

T.C.
MUNZUR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**EĞİTİM SÜRECİNDE BİLGİSAYAR DESTEKLİ KARAR DESTEK
SİSTEMLERİ İLE ALTERNATİFLERİN ÖNCELİKLENDİRİLMESİ**

Yiğit ALIŞAN

YÜKSEK LİSANS TEZİ
MÜHENDİSLİK YÖNETİMİ ANABİLİM DALI

DANIŞMAN
Dr. Öğr. Üyesi Faruk SERİN

TUNCELİ – 2020

T.C.
MUNZUR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**EĞİTİM SÜRECİNDE BİLGİSAYAR DESTEKLİ KARAR DESTEK
SİSTEMLERİ İLE ALTERNATİFLERİN ÖNCELİKLENDİRİLMESİ**

Yiğit ALIŞAN
(18110901106)

YÜKSEK LİSANS TEZİ
MÜHENDİSLİK YÖNETİMİ ANABİLİM DALI

DANIŞMAN
Dr. Öğr. Üyesi Faruk SERİN

TUNCELİ – 2020

**T.C.
MUNZUR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**EĞİTİM SÜRECİNDE BİLGİSAYAR DESTEKLİ KARAR DESTEK
SİSTEMLERİ İLE ALTERNATİFLERİN ÖNCELİKLENDİRİLMESİ**

**Yiğit ALIŞAN
YÜKSEK LİSANS TEZİ
MÜHENDİSLİK YÖNETİMİ ANABİLİM DALI**

Bu tez 24/06/2020 tarihinde aşağıdaki jüri üyeleri tarafından **oybirliği** ile kabul edilmiştir.

İmza:.....

Dr. Öğr. Üyesi Faruk
SERİN
(Munzur Üniversitesi)

DANIŞMAN

İmza:.....

Dr. Öğr. Üyesi Kenan
İNCE
(İnönü Üniversitesi)

ÜYE

İmza:.....

Dr. Öğr. Üyesi Özal
YILDIRIM
(Munzur Üniversitesi)

ÜYE

Bu tez, Enstitümüz Mühendislik Yönetimi Anabilim Dalı'nda hazırlanmıştır.

Doç. Dr. Olcay KAPLAN İNCE
Enstitü Müdürü
İmza ve Mühür

NOT: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı “Fikir ve Sanat Eserleri Kanunu”ndaki hükümlere tabidir.

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi ve hiçbir şekilde “intihal içermediğini” beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

Yiğit ALIŞAN

Dr. Öğr. Üyesi Faruk SERİN

TEŐEKKÖR

Danışman hocam Sayın Dr. Öğr. Üyesi Faruk SERİN başta olmak üzere, tezimin her aşamasında desteklerini benden esirgemeyen aileme ve arkadaşlarıma sonsuz teşekkürlerimi sunarım.

Yiğit ALIŐAN
TUNCELİ- 2020



İÇİNDEKİLER

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ	I
TEŞEKKÜR.....	II
İÇİNDEKİLER.....	III
ŞEKİLLER LİSTESİ	V
TABLolar LİSTESİ	VI
KISALTMALAR LİSTESİ.....	VII
SEMBOLLER LİSTESİ	VIII
ÖZET	IX
ABSTRACT	X
1. GİRİŞ.....	1
1.1. Literatür Taraması.....	3
1.2. Tezin Katkısı ve Organizasyonu	5
2. MATERYAL VE METOT.....	7
2.1. Verilerin İnternet Ortamında Barınması	9
2.1.1. HTML işaretleme dili ve sözdizimi özellikleri.....	9
2.1.2. HTML etiket, öge ve özellikleri	10
2.1.3. Cascading style sheets	10
2.1.4. Xpath (XML adresleme dili)	10
2.2. Python Dili ve Özellikleri.....	11
2.2.1. Karakter ifadeler	15
2.2.2. Değişkenler.....	15
2.2.3. Veri tipleri	16
2.2.4. Şart kontrol komutları.....	18
2.2.5. Döngü komutları.....	19
2.2.6. Fonksiyonlar	20
2.3. Kütüphane Modülleri.....	20
2.3.1. BeautifulSoup kütüphanesi.....	21
2.3.2. LXML kütüphanesi.....	23
2.3.3. Requests kütüphanesi	24
2.3.4. Selenium kütüphanesi.....	24
2.3.5. Numpy kütüphanesi.....	27
2.3.6. Openpyxl kütüphanesi	29
2.3.7. MySQL.connector kütüphanesi	30
2.3.8. FuzzyWuzzy kütüphanesi	31
2.3.9. Doğal dil işleme.....	33

2.3.10. Jpype kütüphanesi	34
2.4. Çok Kriterli Karar Verme Yöntemleri	35
2.4.1. TOPSIS yöntemi	35
2.4.2. VIKOR yöntemi	37
2.4.3. ELECTRE yöntemi	38
2.5. Verilerin Toplanması ve Düzenlenmesi	40
2.5.1. Çevrimiçi iş ilanlarının toplanması.....	41
2.5.2. Ders öğrenme çıktılarının toplanması.....	42
2.5.3. Yararsız kelime kökleri sözlüğü.....	42
2.5.4. Bilgisayar terimleri sözlüğü	42
3. BULGULAR VE TARTIŞMA	44
4. SONUÇ VE ÖNERİLER	63
4.1. Öneriler.....	64
4.2. Kısıtlamalar ve Zorluklar.....	64
4.3. Potansiyel Gelecek Çalışmalar	64
5. KAYNAKLAR	66
ÖZGEÇMİŞ	70

ŞEKİLLER LİSTESİ

Şekil 2.1. Önerilen sistemin akış şeması	8
Şekil 2.2. Pycharm’da projeye yeni paket ekleme	14
Şekil 2.3. Pycharm’da python paket listesi	14
Şekil 3.1. Üniversitelerin 2015-2018 yılları arası taban puanları grafiği.....	45
Şekil 3.2. Uluslararası ilanların ülke dağılımı	46
Şekil 3.3. Ulusal ve uluslararası ilanlarda geçen teknik terimlere ait kelime bulutu	54
Şekil 3.4. Tüm derslerin “Yazılım Uzmanı” pozisyonuna ait nitelikleri karşılama yüzdesi (En yüksek ilk 10 ders)	55
Şekil 3.5. Temel derslerin yazılım uzmanı pozisyonuna ait nitelikleri karşılama yüzdesi..	56
Şekil 3.6. “Yazılım Uzmanı” ve “Software Engineer” ilanlarında geçen teknik terimlere ait kelime bulutu	57
Şekil 3.7. Tüm derslerin “Yazılım Uzmanı” pozisyonuna ait nitelikleri karşılama yüzdesi (En yüksek ilk 10 ders)	59



TABLÖLER LİSTESİ

Tablo 2.1. Yararsız kelime kökleri sözlüğü listesi- örnek kesit.....	42
Tablo 2.2. Bilgisayar Terimleri Sözlüğü Kesiti.....	43
Tablo 3.1. Üniversitelerin kurumsal sitelerinden toplanılan ders sayıları.....	44
Tablo 3.2. Üniversitelerin 2015-2018 yılları arası taban puanları	44
Tablo 3.3. Ulusal ilanlara ait pozisyon listesi	45
Tablo 3.4. Ülkelerin yayınladıkları ilan sayısı	47
Tablo 3.5. Uluslararası ilanlara ait en fazla ilan sayısına sahip ilk 50 pozisyon listesi	47
Tablo 3.6. Ulusal ilanlara ait üniversitelerin iş ilanlarına uygunluk başarı sırası karşılaştırması.....	48
Tablo 3.7. Ulusal ilanlar için algoritma çıktılarına ait ayrıntılı sonuçlar	49
Tablo 3.8. Uluslararası ilanlara ait üniversitelerin iş ilanlarına uygunluk başarı sırası karşılaştırması.....	49
Tablo 3.9. Uluslararası ilanlar için algoritma çıktılarına ait ayrıntılı sonuçlar	50
Tablo 3.10. Ulusal ilanlara ait ilanlarda kullanılan frekans değeri 100 den büyük olan kelimeler	50
Tablo 3.11. Uluslararası ilanlara ait ilanlarda en çok tekrar edilen ilk 50 kelime ve frekans değerleri	51
Tablo 3.12. Ulusal ilanlara ait en çok kullanılan ilk 100 terim ve frekans değerleri	52
Tablo 3.13. Uluslararası ilanlara ait en çok kullanılan ilk 100 terim ve frekans değerleri .	53
Tablo 3.14. Ulusal ilanlara ait “Bilişim/Bilgisayar” ve “Yazılım Uzmanı” pozisyonları karşılaştırması.....	55
Tablo 3.15. Seçmeli derslerin “Yazılım Uzmanı” pozisyonuna göre sıralaması.....	56
Tablo 3.16. Üniversitelerin “Software Engineer” pozisyonuna göre başarı sırası karşılaştırması.....	57
Tablo 3.17. Teknik seçmeli derslerin “Software Engineer” pozisyonuna ait değerler.....	58
Tablo 3.18. Üniversitelerin teknik terim tabanlı genel sıralaması	60
Tablo 3.19. Üniversitelerin tablolar arası karşılaştırma sonuçları	60
Tablo 3.20. Üniversitelerin “Yazılım Uzmanı” pozisyonuna göre sıralaması	61
Tablo 3.21. Üniversitelerin Yazılım Uzmanı pozisyonuna ait tablolar arası karşılaştırma sonuçları.....	61

KISALTMALAR LİSTESİ

AÜ	: Akdeniz Üniversitesi
BS4	: BeautifulSoup 4
CSS	: Cascading Style Sheet
CÜ	: Cumhuriyet Üniversitesi
ÇAKV	: Çok Amaçlı Karar Verme
ÇKKV	: Çok Kriterli Karar Verme
DOM	: Data Object Model
EÜ	: Erbakan Üniversitesi
FÜ	: Fırat Üniversitesi
GOÜ	: Tokat Gazi Osman Paşa Üniversitesi
GPL	: General Public Licence
HÜ	: Harran Üniversitesi
HTML	: Hyper Text Markup Language
IDE	: Integrated Development Environment
İÜ	: İnönü Üniversitesi
İTÜ	: İstanbul Teknik Üniversitesi
İSTÜ	: İskenderun Teknik Üniversitesi
MSKÜ	: Muğla Sıtkı Koçman Üniversitesi
NLP	: Natural Language Processing
ODTÜ	: Orta Doğu Teknik Üniversitesi
SÜ	: Sakarya Üniversitesi
SQL	: Structure Query Language
TYYÇ	: Türkiye Yükseköğretim Yeterlilikler Çerçevesi
URAP	: University Ranking by Academic Performance
VTYS	: Veri Tabanı Yönetim Sistemleri
XML	: Extensible Markup Language

SEMBOLLER LİSTESİ

f	: Kriterler
j	: Alternatif sıra sayısı
i	: Kriter sıra sayısı
r_{ij}	: Normalize edilmiş karar matrisi
v_{ij}	: Ağırlıklandırılmış normalize karar matrisi
w_i	: i nin ağırlık değeri
A^*	: Pozitif ideal çözüm değeri
A^-	: Negatif ideal çözüm değeri
D_j^*	: j nin pozitif ideal çözüme olan uzaklık değeri
D_j^-	: j nin negatif ideal çözüme olan uzaklık değeri
C_j^*	: j nin ideal çözüme olan benzerlik değeri
I'	: Kâr ölçütleri kümesi
I''	: Maliyet ölçütleri kümesi
f_i^*	: En iyi kriter değeri
f_i^-	: En kötü kriter değeri
S_j	: j nin ortalama grup değeri
S^*	: En küçük S_j değeri
S^-	: En büyük S_j değeri
R_j	: j nin en kötü grup değeri
R^*	: En küçük R_j değeri
R^-	: En büyük R_j değeri
Q_j	: j nin maksimum grup yararı değeri
a'	: En iyi alternatif değer
a''	: En iyi ikinci alternatif değer
DQ	: Alternatifler arası fark
$C1$	: Kabul edilebilir avantaj koşulu
$C2$	: Kabul edilebilir istikrar koşulu

ÖZET

İnsanın geleceğini şekillendirmesinde büyük payı olan, en önemli etkenlerden biri eğitimidir. Eğitim sürecinin en temel unsurları öğrenen, öğretene ve ortamdır. Teknoloji ve bilişim alanındaki gelişmeler bu üç unsuru da oldukça etkilemiş durumdadır. Bu gelişmeler bazı mesleklere gereksinimi kaldırırken, yeni meslek olanakları da doğurmaktadır. Eğitim ortamları olan okullar, üniversiteler ve kurslar kendi müfredatlarını ve ders içeriklerini; Öğretene ve öğrenen de öğretme ve öğrenme yöntem ve tercihlerini bu gelişmeler doğrultusunda ve bu gelişmeleri kullanarak güncellemeleri gerekmektedir. Aksi durumda, bu unsurlar kalitesini ve rekabet gücünü kaybetmeye mahkûm kalırlar. Bu durumda, gerekli güncellemelerin yapılabilmesi için yeni araçlara ihtiyaç duyulmaktadır. Bu çalışmada, bu ihtiyaçları giderebilecek bir öneri sistemi sunulmuştur. Önerilen sistem üç aşamadan oluşmaktadır. Birinci aşaması veri toplama aşamasıdır. Bu aşamada, web kazıma yöntemleri kullanılarak internet ortamından veriler toplanmaktadır. İkinci aşamada ise toplanan bu veriler, doğal dil işleme yöntemleri ile anlamlı ve işlenebilir bilgiler haline getirilmiştir. Üçüncü aşamada, bilgisayarlı karar destek sistemleri kullanılarak tercihlerin önceliklendirilmesi sağlanmıştır. Önerilen sistem ile: eğitim ortamları hangi meslek grupları için yeni bölümler açması gerektiğine, ihtiyaç duyulmayan hangi bölümlerin kapatılması gerektiğine ve hangi meslek için hangi dersleri açması gerektiğine; öğretmenler derslerin içerik ve öğretme yönteminin nasıl şekillendireceğine; öğrenenler de hangi ortamı, bölümü, mesleği ve dersleri seçeceğine daha kolay karar verebileceklerdir. Önerilen sistemin test edilmesi için bilgisayar mühendisliğine yönelik bir uygulama gerçekleştirilmiştir. Bunun için, çeşitli iş ilan siteleri üzerinden sektörün çalışanlardan beklentilerinin ve üniversitelerin ilgili bölümlerinin ders içeriklerinin neler olduğuna dair veriler toplandı. Arz talep eşleştirmesi ile üniversiteler ve dersler ELECTRE, TOPSIS ve VIKOR yöntemleri ile sıralandı ve gerekli güncellemeler gösterildi. Yapılan sıralamaların, farklı kriterlere göre sıralama sunan ÖSYM ve URAP verilerine benzer sonuçlar sunması önerilen yöntemin uygulanabilirliğini ve güvenilirliğini göstermektedir.

Anahtar kelimeler: Web kazıma, doğal dil işleme, karar destek sistemi, VIKOR, TOPSIS, ELECTRE, eğitim, çok kriterli karar verme

ABSTRACT

A Computer Assisted Decision Support System for Education Planning

One of the most important factors in shaping the future of the human being is education. The most basic elements of the education process are learners, teachers and educational environment. The developments in technology and informatics have affected reasonably these three elements. These developments are eliminating the demand for certain occupations and create new occupational opportunities. Educational environments such as schools, colleges, universities set their own curriculum and course contents; teachers and learners update their teaching and learning methods and preferences in line with these developments and using these developments. Otherwise, these elements are condemned to lose their quality and competitiveness. Thus, new tools are needed to make the necessary updates. In this study, information management system that can meet these needs has been proposed. The proposed system consists of three stages. The first stage is the data collection stage. At this stage, data is collected from the internet using web scraping methods. In the second stage, the collected data are turned into meaningful and processable information by natural language processing methods. In the third stage, preferences are identified by using computerized decision support systems. Thanks to the proposed system: educational institutions is informed about which new departments should be opened for occupational groups, which departments should be closed due to being unnecessary, which courses should be opened or closed for which occupation; teachers are assisted to decide how to shape the content and method of teaching; learners will be able to decide more easily choose which environment, department, profession and courses. In order to test the proposed system, an application was realized for the computer engineering sector. For this purpose, data were gathered from the various job posting sites, about the expectations of the sector from the employees. In addition, the course contents of the relevant departments of the universities were collected. In order to match supply and demand, universities and courses were ranked using ELECTRE, TOPSIS and VIKOR methods and required updates were shown. The fact that the rankings are similar to the results of OSYM and URAP, which provide ranking according to different criteria, shows the applicability and reliability of the proposed method.

Key words: Web scraping, natural language processing, decision support system, VIKOR, TOPSIS, ELECTRE, education, multi-criteria decision making

1. GİRİŞ

Üniversite tercihi ana strateji, ana amaç ya da nihai amaç değil amaca giden yolda ara amaç ya da benzer bir ifade ile amaçlara ulaşmada rehber görevini üstlenen, kısa veya uzun vadede gerçekleştirilmeye çalışılan hedeftir, kilometre taşıdır (Beceran, 2016).

2018 yılında Türkiye’de üniversitelere öğrenci olarak yerleşme amacıyla girilmesi gereken zorunlu sınav olan Yükseköğretim Kurumları Sınavı’nda (YKS) başarılı olarak üniversite tercihinin hak kazanan aday sayısı 2.260.273’tür (URL-1, 2019). Ülkemizde devlet ve vakıf olmak üzere toplamda 206 adet üniversite ve bu üniversitelerde toplamda 1818 adet fakülte, 460 yüksekokul, 1001 meslek yüksekokulu ve 702 enstitü bulunmaktadır (URL-2, 2019).

Tercihle bulunacak bir adayın üniversite ile ilgili bilgi kaynağı olarak en sık yararlandığı kaynakların başında üniversitelerin internet sayfaları gelmektedir (Simões ve Soares, 2010). Günümüz adayları Y ve Z kuşağında olmalarından dolayı bilgi teknolojilerini aktif şekilde kullanmaktadırlar. Bu nedenle sosyal medyanın üniversitelerin halka ulaşması için en önemli pazarlama aracı haline geldiği sonucuna varabiliriz (Rámháp ve ark., 2015). Üniversite seçiminde rol oynayan baskın nedenler adaydan adaya değişiklik gösterse de temelde 4 (dört) neden bulunmaktadır. Bunlar: kariyer, ilgi, başkalarına yardım etme fırsatı ve okumak için okumadır (Skatova ve Ferguson, 2014). Öğrenciler, üniversiteden ilk olarak “mesleğe hazırlama” daha sonra ise “entelektüel gelişime katkı” işlevini yerine getirmesini beklemektedir (Şahin ve ark., 2011).

Sorbonne Üniversitesinin 800. yıldönümünün kutlandığı 1988 yılında, Avrupa Birliği’nde yükseköğrenimden sorumlu olan Alman, İtalyan, Fransız ve İngiliz bakanların diğer ülkelere Avrupa Yüksek Eğitim Alanı (European Higher Educational Area - EHEA) oluşumu çağrısıyla Bologna Süreci başlamıştır (Ravinet, 2008). Türkiye’de 2001 yılında Bologna Sürecine katılmış olup sürecin hedeflerine yönelik olarak ulusal düzeyde “Yükseköğretim Alanında Ulusal Yeterlilikler Çerçevesi” (Türkiye Yükseköğretim Yeterlilikler Çerçevesi - TYYÇ) geliştirme çalışmalarına başlamıştır. Yeterliliklerin kazanılma derecesi, her ders/modül esnasında ve sonunda uygun ve nesnel yöntemlerle öğrenme çıktıları olarak ölçülmesi olanağı sağlamaktadır. TYYÇ’nin önemli özelliklerinden birisi de işverenlerin istihdam edecekleri öğrencilerin, mezuniyetleri itibarıyla edinmiş oldukları bilgi ve beceriler hakkında bilgiye sahip olmasına imkân sağlamasıdır (URL-3, 2019).

Günümüz gelişen teknolojisi sayesinde, sanayide dijital dönüşümü gerçekleştirecek teknolojiler ucuz şekilde erişilebilir olmaktadır (Tüsiad ve B.C.G., 2017). Dünya Ekonomi Forumu tarafından yapılan araştırmada, “Endüstri 4.0” ve “Dijital Dönüşüm”ün etkileri nedeniyle, günümüzdeki birçok meslek değişmekte ve/veya yok olmaktadır (W.E.F. ve B.C.G., 2018). Örnek olarak Fransa’da internetin yaygınlaşması ve aktif olarak kullanılması, 500.000 düşük ve orta düzey işin yok olmasına, 1.200.000 yüksek düzey yeni işin oluşmasına neden olmuştur (Manyika ve ark., 2017). Gelişen teknoloji aynı zamanda birçok alanda bilginin yarılanma ömrünü de değiştirmektedir. Neimeyer ve ark. (2012), psikoloji alanında bilginin yarılanma ömrünü araştırmak amacıyla çalışmalarda bulunmuştur. Çalışma sonucunda, ilerleyen on yıl içerisinde profesyonel psikoloji alanındaki bilginin yarılanma ömrü 9 yıldan 7 yıla kadar azalması, uzmanlık gerektiren psikanalitik psikoloji alanında 19 yıldan 3.3 yıla kadar ineceği belirtilmiştir. Jones (1966) tarafından yapılan çalışmada, bir mühendisin sahip olduğu bilginin 10 yıllık yarılanma ömrü sonrası bilgilerini güncel tutabilmek için harcaması gereken zamanı haftalık ortalama 5 saat olarak belirtmiştir. Benzer hesaplamanın günümüzde yapılması durumunda, mühendislik alanına ait bilginin yarılanma ömrünün 2.5 – 5 yıl arasında olduğunu ve bireyin kendisini güncel tutabilmesi için haftalık ortalama 10 saatini ayırması gerektiğini belirtmiştir (Parrish, 2018).

Dünya genelinde başlayan dönüşümün sebep olduğu etkilerden biriside, işveren tarafından aranılan nitelikler aynı ünvanlarda dâhi farklı olabilmektedir. Sürekli değişen işveren ihtiyaçları ve bilginin yarılanma sürelerinin gittikçe kısaltmaya başlaması nedeniyle eğitim kurumları da bu hıza ayak uydurmaya çalışmakta, yeni öncelikli bölümler açmakta, bazı bölümleri kapatmakta veya yeniden tanımlamaktadır. Bunların yanında, yeni ders ve ders içerikleri üretilmektedir. Eğitimciler de bunlara göre eğitim modeli ve içerik oluşturmaktadırlar.

Eğitim sisteminin öğrenenleri de üniversite, bölüm, kurs ve ders seçerken bazı kararlar vermesi gerekmektedir. Karar vermede eğer tek bir kriter varsa alternatif olarak buna rahatlıkla karar verilebilir. Fakat karar problemindeki kriter sayısı birden çok ise bu durumda karar verme de zorlaşmaktadır (Göksu ve Güngör, 2008). Çalışmak istediği mesleğe uygun eğitimi almayı düşünen bir bireyin, tercihte bulunabileceği birden fazla seçenek olacaktır. Bu seçenekleri kendi bilgisiyle oluşturmuş olduğu kriterlere göre sıralaması gerekecektir. Tercih aşamasında olan bireyin belirlediği kriterler, istediği meslek açısından objektif ve piyasa şartlarını karşılar nitelikte ise seçmiş olduğu okulda alacağı eğitim sayesinde, mesleği

ile ilgili donanımlı ve bilgili bir şekilde mezun olacaktır. Aksi durumda, mezuniyet sonrası gerekli nitelikleri karşılamaması nedeniyle başarısız, yani işsiz olacaktır.

1.1. Literatür Taraması

Birey, üniversite ve bölüm seçme aşamasında bir çok kriteri göz önünde bulundurmaktadır. Ülkemizde, YÖK tarafından 2018 yılında YKS'ye giren 31.000'den fazla adayla yapılan anket verilerine bakıldığında, üniversite ve bölüm tercih sürecinde, mezuniyet sonrası iş bulma ve istihdam edilme durumu, adaylar için önemli olmaktadır. Adayların %26'sı tercih edeceği bölümün istihdam imkanını, %18'i üniversitenin bulunduğu şehri, %16'sı ise ilgili bölümün taban puanını göz önünde bulundurarak tercihlerini planlamaktadır (URL-18). Joseph ve ark. (2012) tarafından yapılan çalışmada, bireyin üniversite seçiminde dikkate aldığı kriterler özel ve kamu üniversitelerinde farklılık göstermektedir. Özel üniversiteyi seçmeyi düşünen bireylerde üniversitenin seçkinliği ve itibarı, kişisel etkileşim olanakları ve maliyet kriterleri ön plana çıkmaktadır. Kamu üniversitelerini seçen bireylerde ise üniversitenin bulunduğu yer, programın olanakları, barınma olanakları, sosyal programlar gibi kriterler önem kazanmaktadır. Ülkemizde, merkezi sınavlar ile öğrenci seçme ve yerleştirme sistemi bulunmaktadır. Dışsal faktör olarak değerlendirebileceğimiz bu durumda, bireyin sınavlar sonucunda elde etmiş olduğu puanlar üniversite seçiminde oldukça önemli hale gelmektedir. Gonca Telli Yamamoto (2006) tarafından yapılan çalışma sonuçlarına baktığımızda, ülkemizde üniversite seçim aşamasındaki bireyin kararlarını etkileyen önemli kriterlerden bazılarını üniversite puanı, ailenin görüşü, üniversitelerin kurumsal internet sayfaları, üniversite tanıtım fuarları olarak söyleyebiliriz. YÖK'ün yapmış olduğu anketde sorulan sorulardan biriside adayların tercih süresince yararlandıkları kaynaklar ile ilgilidir. Anketi cevaplayan adayların %24.56'sı rehber öğretmenlerden, %16.99'u aile bireylerinden, %15.19'u sosyal medyadan, %14.92'si arkadaşları ve arkadaş çevresinden, % 11.80'i YÖK'ün geliştirdiği ve kullanıma sunduğu YÖK ATLAS programından, % 10.57'si ise üniversitelerin resmi internet sayfalarından yararlandıklarını belirtmişlerdir (URL-18). Oranlara bakılacak olursa, adayların bilgi edinmek için kullanmış oldukları yöntemlerin %37.56'sı doğrudan internet temellidir. Dolaylı olarak öğretmenlerin, ailelerin ve arkadaş çevresinin de internetteki bilgileri kullandığı düşünülürse, bu oranın çok daha yüksek olduğu görülecektir. Fakat üniversite seçimini yapan bireyin eğitim almakta olduğu bölüm ve ilgili iş piyasası hakkında

çoğunlukla eksik bilgiye sahip olduğu görülmektedir. Baker ve ark. (2018) tarafından yapılan çalışmada, örneklem olarak seçilen öğrenci topluluğunun %15'inden daha azının hangi dalların işgücü piyasasında hangi işleri temsil ettiğini doğru şekilde sınıflandırabilmiştir. Ayrıca iş bulma imkanlarını gerçekte olandan %25 daha düşük görmekte ve ortalama maaşı gerçekten % 13 daha fazla olduğunu düşünmektedirler.

Kureková ve ark. (2016) tarafından yapılan çalışmada, küçük ve orta seviye vasıflara sahip işletmeler tarafından verilen çevrimiçi iş ilanlarında aranılan nitelikler analiz edilerek, analiz sonucu ortaya çıkan gereksinimlerin eğitim öğretim düzenlemesi için faydalı olacağı belirtilmiştir. Karaña ve ark. (2016) tarafından yapılan çalışmada, “Sistem Analisti” ünvanına ait çevrimiçi iş ilanları ve bilgi sistemleri uzmanlarına yapılan anket sonuçları incelenerek, iş veren tarafından aranan ve tercih edilen nitelikler belirlenmiştir. Yapılan çalışmanın, ilgili alanda okuyan öğrenciler ve bu alanda çalışmayı düşünen adaylar için, edinilmesi gereken bilgi ve becerileri gösteren faydalı bir kaynak olacağı belirtilmiştir.

Terblanche ve ark. (2001) çalışmasında, beceri kıtlığını önlenmesi gereken acil bir durum olarak görmüş olup, bu sıkıntıyı giderebilmek için işveren taleplerinin doğru şekilde belirlenmesi ve bu taleplere uygun olacak şekilde eğitim, işgücü gelişimi ve göç politikaları geliştirilmesini önermiştir. İşveren taleplerini belirlemede ise kaynak olarak, çevrimiçi iş ilanları toplanmış, analiz edilmiş ve yarı otomatik “İşveren Talep Ontolojisi” isimli araç geliştirmiştir.

Türkiye’deki endüstriyel ürün tasarımı ve endüstriyel tasarım eğitiminin SWOT analizinin yapıldığı çalışmada, lisans ve lisansüstü eğitim programları, akademisyen sayısı, kayıtlı ve mezun sayıları gibi veriler ile referanslar oluşturulmuştur. “Endüstri Tasarımcısı” profili oluşturulurken, çevrimiçi iş ilanları kullanılarak sektördeki talep ve beklentiler belirlenmiştir. SWOT analizi sonrası, endüstri beklentileri ve eğitim modelinin tutarsız olmasının nedenleri ve öneriler irdelenmiştir (Erkarslan, 2013). Hastaneler tarafından, ölümcül hastalığa sahip insanların manevi huzuru sağlanması amacıyla alınmak istenilen papazlara ait çevrimiçi iş ilanları analiz edilerek talep ve tercih edilen nitelikler belirlenmiştir. Yapılan çalışma neticesinde ortaya çıkan sonuçların, papazlık mesleğinin profesyonelleştirilmesi ve geliştirilmesi amacıyla kullanılabileceği önerilmektedir (Cramer ve Tenzek, 2012).

Malezya’da üniversite mezunu işsizler üzerine yapılan çalışmada, ülke genelinde hizmet veren çevrimiçi iş ilanı sitesinden alınan 300 iş ilanı SPSS paket programı ile analiz edilerek, mezunlardan talep edilen nitelikler ve mezunların nitelikleri kıyaslanmış olup, elde

edilen sonuçlar ışığında bazı önerilerde bulunulmuştur (Omar ve ark., 2012). Avustralya’da, 2006 yılına ait, ülke genelinde yayınlanan çevrimiçi iş ilanları analiz edilerek, “Bilgi Sistemleri” alanında iş verenler tarafından talep edilen bilgi ve beceri gibi nitelikler belirlenmiş ve bu alanda verilen eğitim programlarının etkinliklerini, daha iyi anlama ve düzenleme konularına olumlu katkı yapacağını belirtmişlerdir (Kennan ve ark., 2008). Hirudayaraj ve Baker (2018) tarafından yapılan çalışmada, İnsan Kaynakları (İK) birimine ait çevrimiçi iş ilanları analiz edilerek personellerde aranılan nitelikler açıklanmıştır. İK uzmanı yetiştirilmesi sırasında, eğitim programları ve müfredat hazırlanması aşamasında elde edilen niteliklerin dikkate alınması gerektiği vurgulanmıştır. Bir başka çalışmada, endüstri ihtiyaçlarının belirlenmesi için çevrimiçi iş ilanları analiz edilmiştir. Daha sonra yüksek lisans programları incelenerek, ihtiyaçların akademik kısımda hangi alanlara girmiş olduğu tanımlanmış ve geliştirilmesi gereken alanlar ile ilgili birtakım önerilerde bulunulmuştur (Bowers ve ark., 2018).

Çok sayıda yeni mesleklerin ortaya çıkması ve bu nedenle işveren tarafından ihtiyaç duyulan nitelikleri taşıyacak olan personelin, hangi meslek ya da pozisyonda olması gerektiğinin bilinmemesi durumu, işveren tarafından işçi seçimi aşamasında sıkıntı yaratabilmektedir. De Mauro ve ark. (2018) tarafından yapılan çalışmada, günümüzde firmalar tarafından önemli görülen “Büyük Veri Analizi” geniş bir alan olduğu için firmalar, kendilerine uygun personel özelliklerini belirlemede sıkıntı yaşayabilmektedirler. Bu nedenle internet üzerinden toplanan iş ilanları analiz edilerek alt alanlara ayrılmıştır. Ayrıca firmalar ve eğitimciler tarafından kullanılabilecek ortak bir sözlük oluşturulması amaçlanmıştır.

1.2. Tezin Katkısı ve Organizasyonu

Bu çalışmanın katkıları şöyle sıralanabilir:

1. Web kazıma yöntemleri kullanılarak internet ortamından verilerin nasıl toplanabileceğini,
2. Doğal dil işleme yöntemleri ile metin verilerinin nasıl anlamlı ve işlenebilir bilgiler haline getirilebileceğini,
3. Çok kriterli karar destek sistemleri kullanılarak eğitim sürecinde tercihlerin nasıl önceliklendirilebileceğini gösteren bir klavuz oluşturulmuş olup,

Önerilen sistem kullanılarak:

1. Eğitim ortamları olan okullar, kolejler, üniversiteler ve kurs merkezleri hangi meslek grupları için yeni bölümler açması gerektiğine, ihtiyaç duyulmayan hangi bölümlerin kapatılması veya güncellenmesi gerektiğine ve hangi meslek için hangi dersleri açması gerektiğine;
2. Öğretenler derslerin içerik ve öğretme yönteminin nasıl şekillendireceğine;
3. Öğrenenler de hangi ortamı (kurs, okul, üniversite, online-MOOC), bölümü, mesleği ve dersleri seçeceğine

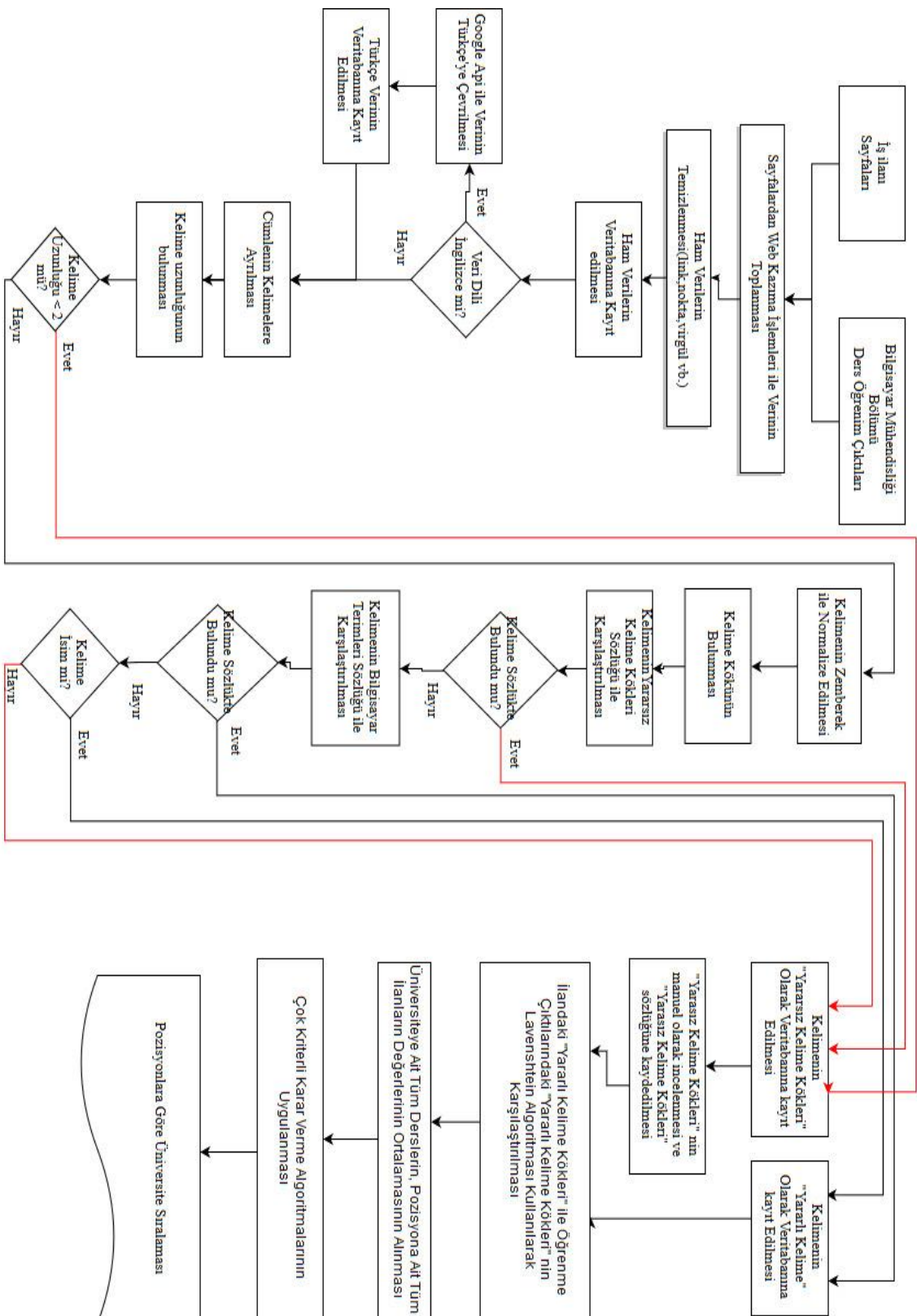
karar vermesine kolaylık sağlayacak ve gelişimlerine pozitif katkı sunacaktır.

Bu tez dokümanı şöyle düzenlenmiştir: Birinci bölümde giriş yapıp literatür ve tezin katkısından bahsedilmiştir. İkinci bölümde ise önerilen sistemi gerçekleştirmek için gerekli olan materyal ve metotlar açıklanmıştır. Üçüncü bölümde, önerilen sistemi gerçeklemek ve test etmek için yapılan örnek uygulamaya ve çıktılarına yer verilmiştir. Son bölümde sırasıyla sonuçlar, öneriler, kısıtlamalar ve potansiyel gelecek çalışmalar hakkında bilgiler sunulmaktadır.

2. MATERYAL VE METOT

Önerilen bilgisayar destekli karar sistemi üç aşamadan oluşmaktadır. Birinci aşaması web kazıma yöntemleri kullanılarak internet ortamından verilerin toplanması aşamasıdır. İkinci aşama elde edilen verilerin, doğal dil işleme yöntemleri ile anlamlı ve işlenebilir bilgiler haline getirilmesidir. Üçüncü aşama çok kriterli karar destek sistemleri kullanılarak tercihlerin önceliklendirilmesidir. Önerilen sistemin akış şeması Şekil 2.1’ de gösterilmektedir. Aşağıda her bir aşamada kullanımına ihtiyaç duyulan materyal ve metotlar detaylı olarak açıklanmıştır.

Bilgisayar programı oluşturma aşamasında önemli adımlardan birisi, bilgisayara yaptırılmak istenilen iş ve işlemler için en uygun programlama dilini belirlemektir. Bir programlama dilinin, güncel ve yaygın kullanıma sahip olması, ulaşılabilir bilgi kaynaklarının yeterli ve kaliteli olması, amaca uygun kütüphanelerinin geliştirilmiş olması, platform bağımsız olması vb. tercih sebebidir. Dünya genelinde, web, masaüstü tabanlı program geliştirme, veritabanı yönetimi, web tasarımı vb. alanlarda amatör veya profesyonel olarak çalışmakta olan kullanıcılardan oluşan ve soru - cevap şeklinde iletişim kurulabilecek, destek sağlanabilecek bir internet sitesi olan Stackoverflow (www.stackoverflow.com) tarafından yapılan ve 90 bin kullanıcının katılım sağladığı “Geliştirici Anketi” sonuçlarına göre, Python programlama dili (%41.7), Java programlama dilini (%41.1) az fark ile geride bırakarak, en hızlı gelişen programlama dili seçilirken, MySQL Veritabanı Yönetim Sistemi %54.00 ile en sık kullanılan veritabanı yönetim sistemi seçilmiştir. Ayrıca Linux işletim sistemi %53.30 ile program geliştiricilerin en çok kullanmış olduğu geliştirme platformu seçilmiştir (URL-4, 2019). Bu nedenlerden ötürü yapılan çalışmada, işletim sistemi olarak Linux (Ubuntu 18.04), programlama dili olarak Python ve ilgili kütüphaneleri, veritabanı yönetim sistemi olarak MySQL/Mariadb kullanılmıştır.



Şekil 2.1. Öneri sistemi akış şeması

2.1. Verilerin İnternet Ortamında Barınması

İnternet üzerinde bulunan veriler sunucularda barınmaktadır. İnternet siteleri HTML (Hypertext Markup Language), CSS (Cascade Style Sheet) gibi işaretleme dilleri kullanarak internet tarayıcılarının yorumlayabileceği şekilde internet üzerinden yayın yapmaktadır. Ayrıca, .Net, python, java, ruby gibi programlama dilleri kullanılarak daha gelişmiş ve kullanıcı etkileşimli sayfalar da oluşturulmakta ve yayınlanmaktadır.

2.1.1. HTML işaretleme dili ve sözdizimi özellikleri

Hipermetin kavramı; “içerisindeki bilgi parçalarının ilişkisini bağlantılar ve işaretler yoluyla gösteren metin” olarak tanımlanabilmektedir (Alessi ve Trollip, 2000). Hipermetinler sadece yazı içermemekle birlikte metin, resim, ses, video, tablo, köprü (link) gibi çok farklı sanal bağlantılar içerebilmektedir. Temel görevi verinin nasıl görüntüleneceği olan Hipermetin İşaretleme Dili (Hypertext Markup Language - HTML), internet sayfalarının yapısını tanımlamaya yarayan standartlara sahip metin işaretleme dilidir.

HTML standartları, World Web Consortium (W3C) tarafından belirlenmekte ve düzenlenmektedir. W3C'nin kurucusu aynı zamanda internetin oluşumunu sağlayan ağ mantığının da mucidi olan Tim Berners-Lee'dir. W3C Ekim 1994 yılında MIT ve Cern bünyesinde kurulmuş ve Dünya çapında ağ (World Wide Web- WWW) standartlarını belirleyen kuruluştur (URL-5, 2019). Son yayınlanan sürümü 5.2 dir. (URL-6, 2019) HTML işaretleme dili “etiket” (tag) ismiyle tabir edilen yapılardan oluşur. Örnek olarak HTML işaretleme dilinde, internet sayfaları 3 temel parçaya ayrılmıştır. Sayfanın üst kısmı “head” etiketi ile başlar (<head>) ve biter (</head>), gövde kısmı “body” etiketi ile başlar (<body>) ve biter (</body>), alt kısmı ise “footer” etiketi ile başlar (<footer>) ve biter (</footer>). Ayrıca internet sayfalarında video, resim, metin, köprü gibi değişik materyallerin kullanılmasını sağlayan etiketler de bulunmakta ve HTML standartlarına uyum sağlayan tüm tarayıcılarda görüntülenebilmektedir.

HTML ile oluşturulan yapılar kendi başına çalışmadığı için programlama dili olarak tanımlanamamaktadır. Bu komutları anlamlandırmak için kullanılan programlara internet tarayıcısı denilmektedir. Günümüzde çok farklı sayıda internet tarayıcısı (Mozilla Firefox, Google Chrome, Internet Explorer vd.) bulunmaktadır. HTML kuralları sayesinde, internet

sayfalarının temel içeriği belirli standartlara bağlanarak oluşturulmakta ve bu sayede sayfalar, her tarayıcıda aynı görsellik sağlanabilmektedir.

Çeşitli programlar aracılığıyla oluşturulan hipermetinlerde, kullanıcının dikkatini çekmek amacıyla farklı yöntemler kullanılabilir. Bunlardan bazıları, bağlantıların metnin geri kalanından farklı renkte olması, eğik, kalın, altı çizili şekilde yazılmaları ya da düğme şeklinde belirtme gibi yöntemlerle sağlanabilir (Jonassen, 1986).

2.1.2. HTML etiket, öge ve özellikleri

HTML işaretleme dilinde kullanılan etiketler (tag) “<” işareti ile başlar ve “>” işareti ile sonlanır. Bu iki işaret arasına amaca uygun etiketler yazılmaktadır. Bazı istisnalar haricinde tüm etiketlerin başlama ve bitiş etiketleri bulunmaktadır. Ayrıca kullanılan etiketlerin başlama ve bitiş arasındaki tüm kısma öge (element) içeriği denilmektedir.

2.1.3. Cascading style sheets

CSS (Cascading Style Sheets), temel olarak sayfa içerisinde bulunan her türlü ögenin (form, buton, arkaplan vb.), görsellik bakımından düzenlenmesini sağlayan bir işaretleme (markup) dilidir. Ayrıca, sayfa içerisinde kullanılan öğelerin bu tip özelliklerini fare hareketlerine (üzerinde, tıklama, tıklamayı bırakma gibi) duyarlı olacak şekilde düzenlenmesine imkân sağlar.

CSS işaretleme dili ile yapılan düzenlemeler HTML sayfasının tamamı için uygulanabileceği (Genel Stil Şablonu) gibi, sadece sayfa içerisinde belirlenen bir nesneye özel olarakta (Yerel stil şablonu) uygulanabilmektedir. CSS komutları, kullanılmak istenilen HTML sayfalarının içerisine yazılabileceği gibi aynı zamanda harici dosya olarakda yazılıp kullanılabilmektedir.

2.1.4. Xpath (XML adresleme dili)

Genişletilmiş işaretleme dili (eXtensible Markup Language - XML), yapı olarak HTML işaretleme diline benzer olmasına rağmen esas amacı, verilerin düzenli şekilde tutulmasını sağlamaktır. HTML işaretleme dili ile arasındaki bir diğer önemli fark ise

kullanıcılar, XML işaretleme dilinde amaçlarına uygun olarak yeni etiketler tanımlayabilmektedirler.

XML adresleme dili olan Xpath, W3 organizasyonu tarafından ilk sürümü 1999 yılında çıkarılmış olup kararlı son sürümü ise 2017 yılında çıkarılmıştır (URL-7, 2019). Xpath, XML formatında olan bir belge içerisinde rahatça gezinebilmek amacıyla belgenin sözdizimsel yapısını düğümler ile belirterek ağaç yapısı şeklinde ortaya koymaktadır. Bu sayede XML olarak yazılmış bir belge soy ağacına benzer yapıya getirilir ve öğeler (element) arası geçişler ebeveyn (parent), çocuk (child) ilişkisi kullanılarak gerçekleştirilir. XPath, XSL Dönüşümleri [XSLT] ve XPointer [XPointer] arasında paylaşılan işlevsellik için ortak bir sözdizimi ve anlamsallık sağlama çabasının bir sonucudur. XPath'ın ana amacı, bir XML belgenin parçalarını birbirlerine adreslemektir (URL-7, 2019).

Örnek HTML kod bloğu:

```
<HTML>
<head>
<title>Yiğit ALIŞAN'ın İnternet Sayfası</title>
</head>
<body>
<tr>
<td></td><td>
<h1>Burası Yazının Başlığı</h1>
<a href="https://www.yigitalisan.com.tr"> Bu bir linktir.</a> </td><td>
<div>

</div></td></tr><td>
<button>Butona tıklayınız</button></td>
</body>
</HTML>
```

Örnekte içeriği gösterilen HTML belgesinde bulunan öğelere ait Xpath yolları:

Buton: /html/body/button

Link: /html/body/a

Resim: /html/body/div/img

Başlık: /html/body/h1 şeklindedir.

2.2. Python Dili ve Özellikleri

Python, 1991 yılında Guidio Van Rossum tarafından tasarlanmış ve şu anda Python Yazılım Vakfı tarafından geliştirmeleri devam etmekte olan “C”, “Perl”, “Lisp”, “Java”, “ABC”, “ALGOL 68” gibi programlama dillerinden etkilenmiş ve “Ruby”, “Comfy”,

“Javascript” gibi dilleri etkilemiş, kaynak kodları herkes tarafından erişilebilir olan, platform bağımsız, nesne yönelimli, yüksek seviyeli bir programlama dilidir.

Program yaşam döngüsünde, önemli olan noktalardan birisi de okunabilirliktir. Okunabilirlik, yazılmış bir programın komutlarının farklı programcılar tarafından rahatlıkla anlaşılabilir olması, ekleme ve çıkarma yapabilme durumu olarak açıklanabilir. Tasarlanma aşamasında okunabilirlik özelliğine oldukça önem verilen Python programlama dilinde, ifade blokları oluşturulurken ayraç karakterleri yerine, boşluktan oluşan girintiler kullanılır.

Python’ın popüler bir programlama dili olmasının temel nedenlerinden birisi de yorumlanan bir programlama dili olmasıdır. Birçok yüksek seviyeli programlama dilinde, komutlar çalıştırıldığı zaman bilgisayarın anlayabileceği yapıya dönüştürülmektedir. Derleme olarak adlandırılan dönüştürme işlemini gerçekleştirmek için kullanılan programlara derleyici denilmektedir. Sonraki aşamada ise kullanıcıya komutların çıktısını vermek içinde tersine derleme işlemi yapılmaktadır. Bu nedenlerden ötürü programlama diline özgü derleyicilere ihtiyaç duyulmaktadır. Derleyiciler ise işletim sistemlerine göre değişmektedir. Python ise yorumlayıcı kullandığı için komutlar çalıştırıldığı anda okunur ve işletilmeye başlanır.

Python programlama dilinin en çok tercih edilen diğer programlama dillerine göre kolay okunabilir olduğu, ekrana “Yiğit ALIŞAN.” yazdırılmak istenilmesi durumunda, yazılması gereken asgari komutları aşağıdaki örneklerde görülmektedir.

Python programlama diline ait örnek kod bloğu:

```
print("Yiğit ALIŞAN.")
```

Java programlama diline ait örnek kod bloğu :

```
public class MerhabaDunya
{
    public static void main(String[] args)
    {
        System.out.println("Yiğit ALIŞAN.");
    }
}
```

C programlama diline ait örnek kod bloğu :

```
#include <stdio.h>
int main(void)
{
    printf("Yiğit ALIŞAN.");
}
```

Python, en basit işlemlerden, üst düzey işlemlere kadar kullanılabilirliği kolay olan kütüphaneler sunmaktadır. Ayrıca ihtiyaç halinde “Matlab”, “R” veya “C” gibi dillerle bağlantılı olarak çalışabilmektedir.

Python açık kaynak kodlu ve GPL (Genel Kamu Lisansı - General Public Licence) destekli Python Yazılım Vakfı Lisansı'na sahip olduğu için kullanıcılara, ücretsiz şekilde edindikleri Python'ı kullanarak geliştirdikleri programları, ticari amaçlı lisanslayarak dağıtım yapabilme imkânı sunmaktadır (URL-8, 2019).

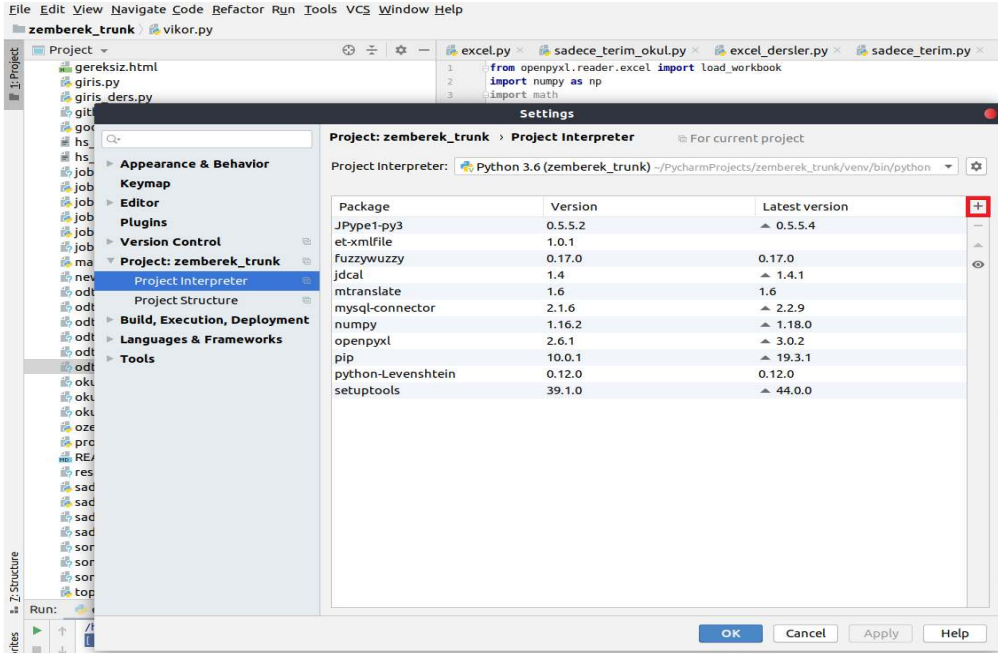
TIOBE (The Importance Of Being Earnest) tarafından yapılan araştırmada, arama motorlarına ait veriler analiz edilerek, arama motorları aracılığıyla aranılan programlama dilleri güncel sıralamasında Python, Java, C ve C++ dan sonra en çok kullanılan genel programlama dili olarak 4. sırada bulunmaktadır (URL-9, 2019). Google Trends verilerine dayanılarak yapılan farklı bir çalışmada, sorgu index sayılarına bakıldığında, Python'ın ilk sırada olan popüler bir programlama dili olduğu sonucuna ulaşılmıştır (URL-8, 2019).

Python'da yazılan komutların yorumlanması için bir yorumlayıcı programa ihtiyaç vardır. Python içerisinde bütünleşik bulunan bütünleşmiş geliştirme ortamı en basit yorumlayıcılardan birisidir.

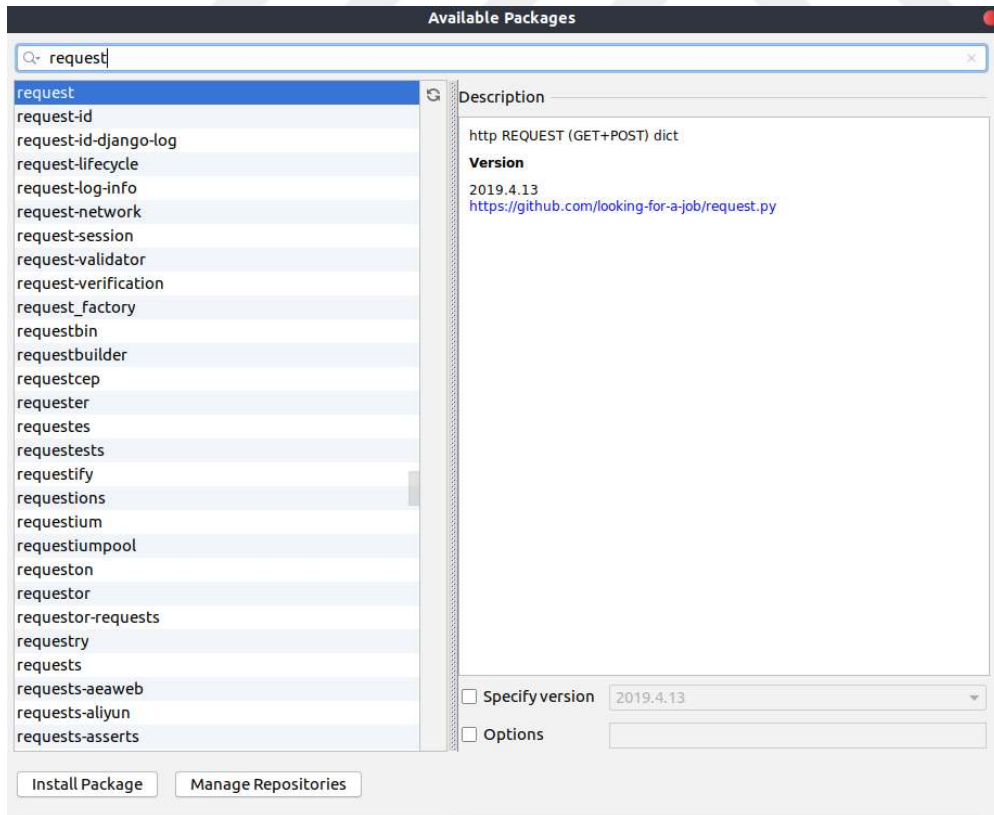
Pycharm ise öncelikli kullanımı Python programlama dili olmak üzere betik dillerini destekleyen, ücretsiz olarak topluluk (community) ve ücretli olarak profesyonel (professional) sürüme sahip JetBrains tarafından geliştirilen bir tümleşik geliştirme ortamı (Integrated Development Environment - IDE) yazılımıdır.

Pycharm, bilgisayara kurulmadan önce Python programlama diline ait temel paketlerin bilgisayarda kurulu olması gerekmektedir. Kodları otomatik olarak tamamlama, güvenli ve program yapısını bozmayacak şekilde yeniden adlandırma ve silme (refactoring), veri tipi gösterimi, SQL söz dizimlerine özel sorgu tamamlama, Github ile tam bütünleşme sağlayabilme, paket yönetimi, kod geçmişini görebilme gibi özellikler, Pycharm'ı öne çıkaran, kullanışlı özelliklerden bazılarıdır.

Pycharm üzerinde açılan bir projeye yeni bir paket kurmak için, ayarlar kısmında bulunan projeye ait olan sekmeden “Project Interpreter” sekmesi seçildiğinde, sağ tarafta proje içerisinde bulunan paketler gösterilir. Şekil 2.2'de sağ üst köşede, kırmızı alanda gösterilen “+” işaretine tıklanarak açılan ekranda (Şekil 2.3), mevcutta bulunan paketlere ait liste görülür. Arama yerine kurulmak istenilen paket ismi yazılarak arama sonucu çıkan listeden paket seçilerek yükleme yapılabilir. Şekil 2.3'de sol en alt kısımda, kırmızı alan ile gösterilen “Install Package” butonuna tıklayarak kurulumu gerçekleştirebiliriz.



Şekil 2.2. Pycharm'da projeye yeni paket ekleme



Şekil 2.3. Pycharm'da python paket listesi

2.2.1. Karakter ifadeler

Karakter ifadelerinin (string) programlarda alfabetik karakterleri ya da karakter topluluğunu saklamak amaçlı kullanılan veri tipidir. Python kullanılarak yazılan programda, kullanılan verinin string olduğunu belirtmek için tırnak işaretleri (') arasında yazılması gerekmektedir. Tırnak işareti sayısı bir ('), iki (') veya üç tırnak (') olarak kullanılabilir. Örnek karakter ifadesi kullanımı ve çıktısı:

```
kelime1 = 'Yiğit Alişan'
print(kelime1)
# Yiğit Alişan # çıktı
kelime2 = "Yiğit Alişan"
print(kelime2)
# Yiğit Alişan # çıktı
kelime3 = '''Yiğit Alişan'''
print(kelime3)
# Yiğit Alişan # çıktı
```

şeklindedir.

Python'da karakter ifadeleri ile matematiksel operatörler birlikte kullanılarak işlem yapılabilir. Örnek olarak artı işareti (+), iki farklı karakter bloğunu birleştirme işlemi yapmaktadır.

Örnek:

```
print('Yiğit' + ' ' + 'ALİŞAN')
# Yiğit ALİŞAN # çıktı
```

şeklinde bir çıktı elde edilir.

Matematiksel ifadelerde çarpma işareti yerine kullanılan yıldız işleci (*) ise, karakter ifadesinin ne kadar sayıda tekrar edeceğini belirtmek için kullanılır.

Örnek:

```
kelime = 'sağol' * 3
print(kelime)
# sağolsağolsağol # çıktı
```

şeklinde bir çıktı elde edilir.

2.2.2. Değişkenler

Değişkenler, program içerisinde bir ya da daha fazla kullanılmak amacıyla oluşturulan, içerisinde karakter ve sayısal değerler saklayabilen veri yapılarıdır. Python programlama dilinde, değişken tanımlanırken verilecek olan değişken isimlerinde uzunluk sınırlaması bulunmamakla birlikte bazı kurallar bulunmaktadır. Değişken tanımlarken verilecek değişken isimleri sayılarla başlamamalıdır. Değişken isimleri harf, sayı ya da alt

çizgi () içerebilmektedir. Değişken ismi olarak Python'da kullanılan komut isimleri ('break', 'continue', 'with', 'try', 'class', 'pass', 'False', 'global', 'finally', 'not', 'raise', 'else', 'or', 'assert', 'return', 'elif' vd.) kullanılamamaktadır.

Ayrıca Python programlama dili büyük, küçük harfe duyarlı (case sensitive) bir sözdizimi yapısına sahiptir.

Örnek;
Degisken_1 = "Kelimeler"
degisken_1 = 1881
print(Degisken_1)
Kelimeler #çıktı
print(degisken_1)
1881 #çıktı

2.2.3. Veri tipleri

Veri tipleri, oluşturulan değişkenlerin türlerini belirtmektedir. Veri türleri, karakter ifadeleri, sayısal ifadeler (karmaşık sayılar, kayan noktasız tam sayılar, kayan noktalı ondalıklı sayılar), diziler vb. şeklinde sıralanabilir. Diğer yüksek seviyeli programlama dillerinde temel olarak bulunan temel veri tiplerinin (ilkel veri tipleri - primitive data type (integer, float, string)) yanı sıra Python'a özel veri tipleri de bulunmaktadır. Ayrıca birçok programlama dilinde değişken tanımlarken veri tipini belirtmek zorunludur. Fakat Python programlama dili için veri tipini belirtmek zorunlu değil, isteğe bağlıdır. Önceden tanımlanmadan kullanılan bir değişkenin hangi veri tipinde olduğu type() fonksiyonu kullanılarak öğrenilebilmektedir.

Örnek:
Degisken_1 = 'kelime'
degisken_1 = 1881
print(type(degisken_1))
<class 'int'> #çıktı
print(type(Degisken_1))
<class 'str'> #çıktı

Sayısal Veri Tipleri:

Python dilinde 3 temel sayısal veri tipi bulunmaktadır. Ayrıca Python programlama dilinde tip dönüşüm işlemleri kolay şekilde gerçekleştirilebilir. Sayısal veri tipleri:

- 1- Kayan noktasız tam sayılar
- 2- Kayan noktalı ondalıklı sayılar
- 3- Karmaşık sayılardır.

Tuple Veri Tipi:

Tuple, listeleri içerisinde barındırabilen bir veri tipi olup en önemli özelliği, ilk tanımlandığı şekilde kalmasıdır. Bu nedenle daha sonra ekleme, çıkarma gibi işlemler yapılmamaktadır. Değişken içeriği oluşturulurken normal parantez işareti “()” kullanılması durumunda, Python programlama dili değişkenin veri tipini tuple olarak algılar ve bu şekilde çalışır.

Örnek:

```
degisken_1=('Ey',19,'TÜRK',5,'Milleti',1881)
print(type(degisken_1))
# <class 'tuple'> #çıktı
print((degisken_1))
# ('Ey', 19, 'TÜRK', 5, 'Milleti', 1881) #çıktı
```

Dictionary Veri Tipi:

Dictionary veri tipi, anahtar ve değer ikilisi olarak verileri tutmaktadır. Tanımlama yapılırken süslü parantez “{}” kullanılır ve böylece Python programlama dili değişkenin dictionary veri tipinde olduğunu otomatik olarak yorumlar.

Örnek:

```
sozluk = {"Merhaba": "Hello", "Dünya": "World"}
print(sozluk["Merhaba"])
# Hello #çıktı
print(sozluk["Dünya"])
# World #çıktı
```

Set Veri Tipi:

Tam olarak matematikteki kümelere özgü özelliklerin kullanılabildiği veri tipidir. Set veri tipi şeklinde tanımlanmış değişkenleri kullanarak kümelere özgü olan kesişim, birleşim, fark gibi matematiksel işlemler gerçekleştirilebilmektedir.

List Veri Tipi:

List (liste) veri tipi, içerisinde farklı türden veri tutulabilmektedir. Köşeli parantez “[]” kullanılarak değişken oluşturulması durumunda Python programlama dili, değişken veri tipini List olarak yorumlamaktadır. List içerisindeki elemanlara ekleme, çıkarma, değiştirme, sıralama, ters çevirme, çoğaltma gibi işlemler yapılabilmektedir. List veri tipleri içerisindeki elemanların sıra sayıları ilk eleman 0. sırada olacak şekilde hesaplanarak çağrılır. İlk elemanın 0. sırada olması nedeniyle son elemana ait sıra sayısı her zaman, eleman sayısı - 1 şeklindedir.

Örnek:

```
degisken_1 = ['Pi', 'sayısının', 'değeri']
print(type(degisken_1))
# <class 'list'> # çıktı
degisken_1.append(3.14)
print(degisken_1)
# ['Pi', 'sayısının', 'değeri', 3.14] #çıktı
del degisken_1[1]
print(degisken_1)
# ['Pi', 'değeri', 3.14] # çıktı
```

2.2.4. Şart kontrol komutları

Herhangi bir iş gerçekleştirmek için yazılan programlar her zaman tek olasılık düşünülerek yazılmamalıdır. Bunun nedenleri olarak, kullanıcı tarafından girilecek bilgilere göre işlem yapılması veya programın bağımlı olduğu çevresel kaynaklardan oluşan problem sonucu hataya düşme ihtimali vb. örnekler verilebilir. Bu gibi nedenlerden ötürü şartlı dallanma ifadeleri bulunmaktadır.

If Şart Deyimi:

If (eğer) şart deyiminin temel kullanım amacı, program içerisinde belirtilen bir şart durumunun oluşup oluşmamasının kontrol edilmesi ve sonuca göre farklı komut bloklarının çalışmasını ya da çalışmamasını sağlamaktır.

If komutu sonrası yazılan şart sağlanıyorsa if bloğu içerisindeki komutlar çalıştırılmaktadır. Aksi durumda if bloğu içerisindeki komutlar yorumlayıcı tarafından okunmadan atlanmaktadır.

Elif Şart Deyimi:

Yazılan program içerisinde birden fazla şartın kontrol edilmesi durumunda kullanılır. Else şart deyimine benzer olarak if şart deyim bloğundan sonra kullanılır. Else şart deyimi arasındaki tek fark ise tıpkı if şart deyim bloğu sonrası şart sınaması yapılabildiği gibi, elif şart deyimi sonrasında da şart sınaması yapılabilmektedir.

Else Şart Deyimi:

Else (değilse) şart deyimi, program içerisinde tek başına bir anlam ifade etmemekte ve kullanılamamaktadır. If veya Elif şart deyimi bloğundan sonra, bir defaya mahsus olmak üzere kullanılabilmektedir. Kullanım amacı ise if şart deyimi bloğundaki şart sağlanmadığı durumda, yorumlayıcı else şart deyimi bloğu içerisindeki komutları çalıştırmaktadır.

2.2.5. Döngü komutları

Program yazma esnasında, bazı kod bloklarının belirtilen sayıda tekrar edilmesi ihtiyacı oluşabilmektedir. Bu gibi durumlarda tekrar edilmesi istenilen kod blokları döngü adı verilen komutlar arasına yazılarak istenilen sayıda tekrarı sağlanabilmektedir.

For Döngüsü:

Basit ve anlaşılabilir olması nedeniyle programlama dillerinde en fazla tercih edilen döngü komutu olan for döngüsü sayesinde istenilen kod bloğu, istenildiği kadar tekrar edilebilmektedir.

Örnek:

```
degisken_1=['Ey',19,'TÜRK',5,'Milleti',1881]
print(type(degisken_1))
# <class 'list'> #çıktı
for i in degisken_1: # veya for i in range(0,Len(degisken_1)):
    print(degisken_1)
...
['Ey', 19, 'TÜRK', 5, 'Milleti', 1881]
['Ey', 19, 'TÜRK', 5, 'Milleti', 1881]
['Ey', 19, 'TÜRK', 5, 'Milleti', 1881]
['Ey', 19, 'TÜRK', 5, 'Milleti', 1881]
['Ey', 19, 'TÜRK', 5, 'Milleti', 1881]
['Ey', 19, 'TÜRK', 5, 'Milleti', 1881]
''' #çıktı
```

Örnekteki for döngüsü içerisinde kullanılan “i” değişkeni değiştirilmediği sürece 0 dan başlamakta ve her defasında bir artmaktadır. Döngü, list veri tipinde olan “degisken_1” değişkeninin eleman sayısına ulaşınca kadar bu artış devam etmekte ve son olarak döngüden çıkılmaktadır.

While Döngüsü:

While döngü komutu ile istenilen bir kod bloğu belirtilen şart sağlandığı sürece tekrarlanmaktadır. Şart sağlanmadığı zaman döngü sonlandırılmaktadır.

Örnek:

```
degisken_1 = ['Ey', 19, 'TÜRK', 5, 'Milleti', 1881]
sayac = 5
while sayac > 1 :
    print(degisken_1)
    sayac -= 1
...
['Ey', 19, 'TÜRK', 5, 'Milleti', 1881]
['Ey', 19, 'TÜRK', 5, 'Milleti', 1881]
['Ey', 19, 'TÜRK', 5, 'Milleti', 1881]
['Ey', 19, 'TÜRK', 5, 'Milleti', 1881]
''' # çıktı
```

Örnekte değer ataması yapılan “sayac” değişkeni, 1 sayısından büyük olduğu sürece while döngüsü içerisindeki komutlar çalıştırılmaya devam etmektedir. Her çalıştırılmasında ekrana yazı yazılacak ve sayac değişkenine ait değer bir azaltılmaktadır. Sayac değişkenine ait değerin 1 olması durumunda ise döngüden çıkılmaktadır.

2.2.6. Fonksiyonlar

Program yazma aşamasında, birden fazla yerde kullanılacak olan kod bloklarını her defasında tekrar tekrar yazmak programın okunabilirliğini düşürmektedir. Bu nedenle bu tip kod blokları için fonksiyon (function) kullanılmaktadır. Fonksiyonlar, program içerisinde sıklıkla kullanılacak olan bir veya daha fazla satırdan oluşan kod bloklarını, her istenildiğinde çağrılacak şekilde düzenlemeye yarar. Python programlama dilinde, fonksiyonlar “def” komutu ile tanımlanmaktadır. Her fonksiyonun benzersiz bir adı olmak zorundadır. Ayrıca, fonksiyon içerisinde istenildiği kadar değişken tanımlamaları yapılabilmektedir. Python programlama dili içerisinde bulunan ve sıklıkla kullanılan bazı fonksiyonlara örnek olarak “print()”, “type()”, “input()”, “len()” fonksiyonları gösterilebilir.

Örnek:

```
def toplama(a,b):  
    return a+b #şeklinde girilen iki sayıyı toplayan bir fonksiyon oluşturuldu.  
print(toplama(1,8))  
# 9 çıktı
```

Örnekte tanımlanan toplama fonksiyonu, iki parametreye değer yüklenerek çağırılması sonucu, iki parametreye girilen değerlerin toplamını döndürmektedir.

2.3. Kütüphane Modülleri

Kütüphane modülleri, program yazma aşamasında ihtiyaç duyulabilen bazı işlevleri yerine getirmeyi sağlayan, içerisinde yapılaş amacına uygun olan sınıfları, fonksiyonları, nitelikleri vb. barındıran ve ihtiyaç duyulmayan kısımların çıkartılması sonucu hızlı, okunabilirliği yüksek bir yapıda olan ve bu sayede kaynakları etkin kullanan hazır araçlardır.

Python, açık kaynak kodlu ve yaygın olarak kullanılan bir programlama dili olması nedeniyle çok sayıda hazır kütüphaneye sahiptir. Bu sayede yazılan programlarda tekrar eden komut sayıları azalmakta ve kütüphaneler tekrar tekrar kullanılabilir.

Örnek:

```
from selenium import webdriver
import time
import BeautifulSoup as bs4
```

Örnekte gösterildiği şekilde program içerisinde kullanılacak olan kütüphaneler, programın en başında projeye eklenmek zorundadır. “import” komutu ile kütüphaneye ait tüm modüller eklenebileceği gibi, “from ... import ...” komut yapısı kullanılarak kütüphane içerisinde bulunan tek bir modül de direk eklenebilmektedir. Ayrıca projeye eklenen kütüphaneye “import ... as ...” yapısını kullanarak takma ad (alias) tanımlanabilir ve proje boyunca takma ad kullanılarak işlemler gerçekleştirilebilmektedir.

2.3.1. BeautifulSoup kütüphanesi

BeautifulSoup, internet sayfalarını, HTML ve XML dosyalarını okuyarak ve komutları ayrıştırarak ağaç yapısına çeviren ve bu sayede istenilen kısımları, doğrudan veya düzenli ifadeler oluşturularak alınmasını, değiştirilmesini, eklenmesini sağlayan hızlı ve aynı zamanda esnek bir kütüphanedir. BeautifulSoup’un 4. sürümü (BS4) güncel ve kullanılması teşvik edilen bir sürüm olup girdileri Unicode’a, çıktıları ise otomatik olarak UTF-8 formatına çevirir. Bu sayede dosyalarla ilgili çeviri sıkıntısının önüne geçilmiştir.

BS4’ün güçlü bir kütüphane olmasını sağlayan en önemli özelliklerinden birisi, lxml ve Html5lib kütüphanelerini iyi şekilde kullanmasıdır. Bu sayede bozuk bir yapıya sahip HTML veya XML dosyalarını da kolay şekilde ayrıştırabilmektedir. BS4, girdi olarak gönderilen veriyi ayrıştırarak bir ayrıştırma ağacı (parse tree) oluşturur. Oluşturulan ayrıştırma ağacında istediğimiz verilere ulaşabiliriz.

BS4 kütüphanesi, Python programlama diline ait kütüphane listesinde bulunmakta ve bu sayede istenilen projeye kolay şekilde eklenebilmektedir. Bu işlem sonrasında kullanılmak istenilen sayfaya “import” komutu ile eklenmesi yeterlidir.

Örnek:

```
from bs4 import BeautifulSoup

ornek_dosya = """
<HTML><head><title>BeautifulSoup</title></head>
<body>
<p class="baslik"><b>BeautifulSoup</b></p>
```

```

<p class="aciklama"> BS4 kütüphanesi hızlı ve güçlü olmasını
<a href="http://www.yigitalisan.com.tr/lXML" class="programlama"
id="kopru1">lXML</a>,
<a href="http://www.yigitalisan.com.tr/HTML5lib" class="programlama"
id="kopru2">HTML5lib</a> gibi
python işleyicilerini kullanmasına borçludur</p>
<p class="aciklama">...</p>
"""

soup = BeautifulSoup(HTML_dosyasi, 'html.parser')
print(soup.prettify())
# HTML çıktısı aşağıda gösterilmektedir.
# <HTML>
# <head>
# <title>
#     BeautifulSoup
# </title>
# </head>
# <body>
# <p class="baslik">
# <b>
#     BeautifulSoup
# </b>
# </p>
# <p class="aciklama">
#     BS4 kütüphanesi hızlı ve güçlü olmasını
#     <a href="http://www.yigitalisan.com.tr/lxml" class="programlama"
id="kopru1">
#     lXML
#     </a>
#     ,
#     <a href="http://www.yigitalisan.com.tr/Html5lib" class="programlama"
id="kopru2">
#     HTML5lib
#     </a>
#     gibi python işleyicilerini kullanmasına borçludur.
# </p>
# <p class="aciklama">
#     ...
# </p>
# </body>
# </HTML>

```

Örnekte girdi olarak kullanılan “ornek_dosya” isimli HTML dosyası, BS4 kütüphanesi tarafından okunup ayrıştırılmakta ve ağaç yapısına çevrilmektedir. Daha sonra ağaç yapısını oluşturan dallar içerisinde, ebeveyn (parent), kardeş (sibling) ilişkisi dikkate alınarak istenilen şekilde arama, ekleme ve çıkarma yapılabilir hale getirilmektedir. Ayrıca arama, seçme işlemlerinde HTML etiketleri kullanılarak tüm dosya içerisinde bulunan metin, köprü (link) vb. bilgiler çekilebilmektedir.

Örnek:

```

soup.title
# <title>BeautifulSoup</title>

```

```

soup.title.name
# u'title'
soup.title.string
# u'BeautifulSoup'
soup.title.parent.name
# u'head'
soup.p
# <p class="baslik"><b>BeautifulSoup</b></p>
soup.p['class']
# u'baslik'
soup.a
# <a href="http://www.yigitalisan.com.tr/lxml" class="programlama"
id="kopru1">lxml</a>
soup.find_all('a')
# [<a href="http://www.yigitalisan.com.tr/lxml" class="programlama"
id="kopru1">lxml</a>,
# <a href="http://www.yigitalisan.com.tr/Html5lib" class="programlama"
id="kopru2">Html5lib</a>]
soup.find(id="kopru2")
# <a href="http://www.yigitalisan.com.tr/Html5lib" class="programlama"
id="kopru2">Html5lib</a>
for link in soup.find_all('a'):
    print(link.get('href'))
# http://www.yigitalisan.com.tr/lxml
# http://www.yigitalisan.com.tr/Html5lib

```

2.3.2. LXML kütüphanesi

Lxml kütüphanesi, Python programlama dilinde özellikle XML işaretleme dili kullanılarak oluşturulmuş olan dosyaların yanısıra HTML işaretleme dili kullanılarak oluşturulan dosyalarda doğrulama, düzenleme ve parçalama işlemlerini gerçekleştirmek için tercih edilir. Ayrıca HTML dilindeki dosyaları hızlı ve düzenli şekilde parçalama işine de yarar. Bu nedenlerden ötürü web kazıma işlemlerinde en çok kullanılan kütüphanelerden birisidir.

Lxml kütüphanesi, Xpath metodunu kullanarak Xpath değişkenlerini destekler. Bu özellik sayesinde internet sayfalarında bulunan veriler toplanmak (web scraping, web crawling vb.) istenildiği zaman, internet tarayıcılarının özellikleri kullanılarak, internet sayfası içerisindeki herhangi bir nesnenin, resmin, bağlantının ya da metnin Xpath yolu kolayca öğrenilebilmekte ve bu kolaylık sayesinde CSS veya HTML etiketleri kullanılarak düzenli ifade oluşturma ile zaman kaybedilmemektedir.

Örnek:

```

import requests
r = requests.get('http://www.yigitalisan.com.tr/')
ornek_dosya= lxml.html.fromstring(r.content)
linkleri_topla=ornek_dosya.xpath("/html/body/div/a")

```

Örnekte, “ornek_dosya” isimli HTML sayfasında, “HTML” etiketi içerisinde bulunan “body” etiketinin çocuğu (sibling) olan tüm “div” etiketleri içerisindeki “a” etiketlerini toplama işlemi yapılmaktadır.

2.3.3. Requests kütüphanesi

İnternet tarayıcıları, kullanıcıların adres çubuğuna yazmış oldukları internet sayfalarını getirmek için internet sayfalarının bulundukları sunuculara istekte bulunurlar ve daha sonra sunucu tarafından cevap olarak gelen komutları derleyerek kullanıcının ekranında göstermektedirler. Web kazıma işlemi yapmak istenilen sayfalara ulaşma işleminin, program tarafından hızlı ve otomatik şekilde yapılabilmesi için kullanılan Python programlama dili kütüphanesi ise Requests'tir.

Program içerisinde requests kütüphanesi kullanılarak, internet tarayıcılarına ihtiyaç duyulmadan, otomatik ve sürekli olacak şekilde bir ya da daha fazla internet sayfasına istek veya parametre gönderilebilmektedir.

Örnek:

```
import requests
r = requests.get('http://www.yigitalisan.com.tr/')
r.text
r = requests.get('http://www.yigitalisan.com.tr/')
r.headers
```

Örnekte ilk olarak, Requests kütüphanesi import edilmektedir. Daha sonra kütüphaneye ait “get” metodu ile internet sayfasına (http://www.yigitalisan.com.tr/) talep gönderilmektedir. İnternet sayfası tarafından verilen cevap, “r” değişkenine aktarılarak “r.text” komutu ile, gelen cevap içerisindeki “text” bilgilerine ulaşılmaktadır. Aynı işlem basamakları gelen cevap içerisindeki “header” bilgisine ulaşmak için tekrarlanmaktadır.

2.3.4. Selenium kütüphanesi

Günümüzde internette gezinmek için ihtiyaç duyulan ilk araç internet tarayıcısıdır. Kullanıcı ziyaret etmek istediği internet sayfasına ait adresi, internet tarayıcılarının adres çubuğuna yazarak kolaylıkla erişim sağlayabilmektedir.

Selenium, internet tarayıcılarını, programların otomatik olarak ihtiyaç duyulan şekilde kullanması işlemini gerçekleştiren, internet tarayıcısı ve program arasında iletişim sağlayan ara programdır. Selenium, yeni yazılmış ya da var olan bir internet sayfasının performans, güvenlik gibi test işlemleri için kullanılabileceği gibi aynı zamanda bir internet sayfasındaki herhangi bir elemente tıklama işlemini de otomatikleştirebilmektedir. Python, .Net, Ruby, Java gibi güncel üst seviye programlama dillerini destekleyen Selenium, ayrıca açık kaynak kodlu yapısı sayesinde birçok platformda kolaylıkla çalışabilmektedir. Selenium otomasyon aracının IDE versiyonuna ait komut satırı çalıştırıcısı kullanılarak, herhangi bir tarayıcı/işletim sistemi kombinasyonunda paralel olarak çalışabilmektedir (URL-10, 2019) . Ayrıca program yazmaya ihtiyaç duymadan IDE tarafından desteklenen internet tarayıcılarına entegre edilerek internet sayfalarında istenilen işlemler gerçekleştirilebilmektedir. Selenium web drive aracı, internet sayfaları içerisindeki elementlere ulaşmak için Xpath yolu, ID değeri, bağlantının metin bilgisi, DOM (Data Object Model) ve element isim özelliği gibi çeşitli yer bulucu kullanılmasına imkân sağlamaktadır.

Selenium web drive aracını kullanmak için, iletişim kuracağı internet tarayıcısına özgü eklentinin ayrıca programa eklenmesi gerekmektedir. Örnek olarak “Mozilla Firefox” isimli internet tarayıcısı için kullanılan Selenium web driver eklentisinin ismi "Gecko Driver" olarak geçmektedir. Mozilla Firefox, Google Chrome, Internet Explorer gibi popüler internet tarayıcılarına ait internet sürücüler (web driver) bulunmaktadır.

Örnek HTML sayfası kesiti:

```
</div>
</div>
<div class="login-info-boxes">
<div class="txt clearfix">
<input name="ctl00$MasterPageBody$lgnUserName" type="text" id="lgnUserName"
placeholder="Kullanıcı Adı veya E-posta" maxlength="100" />
</div>
</div>
<div class="remember-me">
....
</div>
</div>
<!--ilan listeleme-->
<div class="dipnot">
</div>
<div class="pagination">
```

```

<a id="lnkPrevPage" onclick="dataLayer.push({'Category':'Is
Ara','Action':'Sayfa','Label':'Geri','event':'gaEvent'}); " class="nav prev
disabled"></a>
<a href="javascript:;" id="lnkCurrentPage" class="nav cur">1</a>
<a href="javascript:;" id="lnkNextPage" onclick="dataLayer.push({ 'Category':
'Is Ara', 'Action': 'Sayfa', 'Label': 'İleri', 'event': 'gaEvent' });
updateCriteria(2)"" class="nav next "></a>
<a id="lnkLastPage" class="last">100 sayfa</a>
<input type="hidden" id="hdnCurrentPage" name="hdnCurrentPage" value="1" />
<noscript>
</noscript>
...

```

Örnek Python programlama dili kod bloğu:

```

1.from selenium import webdriver
2.tarayici=webdriver.Firefox()
3.tarayici.get('https://www.yigitalisan.com.tr/login.aspx')
4.kull_ad = tarayici.find_element_by_id("lgnUserName")
5.kull_ad.send_keys("yigit.alisan@linux.org.tr")
6.tarayici.execute_script("window.scrollTo(0, 500)")
7.tarayici.find_elements_by_xpath("//*[@id='lnkNextPage']")[0].click()

```

Örnek kod bloğu satır açıklamaları:

1. adımda, Selenium otomasyon aracının Python programlama diline özgü kütüphanesi içerisindeki "webdriver" modülü projeye eklenmektedir.
2. adımda, "tarayici" isimli değişkene "webdriver" modülünün Mozilla Firefox eklentisi özelliği kazandırılmıştır.
3. adımda, eklentinin "get" metodu sayesinde program tarafından bilgisayarda kurulu bulunan Mozilla Firefox internet tarayıcısı kullanılarak açılan yeni sekmede "https://www.yigitalisan.com.tr/login.aspx" adresine giriş yapılmıştır.
4. adımda, açılan sayfa içerisinde aranılan elementin adı girilerek (lgnUserName) "find_element_by_id" metodu aracılığıyla bulunup değeri, "kull_ad" isimli değişkene aktarılmıştır.
5. adımda, "send_keys" metodu yardımıyla, bulunan elemente (Textbox) değer olarak "yigit.alisan@linux.org.tr" metninin yazılması sağlanmıştır.
6. adımda, "execute_script" metodu kullanılarak, işlem yapılan internet tarayıcısının özelliklerinden "window.scrollTo()" metodu çalıştırılarak, internet tarayıcısının kaydırma çubuğu 500 px aşağıya çekilmiştir. Böylece, sayfayı aşağı kaydırınca otomatik olarak yükleme (scroll down auto load data) özelliği olan sayfalarda, program otomatik olarak sayfayı aşağı, yukarı kaydırarak verilere erişebilmektedir.
7. adımda, sayfa içerisinde "lnkNextPage" isimli element (button) Xpath değeri kullanılarak bulunmuş olup, "click()" metodu ile tıklama işlemi yapılmıştır.

2.3.5. Numpy kütüphanesi

Numpy (Numerical Python), bilimsel hesaplamaları hızlı şekilde yapan, matrisler ile çalışma imkânı sağlayan, ayırık Fourier dönüşümü, trigonometrik, üstel fonksiyon, logaritmik gibi ileri matematiksel işlemler yapmak için kullanılan matematik kütüphanesidir. Numpy dizileri temelinde oluşturulmuş kütüphanede, python programlama dilindeki “list” lerden farklı olarak numpy dizilerinin tüm elemanları aynı veri tipine sahip olmak zorundadır. Bu özellikten dolayı numpy dizileri hız ve işlevsellik açısından python programlama dilindeki “list” lerden daha iyidir.

Numpy kütüphanesi, Python programlama diline ait kütüphane listesinde bulunmakta ve bu sayede istenilen projeye kolay şekilde eklenebilmektedir. Numpy kütüphanesi kullanılarak oluşturulan diziler sabit boyutlu olup, python programlama dilindeki liste veri tipinde olduğu gibi dinamik olarak boyut değiştirme işlemi yapılmamaktadır.

Örnek:

```
import numpy as np
numpy_dizisi = np.array([0,1,2,3,4,5]) # array formatında numpy_dizisi isimli
dizi tanımlanmıştır.
print(numpy_dizisi) # numpy_dizisi nin çıktısı ekrana verilmiştir.
0 1 2 3 4 5 # Dizinin eleman çıktısı
print(numpy_dizisi.ndim) # “ndim” komutu kullanılarak, numpy_dizisi nin boyutu
ekrana verilmiştir.
1 # Dizinin boyut çıktısı
print(numpy_dizisi.shape) # “shape” komutu kullanılarak, numpy_dizisi nin satır
ve sütun sayısına ait bilgiler ekrana verilmiştir.
(6,) # Dizinin satır ve sütun sayısı çıktısı
numpy_dizisi.reshape((2,3)) # “reshape” komutu kullanılarak, numpy_dizisi nin
boyutu 2x3 şeklinde düzenlenmiştir.
print(numpy_dizisi)
[[0,1,2]
 [3,4,5]] # Yeniden boyutlandırılan dizinin elemanlarına ait çıktı
print(numpy_dizisi.ndim)
2 # Yeniden boyutlandırılan dizinin boyut çıktısı
print(numpy_dizisi.shape)
(2,3) # Yeniden boyutlandırılan dizinin satır ve sütun sayısı çıktısı
```

Örnekte görüleceği üzere ilk olarak “import” komutu kullanılarak Numpy kütüphanesi çalışmaya eklenmektedir. Numpy kütüphanesini projede kullanmak istediğimizde kütüphane ismi yerine tanımlamış olduğumuz “np” ismini (alias) kullanabiliriz. Numpy kütüphanesindeki “array” komutu ile dizi tanımlama işlemi yapılır. Oluşturulan dizinin boyutunu öğrenmek için numpy kütüphanesinin “ndim” fonksiyonundan yararlanılır. Oluşturulan dizinin satır ve sütun sayısını bulmak için numpy kütüphanesinin “shape” fonksiyonundan yararlanılır. Numpy kütüphanesi kullanılarak oluşturulan dizilerde satır ve sütun sayısını değiştirmek için “reshape” fonksiyonu kullanılır. Burada önemli olan nokta matris eleman sayısının aynı kalmak zorunda olmasıdır.

Numpy ile oluşturulan tek veya çok boyutlu diziler üzerinde toplama, çıkarma gibi temel işlemlerin yanı sıra matrislerde tersine çevirme, transpozunu alma, satır veya sütunları toplama, standart sapma, varyans, medyan gibi işlemleri de kolaylıkla yerine getirebilecek fonksiyonlarda bulunmaktadır.

Örnek:

```
import numpy as np
matris = np.array([0,1,2,3,4,5,6,7,8,9])
print(matris.max())
# 9 # Matris içerisindeki en büyük sayı değeri
print(matris.min())
# 0 # Matris içerisindeki en küçük sayı değeri
print(matris.sum())
# 45 #Matristeki elemanların toplam değeri
print(matris.mean())
# 4.5 # Matrisin orta değeri
print(np.median(matris))
# 4.5 #Matrisin medyan değeri
print(matris.var())
# 8.25 #Matrisin varyans değeri
print(matris.std())
# 2.8722813232690143 #Matrisin standart sapma değeri
```

2.3.6. Openpyxl kütüphanesi

Program çıktıları veritabanlarına yazılabileceği gibi aynı zamanda Excel ya da metin dosyalarına da yazılabilmektedir. Microsoft Excel programı, XML altyapısını kullanması nedeniyle büyük boyutlu verileri çok hızlı şekilde işleyebilmektedir.

Openpyxl kütüphanesi, Python programlama dili ile yazılan programların Excel dökümanlarını oluşturma, okuma, yazma, silme ve değiştirme gibi işlemleri gerçekleştirmesini sağlamaktadır. Openpyxl kütüphanesi ile binlerce satırdan oluşan excel dökümanları, hızlı şekilde okunabilmekte ve istenilen işlemler, programlar aracılığıyla kolayca gerçekleştirilebilmektedir (URL-11, 2020). Bunun yerine Pandas DataFrame de kullanılabilir.

Openpyxl Excel dökümanlar üzerinde hakimiyet sağlaması nedeniyle stil özellikleri de kolay şekilde kullanılabilmektedir.

Örnek:

```
1. from openpyxl import *
2. from openpyxl.styles import Font, Color, Alignment, PatternFill
3. from openpyxl.utils import get_column_letter
4. kitap = Workbook()
5. kitap.create_sheet("ilksayfa")
6. kitap.get_sheet_by_name("oranlar")
```

Örnek kod bloğu satır açıklamaları:

- 1., 2. ve 3. adımda, temel openpyxl kütüphanesini, stil modülünü ve hücre düzenleme modülü projeye eklenmektedir.
4. adımda, "kitap" isimli değişkene "Workbook" metodu özellikleri yüklenmektedir.
5. adımda, "create_sheet()" metodu yardımıyla "ilksayfa" isimli yeni bir çalışma sayfası oluşturulur.
6. adımda, "get_sheet_by_name()" metodu kullanılarak "kitap" isimli excel dosyası içerisindeki sayfalardan "oranlar" isimli çalışma sayfası aktif sayfa olarak belirlenir.

Örnek:

```
1. oranlar["A5"].fill = PatternFill(start_color="FFE699", fill_type="solid")
2. oranlar["A5"] = "Burası A5 hücresi"
3. oranlar.merge_cells(start_row=5, start_column=A, end_row=5, end_column=B)
```

Örnek kod bloğu satır açıklamaları:

1. adımda, Excel dosyası içerisindeki "oranlar" isimli çalışma sayfasında bulunan "A5" ismine sahip hücrenin arka plan rengi değiştirilmiştir.
2. adımda, "A5" hücresinin içerisine "Burası A5 hücresi" verisi eklenmiştir.

3. adımda, "A5" hücresi ile "B5" hücresi birleştirilmiştir.

2.3.7. MySQL.connector kütüphanesi

Veritabanı Yönetim Sistemleri'nin (VTYS) gelişmesiyle birlikte, depolanmış verilerin işlenmesi sırasında kullanılmak üzere yapılandırılmış sorgu dili olan SQL (Structured Query Language) oluşturulmuştur. SQL sorgu dilinin temel komutları aynı olduğu halde farklı VTYS'lerde farklı işlemlere özgü yeni komutlarda kullanılmaya başlanmıştır.

Arama motoru sonuçları (Google ve Bing), soru-cevap sitelerindeki konu sayıları, iş ilanlarındaki kullanım sıklıkları ve sosyal ağlardaki ilgiler dikkate alınarak oluşturulan en çok kullanılan VTYS listesi Oracle, MySQL, Microsoft SQL Server, PostgreSQL şeklinde sıralanmaktadır (URL-1, 2020). En çok tercih edilen VTYS listesinde ikinci sırada bulunan MySQL ilişkisel bir veri tabanı olup, tüm verileri tek bir tabloya ya da veri tabanına yıkmak yerine farklı tablolarda ya da farklı veri tabanlarında düzenli olacak şekilde saklamaktadır. VTYS'ler için standart bir dil haline gelen SQL sorgu dili kullanılarak işlem yapılmaktadır.

Michael Monty Widenius tarafından yazılan MySQL ilk ortaya çıktığı tarih olan 1995 yılında hem GPL hem de ayrı bir lisans ile kullanılmaktaydı. 2008 yılında Sun firması tarafından satın alınan MySQL'in yapısı değiştirilmiştir. Bu durum üzerine 2009 yılında Michael Monty Widenius tarafından MySQL kopyalanarak (fork - çatallama), Mariadb isimli ve sadece Genel Kamu Lisansı altında dağıtılabilen yeni bir VTYS oluşturulmuştur. Oracle firmasının ise 2010 yılında Sun firmasını almasıyla birlikte MySQL VTYS'nin yapısı daha da fazla değiştirilmiştir.

MySQL.connector kütüphanesi, MySQL VTYS'nin Python ile kullanılabilmesi amacıyla Oracle firması tarafından geliştirilmiştir (URL-13, 2019). MySQL.connector kütüphanesi sayesinde, VTYS içerisinde yapılabilecek tüm işlemler, program içerisinde de yapılabilmektedir.

Örnek kod bloğu :

```
import mysql.connector
1. baglanti = mysql.connector.connect(user='yigit', password='alisan',
host='localhost', database='vt_adı')
2. imlec = baglanti.cursor()
3. veri_ekle = "INSERT INTO eleman2.new_Tablo (uni_ad, ders_adi, dersin_amaci,
ders_kredi) VALUES (%s, %s, %s, %d)"
4. eklenecek_veri = ("GaziOsmanpaşa Üniversitesi", "İşletim sistemleri", "Linux
işletim sistemini iyi derecede kullanabilmek", 4)
```

```
5. imlec.execute(veri_ekle, eklenecek_veri)
6. baglanti.commit()
```

Örnek kod bloğu satır açıklamaları:

1. satırda, kütüphanenin "connect" metodu kullanılarak kullanıcı adı, parolası, bulunduğu yer ve ismi yazılan veri tabanına bağlantı yapılmaktadır.
2. satırda, bağlantı kurulan veri tabanı içerisindeki tablolarda, hızlı ve rahat şekilde, aranılan verilere ulaşmak amaçlı kullanılacak olan "cursor" metodu "imlec" değişkenine aktarılmaktadır.
3. satırda, veritabanına ekleme yapılması için gerekli olan standart SQL dilinin "insert into" komutu kullanılarak sorgu cümlesi oluşturulmaktadır.
4. satırda, eklenilmesi istenilen verier "eklenecek_veri" isimli değişkene aktarılmaktadır.
5. satırda, "execute" metodu ile yazılan komutun ve eklenilmek istenilen verilerin birleştirilmesi sağlanmaktadır.
6. satırda, "execute" metodu kullanılarak son şeklini alan sorgu cümlesi "commit" metodu kullanılarak veritabanı tarafında çalıştırılmaktadır.

2.3.8. FuzzyWuzzy kütüphanesi

Bilişim sistemlerinde sözcük arama, tarama işlemleri esnasında aranılan sözcüğün hali hazırda var olan bir sözcük ya da cümle grubu içerisindeki sözcükler ile olan benzerliğinin belirlenmek istenilmesi durumunda mesafe, uzaklık algoritmaları kullanılabilir. Bu tip işlemleri gerçekleştirmeye yarayan algoritmalar, günümüzde kelime işlemci tipi programlarda sıklıkla karşılaşıldığı gibi arama motorlarında, arama yapılacak sözcüklerin yanlış yazımı sonucu arama motorları tarafından düzeltilmiş halinin önerilmesi işleminde, virüs tarama programlarında, intihal programlarında vb. sıklıkla kullanılmaktadır.

Levenshtein Mesafesi:

Levenshtein, iki veri dizisini harf harf sıralı şekilde karşılaştırma işlemi yaparak bir veri dizisinin karşılaştırma yapılan diğer veri dizisine çevrilmesi sırasında yapılacak işlemlerin maliyetini hesaplayan mesafe algoritmasıdır (Novikov, 1965). Levenshtein algoritması (Denklem 2.1), iki veri dizisini matris yapısı şekline getirerek, sıralı şekilde karakter karakter kontrol işlemine tabi tutar. Kontrol aşamasında, karakterin veri dizisindeki sırası dikkate alınarak karakter ekleme (Insertion), çıkarma (Deletion), yer değiştirme

(Substitution) işlemleri sonucunda, iki veri dizisini aynı olacak şekilde işlemlere tabi tutar ve oluşan işlem maliyetini hesaplayarak iki veri dizisi arasındaki yakınlık, benzerlik derecesini bulur. İki veri dizisinin aynı olması durumunda maliyet değeri 0 bulunur.

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \min(i,j) = 0, \\ \min \begin{cases} lev_{a,b}(i-1,j) \\ lev_{a,b}(i,j-1) \\ lev_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{diğer} \end{cases} \quad (2.1)$$

Algoritma çalışma yapısı, girilen ilk kelimenin ikinci kelimeye dönüştürülmesi için gerekli olan işlem adım sayısını vermektedir. Girilen ilk kelimenin, ikinci kelimeye dönüştürülmesi sırasında yapılabilecek işlemler Silme (Deletion), Yer Değiştirme (Substitution), Ekleme (Insertion) şeklindedir. Yapılan her bir işlem bir (1) maliyet değerine eşit olacak şekilde, algoritma sonucu toplam maliyet çıktısı verilmektedir (Novikov, 1965).

FuzzyWuzzy Kütüphanesi Kullanımı:

Fuzzywuzzy kütüphanesi, Levenshtein mesafe algoritmasını kullanarak iki veri dizisini karakter bazında karşılaştırma yapmakta ve çıktı olarak 100 üzerinden sayısal bir mesafe değeri vermektedir. İki karakter dizisinin aynı olması durumunda çıktı değeri 100'dür.

6 farklı metodu bulunan Fuzzywuzzy Kütüphanesi ile her uzunluktaki cümleler üzerinde işlem yapılabilir. Karşılaştırma sırasında cümle içerisinde bulunan veri dizileri haricindeki karakterlerin göz önünde bulundurulmak istenilmemesi durumunda "partial_ratio" metodu kullanılır, aksi durumda ise "ratio" metodu kullanılarak cümle dizisi içerisindeki tüm karakterler dikkate alınarak işlem yapılır (Örnek 2.1).

Cümle dizisi içerisinde bulunan veri dizilerinin, bulundukları yerler göz ardı edilmek istenilirse "token_sort_ratio" metodu kullanılır, aksi durumda ise "ratio" metodu kullanılır (Örnek 2.2).

Cümle dizisi içerisinde birden fazla kez geçen veri dizisi, göz ardı edilmek istenilirse "token_set_ratio" metodu kullanılır, aksi durumda ise "token_sort_ratio" metodu kullanılır (Örnek 2.3).

Örnek 2.1:

```
print(fuzz.ratio("bu bir denemedir", "bu bir denemedir!"))
# 97 #çıkıtı
print(fuzz.partial_ratio("bu bir denemedir", "bu bir denemedir!"))
# 100 #çıkıtı
```

Örnek 2.2:

```
print(fuzz.ratio("bu bir denemedir", "bir bu denemedir"))
# 91 #çıktı
print(fuzz.token_sort_ratio("bu bir denemedir", "bir bu denemedir"))
# 100 #çıktı
```

Örnek 2.3:

```
print(fuzz.token_sort_ratio("bu bir denemedir", "bu bu bir denemedir"))
# 84 #çıktı
print(fuzz.token_set_ratio("bu bir denemedir", "bu bu bir denemedir"))
# 100 #çıktı
```

2.3.9. Doğal dil işleme

Doğal Dil İşleme de (Natural Language Processing - NLP) asıl amaç, konuşma dilinin anlama, işleme, yorumlama ve üretme aşamalarını makinelere öğretmek sistemlerin ses ve metin ile iletişim kurmasını sağlamaktır.

Doğal dil işleme, yedi farklı aşamaya ayrılmıştır (Liddy, 2001).

1. Sözlüksel: Bu aşamada kelimenin anlamı analiz edilir, yorumlanır.
2. Anlamsal: Bu aşamada cümlelerin belirttiği anlamlar üzerinde çalışma yapılır.
3. Sesbilim : Bu aşamada kelimenin seslendirme yapısı analiz edilir.
4. Biçimbilim : Bu aşamada kelime ile birlikte eklentiler analiz edilir.
5. Pragmatik: Bu aşamada içinde bulunulan ortam göz önünde bulundurularak cümle yorumlanır.
6. Söylem: Bu aşamada birden fazla cümleden oluşan uzun söylemler analiz edilir.
7. Sözdizimsel: Bu aşamada cümleyi oluşturan kelimelerin cümledeki yerine göre analizi yapılır.

2.3.9.1. Zemberek - NLP kütüphanesi

1999 yılında “Tspell” ismi ile başlayan, 2004 yılında “Zemberek” ismini alan proje, Apache 2.0 lisansı altında, Türkçe dili üzerine yapılan doğal dil işleme çalışmaları için kullanılabilen, içerisinde 23 binden fazla sınıflandırılmış kelime kökü ve en az 7000 özel isim adı barındıran, Java programlama dili kullanılarak yazılmış, açık kaynak kodlu Türkçe doğal dil işleme kütüphanesidir. 2014 yılı itibariyle ismini “Zemberek-Nlp” olarak güncellemiştir (Akın ve Akın, 2007).

Açık kaynak kodlu olan Zemberek-Nlp kütüphanesi var olan programlar içerisinde kütüphane olarak eklenebilmektedir. Libre Office, Open Office programlarında bulunmakta ve Türkçe imla ve denetim işlemleri için kullanılmaktadır. Ayrıca açık kaynak kodlu işletim sistemlerinin resmi depolarında kendisine yer bulan Zemberek-Nlp kütüphanesi, popüler internet tarayıcıları için (Mozilla Firefox, Google Chrome gibi) Türkçe imla denetim eklentisi de sunmaktadır (URL-14, 2019).

Örnek:

Metin Normalizasyonu

girdi >>> Ey Trk Genli!

çıktı >>> Ey Türk Gençliği!

2.3.10. Jpype kütüphanesi

Belirli programlama dillerinde kullanılmak üzere yazılan kütüphane, modül, eklenti gibi programlar farklı programlama dillerinde yazılan programların içerisinde doğrudan kullanılamamaktadır.

Jpype, Python programlama dili kullanılarak yazılan programın içerisinde Java programlama diline ait olan kütüphanelere erişimi sağlayan köprü görevini yürüten açık kaynak kodlu bir Python programlama dili kütüphanesidir (URL-16, 2019).

Jpype kütüphanesinin kullanılabilmesi için, kullanılacak olan bilgisayarda Java programlama diline ait temel paketlerin eksiksiz şekilde kurulu olması gerekmektedir.

Örnek kod bloğu:

```
1. import jpype
2. jpype.startJVM("/usr/lib/jvm/java-11-openjdk-amd64/lib/server/libjvm.so",
Djava.class.path=/home/alisan/zemberek-tum-2.0.jar", "-ea")
3. Tr = jpype.JClass("net.zemberek.tr.yapi.TurkiyeTurkcesi")
4. tr = Tr()
5. Zemberek = jpype.JClass("net.zemberek.erisim.Zemberek")
6. zemberek = Zemberek(tr)
...
7. kok_aranan_ozellik = icerik_aranan_ozellik[3]
...
8. jpype.shutdownJVM()
```

Örnek kod bloğu satır açıklamaları:

1. satırda, Python programlama dili kullanılarak hazırlanan projeye, “import” komutu ile jpype kütüphanesi eklenmektedir.

2. satırda, jpype kütüphanesinin “startJVM” metodu kullanılarak Java programının çalışabilmesi için bilgisayara kurulu Java programının “libjvm.so” paketinin yolu ve proje içerisinde erişilmek istenilen, Java programlama dili ile oluşturulmuş olan kütüphanenin bilgisayarda bulunduğu dizin yazılmaktadır.
3. satırda, “Tr” isimli değişkene jpype kütüphanesinin “JClass” metodu kullanılarak, kütüphane içerisinde bulunan, kullanılmak istenilen sınıfın özellikleri yüklenmekte ve kullanıma hazır hale getirilmektedir.
4. satırda “tr” isimli değişkene, 3. satırda “Tr” değişkenine aktarılan “TurkiyeTurkcesi” sınıfı tanımlanmaktadır.
5. Satırda “Zemberek” isimli değişkene, kütüphaneye erişim için kullanılacak sınıfı tanımlanmaktadır. Bu sayede Zemberek ile bağlantı sağlanabilmektedir.
6. satırda, program içerisinde “TurkiyeTurkcesi” sınıfına erişim için kullanılacak “zemberek” isimli değişken tanımlanmaktadır.
7. satırda, işlem yapılan kelimeye ait kökün sıfat olup olmadığına bakılmaktadır. Bu satırda kullanılan 3 numarası, “Zemberek” kütüphanesinde sıfatı temsil etmektedir.
8. satırda, jpype kütüphanesinin çalıştırılması sonlandırılmaktadır.

2.4. Çok Kriterli Karar Verme Yöntemleri

Çok kriterli karar verme (ÇKKV), çözülmek istenilen problemde, kullanılacak kriterlerin önemleri amaca uygun şekilde ağırlıklandırılarak, çözüm olarak görülen alternatifler arasından seçim işleminin gerçekleştirilmesidir. Çok kriterli karar verme yöntemleri Hwang ve Yoon (1981) tarafından ÇKKV ve Çok Amaçlı Karar Verme (ÇAKV) olarak ikiye ayrılmış olup ÇKKV problemleri, alternatifleri önceden belirlenmiş, genellikle sınırlı ve kriterler açısından mümkün olduğunca az ödün verilerek sonuca ulaşılabilecek problemler olarak tanımlanmaktadır (Hwang ve Yoon, 1981). Bu çalışmada ÇKKV yöntemleri olarak TOPSIS, VIKOR ve ELECTRE yöntemleri kullanılmıştır.

2.4.1. TOPSIS yöntemi

ÇKKV yöntemlerinden birisi olan TOPSIS’te, alternatifler arasından seçilecek seçenek, pozitif yönden ideal çözüme en yakın olmalı ve aynı zamanda negatif yönden ideal çözüme en uzak olmalıdır (Opricovic ve Tzeng, 2002).

TOPSIS işlem basamakları aşağıdaki gibidir:

Denklem (2.2) kullanılarak r_{ij} normalize değerleri hesaplanır. Denklem (2.3) kullanılarak karar matrisindeki normalize değerlerin ağırlıklandırılması için v_{ij} değeri hesaplanır. Matrisin v_{ij} değerleri hesaplandıktan sonra, pozitif (A^*) (Denklem 2.4) ve negatif (A^-) (Denklem 2.5) ideal değerlerin hesaplanır. Denklemlerde kullanılan, i değişkeni kriter sıra sayısını, w_i i . sıradaki değişkene ait ağırlık değerini, j değişkeni alternatif sıra sayısını, J değişkeni toplam alternatif sayısını, n değişkeni toplam kriter sayısını, I' değişkeni kâr ölçütleri kümesini, I'' değişkeni maliyet ölçütleri kümesini ifade etmektedir.

$$r_{ij} = f_{ij} / \sqrt{\sum_{j=1}^J f_{ij}^2} \quad (2.2)$$

$$v_{ij} = w_i r_{ij} \quad (2.3)$$

$$A^* = \{v_1^*, \dots, v_n^*\} = \left\{ \left(\max_j v_{ij} \mid i \in I' \right), \left(\min_j v_{ij} \mid i \in I'' \right) \right\} \quad (2.4)$$

$$A^- = \{v_1^-, \dots, v_n^-\} = \left\{ \left(\min_j v_{ij} \mid i \in I' \right), \left(\max_j v_{ij} \mid i \in I'' \right) \right\} \quad (2.5)$$

Herbir çözümün ideal çözümlere olan uzaklığını hesaplamak için Öklid kullanılır. Pozitif ideal çözüme olan uzaklık hesabı Denklem (2.6) ile negatif ideal çözüme olan uzaklık hesabı ise Denklem (2.7) ile yapılır.

$$D_j^* = \sqrt{\sum_{i=1}^n (v_{ij} - v_j^*)^2} \quad (2.6)$$

$$D_j^- = \sqrt{\sum_{i=1}^n (v_{ij} - v_j^-)^2} \quad (2.7)$$

İdeal çözüme olan benzerliklerin hesaplanması kısmında ise bir önceki aşamada elde edilen değerler kullanılarak Denklem (2.8) ile hesaplanır.

$$C_j^* = D_j^- / (D_j^* + D_j^-) \quad (2.8)$$

Son adımda ise elde edilen değerler, benzerlik değerleri temel alınarak en yüksekten en düşüğe doğru sıralanır ve ilk sıradaki alternatif önerilir (Opricovic ve Tzeng, 2002).

2.4.2. VIKOR yöntemi

VIKOR yönteminin en etkin özelliklerinden birisi, karar verecek olan birey, sistem tasarım aşamasının başlangıcında tercihlerini net şekilde belirtmemesi durumunda bile çok kriterli karar vermede etkin olabilmesidir. Uzlaşık çözüm temelli bu yöntem için asıl amaç, çoğunluk için en üst seviyede grup yararını, karşıt durumdakiler için ise en alt seviyede zararı sağlamasıdır (Opricovic ve Tzeng, 2002).

VIKOR işlem basamakları aşağıdaki gibidir:

VIKOR yönteminde ilk adım olarak herbir kriter için Denklem (2.9) kullanılarak en iyi kriter (f_i^*), Denklem (2.10) kullanılarak en kötü kriter (f_i^-) belirlenir.

$$f_i^* = \max_j f_{ij} \quad (2.9)$$

$$f_i^- = \min_j f_{ij} \quad (2.10)$$

(f_i^*) ve (f_i^-) değerlerinin hesaplanması sonrasında Denklem (2.11) kullanılarak tüm alternatifler için ortalama grup değerleri (S_j) ve Denklem (2.12) kullanılarak en kötü grup değerleri (R_j) hesaplanır.

$$S_j = \sum_{i=1}^n w_i (f_i^* - f_{ij}) / (f_i^* - f_i^-) \quad (2.11)$$

$$R_j = \max_i [w_i (f_i^* - f_{ij}) / (f_i^* - f_i^-)] \quad (2.12)$$

Denklem (2.13) ile maksimum grup yararı (Q_j) hesaplanır. Denklemdeki v değeri maksimum grup yararını hesaplayabilmek için kullanılan ağırlık değerini ifade etmektedir. Ağırlık değeri olan v için genel olarak 0,5 kullanılır.

$$Q_j = \frac{v(S_j - S^*)}{(S^- - S^*)} + (1 - v)(R_j - R^*) / (R^- - R^*) \quad (2.13)$$

$$S^* = \min_j S_j, \quad S^- = \max_j S_j, \quad R^* = \min_j R_j, \quad R^- = \max_j R_j$$

Yapılan işlemlerin sonucunda tüm alternatifler için hesaplanmış olan S_j, R_j ve Q_j değerleri küçükten büyüğe olacak şekilde sıralanır. En küçük Q_j değerine sahip olan alternatif en iyi seçenek olarak belirlenir. Sıralama sonrası en iyi seçenek olarak belirlenen alternatifin, kesinleşmesi için “Kabul Edilebilir Avantaj” (C1) ve “Kabul Edilebilir İstikrar”

(C2) koşullarını sağlaması gerekmektedir. Alternatifin, kabul edilebilir avantaj koşulunu (C1) sağlayabilmesi için Denklem (2.14) kullanılmaktadır. Bu denklemde a' en küçük Q_j değerine sahip yani en iyi alternatifi, a'' değeri ise en iyi ikinci alternatifi göstermektedir. Burada $DQ = 1/(J-1)$ olmak üzere, denklemdeki DQ değerinin hesaplanmasında kullanılan J değeri ise alternatiflerin sayısını göstermektedir.

$$Q(a'') - Q(a') \geq DQ \quad (2.14)$$

Alternatifin, kabul edilebilir istikrar koşulunu (C2) sağlayabilmesi için, en küçük Q değerine sahip olan alternatifin S ve R sonuçlarına göre sıralamada da en az birisinde en iyi sonucu alması gerekmektedir.

$C1$ ve $C2$ koşullarından:

$C1$ koşulunun sağlanamaması durumunda, a' ve a'' alternatifleri dikkate alınarak, $C2$ koşulunun sağlanamaması durumunda, $a^{(M)}$ alternatifi dikkate alınarak Denklem (2.15) kullanılarak işlem yapılır. (Opricovic ve Tzeng, 2002)

$$Q(a^{(M)}) - Q(a') < DQ \quad (2.15)$$

$$a^{(M)} = a^{(i)}, a^{(j)}, \dots, a^{(M)}$$

2.4.3. ELECTRE yöntemi

ELECTRE (Elimination and Choice Translating Reality) yöntemi, Benayoun tarafından 1966 yılında literatüre kazandırılmıştır. ELECTRE, ÇKKV yöntemlerinden birisi olup, her kriter için alternatifleri kendi arasında ikili şekilde üstünlük bakımından kıyaslar.

ELECTRE işlem basamakları aşağıdaki gibidir:

İlk adım olarak “Karar Matrisi” oluşturulur. Karar matrisinde, alternatifler matrisin satırını (m) oluşturmakta, kriterler ise sütunlarını (n) oluşturmaktadır. Denklem (2.16) kullanılarak Normalize Karar Matrisi (X), oluşturulur.

$$x_{ij} = \frac{a_{ij}}{\sqrt{\sum_{k=1}^m a_{kj}^2}} \quad (2.16)$$

$$X = \begin{bmatrix} x_{11} & \dots & x_{1n} \\ \vdots & \dots & \vdots \\ x_{m1} & \dots & x_{mn} \end{bmatrix}$$

Denklem (2.17)'deki durumu sağlayacak şekilde, Normalize Karar Matrisini oluşturan değerlerin ağırlıklandırılması yapılır. Güncellenen Y Ağırlıklandırılmış Normalize Karar Matrisinde, w ilgili kritere ait ağırlık değerini temsil etmektedir.

$$\sum_{i=1}^n w_i = 1 \quad (2.17)$$

$$Y = \begin{bmatrix} y_{11} & \dots & y_{1n} \\ \vdots & \dots & \vdots \\ y_{m1} & \dots & y_{mn} \end{bmatrix} = \begin{bmatrix} w_1 x_{11} & \dots & w_n x_{1n} \\ \vdots & \dots & \vdots \\ w_1 x_{m1} & \dots & w_n x_{mn} \end{bmatrix}$$

Denklem (2.18) yardımıyla, Ağırlıklandırılmış Normalize Karar Matrisi (Y) kullanılarak uyum seti oluşturulur. Denklem (2.19) da görüleceği üzere uyum setinin tamamlayıcı alt küme elemanları, yani uyum seti haricinde kalan değerler ise uyumsuzluk setini oluşturmaktadır.

$$C_{kl} = \{j | y_{kj} \geq y_{lj}\} \quad (2.18)$$

$$D_{kl} = \{j | y_{kj} < y_{lj}\} \quad (2.19)$$

Uyum ve uyumsuzluk setlerine ait elemanların bulunmasıyla birlikte, denklem (2.20) kullanılarak “Uyum Matrisi”, Denklem (2.21) kullanılarak “Uyumsuzluk Matrisi” oluşturulur. C isimli Uyum Matrisi ve D isimli Uyumsuzluk Matrisi, alternatif nokta sayısı x alternatif nokta sayısı (mxm) boyutundadır.

$$C_{kl} = \sum_{j \in C_{kl}} w_j \quad (2.20)$$

$$C = \begin{bmatrix} - & c_{12} & \dots & c_{1m} \\ c_{21} & - & \dots & \vdots \\ \vdots & \dots & - & \vdots \\ c_{m1} & c_{m2} & \dots & - \end{bmatrix}$$

$$d_{kl} = \frac{\max_{j \in D_{kl}} |y_{kj} - y_{lj}|}{\max_j |y_{kj} - y_{lj}|} \quad (2.21)$$

$$D = \begin{bmatrix} - & d_{12} & \dots & d_{1m} \\ d_{21} & - & \dots & \vdots \\ \vdots & \dots & - & \vdots \\ d_{m1} & d_{m2} & \dots & - \end{bmatrix}$$

“Uyum Baskınlık Matrisi”, Denklem (2.22) yardımıyla hesaplanan uyum eşik değerinin ($\underline{c}$), Uyum Matrisi elemanları ile Denklem (2.23) kullanılarak karşılaştırılması

sonucu 1 ve 0 lardan oluşur. Uyum üstünlük matrisi, alternatif nokta sayısı x alternatif nokta sayısı (mxm) boyutundadır.

$$\underline{c} = \frac{1}{m(m-1)} \sum_{k=1}^m \sum_{l=1, l \neq k}^m c_{kl} \quad (2.22)$$

$$f_{kl} = \begin{cases} 1, & \text{eğer } c_{kl} \geq \underline{c} \\ 0, & \text{eğer } c_{kl} < \underline{c} \end{cases} \quad (2.23)$$

“Uyumsuzluk Baskınlık Matrisi”, Denklem (2.24) yardımıyla hesaplanan uyumsuzluk eşik değerinin ($\underline{d}$), Uyumsuzluk Matrisi elemanları ile Denklem (2.25) kullanılarak karşılaştırılması sonucu 1 ve 0 lardan oluşur. Uyumsuzluk üstünlük matrisi, alternatif nokta sayısı x alternatif nokta sayısı (mxm) boyutundadır.

$$\underline{d} = \frac{1}{m(m-1)} \sum_{k=1}^m \sum_{l=1, l \neq k}^m d_{kl} \quad (2.24)$$

$$g_{kl} = \begin{cases} 1, & \text{eğer } d_{kl} \geq \underline{d} \\ 0, & \text{eğer } d_{kl} < \underline{d} \end{cases} \quad (2.25)$$

Elde edilen Uyum Baskınlık Matrisi (f) ve Uyumsuzluk Baskınlık Matrisi (g) elemanlarını karşılıklı çarpılmasıyla (Denklem 2.26) “Toplam Baskınlık Matrisi” (e) elde edilir.

$$e_{kl} = g_{kl} \times f_{kl} \quad (2.26)$$

Sonuç olarak, elde edilen Toplam Baskınlık Matrisinde e_{kl} nin sonucu 1 ise alternatifler kümesindeki A_k alternatifi, A_l alternatifine tercih edilir. Sonucun 0 olması durumunda ise A_l alternatifi, A_k alternatifine tercih edilir (Triantaphyllou, 2000).

2.5. Verilerin Toplanması ve Düzenlenmesi

Yapılan araştırmada kullanılacak verilerin, azami ölçüde güvenilir ve ülkemiz özelliklerini yansıtmasına gayret edilmiştir. Bu nedenlerden dolayı, ülkemiz sınırları içerisinde bilinen ve sürekli kendisini güncelleyen çevrimiçi iş ilanı sitelerinden 3 tanesinde bulunan güncel iş ilanları kullanılmıştır. Uluslararası ilan toplama aşamasında, ağırlıklı olarak “Teknoloji ve Bilişim” alanına ait ilanlar barındıran çevrimiçi iş ilanı sitesinden yararlanılmıştır. Elde edilen ham verilerin işlenebilmesi için belirli özelliklerde olması gerekmektedir. Bu nedenle ilk olarak ham verilerin düzenlenme işlemi gerçekleştirilmiştir.

Yapılan düzenleme işleminde:

1. Cümle içerisindeki harf harici veriler (Ders kodu, sayılar, noktalama işaretleri) temizlenmiştir.
2. İngilizce olarak toplanılan ders içerikleri Google API kullanılarak otomatik olarak Türkçe'ye çevrilmiştir.
3. Eldeki kelimelerin içerisinde harf uzunluğu "2" den küçük olan kelimeler ayıklanmıştır.
4. Harf uzunluğu 2'den fazla olan kelimeler Zemberek programı yardımıyla normalize edilme işlemine tabi tutulmuş ve böylece yazım yanlışı olan kelimeler düzeltilmiştir.
5. Normalize edilen kelimeler köklerine ayrılmıştır.
6. Elde edilen kelime kökleri içerisinde, manuel olarak oluşturulan "Yararsız Kelime Kökleri Sözlüğü"nde bulunan kökler ayıklanmıştır.
7. Elde edilen kelime köklerinin "Bilgisayar Terimleri Sözlüğü"nde bulunan kelimeler ile karşılaştırılması ve bulunan kelimelerin "Yararlı Kelime" olarak veri tabanına eklenmesi işlemi yapılmıştır.
8. Zemberek-Nlp programı yardımıyla "İsim" olan kelime kökleri haricindeki kelimeler ayıklanmıştır. İsim olan kelimeler "Yararlı Kelime" olarak veri tabanına eklenmiştir.
9. Alternatiflere ve kriterlere ait olan "Yararlı Kelime Kökleri" olarak sınıflandırılan kelimeler FuzzyWuzzy kütüphanesi kullanılarak analiz edilmiştir. Yapılan analiz sonrası kelimelerin değerleri, benzerlik oranı en düşük 0, en yüksek 100 olacak şekilde hesaplanmıştır.
10. Her bir alternatifin puan değerini bulmak için, alternatife ait derslerin sonuç değerlerinin aritmetik ortalaması alınmıştır.

2.5.1. Çevrimiçi iş ilanlarının toplanması

Çevrimiçi iş ilanı sitelerinde, arama yapılırken "Bilgisayar Mühendisi" anahtar kelime olarak kullanılmıştır. Yapılan aramalar sonucunda, elde edilen ilanlar arasında işin bulunduğu şehir, firma, tecrübe yılı vb. değişkenler göz ardı edilmiştir. Arama sonucu elde edilen, fakat temel görevi bilgisayar ile ilgili olmayan (Sekreter, büro personeli, kasiyer, gişe görevlisi vb.) pozisyonlara ait ilanlar elenmiştir. Yapılan ön eleme işlemi sonrasında elde edilen ilanlar işlenmeye hazır hale getirilmiştir.

2.5.2. Ders öğrenme çıktılarının toplanması

Türkiye’de eğitim vermekte olan devlet üniversitelerinden 12 tanesine ait olan “Bilgisayar Mühendisliği Bölümü Dersleri Öğrenme Çıktıları”, üniversitelerin kurumsal internet sayfaları üzerinden web kazıma yöntemi ile elde edilmiştir. Üniversitelerin belirlenmesi aşamasında, özellikle kurumsal internet sayfalarında bulunan ders içerik bilgilerinin mümkün olduğunca düzenli ve eksiksiz olması etkin rol oynamıştır.

2.5.3. Yararsız kelime kökleri sözlüğü

Levenshtein algoritması ile cümleye ait olan kelime köklerinin karşılaştırılması nedeniyle, cümle içerisindeki kelime sayıları önem arz etmektedir.

Tablo 2.1. Yararsız kelime kökleri sözlüğü listesi- örnek kesit

Kelime	Harf	Kelime	Harf
've'	'v'	'mühendis'	'm'
'veya'	'v'	'okul'	'o'
'asker'	'a'	'mezun'	'm'
'muaf'	'm'	'iyi'	'i'
'bölüm'	'b'	'tecrübe'	't'

Daha sağlıklı sonuçların elde edilebilmesi amacıyla, alternatifler ve kriterlere ait veriler içerisinde bulunan ve konu ile ilgili ayırt edici özelliğe sahip olmayan kelimeler belirlenerek toplanmış ve “Yararsız Kelime Kökleri Sözlüğü” oluşturulmuştur (Tablo 2.1). Sözlük taranarak, cümle içerisinde bulunması halinde, manuel olarak verilerin kontrol edilmesi sırasında kolaylık sağlaması amacıyla, veri tabanındaki ilgili cümlelerin satırında bulunan “ayıklanan” sütununa eklenmektedir.

2.5.4. Bilgisayar terimleri sözlüğü

Cümle içerisinde bulunan, bilgisayar bilimine özgü bir terim olan kelime ya da kelime gruplarından Türkçe olmayan kelimelerin, köklere ayrılması ve normalizasyon işlemlerine girmemesi için, Türkçe olan kelime ve kelime gruplarının ayrıştırılmaması için çevrimiçi olarak kullanılabilen “Bilgisayar Terimleri Sözlüğü”nde bulunan 15.000 den fazla

terim (URL-17, 2019), web kazıma yöntemi kullanılarak yerel veritabanına kaydedilmiştir. Ayrıca sözlükte mevcutta bulunan terimlerin tamamı Google api yardımıyla Türkçe'ye çevrilmiştir. Böylece, ilan içerisinde Türkçe ya da İngilizce yazılan terimlerin bulunması kolaylaştırılmıştır. Kelime analizi esnasında ilk olarak Sözlük taranmakta ve teknik terim bulunması halinde (manuel olarak verilerin kontrol edilmesi sırasında kolaylık sağlaması amacıyla yapılmaktadır) veri tabanındaki ilgili cümleye ait satırda bulunan “terim” sütununa eklenmektedir.

Tablo 2.2. Bilgisayar terimleri sözlüğü kesiti

Harf	Terim (İngilizce)	Terim (Türkçe)
'A'	'abbyy'	'abbyy'
'A'	'access'	'erişim'
'B'	'backup'	'yedekleme'
'D'	'data'	'veri'
'D'	'directory'	'dizin'
'H'	'html'	'html'
'H'	'hardware'	'donanım'
'K'	'kernel'	'çekirdek'
'L'	'linux'	'linux'
'P'	'pdf'	'pdf'
'R'	'router'	'yönlendirici'
'S'	'software'	'yazılım'
'S'	'server'	'sunucu'
'V'	'veeam'	'veeam'

Ayrıca, sözlükteki terim sayısının 15000'in üzerinde olması ve neredeyse her kelime için sözlüğün aranması nedeniyle, program yavaş sonuç üreteceği için çözüm olarak, sözlükte arama yapılacak kelimenin ilk harfi alınarak “Harf” sütunu ile karşılaştırılmakta ve sadece ilk harfleri aynı olan kelimeler içerisinde arama işlemi yapılmaktadır. İlgili sözlüğe ait örnek girdiler Tablo 2.2'de gösterilmektedir.

3. BULGULAR VE TARTIŞMA

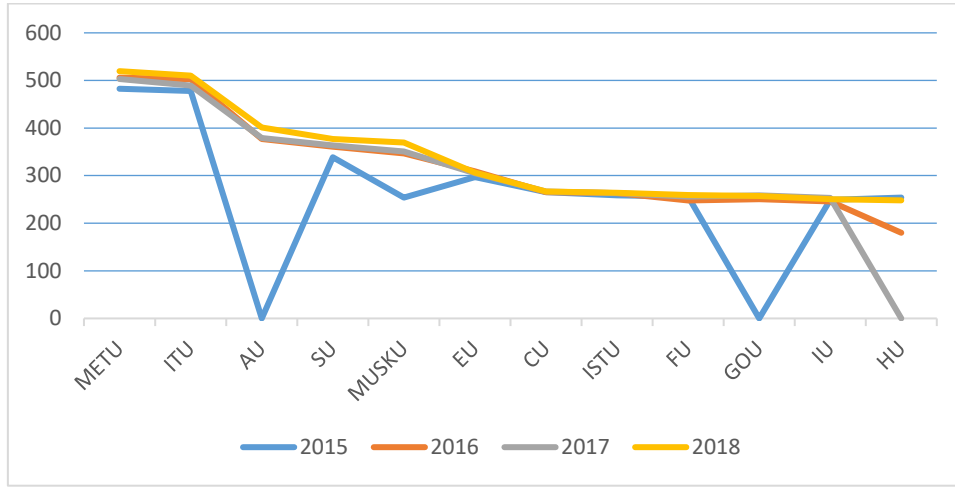
Üniversitelerin, kurumsal çevrimiçi internet sayfalarında bulunan “Bilgisayar Mühendisliği” bölümüne ait derslerin içerikleri, web kazıma yöntemi ile toplanmıştır. Üniversitelere göre elde edilen ders içerik sayıları toplamı Tablo 3.1’de gösterilmektedir. Bu üniversitelerin “Bilgisayar Mühendisliği” bölümüne ait 2015-2018 yılları arasındaki taban puanları ve ortalama puanları Tablo 3.2’de, sayısal bilgilere ait grafik Şekil 3.1’de gösterilmektedir.

Tablo 3.1. Üniversitelerin kurumsal sitelerinden toplanılan ders sayıları

Sıra Nu.	Kısa isim	Üniversite İsmi	Ders Sayısı	Sıra Nu.	Kısa isim	Üniversite İsmi	Ders Sayısı
1	İTÜ	İstanbul Teknik Üni.	39	7	İSTÜ	İskenderun Teknik Üni.	48
2	FÜ	Fırat Üni.	50	8	EÜ	Erbakan Üni.	43
3	HÜ	Harran Üni.	37	9	İÜ	İnönü Üni.	13
4	AÜ	Akdeniz Üni.	30	10	MSKÜ	Muğla Sıtkı Koçman Üni.	63
5	CÜ	Cumhuriyet Üni.	44	11	SÜ	Sakarya Üni.	33
6	GOÜ	GaziOsman Paşa Üni.	27	12	ODTÜ	Orta Doğu Teknik Üni.	40

Tablo 3.2. Üniversitelerin 2015-2018 yılları arası taban puanları

Yıl	ODTÜ	İTÜ	SÜ	HÜ	MSKÜ	CÜ	İSTÜ	FÜ	GOÜ	İÜ	EÜ	AÜ
2018	519,51	510,43	377,00	248,09	369,49	266,70	264,16	259,06	257,00	250,30	306,46	401,30
2017	503,03	489,54	363,31	0,00	350,95	266,44	262,99	256,04	258,31	253,09	304,94	378,90
2016	505,77	502,70	360,75	180,00	346,87	265,29	263,24	247,91	250,66	246,05	309,11	376,68
2015	482,56	477,85	338,40	253,77	324,71	265,44	258,79	255,99	0,00	249,13	297,17	0,00
Ort.	502,717	495,13	359,86	170,46	348,005	265,96	262,29	254,7	255,32	249,64	304,42	385,62



Şekil 3.1. Üniversitelerin 2015-2018 yılları arası taban puanları grafiği

Türkiye’de faaliyette bulunan 3 farklı çevrimiçi iş ilan sitesinden 1816 adet olan iş ilanı verisi toplandı. Toplanan ilanların pozisyonlara göre sınıflandırılması sonucu 120 farklı pozisyon elde edilmiştir. En fazla ilan sayısına sahip ilk 50 pozisyon ve ilan sayılarına ait bilgiler Tablo 3.3’de gösterilmektedir. En fazla ilan sayısı “Diğer” başlıklı kategoridedir. Bu kategori, “Bilgisayar Mühendisliği” anahtar kelimeleri kullanılarak yapılan arama işlemi sonucunda ulaşılmış olup, pozisyon bilgisi boş bırakılan ilanlardan oluşmaktadır.

Tablo 3.3. Ulusal ilanlara ait pozisyon listesi

Sıra	Pozisyon Adı	İlan Sayısı	Sıra	Pozisyon Adı	İlan Sayısı
1	Diğer	1007	26	Web tasarım	3
2	Bilişim/bilgisayar	483	27	Yazılım eğitmeni	3
3	Yazılım uzmanı	35	28	Yönetici asistanı	3
4	Finansal danışman	17	29	Android developer	2
5	Bilgi işlem uzmanı	16	30	Asistan	2
6	Yazılım geliştirme uzmanı	16	31	Bilgi güvenliği uzmanı	2
7	Teknoloji danışmanı	13	32	Bilgi işlem elemanı	2
8	Bilgi işlem sorumlusu	11	33	Bilgi işlem personeli	2
9	Bilgisayar öğretmeni	11	34	Bilgi işlem uzman yardımcısı	2
10	Teknik servis uzmanı	9	35	Bilgi işlem yöneticisi	2
11	Sistem destek uzmanı	7	36	Bilgisayar mühendisi	2
12	Yazılım destek uzmanı	7	37	Bilişim teknolojileri öğretmeni	2
13	Yazılım mühendisi	7	38	Bilişim teknolojileri sorumlusu	2

Tablo 3.3. Ulusal ilanlara ait pozisyon listesi’nin devamı

14	Web yazılım uzmanı	5	39	E-Ticaret satış temsilcisi	2
15	Çağrı merkezi personeli	4	40	Eğitim danışmanı	2
16	Genel müdür	4	41	Eğitmen	2
17	Java yazılım uzmanı	4	42	Erp uzmanı java yazılım uzmanı	2
18	Mobil yazılım uzmanı	4	43	IT mühendisi	2
19	Net yazılım uzmanı	4	44	IT uzmanı	2
20	Arge mühendisi	3	45	Kalite kontrol elemanı	2
21	İş güvenliği uzmanı	3	46	Logo sistem uzmanı	2
22	Logo destek uzmanı	3	47	Network uzmanı	2
23	Saha mühendisi	3	48	Sistem mühendisi	2
24	Seo uzmanı	3	49	Sistem uzmanı	2
25	Sistem ve network uzmanı	3	50	Sistem yöneticisi	2

Bilişim Teknolojileri alanı ağırlıklı olmak üzere uluslararası iş ilanları bulunan bir siteden toplanılan iş ilanlarının ülke bilgisi Şekil 3.2’de, toplam ilan sayıları ise Tablo 3.4’de gösterilmektedir. Uluslararası alana ait iş ilanlarında aranılan 301 pozisyona ait veri toplandı. Bu pozisyonlar içerisinde en fazla ilan sayısına sahip ilk 50 pozisyon Tablo 3.5’ de gösterilmektedir. Tablo 3.5’de görüleceği üzere en fazla ilan sayısı “Software Engineer” pozisyonuna aittir.



Şekil 3.2. Uluslararası ilanların ülke dağılımı

Tablo 3.4. Ülkelerin yayınladıkları ilan sayısı

Sıra Nu.	Ülke	İlan Sayısı	Sıra Nu.	Ülke	İlan Sayısı	Sıra Nu.	Ülke	İlan Sayısı
1	U.S.A.	818	13	Switzerland	15	25	Philippines	2
2	India	77	14	Singapore	14	26	Belgium	1
3	Taiwan	59	15	France	13	27	Chile	1
4	Australia	36	16	S. Korea	11	28	Hong Kong	1
5	U.K.	36	17	Italy	7	29	Hungary	1
6	Canada	34	18	Brazil	6	30	Romania	1
7	Japan	34	19	Spain	5	31	South Africa	1
8	Germany	33	20	Sweden	4	32	U.A.E.	1
9	Israel	28	21	Denmark	3			
10	China	27	22	Netherlands	3			
11	Ireland	23	23	Finland	2			
12	Poland	18	24	Indonesia	2			

Ulusal ve uluslararası ilanlar üzerine yapılan çalışmada, üniversitelere ait ders öğrenme çıktılarından elde edilen veriler ile iş ilanlarında, aranılan elemenda olması talep edilen niteliklere ait veriler karşılaştırılarak TOPSIS, VIKOR ve ELECTRE algoritmalarında kullanılmak üzere sayısal çıktılar elde edilmiştir. Elde edilen çıktılar kullanılarak üniversitelerin, iş ilanları aracılığıyla aranılan elemenda olması talep edilen nitelikleri karşılama sıralaması ile ÖSYM ve URAP değerlerine ait karşılaştırma sonuçları; ulusal ilanlar için Tablo 3.6’da, uluslararası ilanlar için Tablo 3.8’de sunulmuştur. Ayrıca uygulanan algoritmalara ait ayrıntılı sayısal sonuçlar; ulusal ilanlar için Tablo 3.7’de, uluslararası ilanlar için Tablo 3.9’da sunulmuştur.

Tablo 3.5. Uluslararası ilanlara ait en fazla ilan sayısına sahip ilk 50 pozisyon listesi

Sıra Nu.	Pozisyon Adı	İlan Sayısı	Sıra Nu.	Pozisyon Adı	İlan Sayısı
1	Software engineer	249	26	Ios software engineer	8
2	Sales engineer	62	27	Staff software engineer	8
3	Technical program manager	53	28	Front end software engineer	8
4	Cloud technical sol. eng.	41	29	Cloud infrastructure engineer	7
5	Technical solutions engineer	36	30	Big data and analytics cloud consultant	7
6	Product manager	35	31	Partner developer advocate	6
7	Technical sol. consultant	26	32	Software developer	6
8	Customer engineer	23	33	Interaction designer	6
9	Developer advocate	22	34	Quantitative user exper. researcher	6

Tablo 3.5. Uluslararası ilanlara ait en fazla ilan sayısına sahip ilk 50 pozisyon listesi'nin devamı

10	Software eng. manager	19	35	Strategic negotiator	6
11	Partner engineer	18	36	Engineering resident	6
12	Security engineer	16	37	Principal engineer	6
13	Partner technology man.	15	38	Systems engineer	6
14	Test engineer	15	39	Hybrid cloud specialist	6
15	Data center technician	15	40	Infrastructure cloud consultant	6
16	Developer programs eng.	14	41	Information security engineer	6
17	Tech. curriculum dev.	14	42	Lead technical program manager	5
18	User experience research.	12	43	Manufacturing test develop. eng.	5
19	Application engineer	12	44	Application engineering manager	5
20	Business systems analyst	10	45	Director of cloud content and doc.	5
21	Technical writer	10	46	Manager	5
22	Technical specialist	10	47	Ux engineer	5
23	Corporate customer eng.	9	48	Senior developer advocate	5
24	Web solutions engineer	9	49	Senior developer programs engineer	5
25	Sap solutions engineer	9	50	Quantitative user exp. research lead	5

Tablo 3.6. Ulusal ilanlara ait üniversitelerin iş ilanlarına uygunluk başarı sırası karşılaştırması

Üniversiteler	Ösym ¹	Urap ²	TOPSIS	VIKOR	ELECTRE
ODTÜ	1	1	1	1	1
İTÜ	2	2	2	2	2
HÜ	12	11	3	3	3
SÜ	4	5	4	4	3
MSKÜ	5	10	5	5	5
CÜ	8	8	6	6	6
İSTÜ	9	12	7	7	7
FÜ	10	4	8	8	7
GOÜ	11	7	9	9	9
İÜ	7	6	10	10	10
EÜ	6	9	11	11	10
AÜ	3	3	12	12	10

¹ Ösym tarafından yayınlanan 2018 yılı Bilgisayar Mühendisliği Bölümü En Yüksek Yerleştirme Puanı sıralamasına göre oluşturulmuştur.

² URAP sıralamasında ilgili üniversitelerin asıl sıralamaları temel alınarak kendi aralarında sıralanması sonucu oluşturulmuştur.

Tablo 3.7. Ulusal ilanlar için algoritma çıktılarına ait ayrıntılı sonuçlar

Üniversiteler	TOPSIS		ELECTRE		VIKOR			
	Sıra	Değer	Sıra	Değer	Sıra	Q Değeri	S Değeri	R Değeri
ODTÜ	1	1	1	11	1	0	0	0
İTÜ	2	0.59231371	2	10	2	0.49414325	0.40375503	0.00479586
HÜ	3	0.43737978	3	8	3	0.66152182	0.55943538	0.00626235
SÜ	4	0.42262868	3	8	4	0.66486377	0.57679495	0.00617262
MSKÜ	5	0.33155393	5	6	5	0.74032476	0.66722195	0.00666474
CÜ	6	0.32621594	6	5	6	0.75348064	0.6724031	0.00683893
İSTÜ	7	0.27288462	7	4	7	0.81225102	0.72560939	0.00736601
FÜ	8	0.21598726	7	4	8	0.86616211	0.78771213	0.00773848
GOÜ	9	0.11074759	9	2	9	0.95423338	0.89904337	0.00826446
İÜ	10	0.07871338	10	0	10	0.97496558	0.94007767	0.00826446
EÜ	11	0.06314678	10	0	11	0.97830986	0.94669684	0.00826446
AÜ	12	0.02642317	10	0	12	1	0.98962715	0.00826446

*Mevcut tabloda bulunan değerlerin, birbirlerine yakın olması nedeniyle yuvarlama yapılmamıştır.

Tablo 3.8. Uluslararası ilanlara ait üniversitelerin iş ilanlarına uygunluk başarı sırası karşılaştırması

Üniversiteler	Ösym ³	Urap ⁴	TOPSIS	VIKOR	ELECTRE
ODTÜ	1	1	1	1	1
İTÜ	2	2	2	2	2
AÜ	3	3	10	10	7
SÜ	4	5	7	9	7
MSKÜ	5	10	4	4	4
EÜ	6	9	12	12	8
İÜ	7	6	6	7	5
CÜ	8	8	9	8	7
İSTÜ	9	12	5	5	4
FÜ	10	4	8	6	6
GOÜ	11	7	11	11	8
HÜ	12	11	3	3	3

³ Ösym tarafından yayınlanan 2018 yılı Bilgisayar Mühendisliği Bölümü En Yüksek Yerleştirme Puanı sıralamasına göre oluşturulmuştur.

⁴ URAP sıralamasında ilgili üniversitelerin asıl sıralamaları temel alınarak kendi aralarında sıralanması sonucu oluşturulmuştur.

Tablo 3.9. Uluslararası ilanlar için algoritma çıktılarına ait ayrıntılı sonuçlar

Üniversiteler	TOPSIS		ELECTRE		VIKOR			
	Sıra	Değer	Sıra	Değer	Sıra	Q Değeri	S Değeri	R Değeri
ODTÜ	1	1,000000000	1	11	1	0,000000000	0,000000000	0,000000000
İTÜ	2	0,666147633	2	10	2	0,398131103	0,332911707	0,001487260
HÜ	3	0,400290829	3	9	3	0,638168037	0,598652586	0,002174347
MSKÜ	4	0,353140958	4	7	4	0,691892168	0,646875498	0,002364404
İSTÜ	5	0,331121284	4	7	5	0,723538789	0,667431023	0,002501663
İÜ	6	0,188687248	5	4	7	0,868110960	0,815499404	0,002954121
FÜ	8	0,170884854	6	3	6	0,862427173	0,831433773	0,002866208
CÜ	9	0,166661037	7	2	8	0,878356913	0,835676928	0,002954973
SÜ	7	0,178252232	7	2	9	0,916319407	0,830610911	0,003215434
AÜ	10	0,123714382	7	2	10	0,931306912	0,880199551	0,003151978
GOÜ	11	0,071940172	8	0	11	0,969978030	0,937666785	0,003215434
EÜ	12	0,011690392	8	0	12	1,000000000	0,997564486	0,003215434

*Mevcut tabloda bulunan değerlerde, birbirlerine yakın olması nedeniyle yuvarlama yapılmamıştır.

Ulusal ilanların tamamında kullanılan kelimelerin frekans değerleri hesaplanarak, 100 defadan fazla kullanılan ve bağlaç olarak kullanılmayan kelimelere ait sayısal bilgiler Tablo 3.10’da gösterilmektedir. Ulusal ilanlarda, toplamda kullanılan farklı kelime sayısı 1552 olarak bulunmuştur.

Tablo 3.10. Ulusal ilanlara ait ilanlarda kullanılan frekans değeri 100 den büyük olan kelimeler

Kelime	Fr. De.	Kelime	Fr. De.	Kelime	Fr. De.	Kelime	Fr. De.
Konusunda	859	Net	279	Kullanabilen	179	Çözüm	138
Sahip	831	Çalışma	278	İşletme	177	Olmayan	135
Tecrübe	784	Yatkın	268	Ticaret	177	Düşünme	133
Bilgi	763	Analitik	255	Ekip	174	Yapılması	129
Yazılım	716	Yeni	250	Alanında	173	Donanım	129
Mühendis	668	Odaklı	247	Ürün	170	Eğitim	128
Tercihen	641	Takım	247	Sağlamak	169	Becerisi	125
Mezun	609	Teknolojileri	241	Aktif	164	Uyum	122
Geliştirme	576	Müşteri	235	Lisans	162	Windows	119

Tablo 3.10. Ulusal ilanlara ait ilanlarda kullanılan frekans değeri 100 den büyük olan kelimeler’in devamı

İngilizce	407	Olmak	217	Düzeyde	158	Sözlü	111
Web	405	Sistemleri	214	Uygun	157	Yeteneğine	108
Hakim	399	Test	212	Becerileri	154	Edebilecek	106
Proje	378	Bilgisi	211	Etmek	153	Sorumluluk	106
Adaylar	367	Problem	210	Gerekli	151	Yeteneği	106
Çalışmasına	354	Çözme	208	Güçlü	151	Raporlama	105
Takip	351	Elektronik	208	Engeli	146	Sektöründe	104
Deneyimli	349	Yönetim	203	Açık	146	Mvc	103
Teknik	344	Satış	203	Sistem	143	Tercihen	103
Yüksek	322	Veri	193	Geliştirmiş	143	Sistemi	102
Konularında	298	Tamamlamış	189	Seyahat	142	Matematik	100
Tasarım	280	Analiz	181	İkamet	140		
Hakkında	280	Seviyede	180	Görev	139		

Uluslararası ilanlarda kullanılan ve bağlaç olarak kullanılmayan kelimelere ait frekans değerleri hesaplanmıştır. Tablo 3.11’de, frekans değeri en yüksek ilk 50 kelimeye ait değerler gösterilmektedir. Uluslararası İlanlarda, 3976 adet farklı kelime kullanıldığı belirlenmiştir.

Tablo 3.11. Uluslararası ilanlara ait ilanlarda en çok tekrar edilen ilk 50 kelime ve frekans değerleri

Sıra	Kelime	Fr. De.	Sıra	Kelime	Fr. De.
1	Experience	6160	26	Networking	392
2	Computer	1588	27	Following	357
3	Technical	1551	28	Web	329
4	Degree	1460	29	Developing	312
5	Science	1443	30	Security	312
6	Equivalent	1232	31	Distributed	287
7	Qualifications	1125	32	Application	285
8	Practical	1099	33	Learning	277
9	Working	1091	34	Understanding	272
10	Engineering	1020	35	Solutions	272
11	Field	1007	36	Product	265
12	Related	980	37	Business	263
13	Preferred	965	38	Environment	261
14	Systems	883	39	Database	258

Tablo 3.11. Uluslararası ilanlara ait ilanlarda en çok tekrar edilen ilk 50 kelime ve frekans değerleri'nin devamı

15	Manage	864	40	Analysis	254
16	Data	841	41	Communication	245
17	Development	822	42	Teams	243
18	Software	812	43	Customer	236
19	Technology	747	44	Sales	234
20	Bachelor	653	45	Excellent	225
21	Ability	558	46	Machine	225
22	Programming	530	47	Mobile	220
23	Including	516	48	Coding	214
24	Design	477	49	Architecture	173
25	Skills	420	50	Algorithms	143

Bilişim Teknolojileri alanında kullanılan ve ilan içerisinde geçen teknik terimlere ait frekans değerleri çıkarılmıştır. Ulusal ilanlara ait sonuçlardan ilk 100 terim Tablo 3.12' de gösterilmektedir.

Tablo 3.12. Ulusal ilanlara ait en çok kullanılan ilk 100 terim ve frekans değerleri

Terim	Fr. De.	Terim	Fr. De.	Terim	Fr. De.	Terim	Fr. De.
Application	165	Ajax	34	Gsm	17	Browser	9
Sql	165	Wan	34	Code	16	Ccna	9
Javascript	126	Soap	33	Core	16	Ccnp	8
Html5	125	Active	31	Routing	16	Computer	8
Server	121	Bt	31	Vmware	16	Database	8
Asp	107	Apps	28	Adobe	15	Degree	8
Programming	92	Dns	28	Script	15	Excel	8
Java	91	Mysql	27	Postgresql	14	Jsp	8
Css	86	Tcp	27	Hyper-V	13	Remote	8
Network	86	Switch	25	Native	13	View	8
C#	80	Git	24	Nosql	13	Apache	7
Jquery	73	Vpn	23	Security	13	Applet	7
Communications	64	Access	22	System	13	Center	7
Microsoft	61	Storage	22	Data	12	Cluster	7
Framework	60	Execute	21	Developer	12	Dvi	7
Erp	44	Router	21	Hibernate	12	Gonna	7
Firewall	44	Swift	21	Soa	12	Hadoop	7
Android	43	Oop	20	Cisco	11	Http	7

Tablo 3.12. Ulusal ilanlara ait en çok kullanılan ilk 100 terim ve frekans değerleri'nin devamı

Api	43	Dhcp	19	Domain	11	Micro	7
Css3	43	Google	19	Internal	11	Patch	7
Php	43	Jira	19	Photoshop	11	Recovery	7
Office	41	C++	18	SEO	11	Search	7
JSON	38	Exchange	18	Client	10	Ux	7
Bootstrap	35	Iso	18	Embedded	10	Ado	6
Oracle	35	Python	18	UI	10	Adwords	6

Bilişim Teknolojileri alanında kullanılan ve uluslararası ilanlarda kullanılan teknik terimlere ait frekans değerleri çıkarılmıştır. Oluşan sıralamadaki ilk 100 terim, Tablo 3.13' de gösterilmektedir.

Tablo 3.13. Uluslararası ilanlara ait en çok kullanılan ilk 100 terim ve frekans değerleri

Terim	Fr. Değeri	Terim	Fr. Değeri	Terim	Fr. Değeri	Terim	Fr. Değeri
Python	636	Kotlin	29	Cad	11	Odm	7
Java	569	Kubernetes	26	Nosql	10	Oracle	7
C++	523	Mysql	24	Opengl	10	Unix	7
C	513	Matlab	23	Smtip	10	Wifi	7
Javascript	354	UI	22	Soap	10	Chipset	6
Linux	332	Arm	21	Vr	10	Cmos	6
Unix	211	Ajax	19	Angular	9	Dram	6
Sql	210	Html5	19	Dhcp	9	Pcie	6
Paas	132	Nlp	18	Pcb	9	Tlm	6
Iaas	128	Openflow	17	SPI	9	Asics	5
Ip	124	Ansible	16	Cloudsql	8	Cpus	5
Html	113	Chef	16	Css3	8	Elt	5
C#	112	Dsp	16	Gke	8	Json	5
Android	108	Proxy	16	Gsis	8	Ldap	5
Tcp	107	Puppet	16	I2c	8	Mvp	5
Perl	88	Vpn	16	Ipv6	8	Sip	5
Ms	85	Crm	15	Mi	8	Spss	5
Css	74	Isp	14	Po	8	Eda	4
Http	61	Chip	13	Rdbms	8	Ha	4
Dns	57	Dmz	13	Saml	8	Https	4
Hadoop	57	X86	13	Angularjs	7	IEEE	4
Ruby	49	Bios	12	Cdma	7	Mongodb	4
Sap	49	Camtasia	12	Directx	7	Opencl	4

Tablo 3.13. Uluslararası ilanlara ait en çok kullanılan ilk 100 terim ve frekans değerleri'nin devamı

Php	43	Mpls	12	Fpga	7	Ospf	4
Xml	37	Plm	12	Ipv4	7	Wlan	4

Tablo 3.12 ve Tablo 3.13’de gösterilen, ulusal ve uluslararası ilanlarda en çok tekrar edilen teknik terimler ile oluşturulan kelime bulutu Şekil 3.3’te gösterilmektedir. Bu kısımda elde edilen bilgiler, mevcut derslere ait içeriklerin güncellenmesine yardımcı olabilmektedir. Ek olarak, bu kısımdaki nitelikleri karşılayabilecek derslerin olmaması durumunda yeni derslerin açılması sağlanabilir.



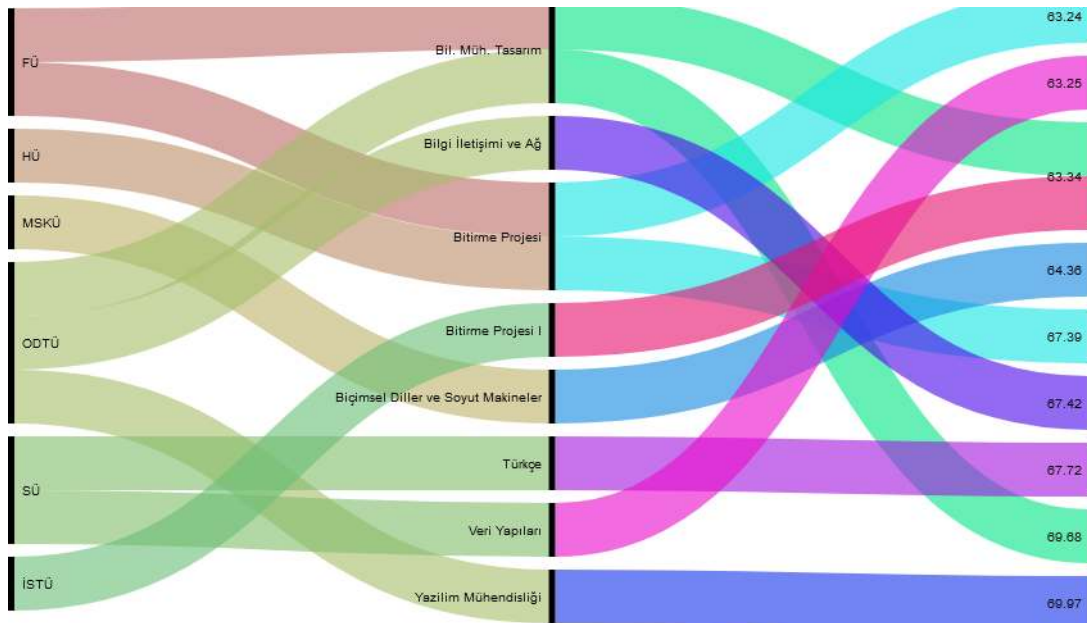
Şekil 3.3. Ulusal ve uluslararası ilanlarda geçen teknik terimlere ait kelime bulutu

Ulusal ilanlara ait verilerin işlenmesi sonucu en fazla ilan sayısına sahip pozisyon 483 ilanla “Bilişim/Bilgisayar” iken uluslararası alanda “Software Engineer” (Yazılım Mühendisi) olmuştur. Ulusal ve uluslararası ilanlar arasında daha iyi karşılaştırma yapılması amacıyla, “Bilişim/Bilgisayar” pozisyonu yerine, 3. sırada bulunan ve uluslararası ilanlarda ilk sırada bulunan “Software Engineer” pozisyonuna yakınlığı nedeniyle “Yazılım Uzmanı” pozisyonu tercih edilmiştir.

Tablo 3.14. Ulusal ilanlara ait “Bilişim/Bilgisayar” ve “Yazılım Uzmanı” pozisyonları karşılaştırması

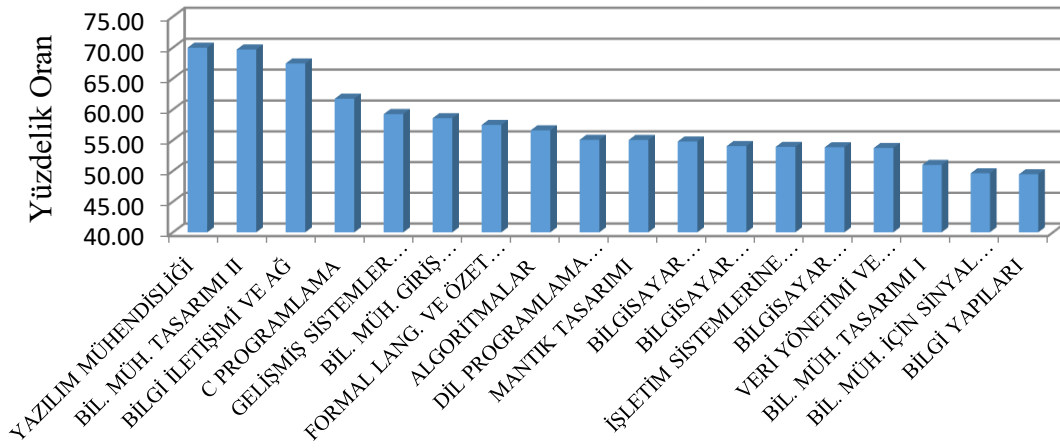
Üniversi- teler	TOPSIS		VIKOR		ELECTRE	
	Yazılım Uzmanı	Bilişim/Bilgi sayar	Yazılım Uzmanı	Bilişim/Bilgi sayar	Yazılım Uzmanı	Bilişim/Bilgi sayar
ODTÜ	1	1	1	1	1	1
İTÜ	2	2	2	2	2	2
SÜ	3	3	3	3	3	3
HÜ	4	4	4	4	4	4
MSKÜ	5	5	5	5	5	5
CÜ	6	6	6	6	6	6
İSTÜ	7	7	7	7	7	7
FÜ	8	8	8	8	8	8
İÜ	9	11	9	11	9	11
EÜ	10	10	10	10	10	10
GOÜ	11	9	11	9	11	9
AÜ	12	12	12	12	12	12

Üniversitelerin bu pozisyonlara ait ilanlardaki nitelikleri karşılama durumuna göre oluşan sıralama Tablo 3.14’ de gösterilmektedir. Üniversiteler herhangi bir pozisyona ait sıralamasını iyileştirmek için, ders içeriklerini güncelleyebilir veya yeni ders açabilirler. “Yazılım Uzmanı” pozisyonuna ait nitelikleri karşılama oranı en yüksek olan üniversite, Tablo 3.14’ de görüldüğü üzere ODTÜ dür.



Şekil 3.4. Tüm derslerin “Yazılım Uzmanı” pozisyonuna ait nitelikleri karşılama yüzdesi (En yüksek ilk 10 ders)

“Yazılım Uzmanı” pozisyonuna göre, tüm üniversitelerde okutulan zorunlu dersler arasında benzerlik oranı en yüksek çıkan ilk 10 ders ve hangi üniversitelerde okutulduğu da Şekil 3.4’de gösterilmektedir. Alüvyon diyagram kullanılarak oluşturulan Şekil 3.4’ü şöyle okuyabiliriz: ODTÜ’de (ilk kolon) okutulan Yazılım Mühendisliği (orta kolon) %69.97 (son kolonda yukarıdan aşağıya doğru olan sıralama küçükten büyüğe doğrudur.) benzerlik oranıyla en yüksek değere sahip olan derstir. Yazılım Uzmanı pozisyonuna ait değerlendirme sonucu ilk sırada bulunan ODTÜ’de okutulan ve alınması zorunlu derslere ait “Yazılım Uzmanı” pozisyonundaki nitelikleri karşılama oranları, Şekil 3.5’de yüzdelik oran şeklinde gösterilmiştir.



Şekil 3.5. Temel derslerin yazılım uzmanı pozisyonuna ait nitelikleri karşılama yüzdesi

Tablo 3.15. Seçmeli derslerin "Yazılım Uzmanı" pozisyonuna göre sıralaması

Seçmeli Ders Adı	TOPSIS		ELECTRE		VIKOR			
	Sıra	Değer	Sıra	Değer	Sıra	Q Değeri	S Değeri	R Değeri
Bulut bilişim	1	1.00	1	9	1	0.00	0.00	0.00
Semantik webe giriş	2	0.81	2	8	2	0.19	0.19	0.19
Bil. görüntü işlemeye giriş	3	0.69	3	7	3	0.31	0.31	0.31
Paralel hesaplamaya giriş	4	0.66	4	6	4	0.34	0.34	0.34
Oop'ye giriş ve sistemler	5	0.63	5	5	5	0.37	0.37	0.37
Bil. bilimi için mantık	6	0.48	6	4	6	0.52	0.52	0.52
Bilimsel hesaplama	7	0.34	7	3	7	0.66	0.66	0.66
Makine öğren. giriş	8	0.28	8	2	8	0.72	0.72	0.72
Veri tabanı yön. sis.	9	0.04	9	1	9	0.96	0.96	0.96
Çoklu ajan sis. – ikin. progr.	10	0.00	10	0	10	1.00	1.00	1.00

*Mevcut tabloda bulunan değerlerde yuvarlama yapılmıştır.

ODTÜ’de “Bilgisayar Mühendisliği” bölümünde okumakta olan ve iş hayatında “Yazılım Uzmanı” pozisyonunda çalışmak isteyen öğrenciler için, “Teknik Seçmeli Dersler” grubunda bulunan dersler üzerinde istatistiksel analiz işlemleri gerçekleştirilerek, ilgili pozisyona ait nitelikleri karşılama oranı %50 ve üstü olan ilk 10 seçmeli ders listesi, Tablo 3.15’ de gösterilmektedir. “Yazılım Uzmanı” pozisyonu ve “Software Engineer” pozisyonuna ait iş ilanlarında geçen teknik terimlere ait kelime bulutu Şekil 3.6’da ayrı ayrı gösterilmektedir.

Şekil 3.6. “Yazılım Uzmanı” ve “Software Engineer” ilanlarında geçen teknik terimlere ait kelime bulutu

Üniversiteler	TOPSIS		ELECTRE		VIKOR			
	Sıra	Değer	Sıra	Değer	Sıra	Q Değeri	S Değeri	R Değeri
ODTÜ	1	1.00	1	11	1	0.00	0.00	0.00
İTÜ	2	0.63	2	10	2	0.37	0.37	0.37
HÜ	3	0.37	3	9	3	0.63	0.63	0.63
MSKÜ	4	0.33	4	8	4	0.67	0.67	0.67
İSTÜ	5	0.31	5	7	5	0.69	0.69	0.69
İÜ	6	0.15	7	5	7	0.85	0.85	0.85
FÜ	8	0.15	6	6	6	0.85	0.85	0.85
CÜ	9	0.14	8	4	8	0.86	0.86	0.86
SÜ	7	0.12	9	3	9	0.88	0.88	0.88
AÜ	10	0.07	10	2	10	0.93	0.93	0.93
GOÜ	11	0.03	11	1	11	0.97	0.97	0.97
EÜ	12	0.00	12	0	12	1.00	1.00	1.00

Uluslararası 1317 adet ilan arasında “Software Engineer” pozisyonuna ait toplamda 249 ilan bulunmaktadır. Üniversitelerin bu pozisyona ait ilanlar ile oluşturulan sıralaması Tablo 3.16’da verilmiştir. Tablo 3.8’de gösterilen uluslararası ilanlara ait üniversitelerin iş ilanlarına uygunluk başarı sırası ile Tablo 3.16’da gösterilen üniversitelerin Software Engineer pozisyonuna göre başarı sırası karşılaştırması arasında farklar görülmektedir.

ODTÜ Bilgisayar Mühendisliği okuyan ve iş hayatında Software Engineer pozisyonunda çalışmak isteyen öğrenciler için, “Teknik Seçmeli Dersler” grubunda bulunan ve öğrenme çıktıları elde edilen dersler üzerinde istatistiksel analiz işlemleri gerçekleştirilerek, pozisyona ait nitelikleri karşılama oranı %50 ve üstü olan seçmeli dersler, Tablo 3.17’de gösterilmektedir.

Tablo 3.17. Teknik seçmeli derslerin “Software Engineer” pozisyonuna ait değerler

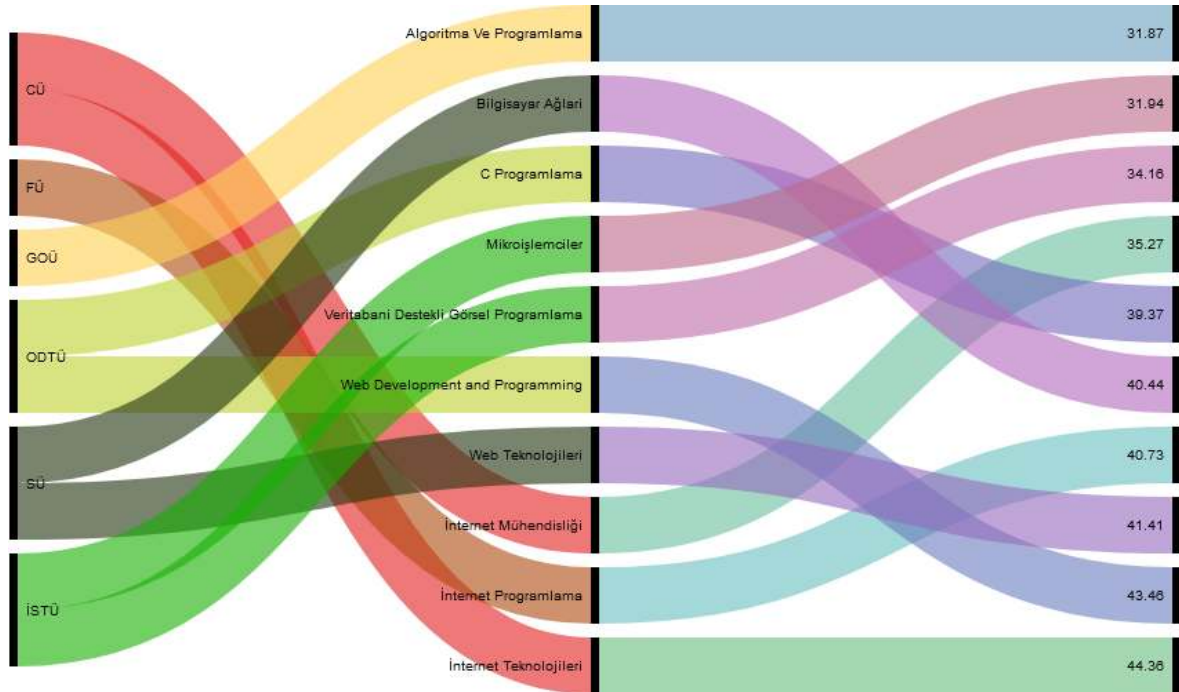
Teknik Seçmeli Dersler	TOPSIS		ELECTRE		VIKOR			
	Sıra	Değer	Sıra	Değer	Sıra	Q Değeri	S Değeri	R Değeri
Cloud computing	1	1.00	1	16	1	0.00	0.00	0.00
Intr. to mainfr. arc. & com.	2	0.84	2	15	2	0.16	0.16	0.16
Intr. to nat. lan. proc.	3	0.82	3	14	3	0.18	0.18	0.18
Intr. to computer vision	4	0.61	4	13	4	0.39	0.39	0.39
Practice of algorithms	5	0.54	5	12	5	0.46	0.46	0.46
Intr. to oop lang and sys.	6	0.50	6	11	6	0.50	0.50	0.50
Intr. to parallel com.	7	0.49	7	10	7	0.51	0.51	0.51
Logic for computer science	8	0.39	8	9	8	0.61	0.61	0.61
Scientific computing	9	0.35	9	8	9	0.65	0.65	0.65
Information systems dev.	10	0.33	10	7	10	0.67	0.67	0.67
Artificial intelligence	11	0.19	11	6	11	0.81	0.81	0.81
Computer graphics ii	12	0.19	12	5	12	0.81	0.81	0.81
Database man. sys.	13	0.17	13	4	13	0.83	0.83	0.83
Soft. dev. with script. lang.	14	0.16	14	3	14	0.84	0.84	0.84
Fundamentals of image proc.	15	0.11	15	2	15	0.89	0.89	0.89
Introduction to bioinfor.	16	0.58	16	1	16	0.94	0.94	0.94
Intr. com. & fortran prog.	17	0.00	17	0	17	1.00	1.00	1.00

*Mevcut tabloda bulunan değerlerde yuvarlama yapılmıştır.

Ulusal ilanlar kullanılarak, tüm üniversitelere ait temel derslerin Yazılım Uzmanı pozisyonunu karşılama oranları, Şekil 3.4’de alüvyal diyagram ile gösterilmektedir. Şekil 3.4’deki sıralamada, Sakarya üniversitesine ait Türkçe dersi, %67.22 benzerlik oranı ile 3. sırada görülmektedir. Bu durum, iş ilanlarından ve öğrenme çıktılarından elde edilen

kelimelerin sadece teknik terimlerden oluşmamasından kaynaklanmaktadır. Dolayısıyla, teknik terim harici kelimeler benzerlik oranının yükselmesine sebep olmaktadır.

Bu olumsuz etkeni ortadan kaldırmak için sadece teknik terimler kullanılarak tekrar istatistiksel analizler yapılmıştır. Yapılan istatistiksel analizler sonucunda oluşan yeni sıralama Şekil 3.7’ de alüvyal diyagram olarak gösterilmektedir. Alüvyal diyagramda görüleceği üzere, sadece teknik terimler göz önünde bulundurularak yapılan değerlendirmede tüm dersler temel teknik derslerden oluşmaktadır. Cumhuriyet Üniversitesine (CÜ) ait İnternet Teknolojileri dersi %44.36 benzerlik oranı ile 1. sırada yer görülmektedir.



Şekil 3.7. Tümü derslerin “Yazılım Uzmanı” pozisyonuna ait nitelikleri karşılama yüzdesi (En yüksek ilk 10 ders)

Toplamda 1816 adet iş ilanının içerisinde en az 1 teknik terim bulunan iş ilanı sayısı 1411 dir. Toplamda 464 ders içeriği içerisinde, teknik terim barındıran içerik sayısı 167 adettir. İş ilanlarında ve ders öğrenme çıktıları içerisinde teknik terim bulunmayanlar elenerek, genel okul sıralaması ve “Yazılım Uzmanı” pozisyonuna ait ders sıralaması sırasıyla Tablo 3.18 ve Tablo 3.20’de sunulmuştur. Şekil 3.7 incelendiğinde, içeriği en uyumlu ders Cumhuriyet Üniversitesi’nin dersi iken, tüm dersleri en uyumlu olan üniversite Fırat Üniversitesi’dir.

Tablo 3.18. Üniversitelerin teknik terim tabanlı genel sıralaması

Üniversiteler	TOPSIS		VIKOR				ELECTRE	
	Değer	Sıra	Q	S	R	Sıra	Değer	Sıra
FÜ	0.854532493	1	0.215974368	0.049747503	0.006790253	3	12	1
SÜ	0.734787235	2	0.082168501	0.17721595	0.004977404	1	11	2
ODTÜ	0.672548005	3	0.18531253	0.270088303	0.005340662	2	10	3
MSKÜ	0.572969849	4	0.500487964	0.407637695	0.007241933	4	7	4
İTÜ	0.516547424	5	0.646480543	0.511464448	0.007905581	6	6	5
CÜ	0.489714726	6	0.566444902	0.453936014	0.007545052	5	2	6
İSTÜ	0.378566024	8	0.891747892	0.657468159	0.009174312	7	2	6
EÜ	0.385363355	7	0.896298129	0.664526967	0.009174312	8	2	6
HÜ	0.277039381	11	0.943417073	0.737622843	0.009174312	9	1	9
AÜ	0.280022696	10	0.959035627	0.761851993	0.009174312	10	1	9
İÜ	0.34587046	9	0.983542141	0.799869081	0.009174312	11	0	11
GOÜ	0.240657456	12	1	0.825400249	0.009174312	12	0	11

*Mevcut tabloda bulunan değerlerde, birbirlerine yakın olması nedeniyle yuvarlama yapılmamıştır.

Sadece teknik terimler dikkate alınarak oluşturulan Tablo 3.18'deki ve tüm içerik ele alınarak oluşturulan Tablo 3.6'daki genel sıralama sonuçları karşılaştırması ve üniversite sıralarındaki değişim Tablo 3.19'da gösterilmektedir. Örnek olarak ulusal ilanlar kullanılarak yapılan sıralama sonucunda ilk sırada bulunan ODTÜ, ulusal ilanlarda ve öğrenme çıktılarında bulunan sadece teknik terimler kullanılarak yapılan, TOPSIS ve ELECTRE algoritması sonucunda 3. sıraya, VIKOR algoritması sonucunda 2. sıraya gerilemiştir.

Tablo 3.19. Üniversitelerin tablolar arası karşılaştırma sonuçları

Üniversiteler	TOPSIS			ELECTRE			VIKOR		
	Tüm	Terim	Değişim	Tüm	Terim	Değişim	Tüm	Terim	Değişim
ODTÜ	1	3	-2	1	3	-2	1	2	-1
İTÜ	2	5	-3	2	5	-3	2	6	-4
HÜ	3	11	-8	3	9	-6	3	9	-6
SÜ	4	2	2	3	2	1	4	1	3
MSKÜ	5	4	1	5	4	1	5	4	1
CÜ	6	6	0	6	6	0	6	5	1
İSTÜ	7	8	-1	7	6	1	7	7	0
FÜ	8	1	7	7	1	6	8	3	5
GOÜ	9	12	-3	9	11	-2	9	12	-3
İÜ	10	9	1	10	11	-1	10	11	-1
EÜ	11	7	4	10	6	4	11	8	3
AÜ	12	10	2	10	9	1	12	10	2

Tablo 3.20. Üniversitelerin "Yazılım Uzmanı" pozisyonuna göre sıralaması

Üniversiteler	TOPSIS		VIKOR				ELECTRE	
	Değer	Sıra	Q	S	R	Sıra	Değer	Sıra
FÜ	1	1	0	0	0	1	11	1
SÜ	0.873762597	2	0.126237403	0.126237403	0.126237403	2	10	2
ODTÜ	0.631430714	3	0.368569286	0.368569286	0.368569286	3	9	3
CÜ	0.514988129	4	0.485011871	0.485011871	0.485011871	4	8	4
İSTÜ	0.434815736	5	0.565184264	0.565184264	0.565184264	5	7	5
MSKÜ	0.393538794	6	0.606461206	0.606461206	0.606461206	6	6	6
GOÜ	0.324854664	7	0.675145336	0.675145336	0.675145336	7	5	7
İTÜ	0.282107181	8	0.717892819	0.717892819	0.717892819	8	4	8
EÜ	0.200928861	9	0.799071139	0.799071139	0.799071139	9	3	9
AÜ	0.143105162	10	0.856894838	0.856894838	0.856894838	10	2	10
HÜ	0.130692344	11	0.869307656	0.869307656	0.869307656	11	1	11
İÜ	0	12	1	1	1	12	0	12

*Mevcut tabloda bulunan değerlerde, birbirlerine yakın olması nedeniyle yuvarlama yapılmamıştır.

Üniversitelerin, ilanlardan elde edilen tüm verilerin kullanılması ile oluşan sıralama (Tablo 3.14) (Tüm) ve ilanlardan elde edilen verilerin içerisinde sadece teknik terim olarak sınıflandırılan verilerin kullanılmasıyla oluşan sıralama (Tablo 3.20) (Terim) sonuçlarıyla birlikte iki sıralamanın karşılaştırılması sonucu oluşan aşağı-yukarı yönlü farklar (Değişim) Tablo 3.21’de gösterilmektedir.

Tablo 3.21. Üniversitelerin Yazılım Uzmanı pozisyonuna ait tablolar arası karşılaştırma sonuçları

Üniversiteler	TOPSIS			ELECTRE			VIKOR		
	Tüm	Terim	Değişim	Tüm	Terim	Değişim	Tüm	Terim	Değişim
ODTÜ	1	3	-2	1	3	-2	1	3	-2
İTÜ	2	8	-6	2	8	-6	2	8	-6
SÜ	3	2	1	3	2	1	3	2	1
HÜ	4	11	-7	4	11	-7	4	11	-7
MSKÜ	5	6	-1	5	6	-1	5	6	-1
CÜ	6	4	2	6	4	2	6	4	2
İSTÜ	7	5	2	7	5	2	7	5	2
FÜ	8	1	8	8	1	8	8	1	8
İÜ	9	12	-3	9	12	-3	9	12	-3
EÜ	10	9	1	10	9	1	10	9	1
GOÜ	11	7	4	11	7	4	11	7	4
AÜ	12	10	2	12	10	2	12	10	2

Örnek olarak ulusal ilanlar kullanılarak yapılan Yazılım Uzmanı pozisyonu sıralama sonucunda ilk sırada bulunan ODTÜ, ulusal ilanlarda ve öğrenme çıktılarında bulunan teknik terimlere kullanılarak yapılan, TOPSIS algoritması sonucunda 3. sıraya, ELECTRE algoritması sonucunda 3. sıraya, VIKOR algoritması sonucunda 3. sıraya gerilediği görülmektedir.



4. SONUÇ VE ÖNERİLER

Üniversitelerin bilgisayar mühendisliği bölümüne ait ÖSYM tarafından yayınlanan 2018 yılı en düşük yerleştirme puanı sıralaması ve URAP tarafından yayınlanan Türkiye Üniversiteleri Sıralaması Listesi'ne göre yapılan sıralar ile bu çalışmada önerilen sistemin yaptığı sıralamar karşılaştırıldığında, benzer sonuçlar ortaya çıktığı görülmektedir. Bazı üniversitelerin sıralamalarında 1-2 sıra aşağı yada yukarı yönlü oynamaların önemli bir nedenide, bireylerin üniversite tercih aşamasında üniversitenin bulunduğu il, sosyal çevre, kendi evine olan uzaklığı vb. farklı kriterleri de göz önünde bulundurarak seçim yapması olduğu düşünülmektedir. Bir diğer neden ise, ÖSYM, URAP ve Önerilen sistemin değerlendirme kriterlerinin ve amacının farklı olmasıdır.

Yapılan deneysel çalışmada, ÇKKV algoritmalarında ağırlık değeri toplamı 1 olacak şekilde, her pozisyona eşit ağırlık değeri verilmiştir. Birey arzu ettiği pozisyona ait ağırlık değerini yükseltip diğer pozisyonların ağırlık değerini düşürerek kendisine özgü bir sıralama elde edebilir. Bu açıdan önerilen sistemin, özgün sıralama sunabilmesi bakımından, birey için diğer sıralama ölçütlerinden daha kullanışlı olabileceği görülmektedir.

Deneysel çalışmada ilk sırada olan ODTÜ'ye ait resmi internet sayfasında, “teknik seçmeli” olan 26 dersin sadece 17 tanesinde öğrenme çıktısına ulaşılabilmektedir. Bu eksiklik diğer üniversiteler için de geçerlidir. Üniversitelerde aynı dersin devamı niteliğinde olan farklı dönem derslerinin içeriklerinin de genel olarak aynı hazırladığı (Veritabanı I ve Veritabanı II gibi) görülmektedir. Bu durum, bireylerin doğru, gerekli ve yeterli bilgiye erişimi konusunda problem teşkil etmektedir. Bu çalışmada önerildiği gibi bir sıralama ölçütü kullanımının yaygınlaşması kurumların bunu önemsemesini sağlayacaktır. Ayrıca, pozisyonlara ait ilanlar içerisinde geçen teknik terimlerin belirlenmesi sayesinde pozisyon ile ilgili edinilmesi gereken yeterlilikler belirlenebilir ve ders içerikleri buna göre güncellenebilir. Bu sayede, kurumlar eksikliklerini giderip sıralamalarını iyileştirebilirler.

Ulusal ilanlara ait pozisyon listesine bakıldığında toplanılan ilanların büyük çoğunluğunun pozisyon bilgisi bilişim/bilgisayar olarak görülmektedir. Bunun nedeni olarak, işverenin aranılan niteliklerin tam olarak hangi pozisyonlara ait olduğu konusu ile ilgili net bilgiye sahip olmaması olduğu düşünülmektedir. Önerilen sistem sayesinde, hangi niteliklerin hangi pozisyonlara uygun olduğu daha iyi anlaşılabilir.

4.1. Öneriler

Yapılan çalışmada her ne kadar üniversite sıralamaları büyük oranda gerçek sıralarını yansıtmakta olsada elde edilen yüzde oranları en iyi üniversite derslerinde dahi %80'in altında kalmaktadır. Ülkemizdeki üniversitelerin Bologna Süreci'ne uyum aşamasında çok yol katettiği gözlenmektedir. Fakat hâlâ sosyal medya ve internet üzerinden tanıtıma yeteri kadar önem gösterilmemesi nedeniyle birçok üniversitenin ders içeriklerine kurumsal internet sayfaları üzerinden ulaşılammış ve çoğu üniversitede de ders içeriklerinin eksik şekilde girildiği gözlenmiştir. Daha iyi bir eğitim için, kurumların önerilen sistemden yararlanarak bu eksikliklerini gidermeleri tavsiye edilmektedir. Ayrıca, öğrencilerin de tercihlerini yaparken önerilen sistemden yararlanmaları faydalarına olacaktır.

Üniversitelerin "İş ve Meslek Danışmanı" sertifikasına sahip çevrimiçi iş ilanı sitelerinde belirtilen nitelikleri karşılama oranlarının düşük olduğu pozisyonlar gözlemlenmektedir. Üniversiteler eğer müfredat ve ders içeriklerini güncelleyerek bu ihtiyaçları gideremiyorsa yeni bölümler açmaları önerilmektedir.

İş literatürüne girmiş olan mesleklere ait pozisyonlar için özel sözlükler oluşturulabilir. Bu sayede eğitim kurumları ders içeriklerini buna göre oluşturabilirler. İşverenlerde, doğru nitelikleri uygun pozisyon için belirtebilirler.

4.2. Kısıtlamalar ve Zorluklar

Önerilen sistemde, yüksek başarı elde etmek için eğitim ortamlarındaki bilgilerin eksiksiz bir şekilde ve iş ilanlarında büyük çoğunluğunun bilinçli ve uygun pozisyonlarda çevrimiçi olarak erişilebilir olması gerekmektedir. Diğer durumlarda veri eksikliğinden kaynaklı daha az başarılı sonuçlar elde edilebilir.

Önerilen sistemin temel zorluğu veri toplama aşamasıdır. Her ne kadar web kazıma yöntemleri ile manuel veri toplama iş yükü hafifletilmiş olsa da her web sitesinin yapısının farklı olması kodlamaların yeniden yapılmasını zorunlu kılmaktadır.

4.3. Potansiyel Gelecek Çalışmalar

Önerilen sisteme yapay zekâ ve büyük veri işleme yöntemleri katılarak daha geniş kapsamlı sonuçlar elde edilebilir. Böylece sadece mevcut durum değerlendirmeleri değil

gelecek durum deęerlendirmeleri de yapılabilir. Yani, gelecek 10 yılda X niteliklere daha yoğun talepler olacak veya Y nitelikler daha az rağbet görecektir gibi öngürülerde de bulunulabilir.



5. KAYNAKLAR

- Akın, A.A., Akın, M.D.,** 2007. Zemberek, an open source nlp framework for turkic languages. *Structure* 10: 1–5.
- Alessi, S.M., Trollip, S.R.,** 2000. *Methods and development . Multimedia for learning*, Pearson, 580s.
- Beceran, E.,** 2016. How much of the sensible choose university: students research study Suleyman Demirel University. *Süleyman Demirel Üniversitesi Vizyoner Dergisi*, 2: 101-110.
- Bowers, M.R., Camm, J.D., Chakraborty, G.,** 2018. The evolution of analytics and implications for industry and academic programs. *INFORMS Journal on Applied Analytics*, 48(6), 487-499.
- Cramer, E.M. Tenzek, K.E.,** 2012. The chaplain profession from the employer perspective: an analysis of hospice chaplain job advertisements. *Journal of Health Care Chaplaincy*, 18(3-4): 133-150.
- De Mauro, A., Greco, M., Grimaldi, M., Ritala, P.,** 2018. human resources for big data professions: a systematic classification of job roles and required skill sets. *Information Processing & Management*, 54(5): 807-817.
- Erkarslan, O.,** 2013. A systematic review of the relations between industrial design education and industry in turkey through SWOT analysis. *The Design Journal*, 16(1): 74-102.
- Göksu, A., Güngör, İ.,** 2008. Bulanık analitik hiyerarşik proses ve üniversite tercih sıralamasında uygulanması. *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 13(3): 1–26.
- Hirudayaraj, M., Baker, R.,** 2018. HRD competencies: analysis of employer expectations from online job postings. *European Journal of Training and Development*, 42(9): 577-596.
- Hwang, C.L. Yoon, K.,** 1981. Methods for multiple attribute decision making. *Multiple Attribute Decision Making: Methods and Applications A State-of-the-Art Survey* (ss. 58-191). Springer Berlin Heidelberg.
- Jonassen, David H.,** 1986. Hypertext principles for text and courseware design. *Educational Psychologist* 21,4: 269-92.
- Jones, T. F.,** 1966. “The dollars and sense of continuing education”. *IEEE Transactions on Aerospace and Electronic Systems* AES-2 2:140-42.

- Karanja, E., M. Grant, D., Freeman, S., Anyiwo, D.,** 2016. Entry level systems analysts: what does the industry want? *Informing Science: The International Journal of an Emerging Transdiscipline*, 19: 141-160.
- Kennan, M. A., Willard, P., Cecez-Kecmanovic, D., Wilson, C. S.,** 2008. Is knowledge and skills sought by employers: a content analysis of australian is early career online job advertisements, *Australasian Journal of Information Systems*, 15(2).
- Kureková, L.M., Beblavý, M., Haita, C., Thum, A.E.,** 2016. Employers' skill preferences across europe: between cognitive and non-cognitive skills. *Journal of Education and Work*, 29(6): 662-687.
- Liddy, E.,** 2001. Natural language processing, *In Encyclopedia of Library and Information Science*, 2.
- Neimeyer, Greg J., Jennifer M. Taylor, Ronald H. Rozensky.,** 2012. The diminishing durability of knowledge in professional psychology: a delphi poll of specialties and proficiencies. *Professional Psychology: Research and Practice* 43 (4): 364-71
- Novikov, P. S.,**1965. Binary codes capable of correcting deletions, insertions, and reversals v. i. levenshtein. *Soviet Physics- Doklady*.
<https://nymity.ch/sybilhunting/pdf/Levenshtein1966a.pdf>
- Omar, N.H., Manaf, A.A., Mohd, R.H., Kassim, A.C., Aziz, K.A.,** 2012. Graduates' employability skills based on current job demand through electronic advertisement. *Asian Social Science*, 8(9): 103.
- Opricovic, S., Tzeng, G.-H.,** 2002. Compromise solution by mcdm methods: a comparative analysis of vikor and topsis. *European Journal of Operational Research*, 156: 445-455.
- Parrish,** 2017 S. *Half life: the decay of knowledge and what to do about it*, Farnam Street
Retrieved from: <https://www.fs.blog/2018/03/half-life>
- Rámháp, S., Szombathelyi, M.K., Májer, E.P.,** 2015. Special coginfo application by non-business organizations: case study on marketing communication of a higher education institution. *2015 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, 483-488.
- Ravinet, P.,** 2008. From voluntary participation to monitored coordination: why european countries feel increasingly bound by their commitment to the bologna process. *European Journal of Education*, 43(3):353-367.
- Simões, C., Soares, A.M.,** 2010. Applying to higher education: information sources and choice factors. *Studies in Higher Education*, 35(4): 371-389.
- Skatova, A., Ferguson, E.,** 2014. Why do different people choose different university degrees? Motivation and the choice of degree. *Frontiers in Psychology*, 5.

- Şahin, İ., Zoraloğlu, Y.R., Fırat, N.Ş.,** 2011. Üniversite öğrencilerinin yaşam amaçları, eğitsel hedefleri üniversite öğreniminden beklentileri ve memnuniyet durumları. *Kuram ve Uygulamada Eğitim Yönetimi Dergisi*, 17(3):429-452.
- Terblanche, C., Wongthongtham, P., Villazon-Terrazas, B.M.,** 2016. Ontology based intelligent system for online employer demand analysis, *Impact of Economic Crisis on Education and the Next-Generation Workforce*, 185–213.
- Triantaphyllou, E.,** 2000. Natural language processing. In *Encyclopedia of Library and Information Science*, 2nd Ed. NY. Marcel Decker, Inc.
- Tüsiad, BCG,** 2017. Türkiye'nin sanayide dijital dönüşüm yetkinliği. *TÜSİAD-T/2017,12 – 589*.
- World Economic Forum BCG,** 2018. *Towards a reskilling revolution: a future of jobs for all*. Insight report (World Economic Forum). Geneva: World Economic Forum. <https://www.weforum.org/reports/towards-a-reskilling-revolution>.
- URL-1,** 2019. https://dokuman.osym.gov.tr/pdfdokuman/2018/YKS/ondeg_yks_rapor_31_072018.pdf. 2018 Yükseköğretim Kurumları Sınavı (YKS) Sonuçları. 09 Nisan 2019.
- URL-2,** 2019. <https://istatistik.yok.gov.tr/>. *Yükseköğretim Bilgi Yönetim Sistemi*. 09 Nisan 2019.
- URL-3,** 2019. <http://tyyc.yok.gov.tr/>. *Türkiye YüksekÖğretim Yeterlilikler Çerçevesi*. 11 Nisan 2019.
- URL-4,** 2019. https://insights.stackoverflow.com/survey/2019/?utm_source=social-share&utm_medium=social&utm_campaign=dev-survey-2019. *Stack Overflow Developer Survey 2019*. 07 Mayıs 2019.
- URL-5,** 2019. <https://www.w3.org/standards/webdesign/htmlcss>. *HTML & CSS - W3C*. 16 Nisan 2019
- URL-6,** 2019. <https://www.w3.org/TR/html52/>. *HTML 5.2*. 16 Nisan 2019
- URL-7,** 2019. <https://www.w3.org/TR/1999/REC-xpath-19991116/>. *XML Path Language (XPath)*. 02 Mayıs 2019.
- URL-8,** 2019. <http://pypl.github.io/PYPL.html>. *PYPL PopularitY of Programming Language index*. 02 Mayıs 2019.
- URL-9,** 2019. <https://www.tiobe.com/tiobe-index/>. *TIOBE Index | TIOBE - The Software Quality Company*. 02 Mayıs 2019.
- URL-10,** 2019. <https://selenium.dev/projects/>. *Selenium Projects*. 29 Aralık 2019.
- URL-11,** 2020. <https://openpyxl.readthedocs.io/en/stable/performance.html>. *Performance — openpyxl 3.0.3 documentation*. 20 Ocak 2020.

- URL-12**, 2020. <https://db-engines.com/en/ranking>. The DB-Engines Ranking ranks database management systems according to their popularity. 21 Ocak 2020.
- URL-13**, 2019. <https://dev.mysql.com/doc/relnotes/connector-python/en/preface.html#legalnotice>. MySQL Connector/Python Release Notes: Preface and Legal Notices. 29 Aralık 2019.
- URL-14**, 2019. <http://zembereknlp.blogspot.com/>. Zemberek NLP. 06 Mayıs 2019.
- URL-15**, 2019. <https://web.archive.org/web/20170726042213/http://jpype.sourceforge.net/> JPyPe - Java to Python integration. 06 Mayıs 2019.
- URL-16**, 2019. <https://web.archive.org/web/20170726042213/http://jpype.sourceforge.net/> 06 Mayıs 2019.
- URL-17**, 2019. <https://www.computerhope.com>. Computer Hope's free computer help. 29 Aralık 2019.
- URL-18**, 2018. https://www.yok.gov.tr/HaberBelgeleri/Haber%20%C4%B0%C3%A7erisindeki%20Belgeler/Dosyalar/2018/yok_un_tercih_anketi_sonuclandi.pdf. Yök'ün yaptığı anket sonuçlandı. 28 Ocak 2020.

ÖZGEÇMİŞ

Adı Soyadı: Yiğit ALİŞAN

Yabancı Dil: İngilizce

Doğum Yeri ve Yılı: Kadirli/1988

E-Posta: yigit.alisan@linux.org.tr

Eğitim ve Mesleki Geçmişi:

- 2015-2018, Munzur Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü
- 2007-2011, GaziOsman Paşa Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Öğretmenliği Bölümü
- 2014-, Munzur Üniversitesi, Bilgi İşlem Daire Başkanlığı, Programcı

Yayınları ve/veya Bilimsel/Sanatsal Faaliyetleri:

- 2019, Bildiri, TRB-1 bölgesindeki online iş ilanlarında aranan niteliklerin web kazıma ve doğal dil işleme ile incelenmesi, 5.Uluslararası bölgesel kalkınma konferansı Malatya.

Mesleki Birlik/Dernek/Kuruluş Üyelikleri:

- 2014-, Linux Kullanıcıları Derneği, Ankara.