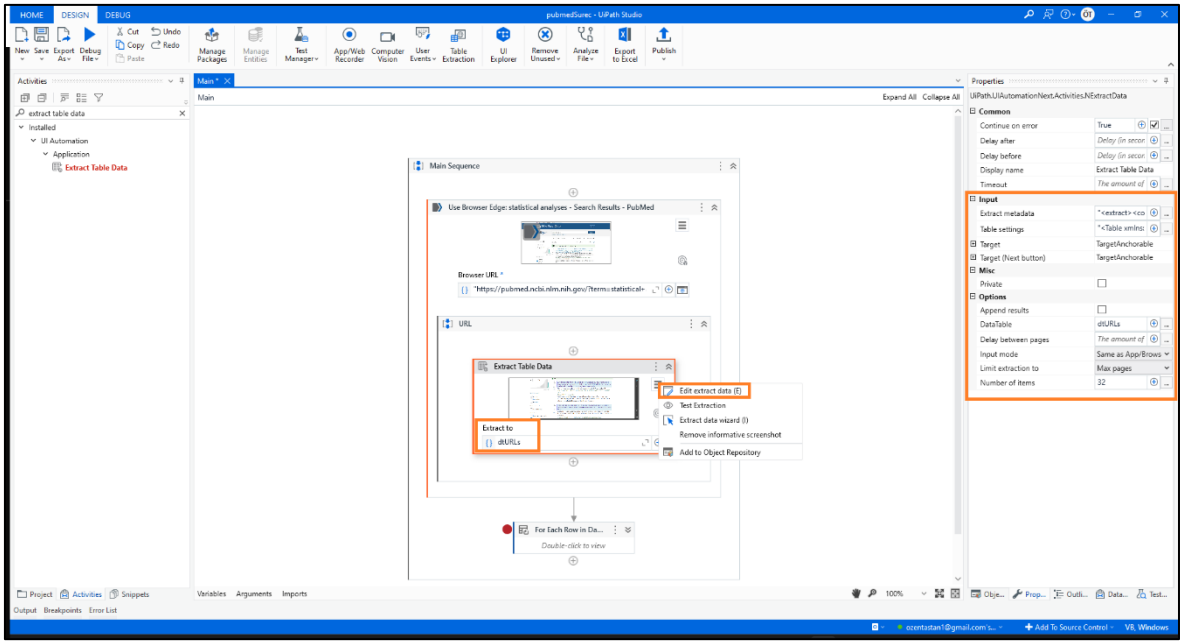
Resim 6.8. PubMed platformu filtreleme aşaması

Resim 6.8’de işaretlenen *URL* kısmında, robotun gideceği adres yer almaktadır. Bu sebeple *URL* kopyalanıp, Resim 6.9’de “Use Application Browser” aktivitesi içerisindeki “Browser URL” bölümüne çift tırnak içerisinde yazılması gerekmektedir. Böylece süreç çalıştırıldığında robot belirtilen *URL*’e gitmektedir.

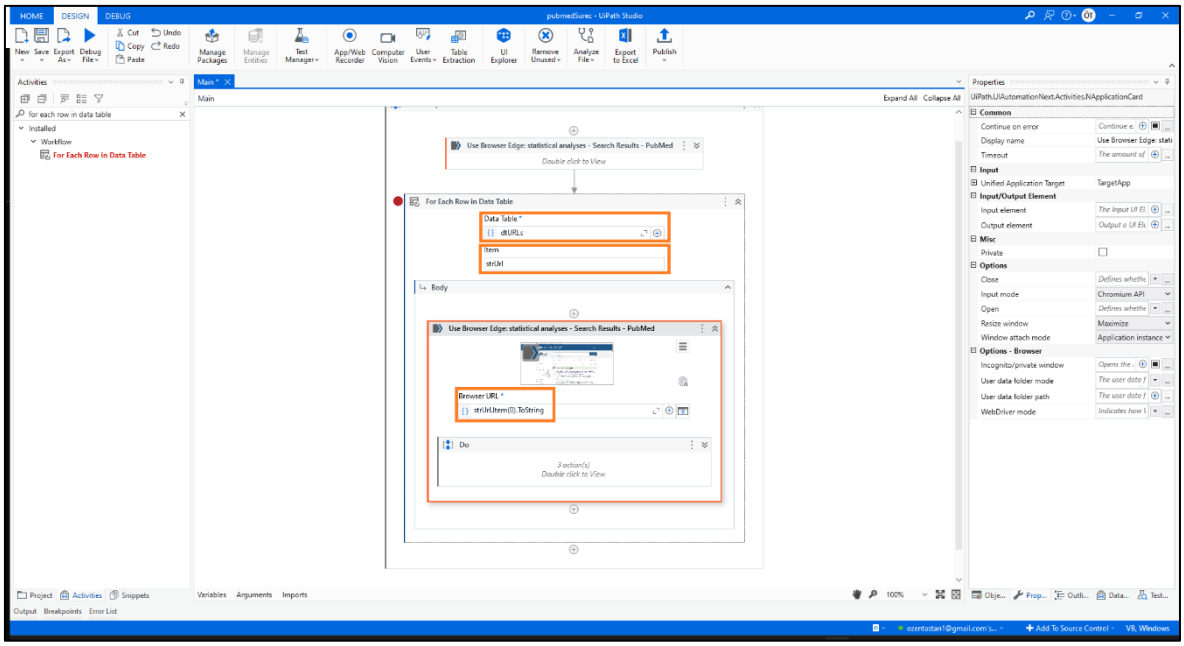


Resim 6.9. PubMed süreci *URL* adresine gitme aşaması

Resim 6.10'da robot, belirtilen *URL*'e gitmektedir ve "Extract Table Data" aktivitesi ile tüm makalelerin *URL*'leri alınarak "dtURLs" adındaki bir "DataTable" üzerine kaydedilmiştir.

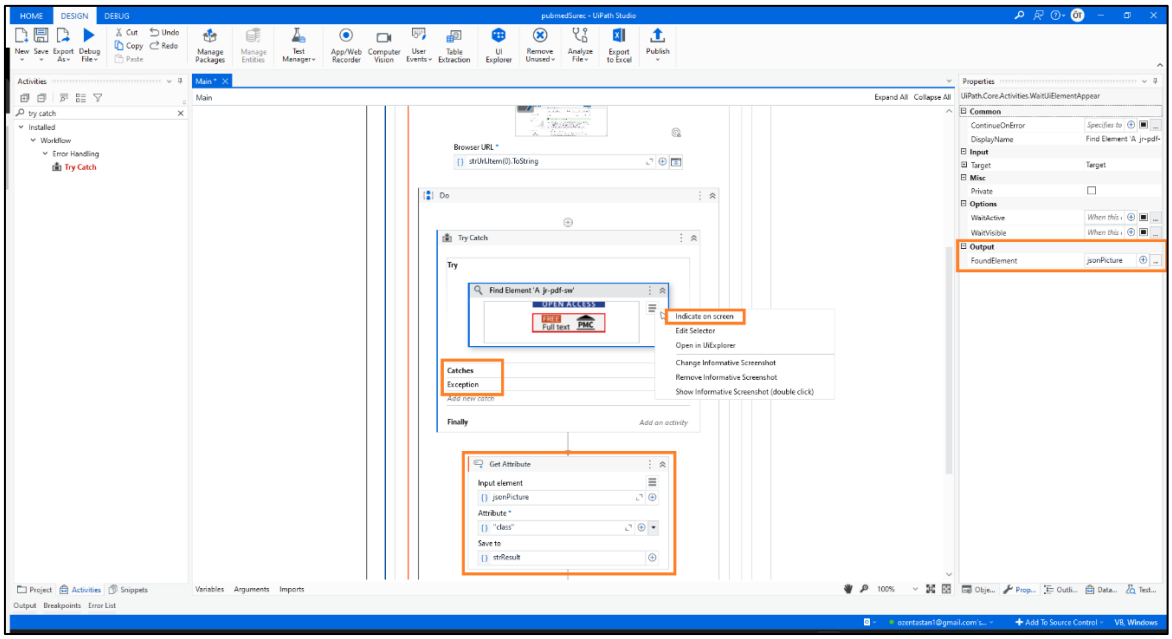


Resim 6.10. PubMed süreci makalelerin *URL*'lerini alma aşaması



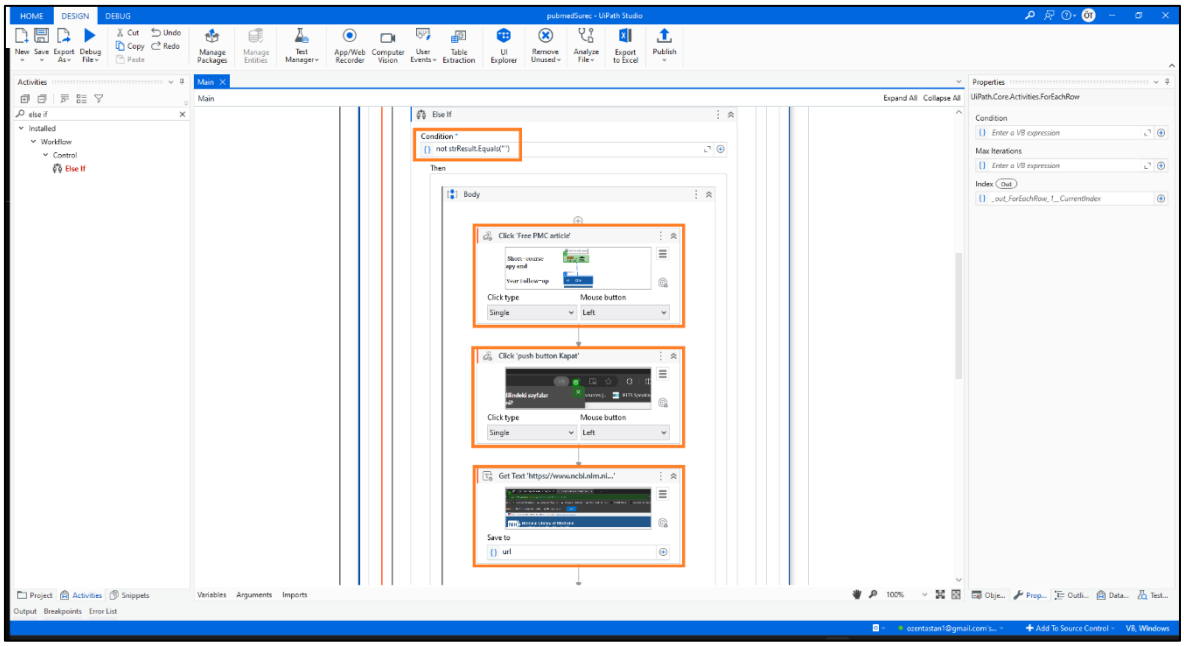
Resim 6.11. PubMed süreci *URL*'den *PDF* indirme birinci aşama

Resim 6.11'de makalelerin *URL*'lerinden oluşan dtURLs tablosunda yer alan her bir makalenin *URL* adresine giderek *PDF* indirmek için ilk adım tamamlanmaktadır. Bu adımda "Data Table" bölümüne "dtURLs" değişkeni, "Item" bölümüne ise kaydedilen *URL*'lerin sırasıyla açılması ve gerekli işlemlerin yapılabilmesi için oluşturulmuş "strURL" değişkeninin tanımlanması gerekmektedir. *URL*'lerin tekrar açılması için "Use Application Browser" aktivitesi kullanılmıştır. "Browser URL" bölümüne girilen "strURL.Item(0).ToString" kodu her bir makalenin *URL* adresinin karakter tipine dönüştürülerek ilgili adrese gidilmesi için yazılmıştır.



Resim 6.12. PubMed süreci *PDF* linklerine tıklama kontrolü aşaması

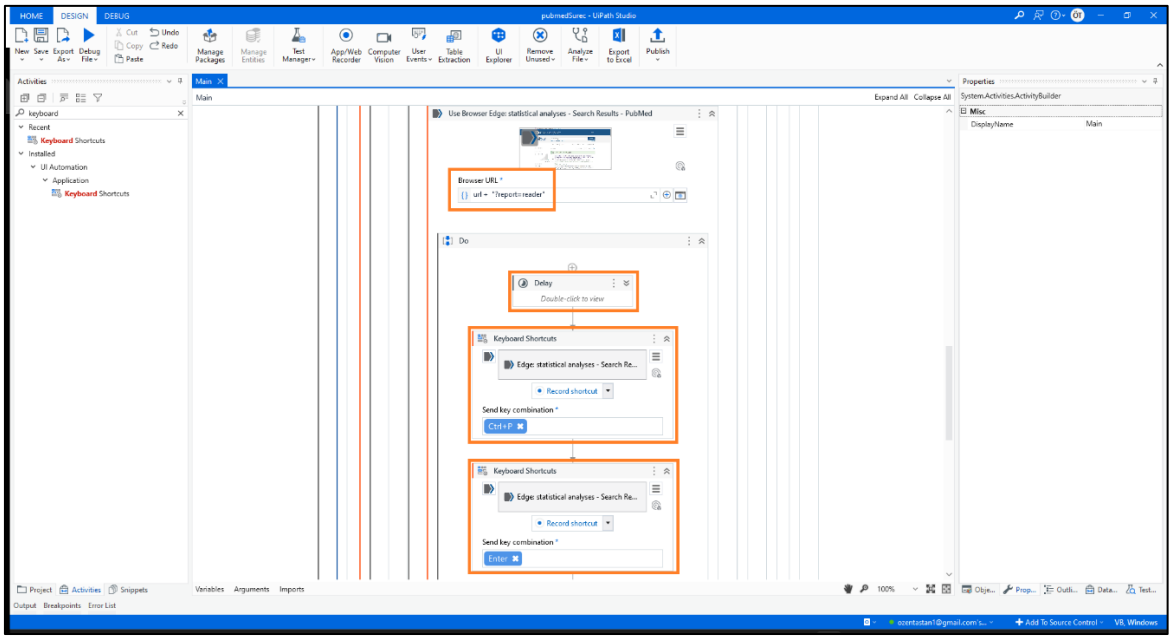
Makale *URL*'leri sayfasındaki "Full Text Links" başlığı altındaki farklı *PDF* indirme butonları mevcuttur. Bu butonlar kullanılarak indirme gerçekleştirilebilmektedir. Resim 6.12'de "Try Catch" aktivitesi içerisinde "Find Element" aktivitesi ile "UiElement" "jsonPicture" değişkenine kaydedilmektedir. "Try catch" aktivitesi içerisinde yapılmasının sebebi ilgili görsel sayfada yer alıyorsa "jsonPicture" değişkenine kaydetmektedir. "Catches" bölümünde görselin sayfada yer almaması durumunda log kayıtlarına hata mesajı basılarak log kayıtlarına hata mesajı yazdırılmaktadır ve bu durum sürecin hatasız devam etmesini sağlamaktadır. "Get Attribute" aktivitesi ile *PDF*'nin indirilebilmesi için seçilecek yol doğrulanmaktadır. "Input element" bölümüne kaydedilen "jsonPicture" değişkeni, "Attribute" bölümüne "class" özelliğinden gidileceği belirtilmektedir ve "strResult" olarak kaydedilmektedir.



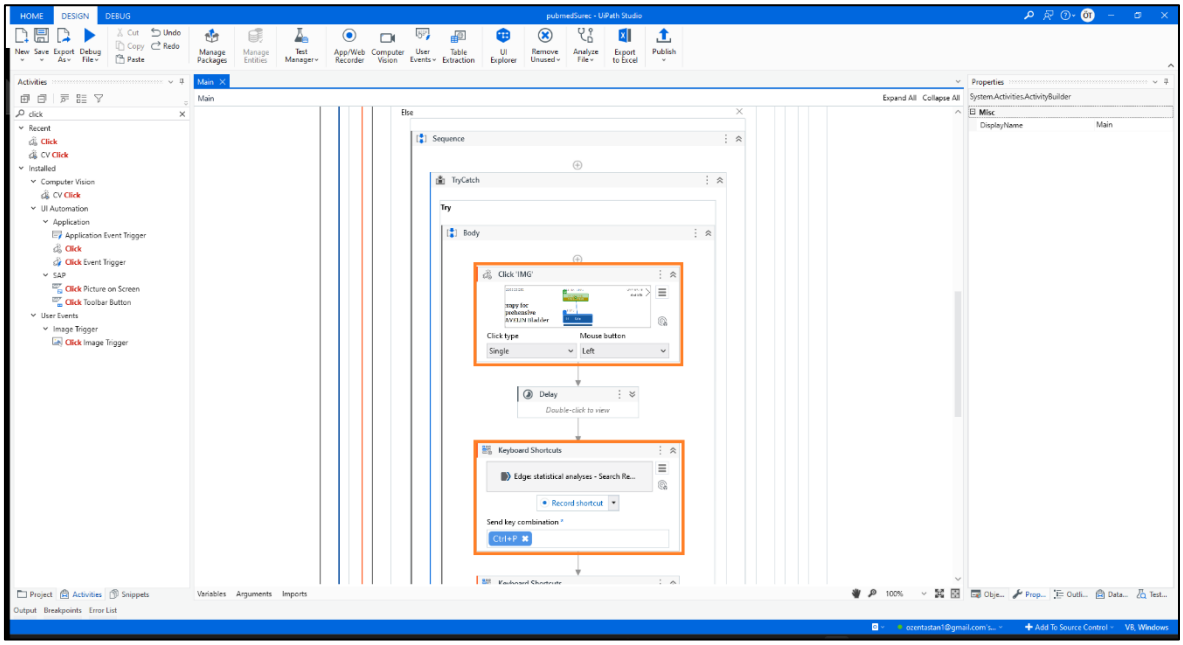
Resim 6.13. PubMed süreci *PDF* indirme linkine tıklama aşaması

Resim 6.13'te "class" özelliğinin kaydedildiği "strResult" değişkeni boş gelmemiş ise indirme işlemlerine devam edilmektedir. Buradaki "not strResult.Equals("")" kodu değişkenin boş gelip gelmediğinin kontrolü için yazılmaktadır. Değişken dolu geldiyse, robotun link görseline tıklaması için "Click" aktivitesi kullanılmıştır. İkinci "Click" aktivitesi internet sayfasında çıkan "pop up"ların kapatılması için eklenmektedir. *PDF*'lere direkt erişim olmadığından dolayı, "Get Text" aktivitesi ile *PDF* link görseline tıklandıktan sonraki *URL*'den kısmını "URL" değişkenine kaydetmektedir.

PDF indirme linki sayfasındaki "PDF" butonunu robot göremediğinden, sayfa incelendiğinde tüm *PDF*'ler için *URL* sonuna sabit olarak "?=report=reader" eklenmektedir. Resim 6.14'de kaydedilen her bir "URL" değişkenine "?=report=reader" eklenerek açılmaktadır. "Keyboard Shortcuts" aktivitesi ile robota "CTRL+P", "ENTER" ve "ENTER" yaptırılarak *PDF*'lerin kaydedilmesi aşaması tamamlanmıştır. Bu süreçte robotun yüksek hızından ve internetin olası düşük hızına bağlı olarak bazı *PDF*'lerin kaydedilemediği görülmüştür ve bunu engellemek için 4'er saniyelik "Delay" aktivitesi eklenmiştir.



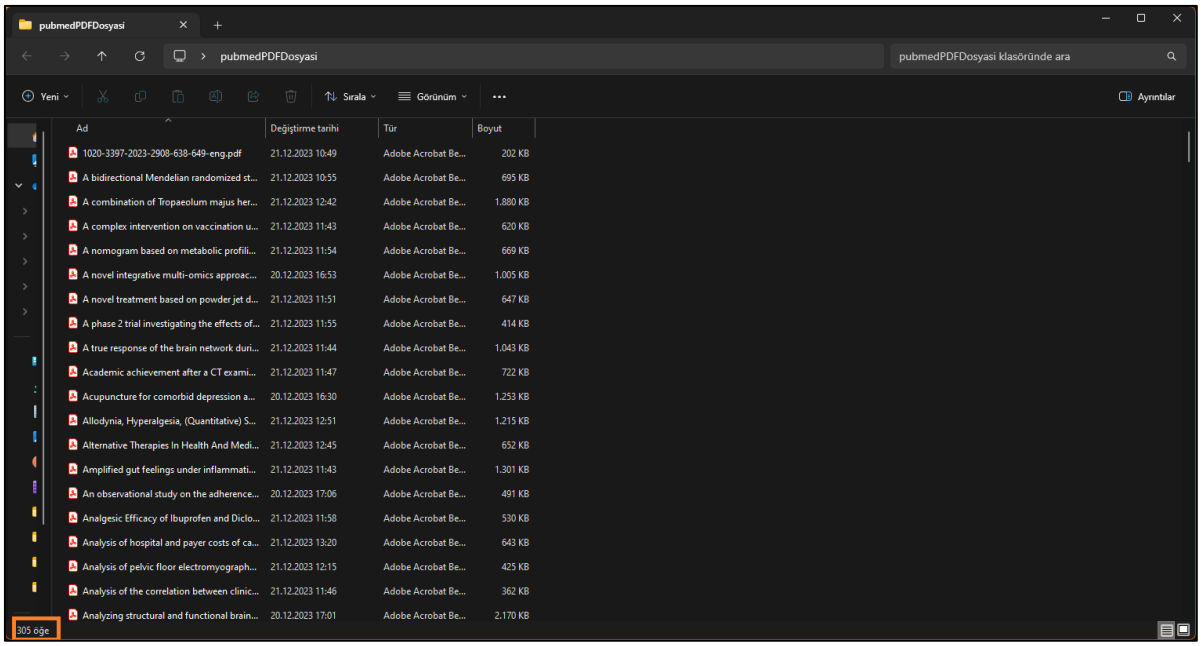
Resim 6.14. PubMed süreci *PDF* kaydetme linki birinci aşama



Resim 6.15. PubMed süreci *PDF* kaydetme linki ikinci aşama

Resim 6.15'te diğer *PDF* indirme linkinin iş akışı gözükmektedir. Birinci görsel linkindeki gibi "Click" aktivitesi ile tıklanmaktadır. "Keyboard Shortcuts" aktivitesi ile robota "CTRL+P", "ENTER" ve "ENTER" yaptırılarak *PDF*'ler kaydedilmektedir. İkinci görsel linkinde bazı durumlarda hata aldığından dolayı "Try Catch" içerisine eklenmiştir. Robotun yüksek hızından ve olası internet yavaşlığından dolayı ise 4'er saniyelik "Delay" aktivitesi

eklenmiştir. Açılan tarayıcı sayfalarının işlemden sonra kapatılması için ise “Close Tab” aktivitesi kullanılmıştır. Tüm bu aşamalardan sonra PubMed platformundan makalelere ait *PDF*’lerin indirilme aşaması tamamlanmış olmaktadır. Resim 6.16’de robotun *PDF* dosyalarını klasöre kaydettiğine dair görsel bulunmaktadır.



Resim 6.16. PubMed süreci *PDF*’lerin kaydedildiği klasör

6.2. Metin Madenciliği Uygulaması

Çalışmada, *Python* programlama dili kullanılmıştır. *Python* programlama dilinin tercih edilmesinin sebebi ara yüzünün diğer programlama dillerine göre daha kullanışlı olması ve işlemlerin daha kolay yapılabilmesidir. Aynı zamanda metin madenciliği için çok geniş bir kütüphaneye sahiptir.

Metin madenciliği için *Anaconda Navigator* üzerinden *Jupyter Notebook IDE*’si kullanılmıştır. *Jupyter Notebook* kullanılmasının sebebi ise kod bloklarının nasıl çalıştığının kodun hemen altında gözükmesidir. *Jupyter Notebook*, birçok programlama dili ile ortak çalışma alanı sağlayan açık kaynak kodlu bir programdır.

Çalışma kapsamında *Python* programında “PyPDF2”, “re”, “pandas”, “os” ve “WordCloud” kütüphaneleri kullanılmıştır. Bu kütüphanelere ilişkin açıklamalar izleyen alt bölümlerde yer verilmiştir.

6.2.1. PyPDF2 kütüphanesi

2005 yılında Mathieu Fenniak tarafından PyPDF *PDF* araç seti olarak piyasa sürülmüştür. PyPDF’nin *PDF* standartlarından çıkan *PDF*’leri okuyamaz duruma gelmesiyle 2011 yılının sonunda “PyPDF2” ortaya çıkmıştır. “PyPDF2”, oldukça geniş bir yelpazedeki değişken *PDF* formatlarındaki örnekleri okuyabilmektedir ve işleyebilmektedir (PyPDF2, 2008).

“PyPDF2” kütüphanesi, *PDF* dosyalarının sayfalarını bölme, birleştirme, dönüştürme ve bilgi çıkarma gibi işlemlerin yapılabileceği ücretsiz ve açık kaynaklı bir *Python* kütüphanesidir. *PDF*’lerden metin verilerinin çıkarılması içinde kullanılmaktadır.

Çalışma kapsamında “PyPDF2” kütüphanesi ile, *PDF*’ler açılmış ve *PDF*’ler içerisinde belirli işlemler yapılmıştır.

6.2.2. re kütüphanesi

“re” (regular expresiones) yani düzenli ifadeler, tüm programlama dillerinin bir parçasıdır. Karmaşık metinleri düzenlemek veya bölmek, metinden bilgi çıkarmak için kullanılmaktadır. Projelerde re kütüphanesi ile gerekli durumlarda peş peşe birden fazla if-else yapısının kullanılması ortadan kaldırılmaktadır (Toprak, 2019).

Çalışma kapsamında re kütüphanesinde “re.search” fonksiyonu ile aratılan istatistiksel yöntemlerin *PDF* içerisinde yer alıp almadığı tespit edilmiştir.

6.2.3. Pandas kütüphanesi

“Pandas” kütüphanesi 2008 yılından itibaren geliştirilmektedir ve başlangıçta finansal veri analizi için geliştirilmiştir. *Python* programlama dili ile bir istatistiksel hesaplama platformu olmuştur. “Pandas” kütüphanesi verileri depolamak ve işlemek için kullanılan iki boyutlu bir veri yapısı olan veri çerçevesi sınıfı sağlamaktadır (McKinney, 2011).

Çalışma kapsamında "pandas" kütüphanesi ile, veri çerçevesi (data frame) oluşturularak kelime listesi üzerinde işlemlerin yapılması ve yeni veri çerçevelerinin oluşturulması aşamasında kullanılmıştır.

6.2.4. os kütüphanesi

Python ile standart olarak gelen "os" kütüphanesi ile, farklı işletim sistemlerindeki dosya ve dizin işlemleri için ortak bir yol üzerinden iletişim kurulmasını sağlamaktadır. Aksi takdirde örneğin Windows için dizinler arasında "\" işareti, Linux için "/" işareti kullanıldığından ayrı kod yazımı gerektirmektedir.

Çalışma kapsamında "os" kütüphanesinden "os.walk()" ve "os.path()" fonksiyonları, dosya ve klasör işlemleri için kullanılmıştır.

6.2.5. WordCloud kütüphanesi

Python ile "WordCloud" kütüphanesi "pip install WordCloud" kodu çalıştırıldıktan sonra yüklenebilmektedir. "Word Cloud" kelime bulutu olarak adlandırılmaktadır. Metin veri setinden elde edilen her bir kelimenin sıklığı ne kadar fazla ise kelime bulutunda yazı büyüklüğü o kadar büyük olmaktadır.

6.2.6. Jupyter Notebook ile metin madenciliği

Çalışma kapsamındaki makale verilerinin 410 adedi DergiPark platformundan, 305 adedi ise PubMed platformundan elde edilmiştir. Her iki platformdan elde edilen yazım dili Türkçe olan makale *PDF* dosyaları "dergiParkPDFDosyasi" klasörü ve yazım dili İngilizce olan makale *PDF* dosyaları ise "PubMedPDFDosyasi" klasörü olmak üzere iki ayrı klasöre kaydedilmiştir.

Makale *PDF* dosyalarında aratılacak istatistiksel yöntemler Türkçe ve İngilizce olmak üzere iki ayrı *PDF* olarak oluşturulmuştur. Çizelge 6.2'de DergiPark platformundan elde edilmiş makalelerde aratılmak üzere yazım dili Türkçe olan istatistiksel yöntemler ve programlar, Çizelge 6.3'te PubMed platformundan elde edilmiş makalelerde aratılmak üzere yazım dili İngilizce olan istatistiksel yöntemler ve programlar yer almaktadır. Yöntemler ve programlar

listesi, manuel araştırma sonucu ile sağlık bilimleri alanında kullanılabilecek yöntemler ve programlar belirlenerek elde edilmiştir.

Çizelge 6.2. DergiPark makalelerinde aratılan istatistiksel yöntemler ve programlar listesi

Cronbach’s Alfa	Birliktelik Analizi	Freeman Halton Testi	Probit Analiz
Mann Whitney U Testi	Box-Jenkins Testi	Kümeleme Analizi	Madde Analizi
t-testi	Bulanık Kümeleme	Mcnemar Testi	z-testi
Anova	Cox Regresyon	Bland-Altman	Kendall’s Tau Testi
Pearson Korelasyon	Çok Boyutlu Ölçekleme	Diskriminant Analizi	k-medoids
Kolmogorov-Smirnov Testi	Dairesel Veri	Trend Analizi	k-ortalama
Shapiro-Wilk Testi	Destek Vektör Makinesi	Uyum Analizi	Friedman Testi
Spearman Korelasyon	Doğrusal Olmayan Regresyon	Zaman Serileri Analizi	Odds Oranı
Kruskal-Wallis Testi	Doğrusal Programlama	Ancova	Manova
Fisher’s Exact Testi	En Yakın Komşuluk	Rasgele Orman	SPSS
Ki-Kare Testi	Genel Doğrusal Model	Ridge Regresyon	NCSS
Regresyon Analizi	Genetik Algoritma	Robust Regresyon	Minitab
Faktör Analizi	GLMM	Sınıflandırma Yöntemi	Jamovi
Cochran’s Q Testi	Güvenilirlik Analizi	Temel Bileşenler Regresyon	Python
Wilcoxon	Hiyerarşik Kümeleme	Wald-Wolfowitz	Matlab
Dunn’s Testi	Hotelling’s T Testi	Yapay Sinir Ağları	R Studio
Meta Analiz	İkili Kümeleme	Yaşam Tablosu Analizi	Stata
Skewness-Kurtosis	Kanonik Korelasyon	Yoğunluk Tabanlı Kümeleme	Knime
Lojistik Regresyon	Karar Ağacı	Özellik Çıkarımı	Rapidminer
Temel Bileşenler Analizi	Logrank Testi	Naive Bayes	Weka
Roc Analizi-Tanı Testi-Kaplan-Meier Testi	Mantel-Haenszel Testi	Panel Veri Analizi	Angoss

Çizelge 6.3. PubMed makalelerinde aratılan istatistiksel yöntemler ve programlar listesi

t-test	Support Vector Machine	Manova	General Linear Model
Chi-square Test	Naive Bayes	Bland-Altman	Mcnemar Test
Fisher's Exact Test	Decision Tree	Principal Component Analysis	Discriminant Analysis
Cochran's Q Test	k-means	Feature Extraction	Box-Jenkins Test
Mann Whitney U Test	Hierarchical Clustering	Hotelling's T Test	Canonical Correlation
Roc Analysis-Diagnostic Test-kaplan meier test	Time Series Analysis	Skewness-Kurtosis	Correspondence Analysis
Anova	Artificial Neural Network	Multidimensional Scaling	Mantel-Haenszel Test
Odds Ratio	Genetic Algorithm	Association Analysis	Item Analysis
Regression Analysis	Nearest Neighbor	Nonlinear Regression	Linear Programming
Shapiro-Wilk Test	k-medoids	Principal Component Regression	SPSS
Wilcoxon Test	GLMM	Ridge Regression	NCSS
Pearson Correlation	Panel Data Analysis	z-test	Minitab
Dunn's Test	Freeman Halton Test	Wald-Wolfowitz Test	Jamovi
Cox Regression	Circular Data	Fuzzy Clustering	Python
Kolmogorov-Smirnov Test	Lİfe Table Analysis	Density Based Clustering	Matlab
Friedman Test	Logrank Test	Reliability Analysis	R Studio
Ancova	Classification Method	Kendall's Tau Test	Stata
Spearman Correlation	Probit Analysis	Meta Analysis	Knime
Factor Analysis	Robust Regression	Cluster Analysis	Rapidminer
Kruskal-Wallis Test	Trend Analysis	Logistic Regression	Weka
Cronbach's Alpha	Biclustering	Random Forest	Angoss

Çalışmanın veri setini oluşturan ve *RPA* robotu ile DergiPark ve PubMed platformlarından indirtilen makale *PDF*'leri "dergiParkPDFDosyasi" ve "PubMedPDFDosyasi" olmak üzere iki ayrı klasörde toplanmıştır.

Uygulama için yazılan *Python* koduna "PyPDF2", "re", "pandas", "os" ve "WordCloud" kütüphaneleri eklenmiştir. Resim 6.17'de gerekli kütüphanelerin eklenmesine ilişkin kod yer almaktadır. Resim 6.18'de ise "PyPDF2" ve "WordCloud" kütüphaneleri için yüklenmesi gereken paketlere ilişkin kod yer almaktadır.

```
import PyPDF2
import re
import pandas as pd
import os
from wordcloud import WordCloud
```

Resim 6.17. Uygulamada kullanılan kütüphanelere ilişkin kod

```
pip install PyPDF2

pip install WordCloud
```

Resim 6.18. "PyPDF2" ve "WordCloud" kütüphaneleri için yüklenmesi gereken paketler

```
#pdfMethodFileObj = open("C://Users//ozent//OneDrive//Masaüstü//yontemListe.pdf", 'rb')
pdfMethodFileObj = open("C://Users//ozent//OneDrive//Masaüstü//yontemList.pdf", 'rb')
pdfText = PyPDF2.PdfReader(pdfMethodFileObj)
for page in pdfText.pages:
    kelimeListesi = page.extract_text()
kelimeListesi = kelimeListesi.split(',')
kelimeListesi
```

Resim 6.19. Yöntemlerin "kelimeListesi" değişkenine atandığı kod

Resim 6.19'da *PDF* formatındaki Türkçe istatistiksel yöntemler listesi "//yontemListe.PDF" ve İngilizce istatistiksel yöntemler listesi ise "//yontemList.PDF" dosya yolundan açtırılmıştır. "PyPDF2" kütüphanesinin sağladığı "PDFReader()" fonksiyonu ile yöntemlerin bulunduğu *PDF* dosyası okutulmuştur. *PDF*'in tüm sayfaları "for döngüsü" ile

“kelimeListesi” değişkenine kaydedilmiştir. Yöntemler arasında “,” bulunduğundan, “kelimeListesi” değişkeninde bulunan yöntemler “,” ile bölünmüştür. Türkçe ve İngilizce makaleler için kod ayrı ayrı çalıştırılmıştır.

```
path = "C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi"
#path = "C://Users//ozent//OneDrive//Masaüstü//pubmedPDFDosyasi"
list_path = []
for(root, dirs, file) in os.walk(path):
    for f in file:
        if'.pdf' in f:
            list_path.append("C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//" + f)
```

Resim 6.20. *PDF*’lerin dosya yolunun “list_path” değişkenine kaydedilmesine ilişkin kod

Resim 6.20’de DergiPark platformundan elde edilen makalelerin bulunduğu dosya yolu “//dergiParkPDFDosyasi” ve PubMed platformundan elde edilen makalelerin bulunduğu dosya yolu ise “//PubMedPDFDosyasi” olarak belirlenmiştir. “list_path” adında boş bir dizi tanımlandıktan sonra “os” kütüphanesinin fonksiyonlarından birisi olan “os.walk()” fonksiyonu kullanılarak, indirilen makale dosyasında bulunan tüm makalelerin dosya yolu “list_path” değişkenine eklenmektedir. Resim 6.21’de makalelerin dosya yollarının eklendiği “list_path” değişkeninin bir dizi olarak kaydedildiğine ilişkin örnek çıktı bulunmaktadır.

```
print(list_path)
```

```
['C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.21020-husbfd.1080741-2281530.pdf', 'C://Users//ozent//OneDriv
e//Masaüstü//dergiParkPDFDosyasi//10.21020-husbfd.1102846-2371774.pdf', 'C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosy
asi//10.21020-husbfd.1110709-2404221.pdf', 'C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.21020-husbfd.1112280-
2410146.pdf', 'C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.21020-husbfd.1112752-2411849.pdf', 'C://Users//oze
nt//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.21020-husbfd.1131495-2489667.pdf', 'C://Users//ozent//OneDrive//Masaüstü//dergi
ParkPDFDosyasi//10.21020-husbfd.1175568-2651861.pdf', 'C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.21020-husb
fd.1210094-2794502.pdf', 'C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.21020-husbfd.1231182-2879234.pdf',
'C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.21020-husbfd.1239425-2905157.pdf', 'C://Users//ozent//OneDrive//
Masaüstü//dergiParkPDFDosyasi//10.21020-husbfd.857797-1500619.pdf', 'C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyas
i//10.22312-sdusbed.1112684-2411571.pdf', 'C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.22312-sdusbed.1115777-
2424278.pdf', 'C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.22312-sdusbed.1124139-2458929.pdf', 'C://Users//oz
ent//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.22312-sdusbed.1197638-2743468.pdf', 'C://Users//ozent//OneDrive//Masaüstü//der
giParkPDFDosyasi//10.22312-sdusbed.1198878-2748851.pdf', 'C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.22312-s
dusbed.1201491-2759956.pdf', 'C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.22312-sdusbed.1206024-2777846.pdf',
'C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.22312-sdusbed.1209305-2791328.pdf', 'C://Users//ozent//OneDriv
e//Masaüstü//dergiParkPDFDosyasi//10.22312-sdusbed.1265473-3012228.pdf', 'C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDos
yasi//10.22312-sdusbed.1270558-3033737.pdf', 'C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.22312-sdusbed.12799
13-3070933.pdf', 'C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.22312-sdusbed.1287546-3102588.pdf', 'C://User
s//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.22312-sdusbed.1293023-3124765.pdf', 'C://Users//ozent//OneDrive//Masaüst
ü//dergiParkPDFDosyasi//10.22312-sdusbed.1293274-3125879.pdf', 'C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//10.2
```

Resim 6.21. “list_path” değişkeninin dizi olarak kaydedildiğine ilişkin örnek çıktı

```
df = pd.DataFrame(columns = kelimeListesi)
df
```

Resim 6.22. “kelimeListesi” içerisindeki yöntemlerden veri çerçevesinin oluşturma kodu

Resim 6.22'de "pandas" kütüphanesinin sağladığı "DataFrame()" fonksiyonu ile, "kelimeListesi" değişkenine kaydedilen tüm yöntemlerin her birinin bir kolon olarak eklenmesi sağlanmıştır. Örnek "df" değişkeni çıktısında da görüldüğü üzere yöntemler kolon olarak eklenmiştir.

```
df.insert(loc=(len(kelimeListesi)),
          column='Dosya Adı',
          value="")

df.insert(loc=(len(kelimeListesi)+1),
          column='Yöntem / Yazılım',
          value="")
```

Resim 6.23. Dosya adı ve Yöntem / Yazılım kolonunun eklenmesine ilişkin kod

Resim 6.23'te "insert()" fonksiyonu ile, oluşturulan veri çerçevesinin sonuna, "list_path" değişkenin den elde edilecek olan "Dosya Adı" kolonu ve makale içerisinde sadece kullanılan istatistiksel yöntemin ve programlama dilinin yazılacağı "Yöntem / Yazılım" kolonu eklenmiştir.

```
def wordExist(liste, x, pdf):
    path = None
    array = []
    arrayBulunanYontem = []
    basename = os.path.basename(x)[:-4]
    for word in liste:
        index = 0
        for page in pdf.pages:
            text = page.extract_text()
            text=text.lower()
            res_search = re.search(word, text)
            if word in text:
                index = 1
                break
            else:
                index = 0
        if(index==1):
            arrayBulunanYontem.append(word)
        array.append(index)
    array.append(basename)
    array.append(arrayBulunanYontem)
    df.loc[len(df)] = array
```

Resim 6.24. Makale içerisinde istatistiksel yöntemlerin aratıldığı kod

Resim 6.24'te makaleler içerisinde istatistiksel yöntemlerin aratılacağı "wordExist()" metodu oluşturulmuştur. Metin madenciliği ön işleme adımlarından olan tüm harflerin küçük harfe dönüştürülme işlemi "text.lower()" fonksiyonu ile gerçekleştirilmektedir. "wordExist()" metodu ile açılan her bir makale *PDF*'inin içerisinde yöntemler tek tek aratılır. "array" , tüm kolonların tutulacağı bir dizidir. Aratılan kelime *PDF*'de yer alıyorsa oluşturulan diziye "1", yer almıyor ise "0" eklenmektedir. *PDF* in sonuna geldikten sonra ise "basename" ile ilgili *PDF*'in adı "array" değişkenine eklenmektedir. "arrayBulunanYontem" dizisi *PDF*'in içinde bulunan yöntemleri içermektedir ve son indeks olarak "array" dizisine eklenmektedir. *PDF* içerisindeki arama tamamlandıktan sonra oluşturulmuş olan dizi veri çerçevesine yeni bir satır olarak eklenmektedir.

```
def pdfAc(liste_adı):
    tur = 0
    for x in list_path:
        tur +=1
        pdfFileObj = open(x, 'rb')
        pdf = PyPDF2.PdfReader(pdfFileObj)
        wordExist(kelimeListesi,x,pdf)
        print("TAMAMLANAN PDF SAYISI")
        print(tur)
        print("**********************************************************************************")
```

Resim 6.25. *PDF*'lere "wordExist" metodunun uygulandığı kod

Resim 6.25'de her bir makale *PDF*'inin bulunduğu klasörden okunmuş olan her bir *PDF*'in açılacağı "pdfAc()" metodu oluşturulmuştur. "pdfAc()" metodu içerisinde her bir *PDF* için, bir önceki adımda yapılan istatistiksel yöntemlerin aratıldığı ve metin ön işleme adımının yapıldığı "wordExist()" metodundaki işlemler uygulanmaktadır. Metot çalıştırıldığında kaçıncı *PDF*'de olduğunun görülmesi adına çıktıya "Tamamlanan PDF Sayısı" bilgisi yazdırılmıştır.

```
df_last2Columns = df[['Dosya Adı','Yöntem / Yazılım']]
df_last2Columns
```

	Dosya Adı	Yöntem / Yazılım
0	10.21020-husbfd.1080741-2281530	[anova, kruskal, pearson, spss]

Resim 6.26. Veri çerçevesinin son iki kolonu çıktı örneği ve kod

Resim 6.26'da oluşturulan veri çerçevesinin son iki kolonu olan "Dosya Adı" ve "Yöntem/Yazılım" kolonlarını içeren yeni bir veri çerçevesi oluşturulmuştur ve örnek çıktısı yer almaktadır.

Resim 6.27'de oluşturulan "df" veri çerçevesinin satır sonuna tüm yöntemlerin kaç tane kullanıldığına ilişkin "Total" satırı eklenmiştir.

```
In [17]: df.loc['total'] = df.sum(numeric_only=True, axis=0)

In [18]: df

Out[18]:
```

	naive	anova	rasgele orman	regresyon	ki - kare	shapiro - wilk	kruskal wallis	wilcoxon	Dosya Adı	Yöntem / Yazılım
0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	10.21020-husbfd.1131495-2489667	[shapiro -wilk]
1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	10.21020-husbfd.1175568-2651861	[anova]
2	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	10.21020-husbfd.1210094-2794502	[kruskal wallis]
3	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	10.21020-husbfd.1231182-2879234	[ki -kare]
4	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	10.21020-husbfd.1239425-2905157	[regresyon, shapiro -wilk]
5	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	10.22312-sdusbed.1112684-2411571	[anova]
6	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	10.22312-sdusbed.1115777-2424278	[anova]
7	0.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	10.22312-sdusbed.1124139-2458929	[anova, shapiro -wilk, kruskal wallis]
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10.22312-sdusbed.1197638-2743468	[]
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10.22312-sdusbed.1198878-2748851	[]
total	0.0	4.0	0.0	1.0	1.0	3.0	2.0	0.0	NaN	NaN

Resim 6.27. Veri çerçevesi satır sonuna toplam satırının eklendiği kod ve çıktı örneği

Resim 6.28'de "iloc[]" fonksiyonu ile toplam satırı ve kolon isimleri arasında kalan satırlar silinmektedir. "drop()" fonksiyonu ile son 2 kolon olan "Dosya Adı" ve "Yöntem/Yazılım" kolonları silinmektedir. Böylece sadece kullanılan istatistiksel yöntemlere ilişkin toplam sayı bilgisi elde edilmiştir. Örnek sonuç verisi Resim 6.28'de gözükmektedir ve "result.xlsx" Excel dosyasına kaydedilmektedir.

```
df_last_row = df.iloc[-1:]
df_last_row.drop(columns=df_last_row.columns[-2:], axis=1,  inplace=True)

df_last_row
```

	naive	anova	rasgele orman	regresyon	ki -kare	shapiro -wilk	kruskal wallis	wilcoxon
total	0.0	4.0	0.0	1.0	1.0	3.0	2.0	0.0

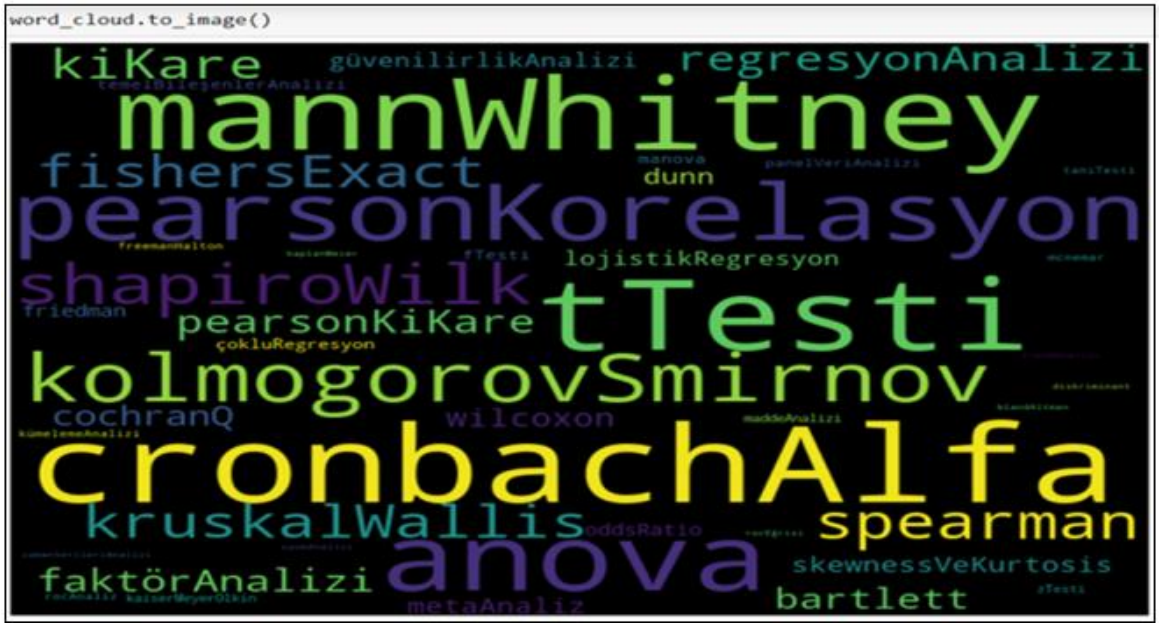
```
df_last_row.to_excel('C://Users//ozent//OneDrive//Masaüstü//result.xlsx')
```

Resim 6.28. Sonuç verinin elde edildiği kod ve çıktı örneği

Resim 6.29'da metin verilerinde aratılan istatistiksel yöntemlere ilişkin sonuç verisinin kaydedildiği Excel dosyası okutulmuştur. "drop()" fonksiyonu ile Excel içerisinde yer alan ve kolon adı bulunmayan "total" yazısının bulunduğu kolon kelime bulutunda kullanılmayacağından kaldırılmıştır. Kelime bulutunun oluşturulması amacıyla gönderilen verinin "string" formatta olması gerekmektedir. Bu nedenle veri çerçevesindeki her bir kolon yani her bir yöntem, kullanılma sayısı kadar "string" formatına "for döngüsü" içerisinde dönüştürülmüştür. "WordCloud" kütüphanesi içerisindeki "WordCloud()" fonksiyonu ile oluşturulacak kelime bulutunun boyutu değiştirilmiştir. "generate()" fonksiyonu ile, dönüştürülen verilerin oluşturduğu "string_values" değişkeni gönderilmiştir. Resim 6.30'da ise "to_image()" ile oluşturulan kelime bulutu görselleştirilmiştir.

```
dfCloud = pd.read_excel('C://Users//ozent//OneDrive//Masaüstü//DPresult2.xlsx')

dfCloud = dfCloud.drop('Unnamed: 0', axis=1)

string_values = ""

for x in dfCloud.columns:
    val = dfCloud[x]
    if(int(val) == 0):
        continue
    for i in range(int(val)):
        string_values = string_values + "," + x
        #print(string_values)

word_cloud = WordCloud(width=700, height=500, collocations=False).generate(string_values)
```

Resim 6.29. Kelime bulutunun oluşturulması için yazılmış kod



Resim 6.30. Kelime bulutunun görsel haline getirildiği kod

7. BULGULAR

DergiPark ve PubMed platformlarından robot yardımıyla indirilen, yazım dili Türkçe ve İngilizce olan tüm makale *PDF*'lerine aratılan istatistiksel yöntemlerin daha doğru bulunması amacıyla, metinlerdeki tüm harfler küçük harfe dönüştürülmüştür. Türkçe ve İngilizce metinlerde aratılmak üzere iki ayrı istatistiksel yöntemler dosyası küçük harflerle yazılarak oluşturulmuştur. Daha sonra bu yöntemler DergiPark makaleleri ve PubMed makaleleri olmak üzere iki *PDF* grubunda aratılmıştır. Aratılan yöntemlerin varlığı ve yokluğuna göre "1" ve "0" olarak bir veri seti oluşturulmuştur. Bu veri setindeki yöntem kolonları altında makalelerdeki varlığına göre oluşturulmuş bir toplam satırı yer almaktadır. Yöntemler ve toplam satırı kullanılarak bir kelime bulutu oluşturulmuştur.

DergiPark platformundan sağlık bilimleri dergilerinde 2023 yılında yayınlanmış, "istatistiksel analiz" içeren araştırma makaleleri filtrelenmiş ve filtreleme sonucu 410 adet makaleye ulaşılmıştır. *Python* üzerinde kod ile makale dosyaları açtırılırken 3 adet *PDF*'si göremediğinden 407 adet makale üzerinde çalışılmıştır. PubMed internet sitesi uluslararası tıp kütüphanesi olduğundan sağlık bilimleri dergileri yer almaktadır. PubMed platformundan 2023 yılında yayınlanmış "statistical analysis" içeren, makale tipi "clinical trial" olan ve makale ulaşılabilirliği "free full text" olan makaleler filtrelenmiş ve filtreleme sonucu 305 adet makaleye ulaşılmıştır. Her iki platform için *RPA* robotu yardımıyla makale *PDF*'lerine ulaşılmış ve bir dosya içerisine kaydedilmiştir.

7.1. Makalelerde İstatistiksel Yöntemlerin Tespit Edilmesi

Çalışma kapsamında, metin madenciliği yöntemi ile DergiPark platformundan 407 adet ve PubMed platformundan 305 adet elde edilmiş makale olmak üzere toplamda 712 adet makale üzerinde çalışılmıştır. Çizelge 7.1'de ve Çizelge 7.2'de DergiPark platformundan elde edilen makalelerde yer alan istatistiksel yöntemlere ve programlara ilişkin dağılım yer almaktadır.

Çizelge 7.1. DergiPark platformundan elde edilen makalelerde yer alan istatistiksel yöntemlere ilişkin dağılım

Yöntem	Frekans	Yöntem	Frekans
Cronbach's Alfa	150	Birliktelik Analizi	0
Mann Whitney U Testi	131	Box-Jenkins Testi	0
t-testi	126	Bulanık Kümeleme	0
Anova	115	Cox Regresyon	0
Pearson Korelasyon	107	Çok Boyutlu Ölçekleme	0
Kolmogorov-Smirnov Testi	87	Dairesel Veri	0
Shapiro-Wilk Testi	78	Destek Vektör Makinesi	0
Spearman Korelasyon	58	Doğrusal Olmayan Regresyon	0
Kruskal-Wallis Testi	54	Doğrusal Programlama	0
Fisher's Exact Testi	52	En Yakın Komşuluk	0
Ki-Kare Testi	45	Genel Doğrusal Model	0
Regresyon Analizi	42	Genetik Algoritma	0
Faktör Analizi	38	GLMM	0
Cochran's Q Testi	22	Güvenilirlik Analizi	0
Wilcoxon	16	Hiyerarşik Kümeleme	0
Dunn's Testi	12	Hotelling's T Testi	0
Meta Analiz	12	İkili Kümeleme	0
Skewness-Kurtosis	12	Kanonik Korelasyon	0
Lojistik Regresyon	10	Karar Ağacı	0
Friedman Testi	8	Kendall's Tau Testi	0
Odds Oranı	8	k-medoids	0
Manova	5	k-ortalama	0
Temel Bileşenler Analizi	5	Logrank Testi	0
Roc Analizi-Tanı Testi-Kaplan-Meier Testi	5	Mantel-Haenszel Testi	0
Panel Veri Analizi	4	Naive Bayes	0

Çizelge 7.1. (devam) DergiPark platformundan elde edilen makalelerde yer alan istatistiksel yöntemlere ilişkin dağılım

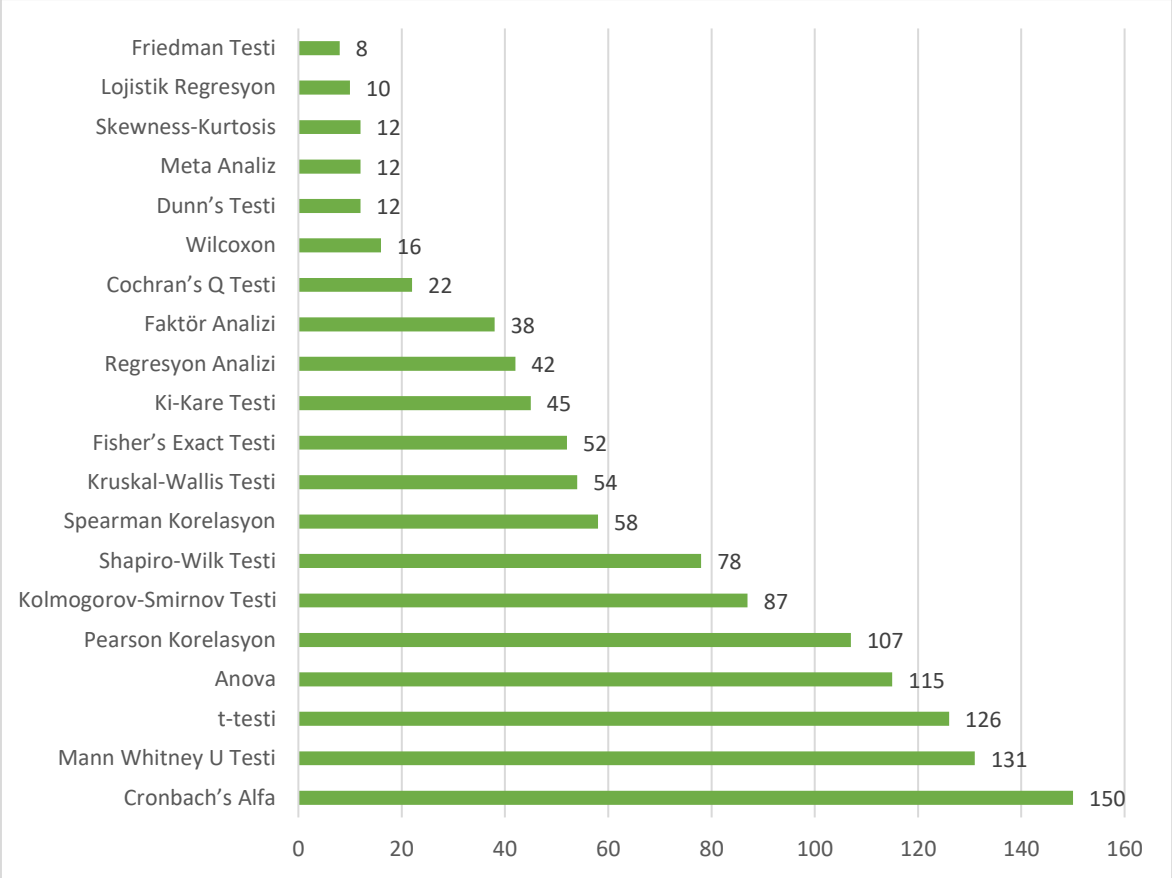
Yöntem	Frekans	Yöntem	Frekans
Madde Analizi	3	Özellik Çıkarımı	0
z-testi	3	Probit Analiz	0
Freeman Halton Testi	2	Rasgele Orman	0
Kümeleme Analizi	2	Ridge Regresyon	0
Mcnemar Testi	2	Robust Regresyon	0
Bland-Altman	1	Sınıflandırma Yöntemi	0
Diskriminant Analizi	1	Temel Bileşenler Regresyon	0
Trend Analizi	1	Wald-Wolfowitz	0
Uyum Analizi	1	Yapay Sinir Ağları	0
Zaman Serileri Analizi	1	Yaşam Tablosu Analizi	0
Ancova	0	Yoğunluk Tabanlı Kümeleme	0

Çizelge 7.2. DergiPark platformundan elde edilen makalelerde yer alan programlara ilişkin dağılım

Program	Frekans	Program	Frekans
SPSS	315	R Studio	0
NCSS	6	Stata	0
Minitab	3	Knime	0
Jamovi	3	Rapidminer	0
Python	1	Weka	0
Matlab	1	Angoss	0

DergiPark platformundan indirilen 407 adet sağlık bilimleri araştırma makalesinde yer alan istatistiksel yöntemler ve programlara ilişkin kullanılma sayıları Excel dosyasına kaydedilmiştir. Kaydedilen Excel dosyasındaki veriler Çizelge 7.1'de ve Çizelge 7.2'de gösterilmiştir. Çizelge 7.1'de görüldüğü üzere, en çok kullanılan beş istatistiksel yöntem

olarak cronbach's alfa değeri 150, mann whitney u testi 131, t-testi 126, anova 110 ve pearson korelasyon 107 kullanım sayılarıyla yer almaktadır. Çizelge 7.2'de ise kullanılan istatistiksel yöntemlerin çoğunlukla *SPSS* programıyla uygulandığı görülmektedir. Şekil 7.1'de DergiPark makalelerinde en çok kullanılan ilk 20 istatistiksel yönteme ilişkin çubuk grafiği yer almaktadır.



Resim 7.1. DergiPark makalelerinde en çok kullanılan ilk 20 istatistiksel yöntem

Çizelge 7.3. PubMed platformundan elde edilen makalelerde yer alan istatistiksel yöntemlere ilişkin dağılım

Yöntem	Frekans	Yöntem	Frekans
t-test	167	Support Vector Machine	1
Chi-square Test	89	Naive Bayes	1
Fisher's Exact Test	77	Decision Tree	1
Cochran's Q Test	75	k-means	1
Mann Whitney U Test	64	Hierarchical Clustering	1

Çizelge 7.3. (devam) PubMed platformundan elde edilen makalelerde yer alan istatistiksel yöntemlere ilişkin dağılım

Yöntem	Frekans	Yöntem	Frekans
Roc Analysis-Diagnostic Test-kaplan meier test	57	Time Series Analysis	1
Anova	50	Artificial Neural Network	1
Odds Ratio	49	Genetic Algorithm	1
Regression Analysis	47	Nearest Neighbor	1
Shapiro-Wilk Test	39	k-medoids	1
Wilcoxon Test	37	GLMM	0
Pearson Correlation	35	Panel Data Analysis	0
Dunn’s Test	20	Freeman Halton Test	0
Cox Regression	18	Circular Data	0
Kolmogorov-Smirnov Test	17	LİIfe Table Analysis	0
Friedman Test	17	Logrank Test	0
Ancova	16	Classification Method	0
Spearman Correlation	15	Probit Analysis	0
Factor Analysis	15	Robust Regression	0
Kruskal-Wallis Test	12	Trend Analysis	0
Cronbach’s Alpha	12	Biclustering	0
General Linear Model	8	Box-Jenkins Test	0
Mcnemar Test	7	Canonical Correlation	0
Discriminant Analysis	5	Correspondence Analysis	0
Random Forest	5	Item Analysis	0
Mantel-Haenszel Test	4	LİInear Programming	0
Meta Analysis	3	Logistic Regression	0
Cluster Analysis	3	Nonlinear Regression	0
Manova	2	Principal Component Regression	0
Bland-Altman	2	Ridge Regression	0

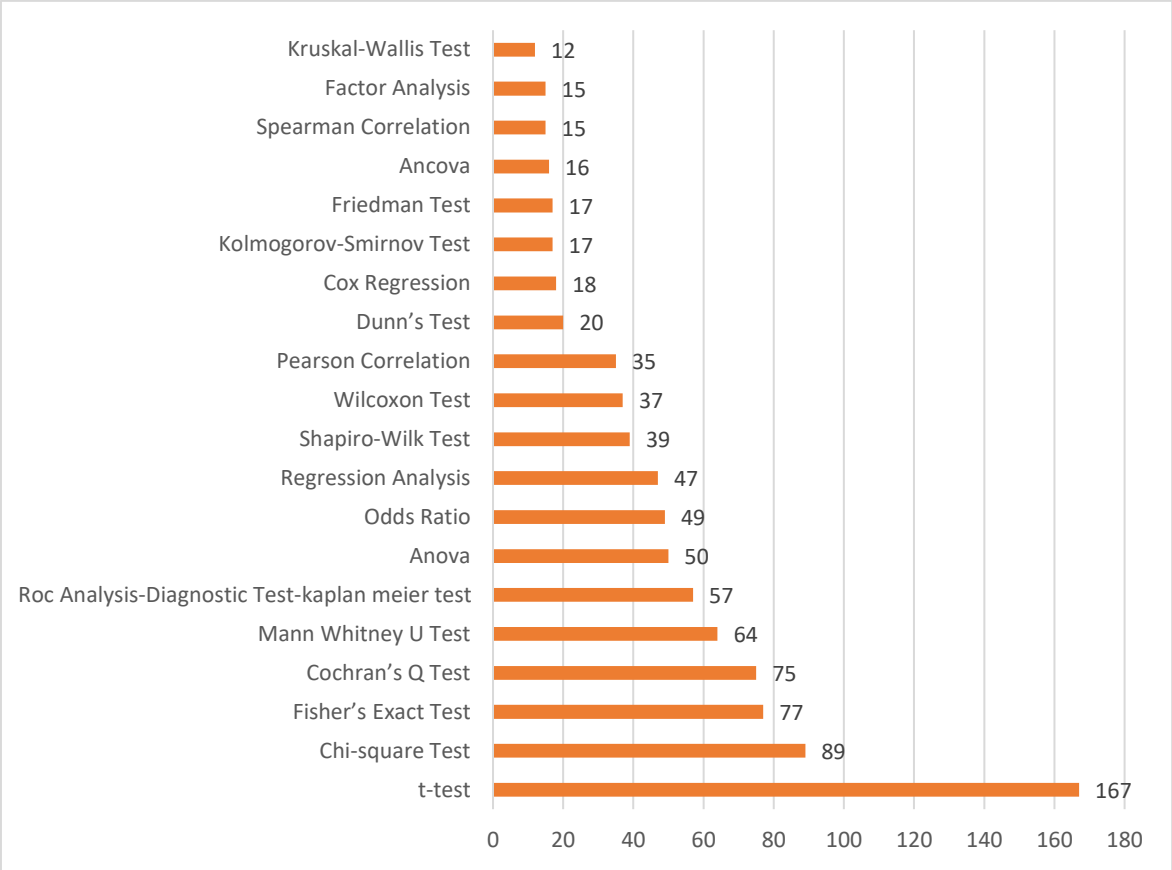
Çizelge 7.3. (devam) PubMed platformundan elde edilen makalelerde yer alan istatistiksel yöntemlere ilişkin dağılım

Yöntem	Frekans	Yöntem	Frekans
Principal Component Analysis	2	z-test	0
Feature Extraction	2	Wald-Wolfowitz Test	0
Hotelling's T Test	2	Fuzzy Clustering	0
Skewness-Kurtosis	2	Density Based Clustering	0
Multidimensional Scaling	2	Reliability Analysis	0
Association Analysis	2	Kendall's Tau Test	0

Çizelge 7.4. PubMed platformundan elde edilen makalelerde yer alan programlara ilişkin dağılım

Program	Frekans	Program	Frekans
SPSS	121	Python	1
Stata	56	Minitab	0
NCSS	5	Knime	0
R Studio	3	Rapidminer	0
Jamovi	2	Weka	0
Matlab	2	Angoss	0

PubMed platformundan indirilen 305 adet sağlık bilimleri klinik araştırma makalesinde yer alan istatistiksel yöntemler ve programlara ilişkin kullanılma sayıları Excel dosyasına kaydedilmiştir. Kaydedilen Excel dosyasındaki veriler Çizelge 7.3'te ve Çizelge 7.4'te gösterilmiştir. Çizelge 7.3'te görüldüğü üzere, en çok kullanılan beş istatistiksel yöntem olarak *t*-test 167, chi-square test 89, fisher's exact test 77, cochran's q test 75 ve mann whitney u test 64 kullanım sayılarıyla yer almaktadır. Çizelge 7.4'te kullanılan istatistiksel yöntemlerin genellikle *SPSS* programıyla uygulandığı görülmektedir. Şekil 7.2'de ise PubMed makalelerinde en çok kullanılan ilk 20 istatistiksel yönteme ilişkin çubuk grafiği yer almaktadır.



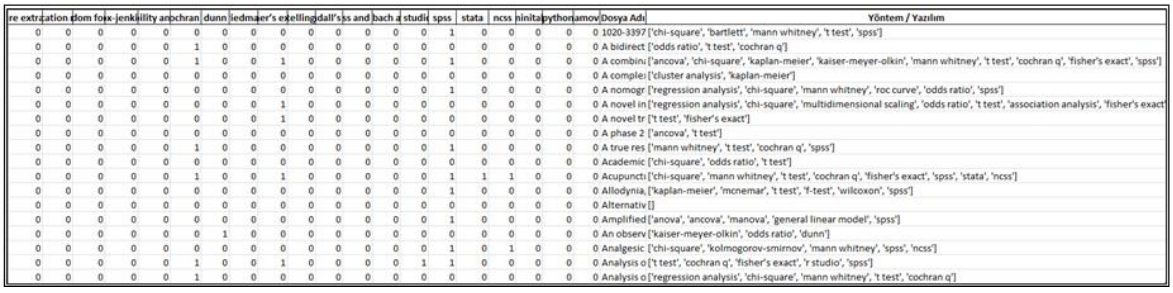
Resim 7.2. PubMed makalelerinde en çok kullanılan ilk 20 istatistiksel yöntem

	alizi	ochran	dunn	edma	aer's e	xtelling	ndall's	ss ve	nbach	studi	spss	ncss	minita	matlat	Dosya Adı	Yöntem / Yazılım
0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	10.37989-gumussagbi	['kolmogorov-smirnov', 'mann whitney', 'meta analiz', 'shapiro-wilk', 't testi', 'skewness ve kurtosis', 'spss']
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10.37989-gumussagbi	[]
2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	10.37989-gumussagbi	['mann whitney', 'spss']
3	0	0	0	0	0	0	0	0	1	0	1	0	0	0	10.38079-igusabder.1	['anova', 'mann whitney', 'spearman', 't testi', 'cronbach alfa', 'spss']
4	0	0	0	0	0	0	0	0	0	0	0	0	1	0	10.38079-igusabder.1	['pearson korelasyon', 'minitab']
5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	10.38079-igusabder.1	['shapiro-wilk', 'wilcoxon', 'spss']
6	0	0	0	0	0	0	0	0	0	0	1	0	0	0	10.38079-igusabder.1	['anova', 'kolmogorov-smirnov', 'mann whitney', 'pearson korelasyon', 'spearman', 'spss']
7	0	0	0	0	1	0	0	0	1	0	1	0	0	0	10.38079-igusabder.1	['mann whitney', 'pearson korelasyon', 'spearman', 'fisher's exact', 'cronbach alfa', 'spss']
8	1	0	0	0	0	0	0	0	1	0	1	0	0	0	10.38079-igusabder.1	['faktör analizi', 'bartlett', 'pearson chi-square', 'temel bileşenler analizi', 'güvenilirlik analizi', 'cronbach alfa',
9	0	0	0	0	1	0	0	0	0	0	1	0	0	0	10.38079-igusabder.1	['anova', 'mann whitney', 'meta analiz', 'spearman', 'odds ratio', 't testi', 'fisher's exact', 'spss']
10	0	0	1	0	0	0	0	0	0	0	1	0	0	0	10.38079-igusabder.1	['wilcoxon', 'dunn', 'spss']
11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	10.38079-igusabder.1	['ki-kare', 'bartlett', 'mann whitney', 'spss']
12	0	0	1	0	0	0	0	0	1	0	1	0	0	0	10.38079-igusabder.1	['ki-kare', 'kruskal wallis', 'mann whitney', 'shapiro wilk', 'shapiro', 'dunn', 'spss']
13	0	0	0	0	0	0	0	0	1	0	1	0	1	0	10.38079-igusabder.1	['faktör analizi', 'madde analizi', 'bartlett', 'pearson korelasyon', 't testi', 'cronbach alfa', 'spss', 'minitab']
14	0	0	0	0	0	0	0	0	0	0	1	0	0	0	10.38079-igusabder.1	['mann whitney', 'shapiro-wilk', 't testi', 'spss']
15	0	0	0	0	0	0	0	0	1	0	1	0	0	0	10.38079-igusabder.1	['pearson korelasyon', 'cronbach alfa', 'spss']
16	0	0	1	0	0	0	0	0	0	0	0	1	0	0	10.38079-igusabder.1	['mann whitney', 'pearson korelasyon', 't testi', 'dunn', 'ncss']
17	0	0	0	0	0	0	0	0	0	0	1	0	0	0	10.38079-igusabder.1	['ki-kare', 'spss']
18	0	0	0	0	0	0	0	0	0	0	1	0	0	0	10.38079-igusabder.1	['pearson korelasyon', 't testi', 'spss']
19	0	0	0	0	0	0	0	1	1	0	1	0	0	0	10.38079-igusabder.1	['anova', 'pearson korelasyon', 't testi', 'skewness-kurtosis', 'cronbach alfa', 'spss']

Resim 7.3. DergiPark platformu makale bazlı örnek çıktı

Resim 7.3'te DergiPark platformundan indirilmiş, istatistiksel yöntemlerin varlığına ve yokluğuna göre "1" ve "0" yazdırılmış yöntemlerin ayrı ayrı kolon olarak bulunduğu, bu makalelere ilişkin *PDF* isimlerinin yer aldığı Dosya Adı kolonu ve ilgili makale *PDF*'sinde bulunan istatistiksel yöntemin/yöntemlerin ve kullanılan programa ilişkin Yöntem/Yazılım kolonunun yer aldığı toplamda 407 satırdan oluşan çıktı bulunmaktadır. Çıktının belirli bir kısmından görüldüğü üzere Yöntem/Yazılım kolonunda yer alan tüm yöntem ve

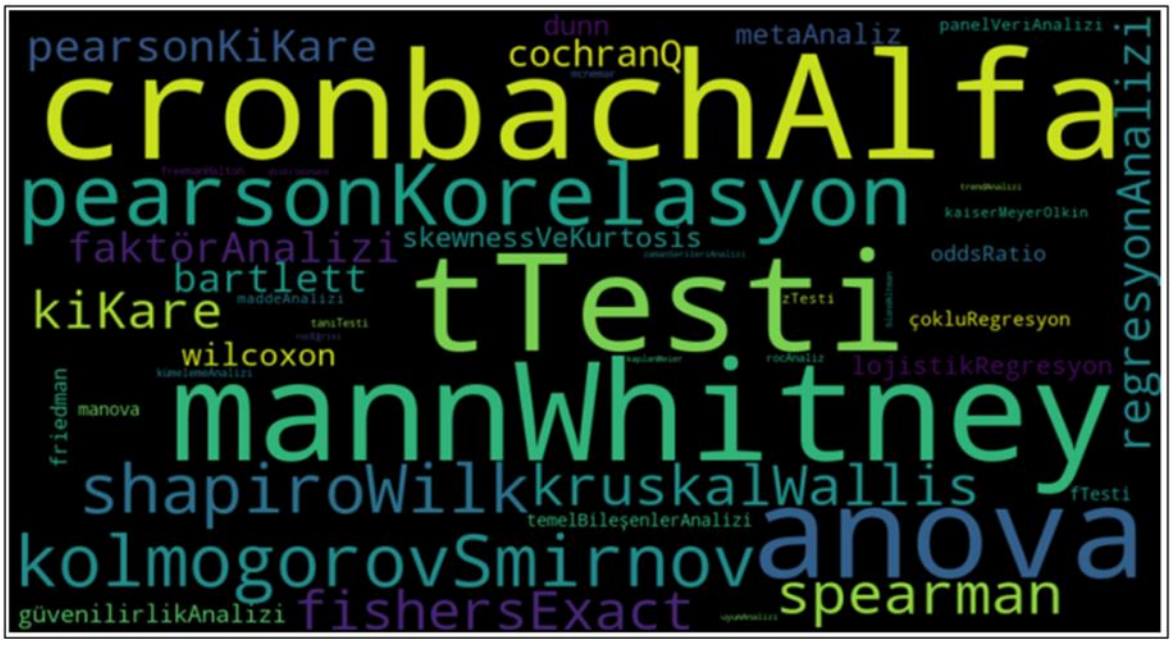
programların bulunduğu kolonlarda, ilgili makalede bulunup bulunmadığına göre “1” ve “0” olarak kodlanmıştır.



re extr	ation	dom fo	x-	jenk	ility an	pchran	dunn	iedma	er's e	xelling	dall's	ss and	bach a	studi	spss	stata	ncss	hinita	python	amov	Dosya Adı	Yöntem / Yazılım
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1020-3397	['chi-square', 'bartlett', 'mann whitney', 't test', 'spss']
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A bidirect	['odds ratio', 't test', 'cochran q']
0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	A combin	['ancova', 'chi-square', 'kaplan-meier', 'kaiser-meyer-olkin', 'mann whitney', 't test', 'cochran q', 'fisher's exact', 'spss']
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A comple	['cluster analysis', 'kaplan-meier']
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	A nomogr	['regression analysis', 'chi-square', 'mann whitney', 'roc curve', 'odds ratio', 'spss']
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	A novel in	['regression analysis', 'chi-square', 'multidimensional scaling', 'odds ratio', 't test', 'association analysis', 'fisher's exact'
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	A novel tr	['t test', 'fisher's exact']
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A phase 2	['ancova', 't test']
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	A true res	['mann whitney', 't test', 'cochran q', 'spss']
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Academic	['chi-square', 'odds ratio', 't test']
0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	Acupunct	['chi-square', 'mann whitney', 't test', 'cochran q', 'fisher's exact', 'spss', 'stata', 'ncss']
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	Allodynia	['kaplan-meier', 'mcnemar', 't test', 'f-test', 'wilcoxon', 'spss']
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Alternativ	[]
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	Amplified	['anova', 'ancova', 'manova', 'general linear model', 'spss']
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	An observ	['kaiser-meyer-olkin', 'odds ratio', 'dunn']
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	Analgesic	['chi-square', 'kolmogorov-smirnov', 'mann whitney', 'spss', 'ncss']
0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	Analysis o	['t test', 'cochran q', 'fisher's exact', 'r studio', 'spss']
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Analysis o	['regression analysis', 'chi-square', 'mann whitney', 't test', 'cochran q']

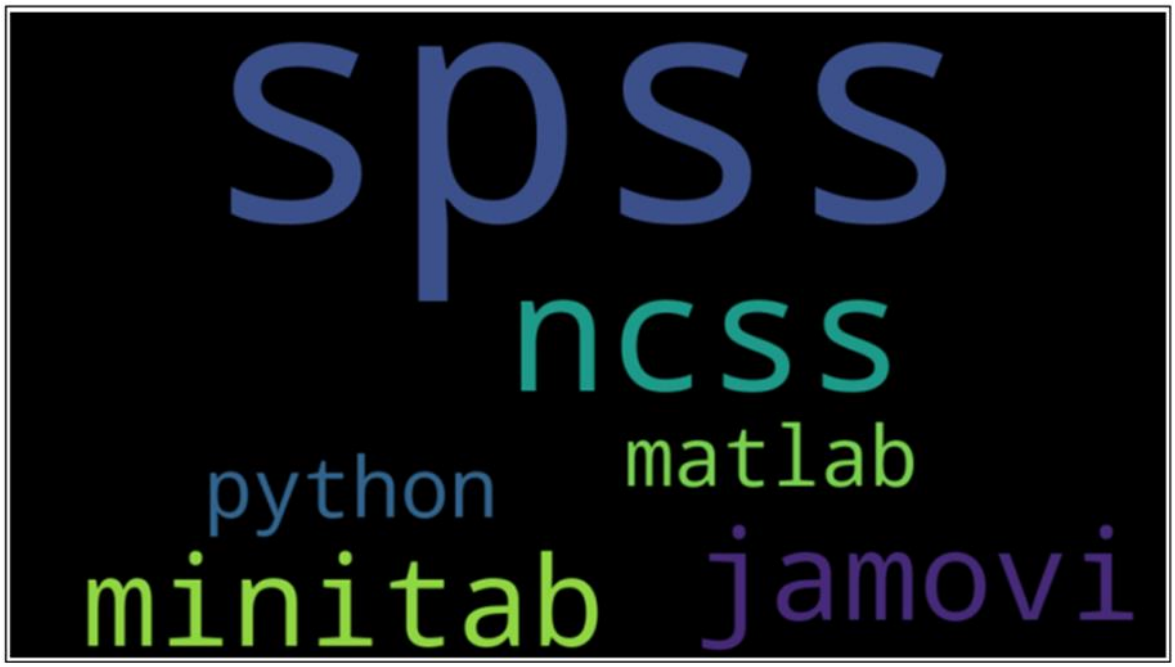
Resim 7.4. PubMed platformu makale bazlı örnek çıktı

Resim 7.4’te PubMed platformundan indirilmiş, istatistiksel yöntemlerin varlığına ve yokluğuna göre “1” ve “0” yazdırılmış yöntemlerin ayrı ayrı kolon olarak bulunduğu, bu makalelere ilişkin *PDF* isimlerinin yer aldığı Dosya Adı kolonu ve ilgili makale *PDF*’sinde bulunan istatistiksel yöntemin/yöntemlerin ve kullanılan programa ilişkin Yöntem/Yazılım kolonunun yer aldığı toplamda 305 satırdan oluşan çıktı bulunmaktadır. Çıktının belirli bir kısmından görüldüğü üzere Yöntem/Yazılım kolonunda yer alan tüm yöntem ve programların bulunduğu kolonlarda, yöntemin ve programın ilgili makaledeki varlığına göre “1” ve “0” olarak kodlanmıştır.



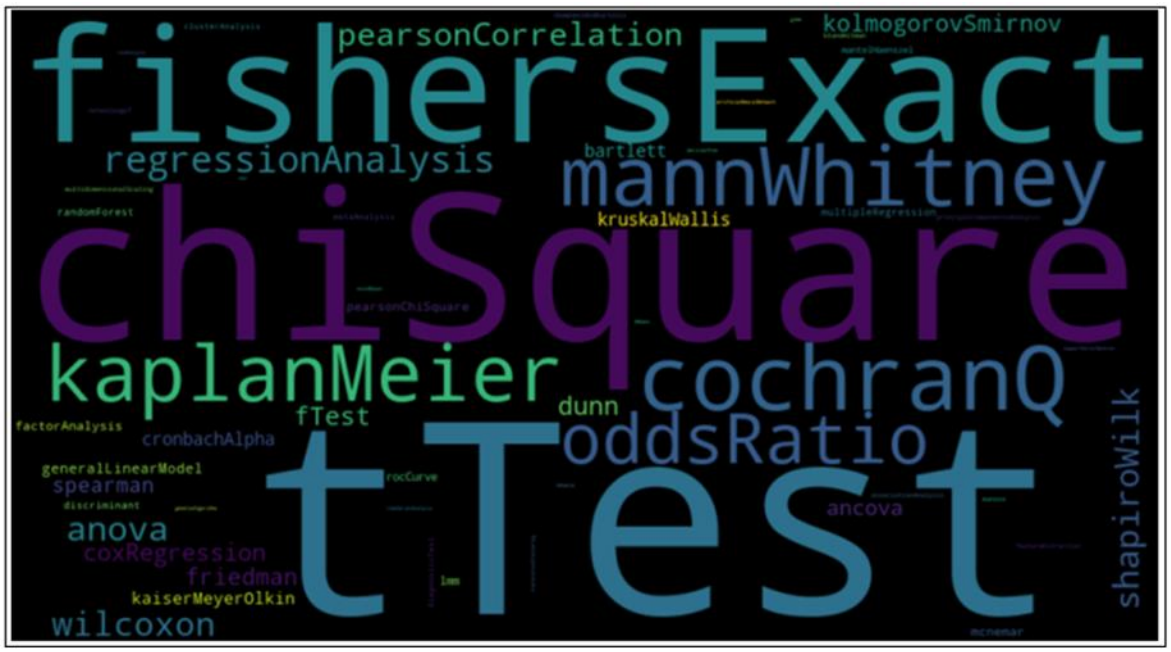
Resim 7.5. DergiPark makalelerinde bulunan istatistiksel yöntemlere ilişkin kelime bulutu

Resim 7.5'te DergiPark makalelerinde tespit edilmiş istatistiksel yöntemlere ve kullanılma sayılarına göre oluşturulmuş kelime bulutu yer almaktadır. Kullanılma sayılarına göre en yüksek beş yöntem olarak cronbach's alfa 150, mann whitney u testi 131, *t*-testi 126, anova 110 ve pearson korelasyon 107 olarak belirlenmişti. Kelime bulutunda bakıldığında, bu beş yöntemin yazı boyutu diğer yöntemlere göre daha büyük olduğu görülmektedir.



Resim 7.6. DergiPark makalelerinde bulunan programlara ilişkin kelime bulutu

Resim 7.6'da DergiPark makalelerinde tespit edilmiş programlara ve kullanılma sayılarına göre oluşturulmuş kelime bulutu yer almaktadır. İstatistiksel yöntemlerin genellikle *SPSS* programıyla uygulandığı belirlenmişti. Kelime bulutunda bakıldığında, *SPSS* programının yazı boyutu diğer programlara göre daha büyük olduğu görülmektedir.



Resim 7.7. PubMed makalelerinde bulunan istatistiksel yöntemlere ilişkin kelime bulutu

Resim 7.7'de PubMed makalelerinde tespit edilmiş istatistiksel yöntemlere ve kullanılma sayılarına göre oluşturulmuş kelime bulutu yer almaktadır. Kullanılma sayılarına göre en yüksek beş yöntem olarak *t*-test 167, chi-square test 89, fisher's exact test 77, cochran's q test 75 ve mann whitney *u* test 64 olarak belirlenmişti. Kelime bulutunda bakıldığında, bu beş yöntemin yazı boyutu diğer yöntemlere göre daha büyük olduğu görülmektedir.

Resim 7.8. PubMed makalelerinde bulunan programlara ilişkin kelime bulutu

Resim 7.8’de PubMed makalelerinde tespit edilmiş programlara ve kullanılma sayılarına göre oluşturulmuş kelime bulutu yer almaktadır. İstatistiksel yöntemlerin çoğunlukla *SPSS* programıyla ve en yüksek ikinci olarak *Stata* programında uygulandığı belirlenmişti. Kelime bulutuna bakıldığında, *SPSS* programının yazı boyutu diğer programlara göre daha büyük olduğu ve ikinci olarak *Stata* programının olduğu görülmektedir.

8. SONUÇ VE ÖNERİLER

Bu çalışmada DergiPark ve PubMed platformlarında sağlık bilimleri dergilerinde yayınlanmış, 2023 yılına ait, “istatistiksel analiz” ve “statistical analyses” içeren ve *RPA* robotunun erişebileceği ücretsiz makaleler indirilip, bu makalelerdeki hangi istatistiksel yöntemlerin ve programların bulunduğu tespit edilmiştir. Çalışma kapsamında literatür araştırması yapılarak, diğer çalışmalara değinilmiş, veri madenciliği ve metin madenciliği hakkında bilgi verilmiş ve süreçleri anlatılmıştır. Daha sonrasında bu çalışmada yapılan uygulama adımlarına yer verilmiştir.

DergiPark platformundan ilgili filtremeler sonucu 407 adet makale ve PubMed platformundan ilgili filtrelemeler sonucu 305 adet makalenin *URL* bilgisine ulaşılmıştır. Bu *URL* bilgisi *RPA* robotuna verilmesi ve indirme aşamaları komutlarının verilmesiyle birlikte her iki platformdan toplamda 712 adet makale indirilmiştir. Bu indirilen makalelere metin madenciliği uygulanarak makale *PDF*’lerinde yer alan istatistiksel yöntemler ve programlar aratılmıştır. Kullanılan yöntemlerin sıklığına göre kelime bulutu uygulaması yapılmıştır.

DergiPark platformundan filtrelenen makalelere uygulanan metin madenciliği uygulaması sonucunda en çok kullanılan yöntemin cronbach’s alfa olduğu ve mann whitney u testinin takip ettiği ve istatistiksel yöntemlerin çoğunlukla *SPSS* programı ile uygulandığı belirlenmiştir.

PubMed platformundan filtrelenen makalelere uygulanan metin madenciliği uygulaması sonucunda en çok kullanılan yöntemin *t*-testi olduğu ve chi-square testinin takip ettiği ve istatistiksel yöntemlerin en çok *SPSS* programı ile uygulandığı ve sonrasında *Stata* programının takip ettiği sonucuna varılmıştır.

DergiPark makalelerinde PubMed makalelerine göre; cronbach’s alfa, mann whitney u testi, anova, kolmogorov-smirnov, shapiro-wilk, kruskal-wallis, korelasyon ve faktör analizi yöntemleri daha çok kullanılmıştır. PubMed makalelerinde ise DergiPark makalelerine göre; ki-kare, t-testi, fisher’s exact, cochran’s q, wilcoxon, mcnemar ve roc analizi yöntemleri daha çok kullanılmıştır.

DergiPark makalelerinde kullanılmış fakat PubMed makalelerinde kullanılmayan yöntemler; lojistik regresyon, panel veri analizi, freeman halton ve madde analizi olarak belirlenmiştir. PubMed makalelerinde kullanılmış fakat DergiPark makalelerinde kullanılmayan yöntemler ise; ancova, genel doğrusal modeller, rasgele orman, çok boyutlu ölçekleme, cox regresyon ve diskriminant analizi olarak belirlenmiştir.

DergiPark ve PubMed platformlarından elde edilen makalelerde kullanılma sayıları birbirine yakın olan yöntemler arasında; regresyon analizi, temel bileşenler analizi, kümeleme analizi, manova ve zaman serileri analizi yer almaktadır.

8.1. Çalışmanın Zorlukları

PubMed platformunda ücretli ve ücretsiz olmak üzere makale ayrımları yer almaktadır. Fakat ücretsiz makaleler seçilmesine rağmen nadirde olsa ücretli makalelere denk gelinmiştir. Bu sebeple *RPA* robotuna *PDF* butonunun varlığının kontrol edilmesi konusunda zorluk yaşanmıştır. PubMed platformunda makale *PDF*'lerine tek bir link üzerinden erişilememektedir. Birkaç web sitesine gidildikten sonra ve çeşitli yollar denenerek (*URL* kısmına *PDF*'in açılabileceği sabit bilgiler girerek vb.) ancak makalelere ilişkin *PDF*'ler bulunmuş ve indirtilmiştir. Bu sebeple *RPA* robotuna makale *PDF*'lerine erişmek amacıyla verilen komutlarda zorluk yaşanmıştır.

PDF üzerinde yöntemlerin aratıldığı aşamada, makale içerisinde yöntemin birden fazla farklı yazım türü olduğuna (Cronbach Alfa, Cronbach's Alfa, Cronbach Alpha, Cronbach's Alpha vb.) rastlanmıştır. Bu sebeple herhangi bir yöntemin yazılabileceği tüm farklı senaryolar belirlendikten sonra *PDF* dosyalarında aratılma yapılmıştır. Bu sebeple yöntemlerin ve programların *PDF* içerisinde aratıldığı metotlar oldukça uzun süre çalışmıştır.

PDF içerisindeki belirli bir bölüm ele alınamadığı için tüm *PDF* içerisinde yöntemler aratılmıştır. Bu sebeple bazı yöntem isimleri bölünerek aratıldığında bir yazarın adı, soyadı ya da kitap adı olarak bulunabilmektedir. Bu durumlar sonuç verisinden silinmiştir.

8.2. Yapılacak Çalışmalara Öneriler

RPA robotlarına komutların daha kolay verilmesi adına veri setlerinin elde edileceği veri kaynağının, olabildiğince tek formatta olması önem arz etmektedir. Bu sebeple veri setini robot yardımıyla elde edilecek çalışmalar için, metin dosyasının indirileceği web sitesinin tüm makaleleri farklı formattaki linkler ile indirilmemesine (PubMed vb.) dikkat edilmelidir. Uygulamaya arayüz geliştirilip masaüstü uygulaması haline getirilebilir.

KAYNAKLAR

Aksu, M., Karaman, E. (2020). FastText ve Kelime Çantası Kelime Temsil Yöntemlerinin Turistik Mekanlar İçin Yapılan Türkçe İncelemeler Kullanılarak Karşılaştırılması, *Avrupa Bilim ve Teknoloji Dergisi*, 20, 311-320.

Allahyari, M., Safaei, S., Pouriyeh, S., Trippe, E., Kochut, K., Assefi, M., Gutierrez, J. (2017). A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques, *In Ldv Forum*, 19-62.

Altunkaynak, B. (2019). *Veri Madenciliği Yöntemleri ve R Uygulamaları*. (2.Baskı). Ankara: Seçkin, 17-35.

Bayer, H. (2011). *Veri Madenciliğinde Bir Metin Madenciliği Uygulaması*. Yüksek Lisans Tezi, Beykent Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 1-2.

Bilgin, A. (2018). *Metin Madenciliği Yöntemleri ile Yazar Tanıma: Divan Edebiyatı Örneği*. Yüksek Lisans Tezi, Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü, Trabzon, 1-2.

Böschen, I. (2021). Evaluation of JATSdecoder as an automated text extraction tool for statistical results in scientific reports, *Scientific Reports*, 11(19525), 1-3.

Çağıltay, N., Tokdemir, G. (2010). *Veritabanı Sistemleri Dersi Teoriden Pratiğe*. (1. Baskı). Ankara: Ada, 14-16.

Çam, E. (2019). *Metin Madenciliğinde Kategorik Değişkenler İçin Benzetim Katsayılarının Kullanılması Üzerine Bir Çalışma*. Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü, İzmir, 1-2.

Döven, S. (2013). *Metin Madenciliği ile Dokümanlar Arasındaki Benzerliklerin Bulunması.* Yüksek Lisans Tezi, Bahçeşehir Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 2-3.

Eroğlu, Ö. (2005). *Hece Tabanlı İstatistiksel Yöntemler ile Yazım Hatası Bulma ve Düzeltme*. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 15-16.

Esen, F. (2009). *Veritabanlarında Bilgi Keşfi: Veri Madenciliği ve Bir Sağlık Uygulaması*. Yüksek Lisans Tezi, İstanbul Üniversitesi Sosyal Bilimler Enstitüsü, İstanbul, 44-45.

Farboudi, S. (2009). *Tıp Bilişiminde İstatistiksel Veri Madenciliği*. Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 29.

Gorunescu, F. (2011). *Data Mining Concepts, Models and Techniques*. Springer, 185-317.

Gürbüz, T. (2022). *Metin Madenciliği Yöntemi ile Araştırma Makalesi Sınıflandırması*. Yüksek Lisans Tezi, Gazi Üniversitesi Bilişim Enstitüsü, Ankara, 4-7.

Ibrahim, B. (2021). Statistical methods used in Arabic journals of library and information science, *Scientometrics,*126, 4383-4385.

İnternet: Deloitte (2018). Deloitte Global RPA Survey. URL:https://www2.deloitte.com/bg/en/pages/technology/articles/deloitte-global-rpa-survey-2018.html / Son Erişim Tarihi: 20.10.2023.

İnternet: Kirkwood, G. (March, 2019). The Impact of RPA on Employee Engagement, a Forrester Consulting Thought Leadership Paper. URL:https://www.uipath.com/blog/rpa/impact-of-rpa-on-employee-engagement-forrester / Son Erişim Tarihi: 20.10.2023.

İnternet: PyPDF2 (2008). History of PyPDF2. URL:https://pyPDF2.readthedocs.io/en/3.0.0/meta/history.html# / Son Erişim Tarihi: 07.11.2023.

İnternet: Toprak, A. H. (Ocak, 2019). Python Re (Regular Expression) Modülü. URL:https://alhydrtprk.medium.com/python-re-regular-expression-mod%C3%BCl%C3%BC-6dec531a8434 / Son Erişim Tarihi: 07.11.2023.

İnternet: UiPath. (2022). What is Robotic Process Automation. URL:https://www.uipath.com/rpa/robotic-process-automation / Son Erişim Tarihi: 20.10.2023.

İnternet: Wu, T. (June, 2020). New Research Shows Worker Concerns About Skills Gap. URL:https://www.uipath.com/blog/digital-transformation/new-research-shows-workers-concerned-skills-gaps / Son Erişim Tarihi: 20.10.2023.

John, G., Kohavi, R., Pfleger, K. (1994). *Irrelevant Feature and the Subset Selection Problem*, Proceedings of 11th International Conference on Machine Learning, 121-129.

Jurafsky, D., Martin, J. (2023). *Speech and Language Processing*. (Third Edition). Chapter 3 1-3.

Kamuk, U. (2020). Spor Bilimleri Alanında Yayınlanan Makalelerde Kullanılan İstatistiksel Yöntemlerin İncelenmesi, *The Journal of Physical Education and Sport Sciences*, 18(3), 73-85.

Kaşıkçı, T., Gökçen, H. (2014). Metin Madenciliği ile e-Ticaret Sitelerinin Belirlenmesi, *Bilişim Teknolojileri Dergisi*, 7(1), 25-26.

Kaydım, C. (2021). *Implementation of Data Mining Algorithms on Rock Mechanics Test Data For Knowledge Discovery*. Yüksek Lisans Tezi, Orta Doğu Teknik Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 34-35.

Liu, C., Sheng, Y., Wei, Z., Yang, Y. (2018). *Research of text classification based on improved TF-IDF algorithm*, 2018 The Institute of Electrical and Electronics Engineers (IEEE) International Conference of Intelligent Robotic and Control Engineering (IRCE), China, 218-222.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C. (2002). Text Classification Using String Kernels, *Journal of Machine Learning Research 2*, 2(02), 419-444.

Madadizadeh, F., Bahariniya, S. (2022). Frequency of the Statistical Methods and Relation with Acceptance Period in Archives of Iranian Medicine Articles: A Review from 2015-2019, *Archives of Iranian Medicine,* 25(4), 267-273.

McKinney, W. (2011). Pandas: A Foundational Python Library for Data Analysis and Statistics. *Python for High Performance and Scientific Computing,* 14(9), 1-9.

Oğuz, B. (2009). *Metin madenciliği teknikleri kullanılarak kulak burun boğaz hasta bilgi formlarının analizi*. Yüksek Lisans Tezi, Akdeniz Üniversitesi Sağlık Bilimleri Enstitüsü, Antalya, 9.

Özkan, Y. (2008). *Veri Madenciliği Yöntemleri*. (1. Baskı). İstanbul: Papatya, 22-26.

Pilavcılar, İ. (2007). *Metin Madenciliği ile Metin Sınıflandırma*. Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 2-3.

Rainardi, V. (2008). *Building Data Warehouse*. (First Edition). Apress, 29-47.

Sgambati, S., Gargiulo, C. (2022). The evolution of urban competitiveness studies over the past 30 years. A bibliometric analysis. *Cities,*103811, 1-2.

Steinbach, M., Tan, P., Kumar, V. (2005). *Introduction to Data Mining*. (First Edition). Pearson, 152-154.

Şeker, Ş. (2012). *Internet Knowledge Engineering,* Turkish Query Engine on Library Ontology, 26-33.

Şeker, Ş., Al-Naami, K. (2013). *Sentimental Analysis on Turkish Blogs via Ensemble Classifier.* Proceedings of the 2013 International Conference on Data Mining, 10-16.

Şeker, Ş., Diri, B. (2010). *TimeML and Turkish Temporal Logic*, Proceedings of International Conference on Artificial Intelligence, 881-887.

Şeker, Ş., Mert, C., Al-Naami, K., Özalp, N., Ayan, U. (2013). *Correrlation between the Economy News and Stock Market in Turkey*, International Journal of Business Intelligence and Review, 1-21.

Şentürk, Z. (2011). *Veri Madenciliği ile Kanser Tanısı*. Yüksek Lisans Tezi, Düzce Üniversitesi Fen Bilimleri Enstitüsü, Düzce, 18-21.

Tekin, M. (2018). *Yazılım Geliştirme Taleplerinin Metin Madenciliği ile Sınıflandırılması ve Önceliklendirilmesi.* Yüksek Lisans Tezi, Maltepe Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 13-14.

Toplu, S. (2019). *Metin Madenciliği ve Sağlık Alanında Bir Uygulama*. Yüksek Lisans Tezi, Düzce Üniversitesi Sağlık Bilimleri Enstitüsü, Düzce, 1-2.

Uçkan, T., Hark, C., Seyyarer, E., Karcı, A. (2019). Ağırlıklandırılmış Çizgelerde Tf-Idf ve Eigen Ayrışımı Kullanılarak Metin Sınıflandırma. *BEÜ Fen Bilimleri Dergisi,* 8(4), 1349-1362.

Ukşul, E. (2016). *Türkiye'de Eğitimde Ölçme ve Değerlendirme Alanında Yapılmış Bilimsel Yayınların Sosyal Ağ Analizi ile Değerlendirilmesi: Bir Bibliyometrik Çalışma*. Yüksek Lisans Tezi, Akdeniz Üniversitesi Eğitim Bilimleri Enstitüsü, Ankara, 7-8.

Ülger, G., Baldemir, R. (2022). Analysis of global publications on tracheostomy between 1980 and 2021, including the impact of COVID-19: a bibliometric overview. *Medicine Palliative Care,* 3(2), 103-110.

Visa, A. (2001). *Technology of Text Mining.* Tampare University of Technology, Germany, 1-11.

Yılmaz, G. (2021). *E-Ticaret Sistemlerinde Yapılan Ürün ve Yorumlarının Metin Madenciliği Uygulaması ile İncelenmesi*. Yüksek Lisans Tezi, İstanbul Aydın Üniversitesi Eğitim Enstitüsü, İstanbul, 1-2.

Yudhistyra, W., Risal, E., Raungratanaamporn, I., Ratanavaraha, V. (2020). Exploring Big Data Research: A Review of Published Articles from 2010 to 2018 Related to Logistics and Supply Chains, *Operations and Supply Chain Management,* 13(3), 134-149.

EKLER

EK-1. Kodlar

```
pip install PyPDF2
pip isal WordCloud
import PyPDF2
import re
import pandas as pd
import os
from wordcloud import WordCloud


#pdfMethodFileObj=open("C://Users//ozent//OneDrive//Masaüstü//yontemListe.PDF",'rb')
pdfMethodFileObj=open("C://Users//ozent//OneDrive//Masaüstü//yontemList.PDF", 'rb')
pdfText = PyPDF2. PdfReader(pdfMethodFileObj)
for page in pdfText.pages:
   kelimeListesi = page.extract_text()
kelimeListesi = kelimeListesi.split(',')
kelimeListesi


path = "C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi"
#path = "C://Users//ozent//OneDrive//Masaüstü//pubMedPDFDosyasi"
list_path = []
for(root, dirs, file) in os.walk(path):
   for f in file:
      if '.pdf' in f:
       list_path.append("C://Users//ozent//OneDrive//Masaüstü//dergiParkPDFDosyasi//"+f)


df = pd.DataFrame(columns = kelimeListesi)
df


df.insert(loc=(len(kelimeListesi)),
         column='Dosya Adı',
         value="")
df.insert(loc=(len(kelimeListesi)+1),
         column='Yöntem / Yazılım',
         value="")
```

EK-1. (devam) Kodlar

```
def wordExist(liste, x, PDF):
    path = None
    array = []
    arrayBulunanYontem = []
    basename = os.path.basename(x)[:-4]
    for word in liste:
        index = 0
        for page in pdf.pages:
            text = page.extract_text()
            text=text.lower()
            res_search = re.search(word, text)
            if word in text:
                index = 1
                break
            else:
                index = 0

if(index==1):
            arrayBulunanYontem.append(word)
        array.append(index)
    array.append(basename)
    array.append(arrayBulunanYontem)
    df.loc[len(df)] = array

def pdfAc(liste_adı):
    tur = 0
    for x in list_path:
        tur +=1
        pdfFileObj = open(x, 'rb')
        pdf = PyPDF2.PdfReader(pdfFileObj)
        wordExist(kelimeListesi,x,pdf)
        print("TAMAMLANAN PDF SAYISI")
        print(tur)
        print("********")

pdfAc(list)

df_last2Columns = df[['Dosya Adı','Yöntem / Yazılım']]
df_last2Columns
df_last2Columns.to_Excel('C://Users//ozent//OneDrive//Masaüstü//result.xlsx')
```

EK-1. (devam) Kodlar

```
df.loc['total'] = df.sum(numeric_only=True, axis=0)
df
df.to_Excel('C://Users//ozent//OneDrive//Masaüstü//result1PBpart5.xlsx')


df_last_row = df.iloc[-1:]
df_last_row.drop(columns=df_last_row.columns[-2:], axis=1,  inplace=True)
df_last_row


df_last_row.to_Excel('C://Users//ozent//OneDrive//Masaüstü//result2PBpart5.xlsx')
dfCloud = pd.read_Excel('C://Users//ozent//OneDrive//Masaüstü//PBresult2prog.xlsx')


dfCloud = dfCloud.drop('Unnamed: 0', axis=1)
dfCloud


string_values = ""
for x in dfCloud.columns:
    val = dfCloud[x]
    print(val)
    if(int(val) == 0):
        continue
    for i in range(int(val)):
        string_values = string_values + "," + x


word_cloud=WordCloud(width=900,height=500,collocations=False)
        .generate(string_values)
word_cloud.to_image()
```

Gazili olmak ayrıcalıktır