T.C.
KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

SINIFLANDIRMA VE VERİ ANALİZİ

WEB SİTE DEDEKTÖRÜ

EMİNE ELİF ERCAN

RESUL BERKEM AYDEMİR

BİTİRME PROJESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

DANIŞMAN: SOYDAN SERTTAŞ

KÜTAHYA 2021

T.C.
KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

VERİ ANALİZİ

WEB SİTE DEDEKTÖRÜ

Emine Elif ERCAN ve Resul Berkem AYDEMİR tarafından hazırlanan proje çalışması .../.../…. Tarihinde aşağıdaki jüri tarafından Kütahya Dumlupınar Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünde Lisans **BİTİRME PROJESİ** olarak kabul edilmiştir.

Proje Danışmanı
Dr. Öğr. Üyesi Soydan SERTTAŞ
Kütahya Dumlupınar Üniversitesi

Jüri Üyeleri
Prof. Dr. ….
Kütahya Dumlupınar Üniversitesi _____

Doç. Dr. ….
Kütahya Dumlupınar Üniversitesi _____

Dr. Öğr. Üyesi ….
Kütahya Dumlupınar Üniversitesi _____

TEŞEKKÜR

Çalışmamız sırasında bizden yardımlarını esirgemeyen, tez konumuzun seçimin de ve yapım aşamasında her türlü desteği bize sağlayan ve her konuda sonsuz sabır gösteren sevgili danışman hocamız **Sayın Dr. Öğr. Üyesi Soydan Serttaş'a**

teşekkürlerimizi bir borç biliriz.

Çalışmamız boyunca bize her türlü maddi ve manevi desteği gösteren sevgili ailelerimize teşekkür ederiz. Deneysel çalışmalarımızda bize yardımcı olan Yüksek lisans Öğrenci arkadaşımız Binnaz KILIÇLI'ya teşekkür ederiz.

İÇİNDEKİLER

Sayfa

SİMGE VE KISALTMALAR

DPÜ	Dumlupınar Üniversitesi
VTYS	Veritabanı Yönetim Sistemi
ÖBS	Öğrenci Bilgilendirme Sistemi
DB	Database (Veritabanı)

ŞEKİL LİSTESİ

Sayfa

GİRİŞ

1.1 Projenin Amaç ve Kapsamı

Proje konusunun adı “Sınıflandırma ve Veri analizi ile web site dedektörü” dür. Veri analizi bilgileri bulma, sonuçlara varma ve karar alma sürecini desteklemek amacı ile verileri inceleme, dönüştürme ve modelleme yapmaktır. Gerekli veriler aynı zamanda bir elemeden geçirilerek yararlı olmayan bilgileri ayrıştırır ve verilerin çıkarıldığı modelleme işlemini kapsar. Toplanan bu bilgiler aşamasında önemli olan sonuca ulaşmaktır. Çıkarılan verilerle birlikte nasıl yol izleneceği ve ne şekilde kullanılacağı netleşir, bu durum; veri analiz sistemi dönüşüm süreci olarak da kabul edilebilir.

Yaygın bir şekilde "içerik denetimi yazılımı" olarak tanımlanan Web filtresi, bir kullanıcının bilgisayarında ziyaret edebileceği web sitelerini kısıtlamak üzere tasarlanmış bir yazılım parçasıdır. Bu filtreler, bir beyaz liste veya kara liste kullanarak çalışabilir. Beyaz liste, yalnızca filtreyi ayarlayan kişinin özel olarak seçtiği sitelere erişime olanak tanırken kara liste, filtreye yüklenen standartların belirlediği istenmeyen sitelere erişimi kısıtlar. Bu programlar, istenen sitenin URL'sine bakar ve site içeriğinde kısıtlanmış anahtar kelimeleri arar, daha sonra bağlantının engellenmesine veya bağlantıya izin verilmesine karar verir. Filtreler genellikle bir tarayıcı uzantısı, bilgisayarda bağımsız bir program veya genel güvenlik çözümünün bir parçası olarak yüklenir. Bununla birlikte, bir İSS veya işletme tarafından, birden çok kullanıcının aynı anda Web erişimini kısıtlamak için ağ tarafına da yüklenebilirler. Bazı arama motorları da arama sonuçlarından istenmeyen sayfaları çıkarmak için basit filtreler kullanır.[1]

Projede yapılacak olan uygulama ile makine öğrenimi algoritmalarından birini kullanarak sonrasında algoritmanın performansını geliştirir. Uygulamamız ile web sitelerde çıkan linklerin ve pop-up ların güvenilir mi değil mi olduğuna ait bütün verileri istenilen şekilde kullanabilmeyi ve raporlayabilmeyi sağlamak amacı ile yazılmış bir uygulama olacaktır. Proje uygulaması içerisinde web sitelerde planlama ve programlamanın, daha kolay, anlaşılabilir ve sağlıklı sonuç üreten, üretilen sonuca göre raporlama sistemi sunan geniş kapsamlı bir uygulama olacaktır.

Günümüzde web sitelerin bütün işleyiş sistemini herhangi bir program kullanmadan tek elden kontrol etmek oldukça zordur. Bu uygulama web sitesinde çıkan linkleri ve pop-up ları incelemek ve planlamayı yapmak adına çok daha kolay bir hâl almasını hedeflemektedir.

Uygulamada ne gibi içerikler vardır;

Phising denilen dolandırıcıların rastgele kullanıcı hesaplarına e-mail veya pop-up gönderdikleri saldırı türleridir. E-postalar , bilinen web sitelerinden veya kullanıcının bankasından, kredi kartı şirketinden e-posta veya internet hizmeti sağlayıcısından gönderilmiş gibi gözükmesidir. Uygulamamızla da bazı linklerin güvenilir olup olmadığını kontrol ederek sınıflandıran geniş kapsamlı bir uygulama geliştireceğiz. Böylelikle bilgi güvenliği, virüsler, siber ataklara karşı koruma sağlayacağız.

Pop-Up Engelleme Yöntemleri Nelerdir?

Bu yöntemlerden birincisi, reklam engelleyiciler kullanmak oluyor. Piyasada çeşitli geliştiriciler tarafından kullanıma sunulan reklam engelleme programları ve eklentileri mevcuttur. Bu eklenti ve programlar, kullanıcılarını sadece pop-up reklamlardan değil bir sayfada karşılaşabilecek çoğu reklamdan kurtarıyor. Fakat bu eklenti ve programlar, bazı kullanıcılarda güvenlik endişesi yaratıyorlar. Bundan dolayı bu kullanıcılar, alternatif bir pop-up engelleme yöntemi arıyorlar.

Buna istinaden bizim uygulamamız ön plana çıkıyor.Çünkü linkleri kontrolden geçirip phising e izin verilmeyerek bilgi güvenliğini öne çıkarıyor.

Özetlemek gerekirse uygulama temel olarak 3 katmandan oluşması hedeflenmektedir.

1- Verilerin analizi ve sınıflandırılması
2- Makine öğrenimi ile algılama
3- Algoritmalar ile performans geliştirme

1.2 Proje Konusunun Anlam ve Önemi

İnternet kullanımı yaygınlaştıkça, kurum çalışanları veya bireysel kullanıcılar daha fazla çevrimiçi olmak, ürün veya hizmetlere erişimde interneti kullanmayı talep etmektedir. Bu noktada internet kullanımının yaygınlaşması ile alışverişlerimiz, bankacılık işlemlerimiz, finansal işlemlerimiz, kurum içi iletişimlerimiz ve benzeri birçok kritik veri internet üzerinde yaygın olarak kullanılmaya başlanmıştır. Doğal olarak bu durum siber saldırganların bakış açısını değiştirerek hedefli saldırıların artırmasına sebep olmuştur.

Web filtreleme yazılımının iki ana müşteri tabanı vardır: Çocuklarının istemedikleri veya uygunsuz buldukları içeriğe erişmesini engellemek isteyen ebeveynler ve çalışanlarının, işleriyle ilgili olmayan web sitelerine erişimini önlemek isteyen işletmeler. Web filtreleri, pornografi veya kumarla ilgili siteler gibi genellikle kötü amaçlı yazılım barındıran sitelere erişimi engelleyeceğinden, kötü amaçlı yazılım için engelleme aracı olarak da yaygın bir şekilde kullanılır. En gelişmiş filtreler, hassas verilerin yayınlanmaması için İnternet üzerinden gönderilen bilgileri bile engelleyebilir.

Web tabanlı bir proxy, yabancı dildeki web sitelerini kullanma veya kişisel bir proxy sunucusu için bir VPN oluşturma gibi web filtreleme yazılımlarını aşan yöntemler vardır. Bu boşluklardan dolayı, ağ yöneticileri veya endişe duyan ebeveynlerin seçtikleri filtrenin yalnızca belirli web sitelerini engellemek veya belirli sitelere izin vermekten daha fazlasını yapabileceğinden emin olması gerekir.

Siber saldırganlar phishing yöntemleri ile bilinçsiz kullanıcıları hedefleyerek büyük zararlara sebep olmaktadır. Phising saldırıları hedefli olarak yapıldığı takdirde ise büyük bir başarı oranına sahiptir. Doğal olarak siber saldırganlar internet tarihinin en eski ve en etkili yöntemlerinden biri olan phishing saldırılarını sıklıkla kullanmaktadır. Sosyal mühendislik saldırıları ile gerçekleştirilen spear phishing saldırıları ise maalesef ki siber saldırganların elinde korunması zor ve tehlikeli bir siber silah olarak kurumları tehdit etmektedir.

Phishing saldırıları hem sosyal mühendislik hem de teknik altyapı kullanılarak gerçekleştirilen bir suç olarak tanımlanır. Yaygın olarak e-posta aracılığıyla gerçekleştirilen bu saldırılar günümüz sosyal ağlarının popüler olması ile evrim geçirerek çok daha büyük kitlelere ulaştığını, virüs worm gibi zararlı kodların yayılmasında etkili rol oynadığı göstermiştir.

Sınıflandırma ve Veri Analiyle web site dedöktörü yapmamızın neden önemli olduğunu, hangi faydaları sağladığını birkaç başlıkta toplamak gerekirse;

- Geçmişe yönelik yapılan analizler yolu ile geleceğin ön görülmesi,
- Mevcut davranış örüntülerinin önceki sonuçlarla kıyaslanması yolu ile gelecekte elde edilmesi istenen durumlar için en doğru stratejilerin üretilmesi,

- Veri analizinin iç görü teknolojisi ile birleştirilerek otomatik optimizasyon ve iyileştirmelerin gerçekleştirilebilmesi,
- Manuel olarak incelenemeyecek kadar büyük veri gruplarının otomatik atama ve analiz yolu ile anlaşılabilir parçalara ayrılması ve pazarlama süreçlerinden maksimum verim alınması mümkündür.

Günümüzde birçok farklı alanda farklı işlev göstersen ve gündelik hayatımızı kolaylaştıran yazılımlar mevcuttur. Bu projen uygulamasının kullanıldığı yazılım alanı ise insanların zararlı yazılımlardan korunması, bilgi güvenliği ve siber ataklara karşı korunmaktır. Bu zamana kadar yapılan çalışmaların incelenmesi sonucu, elde edilen fikirler doğrultusunda kullanılan yazılımların karmaşıklığı oldukça dikkat çekmektedir. Bu konuda gerçekleşecek olan projeden; süreci takip ve kontrol etme konusunda büyük kolaylık ve avantaj getirmesi beklenmektedir.

2 ÖNCEKİ ÇALIŞMALAR

2.1 Veri Nedir?

Bilgi ve veri kavramı genellikle birbiri ile karıştırılmaktadır. Bilgi ve veri ifadeleri çeşitli açılardan açıklanabilir. Genellikle veri ile bilgi arasında farklılık olduğu ve bilgiyi elde etmeye yarayan işlenmemiş ham malzemenin veri olduğu kabul edilir. Öğrenilmek istenilen şeyleri bildikten ve veriyi kullanmaya başladıktan sonra bilgi ifadesi ortaya çıkar. Bilgi şimdi bilinen ve gelecek zamanda verilecek olan kararlar için var olan gerçek bir değerdir ve anlamlı biçimde derlenen ve birleştirilen verilerden oluşur. Bir başka anlamda, bir kaynaktan, bir alıcıya iletilen mesajın içeriğidir ve bu anlamda bilgi, karar verme ile bağlantılıdır ve dolayısıyla veriye göre daha etkin bir kavram olmaktadır.[5]

Şekil 2.1 Veri için Bilgi Hiyerarşisi

2.2 Veri Analizi

Veri analizi, yararlı bilgileri keşfetmek, sonuçları bildirmek ve karar vermeyi desteklemek amacıyla verileri inceleme, temizleme, dönüştürme ve modelleme sürecidir. Veri analizinin birden çok yönü ve yaklaşımı vardır, çeşitli isimler altında farklı teknikleri kapsar ve farklı işletme, bilim ve sosyal bilim alanlarında kullanılır. Günümüz iş dünyasında veri analizi, kararların daha bilimsel verilmesinde ve işletmelerin daha etkin çalışmasına yardımcı olmada rol oynamaktadır.[3]

2.2.1 Veri Analizi Süreci

1. Veri gereksinimleri

Veriler, analizi yönetenlerin (veya analizin bitmiş ürününü kullanacak olan müşterilerin) gereksinimlerine göre belirlenen analiz için girdi olarak gereklidir. Verilerin toplanacağı genel varlık türü, deneysel birim (örneğin, bir kişi veya insan nüfusu) olarak adlandırılır. Bir popülasyona ilişkin spesifik değişkenler (örn. yaş ve gelir) belirlenebilir ve elde edilebilir. Veriler sayısal veya kategorik olabilir (yani sayılar için bir metin etiketi).

2. Veri toplama

Veriler çeşitli kaynaklardan toplanır.Gereksinimler, analistler tarafından verilerin sorumlularına iletilebilir; örneğin, bir kuruluş içindeki Bilgi Teknolojisi personeli.[18] Veriler ayrıca trafik kameraları, uydular, kayıt cihazları vb. dahil olmak üzere ortamdaki sensörlerden de toplanabilir. Ayrıca görüşmeler, çevrimiçi kaynaklardan indirmeler veya belgelerin okunması yoluyla da elde edilebilir.

3. Veri işleme

Ham bilgiyi eyleme dönüştürülebilir zekaya veya bilgiye dönüştürmek için kullanılan istihbarat döngüsünün aşamaları, kavramsal olarak veri analizindeki aşamalara benzer.
Veriler, ilk elde edildiğinde, analiz için işlenmeli veya düzenlenmelidir. Örneğin bunlar, genellikle elektronik tablo veya istatistiksel yazılım kullanılarak daha fazla analiz için verilerin bir tablo formatında (yapılandırılmış veri olarak bilinir) satırlara ve sütunlara yerleştirilmesini içerebilir.

4. Veri temizleme

İşlenip düzenlendiğinde veriler eksik olabilir, kopyalar içerebilir veya hatalar içerebilir. Veri temizleme ihtiyacı, verilerin girilme ve saklanma şeklindeki sorunlardan doğacaktır. Veri temizleme, bu hataların önlenmesi ve düzeltilmesi işlemidir. Ortak görevler arasında kayıt eşleştirme, verilerin yanlışlığının belirlenmesi, mevcut verilerin genel kalitesi, veri tekilleştirme ve sütun segmentasyonu yer alır. Bu tür veri sorunları, çeşitli analitik tekniklerle de tanımlanabilir. Örneğin; finansal bilgilerle, belirli değişkenlerin toplamları, güvenilir olduğuna inanılan ayrı olarak yayınlanan sayılarla karşılaştırılabilir. Önceden belirlenmiş eşiklerin üstünde veya altında olan olağandışı miktarlar da gözden geçirilebilir. Kümedeki veri türüne bağlı olan birkaç tür veri temizleme vardır; bu telefon numaraları, e-posta adresleri, işverenler veya diğer değerler olabilir. Aykırı değer tespiti için nicel veri yöntemleri, yanlış girilme olasılığı daha yüksek görünen verilerden kurtulmak için kullanılabilir. Metinsel veri yazım denetleyicileri, yanlış yazılan sözcüklerin miktarını azaltmak için kullanılabilir. Ancak, kelimelerin kendilerinin doğru olup olmadığını söylemek daha zordur.

5. Keşifsel veri analizi

Veri kümeleri temizlendikten sonra analiz edilebilirler. Analistler, elde edilen verilerde yer alan mesajları anlamaya başlamak için keşifsel veri analizi olarak adlandırılan çeşitli teknikler uygulayabilir. Veri araştırma süreci, ek veri temizliğine veya ek veri taleplerine neden olabilir; bu nedenle, bu bölümün ana paragrafında bahsedilen yinelemeli aşamaların başlatılması.Verilerin anlaşılmasına yardımcı olmak için ortalama veya medyan gibi tanımlayıcı istatistikler oluşturulabilir. Veri görselleştirme aynı zamanda analistin veri içindeki mesajlarla ilgili ek içgörüler elde etmek için verileri grafik formatında inceleyebildiği bir tekniktir.

6. Modelleme ve algoritmalar

Değişkenler arasındaki ilişkileri tanımlamak için verilere matematiksel formüller veya modeller (algoritmalar olarak bilinir) uygulanabilir; örneğin, korelasyon veya nedensellik kullanarak. Genel anlamda, modeller, uygulanan modelin doğruluğuna bağlı olarak bazı artık hatalarla (örneğin, Veri = Model + Hata) veri kümesinde yer alan diğer değişken(ler)e dayalı olarak belirli bir değişkeni değerlendirmek için geliştirilebilir.

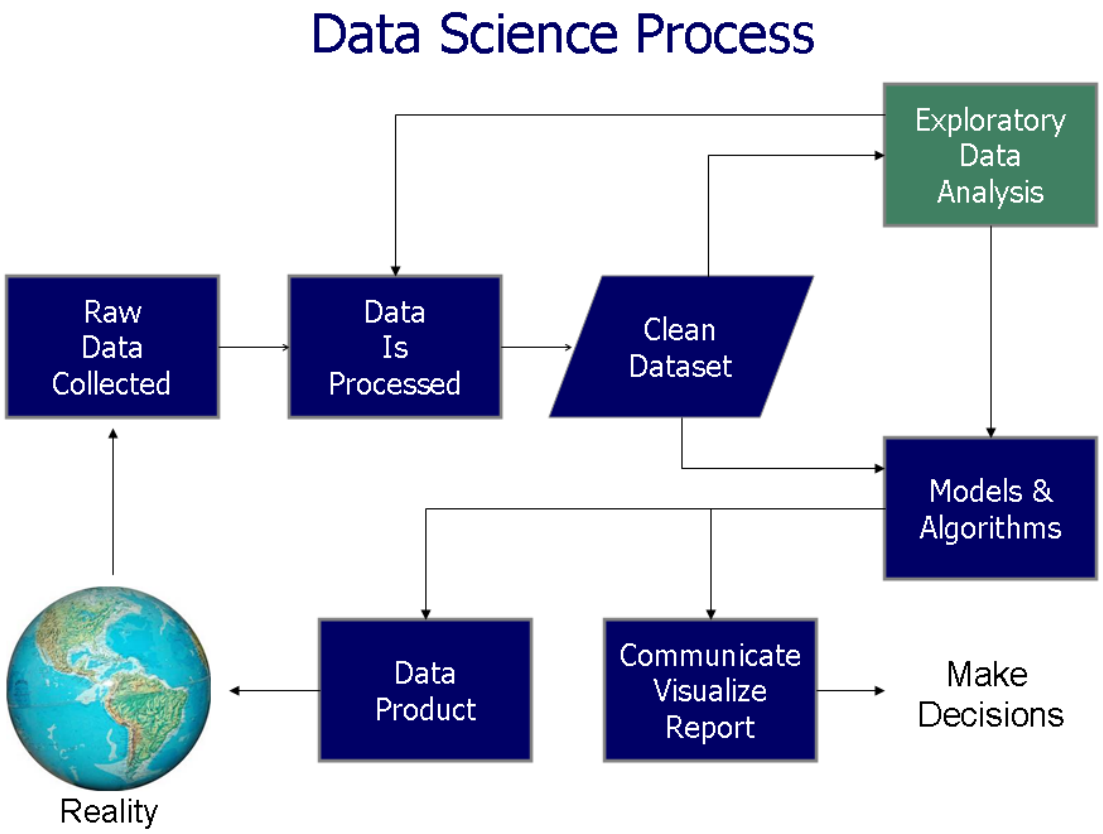
7. Veri ürünü

Bir veri ürünü, veri girdilerini alan ve çıktılar üreten ve bunları çevreye geri besleyen bir bilgisayar uygulamasıdır. Bir modele veya algoritmaya dayalı olabilir. Örneğin, müşterinin satın alma geçmişiyle ilgili verileri analiz eden ve sonuçları müşteriye diğer satın alma işlemlerini önermek için kullanan bir uygulama.

8. İletişim

Veriler analiz edildikten sonra, gereksinimlerini desteklemek için analiz kullanıcılarına birçok formatta rapor edilebilir. Kullanıcılar, ek analizle sonuçlanan geri bildirim alabilir. Bu nedenle, analitik döngünün çoğu yinelemelidir.

Analist, sonuçların nasıl iletileceğini belirlerken, mesajın izleyiciye daha açık ve verimli bir şekilde iletilmesine yardımcı olmak için çeşitli veri görselleştirme tekniklerini uygulamayı düşünebilir.[45] Veri görselleştirme, verilerde yer alan önemli mesajların iletilmesine yardımcı olmak için bilgi ekranlarını (tablolar ve çizelgeler gibi grafikler) kullanır. Tablolar, bir kullanıcının belirli sayıları sorgulamasını ve bunlara odaklanmasını sağlayarak değerli bir araçtır; grafikler (örneğin, çubuk grafikler veya çizgi grafikler), verilerde yer alan nicel mesajları açıklamaya yardımcı olabilir.

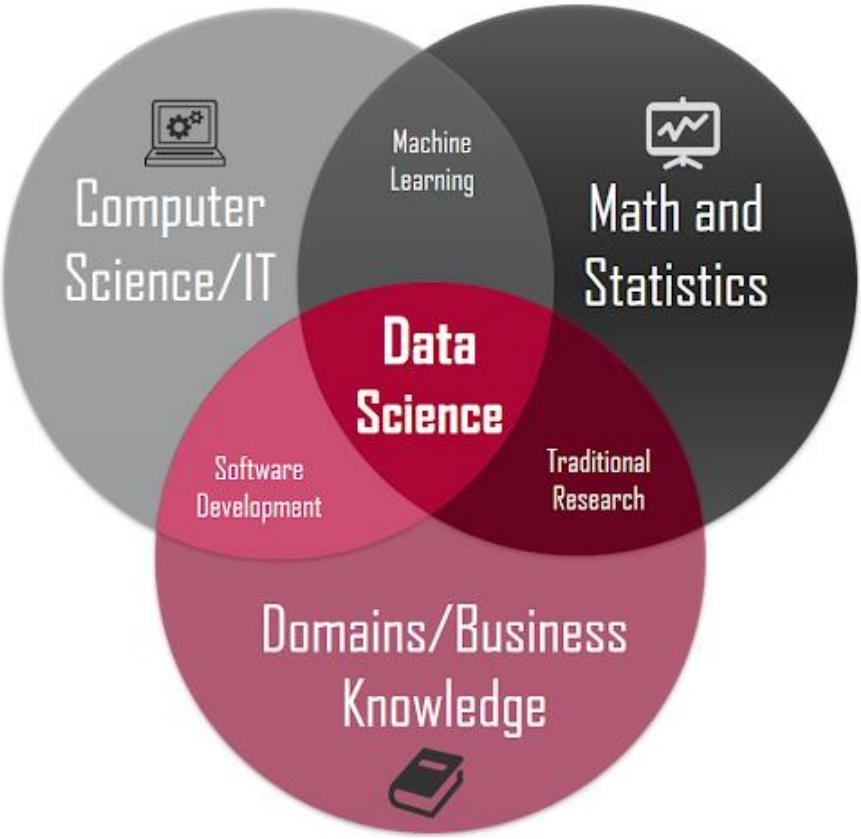


Şekil 2.2- Veri Analiz Süreci

2.3 Veri Bilimi

Veri bilimi, bir dizi ilkeyi, çeşitli algoritmaları, olayları ve büyük veri kümelerinden gelen kullanışlı kalıpları ayıklamak için gerekli süreçleri kapsamaktadır. Bununla birlikte veri bilimi, bu süreçlerde; veri analizini, istatistikleri, makine öğrenmesi ve veri madenciliği gibi alanları ve bunlarla ilgili birçok yöntemi birleştirmek için kullanılan bir kavram olarak belirtilir.

Veri bilimi, makine öğrenmesi ve veri madenciliği kavramları sıklıkla birbirleri yerine kullanılmaktadır. Bu disiplinler arasındaki ortaklık, verilerin analizi yoluyla karar vermenin iyileştirilmesini sağlamaktır. Veri bilimi bu alanlardan beslenmekle birlikte, daha geniş bir kapsama alanına sahiptir. Makine öğrenmesi, veriden örüntü çıkarma algoritmalarının tasarımı ve değerlendirmesine de odaklanır. Veri madenciliği genellikle yapılandırılmış verilerin analizi ile ilgilenir ve ticari uygulamalara vurgu yapar. Veri bilimi ise, tüm bu hususları dikkate almaktadır.



Şekil 2.3- Veri Bilimi

2.4 Veri Tabanı

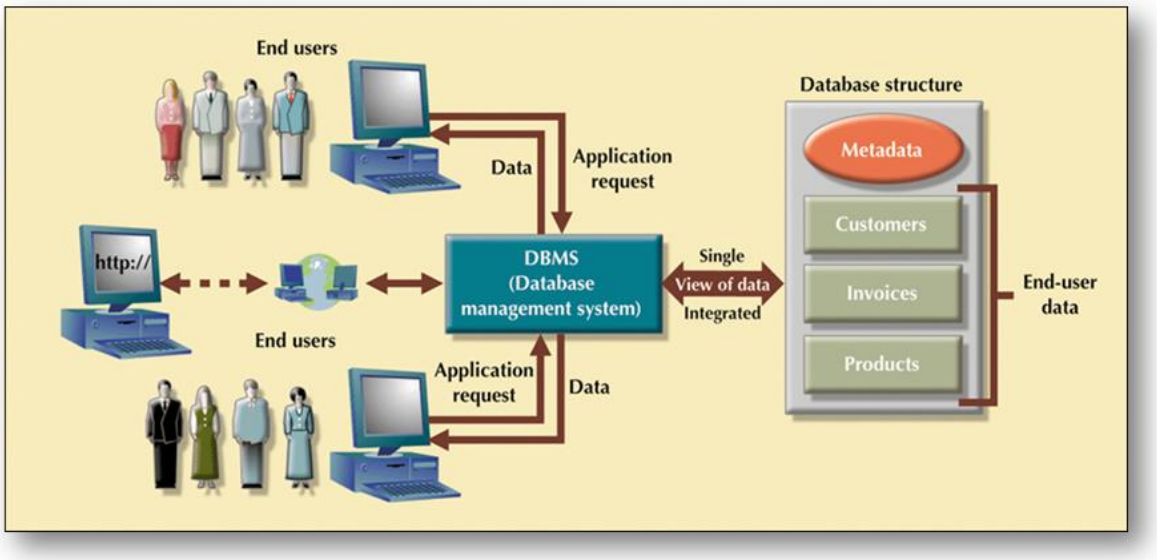
Veri tabanı, birbirleriyle ilişkili olan verilerin tutulduğu, kullanım amacına uygun olarak düzenlenmiş veriler topluluğunun mantıksal ve fiziksel olarak tanımlarının olduğu bilgi depolarıdır. Veri tabanları gerçekte var olan ve birbirleriyle ilişkileri olan bilgilerden oluşur. Veri tabanı, birbirleriyle ilişkili verilerin tekrara yer vermeden, çok amaçlı kullanımına olanak sağlayacak şekilde depolanmasını sağlayan yazılımdır. Kısaca, veri tabanı depolanan bilgiyi verimli ve hızlı bir şekilde kullanan yazılımdır.

Belirli bir konu hakkında toplanmış veriler bir veri tabanı programı altında toplanırlar. Bu verilerden istenildiğinde; toplanılan bilgilerin tümü veya istenilen özelliklere uyanları görüntülenebilir, yazdırılabilir ve hatta bu bilgilerden yeni bilgiler üretilerek bunlar çeşitli amaçlarla kullanılabilir [5].

Şekil 2.4- Veri Tabanını Simgeleyen Görsel

2.5 Veri Tabanı Yönetim Sistemi (VTYS)

Veri tabanı yönetim sistemi, yeni bir veri tabanı oluşturmak, veri tabanı düzenlemek, geliştirmek, bakımını yapmak, yedeğini almak, performansına bakmak gibi pek çok işlemlerin gerçekleştirildiği birden fazla programdan oluşan bir yazılım sistemidir. Veri tabanı yönetim sistemi, kullanıcı ile veri tabanı arasında bir ara birim oluşturur ve veri tabanına her türlü erişimi sağlar. [5]



Şekil 2.5 Veri Tabanı Yönetim Sistemi Çalışma Modeli

2.6 Veri Analitiği

Dünyanın her yerindeki işletmeler, işlem verileri, günlük dosyaları, web sunucuları ve müşteriyle ilgili diğer farklı veriler biçiminde büyük miktarda günlük veri oluşturur. Bu bilgilerin dışında her gün sosyal medya kullanıcıları tarafından da çok büyük miktarda verinin üretildiğini bilmeliyiz. Genellikle şirketler, oluşturulan tüm verilerden bir değer yaratmak ve bu nedenle etkin iş kararları almak için yararlanmaya ihtiyaç duyarlar. Başka bir deyişle, veri analitiğinin büyük veri kümelerini keşfetme ve ardından analiz etme yöntemi olduğu söylenebilir. Bunlar, kalıpları, gizli eğilimleri toplamak, yeni korelasyonları keşfetmek ve değerli iç görülerle bir sonuca varmak için kullanılır. Bu da insanların iş tahminleri yapmak

için stratejik iş kararları almalarına izin vererek hızı ve iş verimliliğini artırır. Bunun iş temposunu ve verimliliğini arttırdığı söylenebilir.[2]

2.6.1 Veri Analitiği Nasıl Kullanılır?

- Verimli İşlemler Yapın: Veri analitiği sayesinde süreçlerinizi kolayca takip edebilir, üretimi artırabilir ve tasarruf edebilirsiniz. Hedef kitlenin arzusuna yönelik gelişmiş anlayışla birlikte, farklı içeriklerin yanı sıra reklam oluşturmak için daha az zaman harcarsınız.
- Müşteri Hizmetlerinin kalitesini artırın: Veri analitiğini kullanarak müşteri hizmetlerinizi onların ihtiyaçlarına göre ayarlayabilirsiniz. Dahası, veri analitiği, müşterilerle daha güçlü ilişkiler kurmanın yanı sıra kişiselleştirme sunar. Ayrıca, analiz edilen veri setleri, müşterilerin ilgi alanları, endişeleri ve daha fazlası hakkında daha fazla bilgi sağlayabilir. Diğer bir deyişle, ürün ve hizmetler için daha detaylı ve hedefe yönelik öneriler vermenizi sağlar.
- Gelişmiş Karar Verme: Veri Analitiği, tahmine dayalı kararlardan ve manuel görevlerden kurtulmanıza yardımcı olur. Böylece şirketler ve kuruluşlar, doğru kararlar almak için bu süreçten elde ettikleri bilgileri pratik olarak kullanabilirler ve bu da daha iyi sonuçların yanı sıra müşteri memnuniyetini de beraberinde getirir.
- Etkili Pazarlama: Veri analitiği değerli iç görüler hakkında bilgi verebileceğinden, onu optimum sonuçlara ulaşmak için kullanabilirsiniz.

2.6.2 Veri analitiği süreci, takip edilmesi gereken alt adımları içerir:

- Problemi anlayın: İş konusunu anlamak, organizasyonel hedefleri belirlemek ve stratejik çözümü planlamak, analitik sürecindeki ilk adım olmalıdır. Kuruluşlar

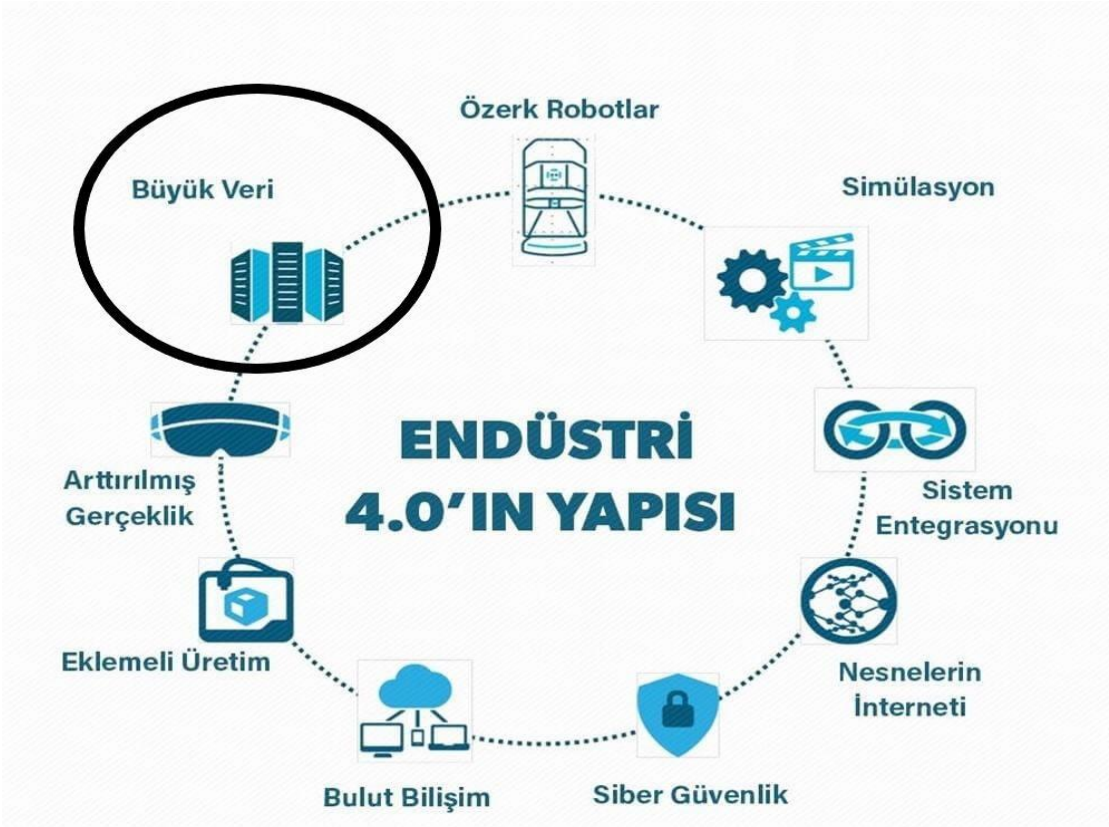
genellikle ürünlerin iadesini tahmin etme, uygun ürün önerileri verme, siparişlerin iptali, sahtekarlıkları belirleme ve araç rotasını optimize etme gibi sorunlarla karşılaşmaktadır.

- Verilerin Toplanması: Verilerin toplanması ikinci adımdır. Son birkaç yıla ait işlemsel iş verilerini ve müşteriyle ilgili bilgileri toplamak önemlidir. Geçmiş veriler, işletmenin geleceğini şekillendirmede önemli bir rol oynar.
- Verilerin Temizlenmesi: Topladığınız tüm veriler sıklıkla düzensiz ve dağınık olacaktır. Bu nedenle, bunun gibi verilerin veri analizi yapmak için kullanılması uygun değildir. Yapmanız gereken, istenmeyen, fazlalık ve eksik değerlerden kurtulmak için verileri temizlemek olmalıdır. Bu onu analize hazır hale getirir.
- Veri Keşfi ve Analizi: Son adımdan önceki adım, keşifsel veri analizini gerçekleştirmektir. Bunu yapmak için, bu verileri analiz etmek, görselleştirmek ve gelecekteki sonuçları tahmin etmek için veri görselleştirme ve iş zekâsı araçlarından, veri madenciliği tekniklerinden ve tahmine dayalı modellemeden yararlanabilirsiniz. Bu yöntemlerden yararlanarak, belirli bir özelliğin diğer değişkenlere göre etkisini ve ilişkisini görebilirsiniz.
- Sonuçları anlamlandırın: Son adım, sonuçları yorumlamak ve sonuçların beklentilerinizi karşılayıp karşılamadığını onaylamaktır. Bunun dışında, içgörü kazanmanıza yardımcı olacak görünmeyen kalıpları ve gelecekteki eğilimleri öğrenebilirsiniz. Bu iç görüler, ilgili veriye dayalı karar verme konusunda sizi güçlendirecektir. [6]

2.7 Endüstri 4.0 ve Veri Analitiği

Son yıllarda, Dördüncü Sanayi Devrimi veya Akıllı Üretim olarak da bilinen Endüstri 4.0 (I4.0) kavram ve teknolojilerinin tanıtılmasıyla endüstriyel ortam kökten değişti. Bilgi ve iletişim teknolojilerini (BİT) ve ileri endüstriyel teknolojileri siber-fiziksel sistemlere dönüştürerek dijital, akıllı ve sürdürülebilir bir işletme yaratmayı hedefliyor. Ana I4.0'ın anlamı, ürünleri, makineleri ve insanları dış çevre ile birbirine bağlamakta ve üretim, bilgi teknolojisi ve interneti birleştirmede yatmaktadır. [7]

Veri analitiği ile Endüstri 4.0 teknolojisini kullanan firmalara doğru bilgi aktarımı sağlanarak firmalar phishing saldırılarından korunarak sektöre dair risk en aza indirgenmiş olur.



Şekil 2.6 Endüstri 4.0'ın Yapısı

2.8 IoT ve Veri Analitiği

Nesnelerin İnterneti (IoT), fiziksel nesnelerin, binaların, araçların ("bağlı cihazlar" veya "akıllı cihazlar" olarak da anılır) ve gömülü elektronik, yazılım, sensörler ve bağlantıya sahip diğer şeylerin (nesnelerin birbirine bağlanmasına izin veren) birbirine bağlanmasıdır. 2013 yılında, Nesnelerin İnterneti Üzerindeki Küresel Standartlar Girişimi (IoT-GSI), IoT'yi "geliştirme aşamasındaki mevcut ve birlikte çalışabilir bilgi ve iletişim teknolojilerine dayalı olarak gelişmiş ağ hizmetleri (fiziksel ve sanal) sağlayan küresel bir bilgi toplumu altyapısı"

olarak tanımladı. Bu amaçla “şey” terimi, fiziksel dünyanın bir nesnesi veya tanımlanabilen ve iletişim ağlarına entegre edilebilen bir kelimedir (sanal şeyler). IoT, nesnelere mevcut ağ altyapısı aracılığıyla uzaktan erişilmesine ve kontrol edilmesine izin vererek, fiziksel dünya ve bilgisayar sistemlerinin daha doğrudan entegrasyonu için fırsatlar yaratarak, azaltılmış insan müdahalesi ile artan verimlilik, doğruluk ve ekonomik faydalar sağlar. IoT, sensörler ve sürücüler tarafından arttığında, teknoloji, akıllı şebekeler, sanal enerji santralleri, akıllı evler, akıllı ulaşım ve akıllı şehirler gibi teknolojileri de içeren daha genel bir siber fiziksel sistem sınıfının bir anında haline geldi. Her şey gömülü bir bilgisayar sistemi aracılığıyla benzersiz bir şekilde tanımlanabilir ve her şey mevcut İnternet altyapısı içinde birlikte çalışabilir.

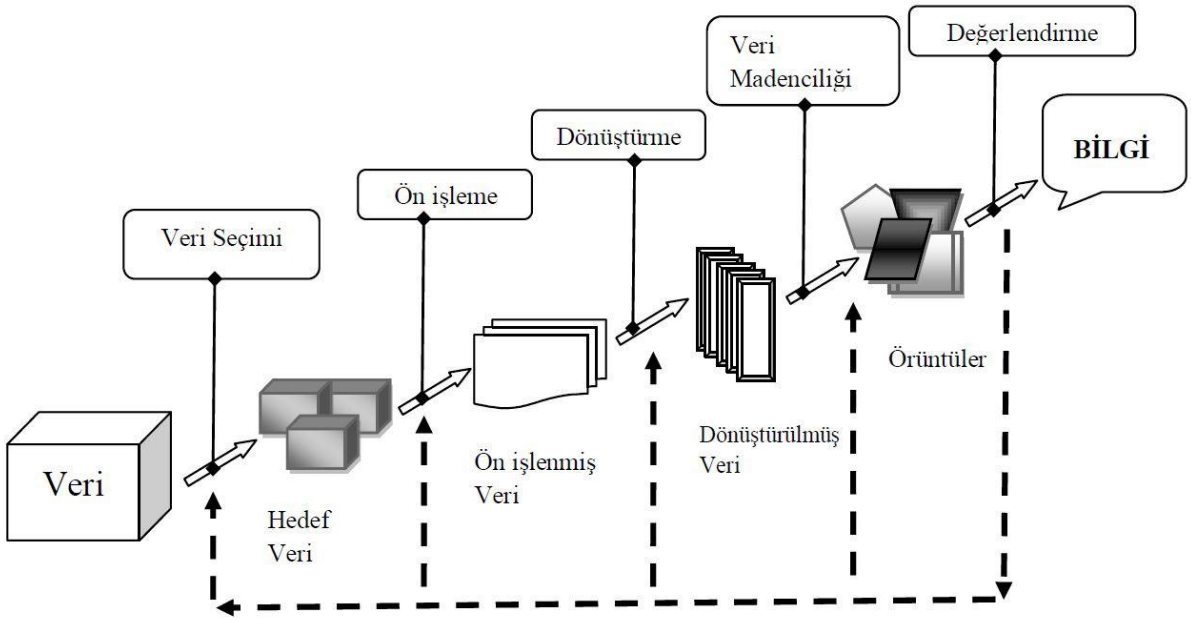
'Nesnelerin İnterneti' terimi 1999'da Kevin Ashton tarafından önerildi. İnternet kavramı ilk olarak MIT'deki Otomatik Kimlik Merkezi Merkezi aracılığıyla pazarla bağlantılı olarak popüler hale getirildi, ancak kavram 1991'den beri tartışılıyor. Radyo Frekansı Tanımlama (RFID) ve yayın analizi, ilk zamanlarda Nesnelerin İnterneti için bir ön koşul olarak görülüyordu. Günlük yaşamdaki tüm nesneler ve insanlar tanımlayıcılarla donatılmış olsaydı, bir bilgisayarda saklanırdı. RFID kullanımına ek olarak, yaklaşık iletişim alanları, barkod, QR kod ve dijital filigran gibi teknolojiler aracılığıyla etiketleme yapılabilmektedir.[8]

IoT teriminin proje açısından anlamı şudur: Oldukça geniş bir pazardan elde edilen binlerce terabaytlık kullanıcı verisinin, bilgisayarlar yardımı ile veri analitiği ile işlenebilmesi ve kullanıcı ihtiyaçlarının ne yönde olduğunun, hangi sitenin kötü amaçlı bir site olduğunun tespit edilmesidir.

Şekil 2.7 IoT Modeli

2.9 Veri Madenciliği

Veri miktarlarındaki artışla beraber, verilerden anlamlı bilgilerin elde edilmesini sağlayacak analiz tekniklerini içeren veri madenciliği kavramı ortaya çıkmıştır. Kavram literatürde, veri tabanlarından bilgi keşfi (knowledge discovery from databases) gelişmiş veri analizi (advanced data analysis), bilgi madenciliği (knowledge mining) ve makine öğrenmesi (machine learning) gibi adlarla tanımlanmaktadır. Gelişimi 1980'li yıllara dayanan makine öğrenmesi, yapay zekânın bir alt kümesi olarak insan beyninin çalışma prensibini taklit ederek, verilerden anlam çıkarmayı sağlayan teknikler bütünüdür. Bilinen istatistiksel ve ekonometrik yöntemlerle veri analizinde amaç; tahmin, özetleme, değerlendirme/keşif ve hipotez testi olmakla birlikte veri madenciliğinde tahmin, veri özetleme ve veriden farklı örüntüler çıkarma temel amaçlardandır ve analiz dahilinde doğru model bilinmemekle birlikte araştırmanın amacı doğru modeli keşfetmektir.[9]



Şekil 2.8 Veri Madenciliği Süreçleri

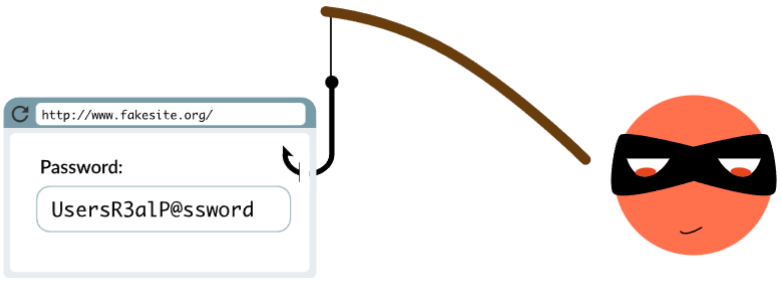
2.10 Phishing

Kimlik avı, kandırmaya çalışan e-postayı silah olarak kullanan bir siber saldırıdır. Amaç, e-posta alıcısını, mesajın istediği veya ihtiyaç duyduğu bir şey (örneğin bankalarından gelen bir talep veya şirketlerinden birinden gelen bir not) olduğuna inandırmak ve bir bağlantıya tıklamak veya bir ek indirmek için kandırmaktır.

Kimlik avını gerçekten ayırt eden şey, mesajın aldığı biçimdir: saldırganlar, bir tür güvenilir varlık, genellikle gerçek veya makul bir şekilde gerçek bir kişi veya kurbanın iş yapabileceği bir şirket gibi davranır. 1990'lara dayanan en eski siber saldırı türlerinden biridir

ve kimlik avı mesajları ve teknikleri giderek daha karmaşık hale geldiğinden hala en yaygın ve zararlı olanlardan biridir.

"Phish", tıpkı yazıldığı gibi telaffuz edilir, yani "fish"(balık) kelimesi gibi. Analojisi, bir olta balıkçısının oraya yemli bir kanca fırlatmasına (kimlik avı e-postası) ve denizdeki bir balığın(sizin) ısırmasını ummasına benzer. Terim, 1990'ların ortalarında, AOL kullanıcılarını giriş bilgilerinden vazgeçmeleri için kandırmayı amaçlayan bilgisayar korsanları arasında ortaya çıktı. "ph", tuhaf bir bilgisayar korsanı heceleme geleneğinin bir parçasıdır ve muhtemelen "telefon phreaking" in kısaltması olan "phreaking" teriminden etkilenmiştir; bu, ücretsiz telefon görüşmeleri almak için telefon ahizelerine ses tonlarını çalmayı içeren erken bir bilgisayar korsanlığı biçimidir.[10]



2019 Verizon Veri İhlali Araştırmaları Raporuna göre, geçen yılki tüm ihlallerin yaklaşık üçte biri kimlik avını içeriyordu. Siber casusluk saldırıları için bu sayı %78'e çıkıyor. 2019 için en kötü kimlik avı haberi, iyi üretilmiş, kullanıma hazır araçlar ve şablonlar sayesinde faillerinin çok, çok daha iyi hale gelmesidir.[11]

Bazı kimlik avı dolandırıcılıkları, büyük ses getirecek kadar başarılı oldu:

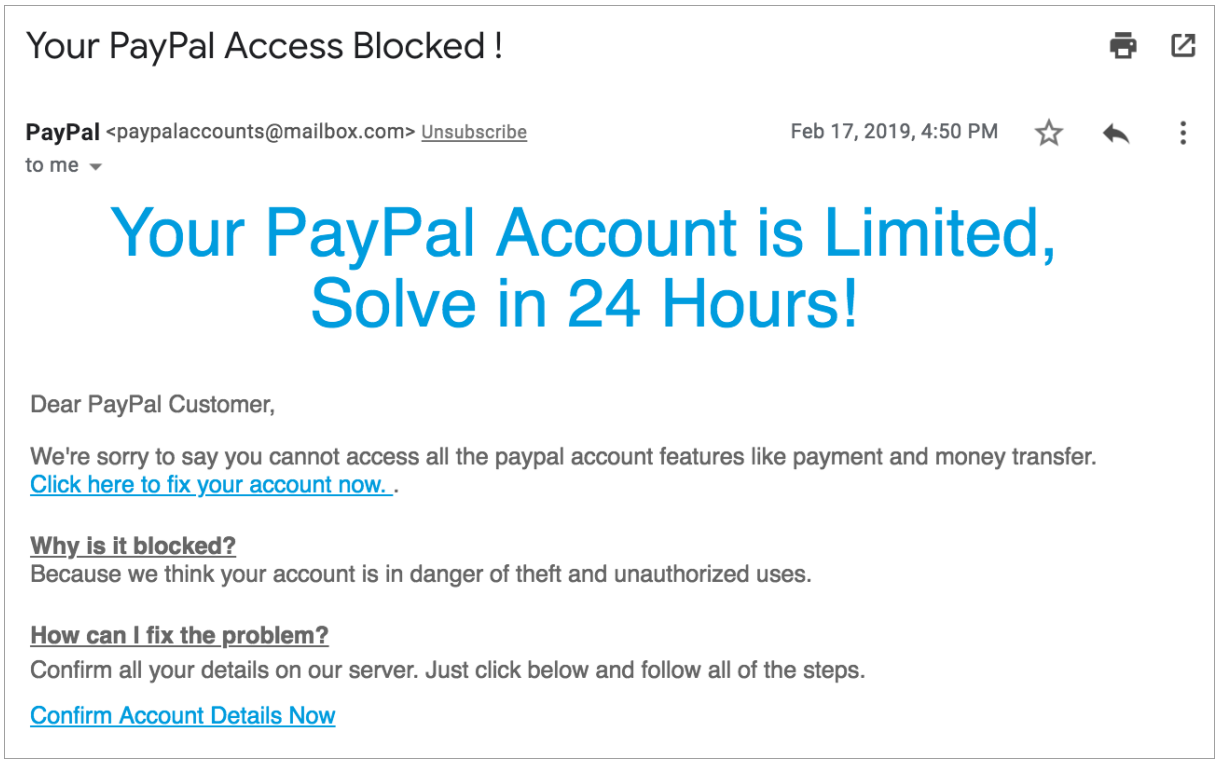
- 2016 yılında, Kansas Üniversitesi çalışanları bir kimlik avı e-postasına yanıt verdi ve maaş çeki depozito bilgilerine erişim vererek maaşlarını kaybetmelerine neden oldu.[10]

- Belki de tarihteki en önemli kimlik avı saldırılarından biri, bilgisayar korsanlarının Hillary Clinton kampanya başkanı John Podesta'nın Gmail şifresini sunmasını sağlamayı başardığı 2016 yılında gerçekleşti.[12]

- Bir dizi ünlünün iCloud fotoğraflarının halka açıklandığı "fappening" saldırısının, başlangıçta Apple'ın iCloud sunucularındaki güvensizliğin bir sonucu olduğu düşünülüyordu, ancak aslında bir dizi başarılı kimlik avı girişiminin ürünüydü.[13]

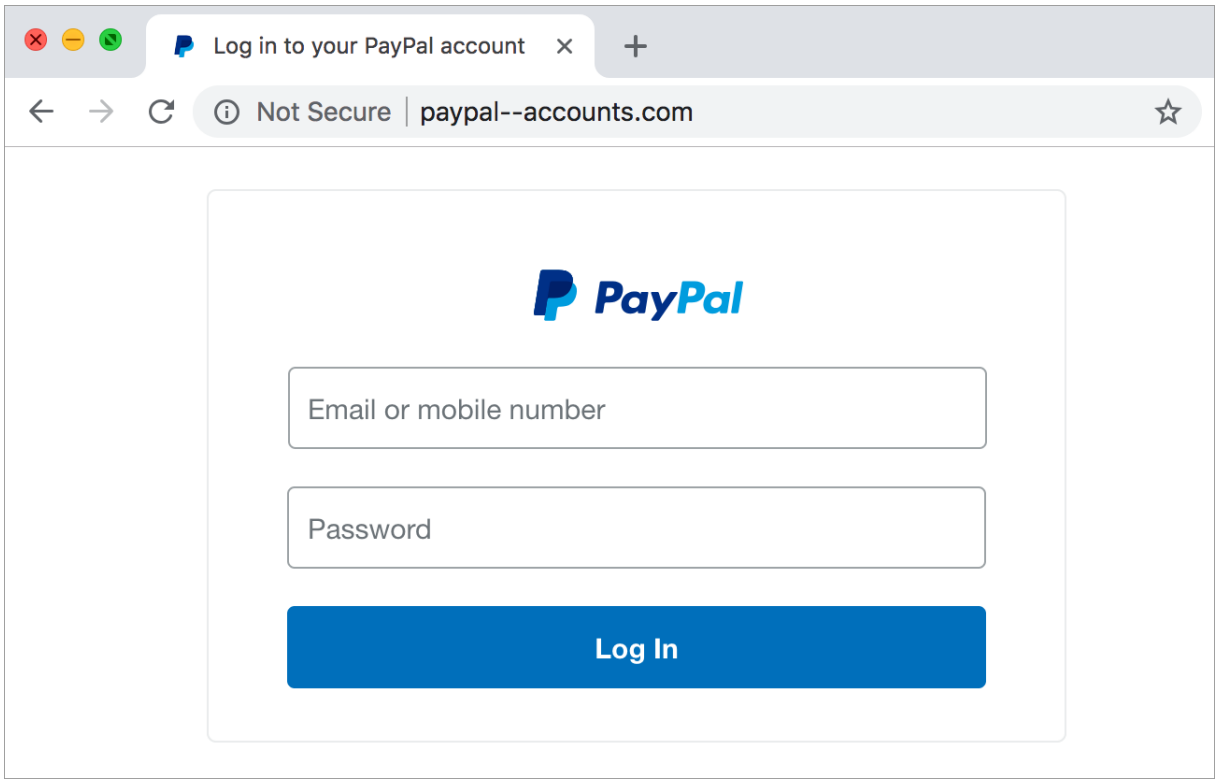
2.10.1 Örnek Bir Saldırı

Kimlik avı saldırısı, genellikle bir bankacılık web sitesi veya çevrimiçi mağaza gibi meşru bir web sitesinden geldiğini iddia eden bir e-postayla başlar:



Kimlik avı e-postasının ekran görüntüsü. Konu satırında "PayPal Erişiminiz Engellendi!" yazıyor. E-posta, "PayPal paypalaccounts@mailbox.com " adresinden gelmektedir . PayPal'dan geldiğini iddia eden bu e-posta başlığında "PayPal Hesabınız Sınırlıdır, 24 Saatte Çözün!" ve mailde "Değerli PayPal Müşterisi, ödeme ve para transferi gibi tüm paypal hesabı özelliklerine erişemediğinizi üzülerek bildiririz. Hesabınızı şimdi düzeltmek için buraya tıklayın. Aşağıya tıklayın ve tüm adımları izleyin. Hesap Ayrıntılarını Şimdi Onaylayın." yazmaktadır.

E-postanın amacı, kullanıcıdan özel veriler elde etmektir, bu nedenle alıcıdan kişisel bilgilerle yanıt vermesini ister veya orijinal siteye oldukça benzeyen bir web sitesine bağlantı verir:



Bir kimlik avı web sitesinin ekran görüntüsü. Web tarayıcısı "PayPal hesabınızda oturum açın" web sayfası başlığını gösteriyor. Adres çubuğu "paypal--accounts.com" gösteriyor. Ekranın ana alanı, PayPal logolu oturum açma kutusunu içerir: e-posta veya cep telefonu numarası için bir giriş alanı, bir şifre giriş alanı ve bir "Oturum aç" düğmesi bulunmaktadır.

PayPal giriş ekranı olduğunu iddia eden bu web sitesinde kullanıcı ikna olur ve siteye özel bilgileri girerse, bu veriler artık saldırganın elindedir. Eğer kullanıcı giriş bilgilerini girdiyse, saldırgan bu bilgileri gerçek web sitesine giriş yapmak için kullanabilir veya kullanıcı kredi kartı bilgilerini verdiyse, kredi kartını herhangi bir yerde alışveriş yapmak için kullanabilir.[14]

3 MATERYAL VE METOTLAR

3.1 KULLANILACAK MATERYALLER

3.1.1 Anaconda

Anaconda ücretsiz ve açık kaynaklı, Python ve R programlama dillerinin bilimsel hesaplama kullanımında paket yönetimini kolaylaştırmayı amaçlayan açık kaynaklı bir programdır. Anaconda programı Windows,Linux ve MacOS işletim sistemlerinde kullanılabilen binlerce veri bilim paketi ve kitaplığı içerir. Anaconda, 1.500'den fazla paketin yanı sıra conda paketi ve sanal çevre yöneticisiyle birlikte gelir. Ayrıca komut satırı arabirimine (CLI) grafiksel bir alternatif olarak bir GUI, Anaconda Navigator içerir. Bir program çalıştırmak için birçok bilimsel paket, diğer paketlerin belirli sürümlerine bağlıdır. Veri bilimcileri genellikle birçok paketin birden çok sürümünü kullanır ve bu farklı sürümleri ayırmak için birden çok ortam kullanır. Komut satırı programı conda, hem paket yöneticisi hem de ortam yöneticisidir. Bu, veri bilimcilerin her paketin her sürümünün ihtiyaç duyduğu tüm bağımlılıklara sahip olduğundan ve doğru çalıştığından emin olmasına yardımcı olur. Anaconda Navigator, bir terminal penceresinde conda komutları yazmaya gerek kalmadan paketler ve ortamlarla çalışmanın kolay, "point and click" yöntemidir. Anaconda Navigator istenilen paketleri bulmak, bir ortama kurmak, paketleri çalıştırmak ve güncellemek için kullanılabilir.[15]

3.1.2 Python

Python, nesne yönelimli üst düzey bir programlama dilidir. Dinamik yazma ve dinamik bağlama ile birleştirilmiş yüksek düzeyde yerleşik veri yapıları, onu uygulama geliştirme için ve ayrıca mevcut bileşenleri birbirine bağlamak için çok uygun bir programlama dilidir. Python basit, öğrenmesi kolay sözdizimi okunabilirliğine sahiptir ve bu nedenle program bakımı daha kolaydır. Python, program modülerliğini ve kodun yeniden kullanımını teşvik eden modülleri ve paketleri destekler. Python yorumlayıcısı ve kapsamlı standart kitaplık, tüm büyük platformlar için ücretsiz olarak mevcuttur ve ücretsiz olarak dağıtılabilir.

Programcılar, sağladığı artan üretkenlik nedeniyle genellikle Python'dan memnun kalırlar. Derleme adımı olmadığından, düzenleme-test-hata ayıklama döngüsü inanılmaz derecede hızlıdır. Python programlarında hata ayıklamak kolaydır: bir hata veya hatalı giriş asla bir segmentasyon hatasına neden olmaz. Bunun yerine, yorumlayıcı bir hata keşfettiğinde bir istisna oluşturur. Program istisnayı yakalamadığında, yorumlayıcı bir yığın izi yazdırır. Kaynak düzeyinde bir hata ayıklayıcı, yerel ve global değişkenlerin incelenmesine, rastgele ifadelerin değerlendirilmesine, kesme noktalarının ayarlanmasına, kodda bir seferde bir satır adım adım ilerlemeye vb. izin verir. Hata ayıklayıcı, Python'un iç gözlem gücüne tanıklık ederek Python'un kendisinde yazılmıştır. Öte yandan, genellikle bir programda hata ayıklamanın en hızlı yolu, kaynağa birkaç yazdırma ifadesi eklemektir: hızlı düzenleme-test-hata ayıklama döngüsü bu basit yaklaşımı çok etkili kılar.[16].

Python programlama dilini neden tercih ettiğimizi iki başlıkta toplamak gerekirse;

- Python, hem basitliğe hem de okunabilirliğe odaklanırken, aynı zamanda veri analizleri için çok sayıda yararlı seçenek sunar. Yalnızca birkaç satır kodla karmaşık senaryolar için etkili çözümler oluşturmak için nispeten basit sözdizimini kolayca kullanılabilir.

- Python hızlıca uygulanabilir, bu da makine öğrenimi alanında çalışan mühendislerin bir fikri hemen doğrulamasına yardımcı olur. Python'un makine öğrenimi için tercih edilen dil olmasının ana nedenlerinden biri, birçok kütüphaneye erişimidir.

3.1.3 Pandas

Pandas, veri analizi için bir Python kütüphanesidir. 2008 yılında Wes McKinney tarafından güçlü ve esnek bir nicel analiz aracına duyulan ihtiyaçtan yola çıkılarak başlatılan pandas, en popüler Python kitaplıklarından biri haline geldi. Son derece aktif bir katkıda bulunanlar topluluğuna sahiptir.

Pandas, veri görselleştirme için matplotlib ve matematiksel işlemler için NumPy olmak üzere iki temel Python kitaplığının üzerine inşa edilmiştir. Pandas, bu kitaplıklar üzerinde bir sarmalayıcı görevi görerek, matplotlib'in ve NumPy'nin yöntemlerinin çoğuna daha az kodla erişmenizi sağlar. Örneğin, pandas' .plot() işlevi, birden çok matplotlib yöntemini tek bir yöntemde birleştirerek birkaç satırda bir grafik çizmenize olanak tanır.

Pandas önce, çoğu analist veri toplama ve hazırlama için Python'u kullandı ve ardından iş akışlarının geri kalanı için R gibi daha alana özgü bir dile geçti. Pandas, analitik görevleri kolaylaştıran ve araçları değiştirme ihtiyacını ortadan kaldıran verileri depolamak için iki yeni nesne türü tanıttı: Liste benzeri bir yapıya sahip olan Seriler ve tablo şeklinde bir yapıya sahip olan DataFrame'ler.[17]

3.1.4 NumPy

NumPy, Python'da bilimsel hesaplama için temel pakettir. Çok boyutlu bir dizi nesnesi, çeşitli türetilmiş nesneler (maskelenmiş diziler ve matrisler gibi) ve diziler üzerinde matematiksel, mantıksal, şekil işleme, sıralama, seçme, I/O, ayrık Fourier dönüşümleri, temel lineer cebir, temel istatistiksel işlemler, rastgele simülasyon ve çok daha fazlası dahil olmak üzere hızlı işlemler için çeşitli rutinler sağlayan bir Python kitaplığıdır.

NumPy paketinin merkezinde ndarray nesnesi bulunur. Bu, performans için derlenmiş kodda gerçekleştirilen birçok işlemle birlikte, homojen veri türlerinin n-boyutlu dizilerini kapsar. NumPy dizileri ile standart Python dizileri arasında birkaç önemli fark vardır:

- NumPy dizileri, Python listelerinin (dinamik olarak büyüyebilen) aksine, oluşturma sırasında sabit bir boyuta sahiptir. Bir ndarray'in boyutunu değiştirmek, yeni bir dizi oluşturacak ve orijinali silecektir.

- NumPy dizisindeki öğelerin hepsinin aynı veri türünde olması gerekir ve bu nedenle bellekte aynı boyutta olacaktır. İstisna: Bir nesne(Python, NumPy dahil) dizilerine sahip olabilir, böylece farklı büyüklükteki elemanların dizilerine izin verilir.

- NumPy dizileri, çok sayıda veri üzerinde gelişmiş matematiksel ve diğer türdeki işlemleri kolaylaştırır. Bu tür işlemler Python'un yerleşik dizileri kullanılarak mümkün olandan daha verimli ve daha az kodla yürütülür.

- Sayıları giderek artan bilimsel ve matematiksel Python tabanlı paketler NumPy dizilerini kullanıyor; Bunlar tipik olarak Python dizisi girdisini desteklese de, bu girdileri işlemeden önce NumPy dizilerine dönüştürürler ve genellikle NumPy dizilerinin çıktısını alırlar. Başka bir deyişle, günümüzün bilimsel/matematiksel Python tabanlı yazılımlarının çoğunu (hatta belki de çoğunu) verimli bir şekilde kullanmak için, yalnızca Python'un yerleşik dizi türlerini nasıl kullanacağınızı bilmek yetersizdir. Ayrıca NumPy dizilerinin nasıl kullanılacağını da bilmek gerekir.[18]

3.1.5 Matplotlib ve Pyplot

Matplotlib, Python ve onun sayısal uzantısı NumPy için bir çapraz platform, veri görselleştirme ve grafik çizim kitaplığıdır. Bu nedenle, MATLAB'a uygun bir açık kaynak alternatifi sunar. Geliştiriciler, grafikleri GUI uygulamalarına yerleştirmek için matplotlib'in API'lerini (Uygulama Programlama Arayüzleri) de kullanabilir. Bir Python matplotlib script'i, çoğu durumda bir görsel veri grafiği oluşturmak için gereken tek şey birkaç kod satırı olacak şekilde yapılandırılmıştır. matplotlib komut dosyası katmanı iki API'yi kaplar:

- Pyplot API, matplotlib.pyplot tarafından tepesinde bulunan Python kod nesnelerinin bir hiyerarşisidir.
- Pyplot'tan daha fazla esneklikle birleştirilebilen bir "Nesneye Yönelik" API nesneleri koleksiyonu. Bu API, Matplotlib'in arka uç katmanlarına doğrudan erişim sağlar.

Pyplot API, uygun bir MATLAB tarzı durum bilgisi içeren ara yüze sahiptir. Aslında matplotlib, başlangıçta MATLAB için bir açık kaynak alternatifi olarak yazılmıştır. Nesneye Yönelik API ve ara yüzü, pyplot'tan daha özelleştirilebilir ve güçlüdür, ancak kullanımı daha zor kabul edilir. Sonuç olarak, pyplot arabirimi daha yaygın olarak kullanılır. Matplotlib'in pyplot API'sini anlamak, grafiklerle nasıl çalışılacağını anlamanın anahtarıdır:

- matplotlib.pyplot.figure: "Figure" en üst düzey kapsayıcıdır. Bir veya daha fazla eksen dahil olmak üzere bir çizimde görselleştirilen her şeyi içerir.
- matplotlib.pyplot.axes: Eksenler, bir çizimdeki öğelerin çoğunu içerir: Axis, Tick, Line2D, Text, vb. ve koordinatları ayarlar. Verilerin çizildiği alandır. Eksenler, X Ekseni, Y Ekseni ve muhtemelen bir Z Ekseni de içerir.[19]

3.1.6 Scikit-Learn

Scikit-learn (Sklearn), Python'da makine öğrenimi için en kullanışlı ve sağlam kütüphanedir. Python'da bir tutarlılık ara yüzü aracılığıyla sınıflandırma, regresyon, kümeleme ve boyutsallık azaltma dahil olmak üzere makine öğrenimi ve istatistiksel modelleme için bir dizi verimli araç sağlar. Büyük ölçüde Python ile yazılan bu kütüphane NumPy, SciPy ve Matplotlib üzerine kurulmuştur. Scikit-learn kitaplığı, verileri yüklemeye,

işlemeye ve özetlemeye odaklanmak yerine, verileri modellemeye odaklanır. Sklearn tarafından sağlanan en popüler model gruplarından bazıları şunlardır:

- Denetimli Öğrenme Algoritmaları: Doğrusal Regresyon, Destek Vektör Makinesi (SVM), Karar Ağacı vb. gibi neredeyse tüm popüler denetimli öğrenme algoritmaları, scikit-learn'in bir parçasıdır.
- Denetimsiz Öğrenme Algoritmaları: Öte yandan, kümeleme, faktör analizi, PCA'dan (Temel Bileşen Analizi) denetimsiz sinir ağlarına kadar tüm popüler denetimsiz öğrenme algoritmalarına da sahiptir.
- Kümeleme: Bu model, etiketlenmemiş verileri gruplamak için kullanılır.
- Çapraz Doğrulama: Görünmeyen veriler üzerinde denetlenen modellerin doğruluğunu kontrol etmek için kullanılır.
- Boyutsallık Azaltma: Özetleme, görselleştirme ve özellik seçimi için daha fazla kullanılabilecek verilerdeki özniteliklerin sayısını azaltmak için kullanılır.
- Topluluk Yöntemleri: Adından da anlaşılacağı gibi, birden çok denetlenen modelin tahminlerini birleştirmek için kullanılır.
- Özellik Çıkarma: Görüntü ve metin verilerindeki nitelikleri tanımlamak için verilerden özellikleri çıkarmak için kullanılır.
- Özellik Seçimi: Denetimli modeller oluşturmak için faydalı öznitelikleri belirlemek için kullanılır.
- Açık Kaynak: Açık kaynak kodlu bir kütüphanedir ve ayrıca BSD lisansı altında ticari olarak kullanılabilir.[20]

3.1.7 Seaborn

Seaborn, Python'da istatistiksel grafikler yapmak için bir kütüphanedir. Matplotlib için üst düzey bir arayüz sağlar ve “pandas” veri yapılarıyla entegre olur. Seabornlibrary'deki işlevler, verilerle ilgili soruları yanıtlayabilecek grafiklere dönüştürmeyi kolaylaştıran bildirime dayalı, veri kümesi odaklı bir API sunar. Bir veri kümesi ve oluşturulacak çizimin bir özelliği verildiğinde, seaborn veri değerlerini otomatik olarak renk, boyut veya stil gibi görsel niteliklerle eşleştirir, dahili olarak istatistiksel dönüşümleri hesaplar ve grafiği bilgilendirici eksen etiketleri ve bir gösterge ile süsler. Birçok işlev, koşullu veri alt kümeleri arasında veya bir veri kümesindeki farklı değişken çiftleri arasında karşılaştırmalar sağlayan birden çok panelli rakamlar üretebilir. seaborn, bilimsel bir projenin yaşam döngüsü boyunca faydalı olacak şekilde tasarlanmıştır. Seaborn, minimum argümanla tek bir işlev çağrısından eksiksiz grafikler üreterek hızlı prototip oluşturmayı ve keşifsel veri analizini kolaylaştırır. Altta yatan matplotlib nesnelerini açığa çıkarmanın yanı sıra kapsamlı özelleştirme seçenekleri sunarak, cilalı, yayın kalitesinde rakamlar oluşturmak için kullanılabilir.[21]

3.1.8 Keras

Keras, Python ile yazılmış, makine öğrenimi platformu TensorFlow'un üzerinde çalışan bir derin öğrenme API'sidir. Hızlı denemeyi mümkün kılmaya odaklanılarak geliştirilmiştir. Fikirden sonuca mümkün olduğunca hızlı gidebilmek, iyi araştırma yapmanın anahtarıdır.

Keras:

- Keras, sizi problemin gerçekten önemli olan kısımlarına odaklanmaktan kurtarmak için geliştiricinin bilişsel yükünü azaltır.
- Keras, karmaşıklığın aşamalı olarak ifşa edilmesi ilkesini benimser: basit iş akışları hızlı ve kolay olmalı, keyfi olarak gelişmiş iş akışları ise halihazırda öğrendiklerinizin üzerine inşa edilen net bir yol aracılığıyla mümkün olmalıdır.
- Keras, sektör gücünde performans ve ölçeklenebilirlik sağlar: NASA, YouTube veya Waymo gibi kuruluşlar ve şirketler tarafından kullanılır.

3.1.8.1 Tensorflow 2 ve Keras

TensorFlow 2, uçtan uca, açık kaynaklı bir makine öğrenimi platformudur. Diferansiyellenebilir programlama için bir altyapı katmanı olarak düşünebilirsiniz. Dört temel yeteneği birleştirir:

- CPU, GPU veya TPU üzerinde düşük seviyeli tensör işlemlerini verimli bir şekilde yürütme.
- Keyfi türevlenebilir ifadelerin gradyanını hesaplama.
- Hesaplamayı yüzlerce GPU'dan oluşan kümeler gibi birçok cihaza ölçeklendirme.
- Programları ("grafikleri") sunucular, tarayıcılar, mobil ve gömülü cihazlar gibi harici çalışma zamanlarına aktarma.
- Keras, TensorFlow 2'nin üst düzey API'sidir: Modern derin öğrenmeye odaklanan, makine öğrenimi sorunlarını çözmek için ulaşılabilir, son derece üretken bir arayüz. Yüksek yineleme hızıyla makine öğrenimi çözümleri geliştirmek ve göndermek için temel soyutlamalar ve yapı taşları sağlar.

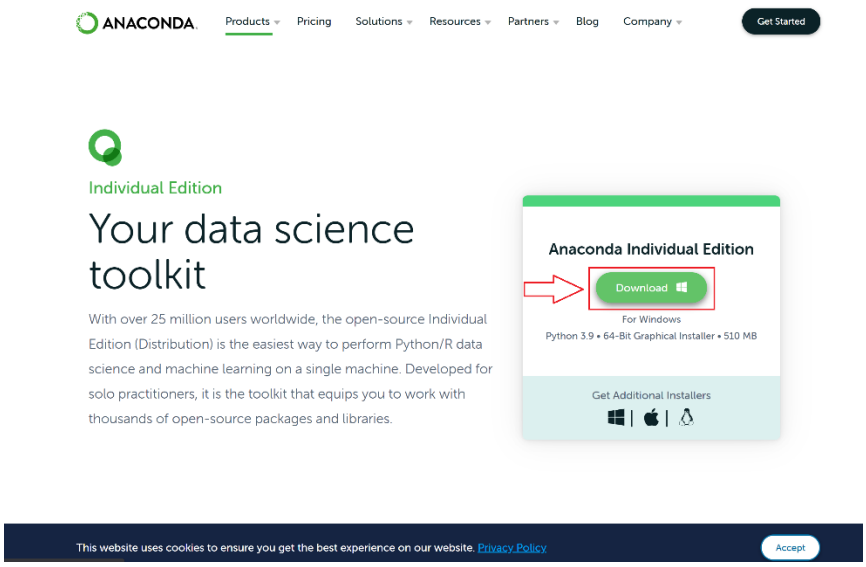
Keras, mühendislere ve araştırmacılara TensorFlow 2'nin ölçeklenebilirlik ve platformlar arası özelliklerinden tam olarak yararlanmaları için yetki verir: Keras'ı TPU'da veya büyük GPU kümelerinde çalıştırabilir ve Keras modellerinizi tarayıcıda veya mobil cihazlarda çalışacak şekilde dışa aktarabilirsiniz.[22]

3.2 Kullanılacak Materyallerin Kurulumu

3.2.1 Anaconda Kurulumu

1. https://www.anaconda.com/products/individual web sitesine girin.

2. İşletim sistemine uygun Anaconda sürümünü indirin.



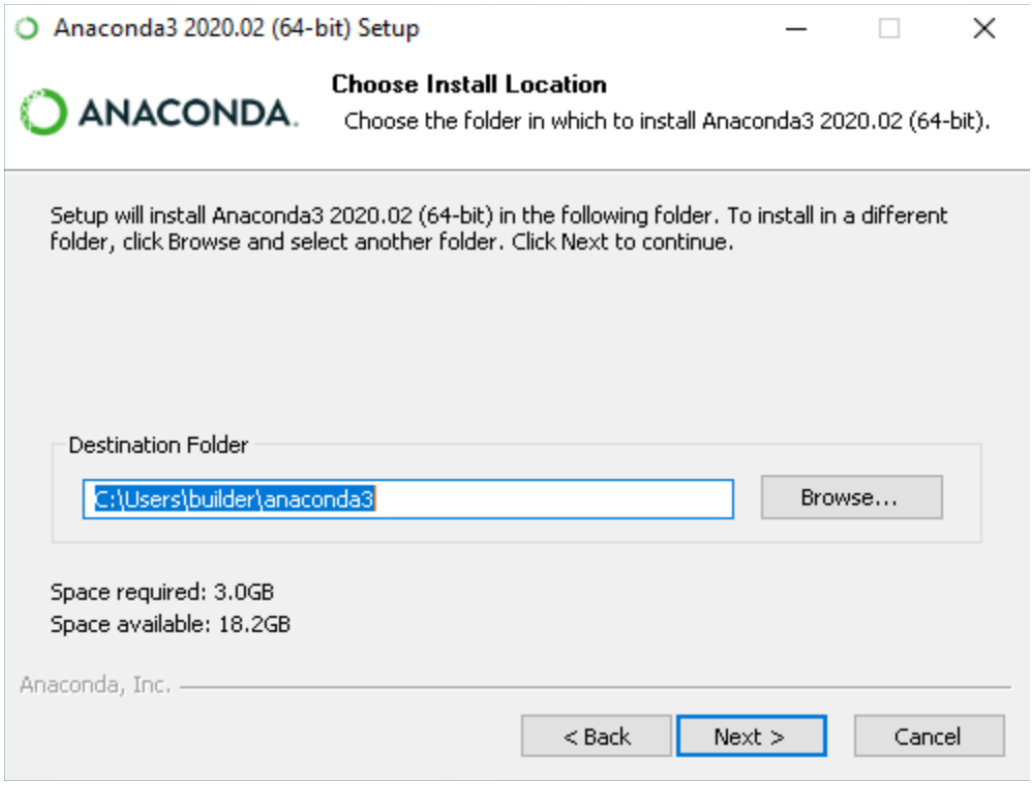
3. Yüklemeyi başlatmak için inen yükleme dosyasına çift tıklayın. Not: Eğer yükleme sırasında sorunlarla karşılaşırsanız, yükleme sırasında virüsten koruma yazılımınızı geçici olarak devre dışı bırakın ve yükleme tamamlandıktan sonra yeniden etkinleştirin. Tüm kullanıcılar için yüklediyseniz, Anaconda'yı kaldırın ve yalnızca kullanıcınız için yeniden yükleyin ve yeniden deneyin.

4. “Next”e tıklayın.

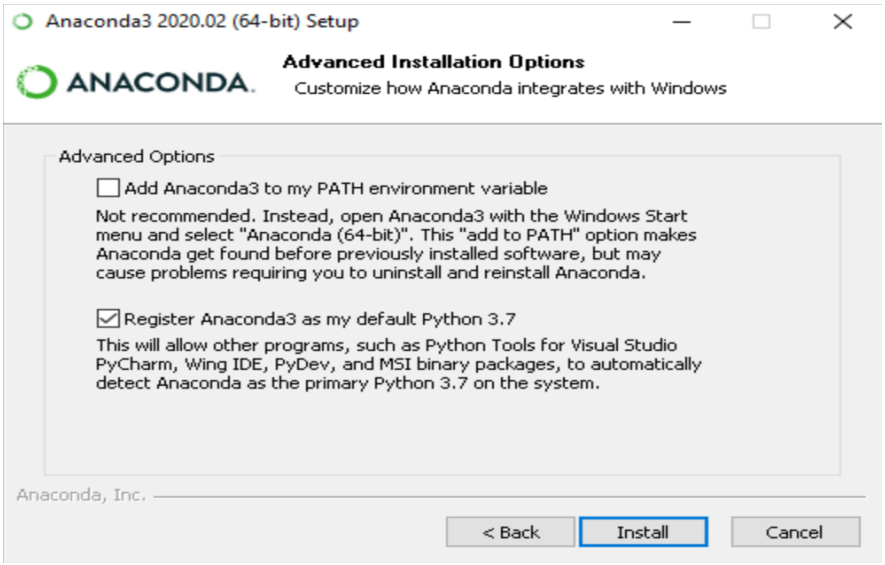
5. Lisans koşullarını okuyun ve “Accept” düğmesine tıklayın.

6. Tüm kullanıcılar için yükleme yapmıyorsanız (Windows Yönetici ayrıcalıkları gerektirir) “Just Me” için bir yükleme seçin ve “Next” düğmesine tıklayın.

7. Anaconda'yı kurmak için bir hedef klasör seçin ve “Next” düğmesine tıklayın.



8. PATH ortam değişkeninize Anaconda eklenip eklenmeyeceğini seçin. Diğer yazılımları etkileyebileceğinden, PATH ortam değişkenine Anaconda eklememenizi öneririz. Bunun yerine, Başlat Menüsünden Anaconda Navigator'ı veya Anaconda İstemi'ni açarak Anaconda yazılımını kullanın.



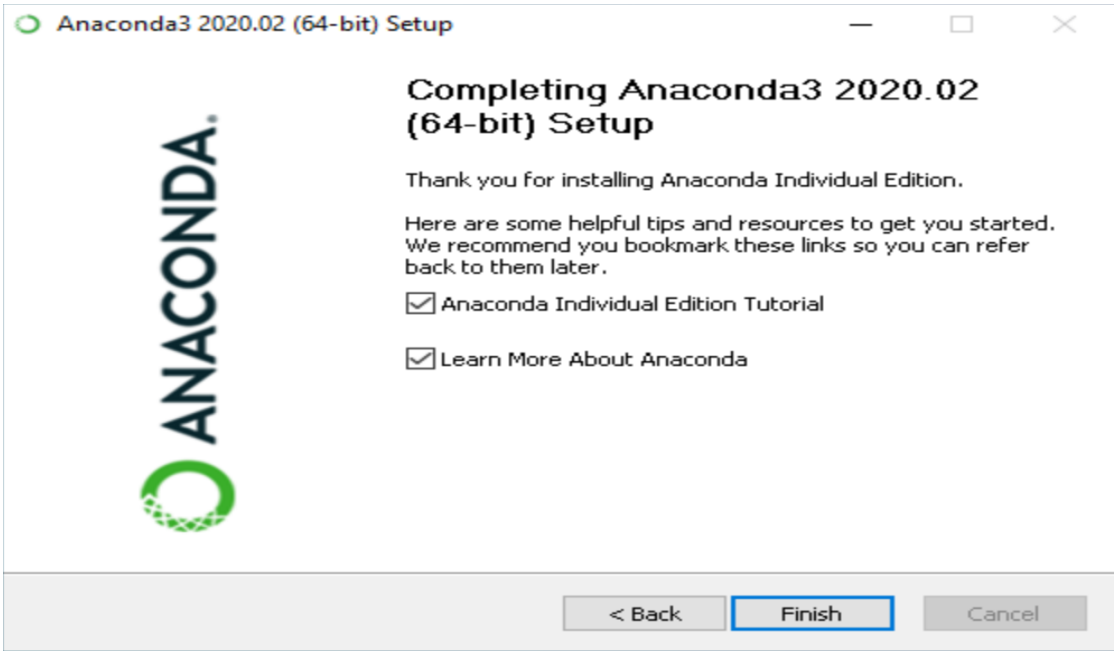
9. Anaconda'yı varsayılan Python'unuz olarak kaydedip kaydetmeyeceğinizi seçin. Anaconda'nın birden çok sürümünü veya Python'un birden çok sürümünü yüklemeyi ve çalıştırmayı planlamıyorsanız, varsayılanı kabul edin ve bu kutuyu işaretli bırakın.

10. “Install” düğmesini tıklayın. Anaconda'nın kurduğu paketleri izlemek istiyorsanız, “Show Details” düğmesine tıklayın.

11. “Next” düğmesine tıklayın.

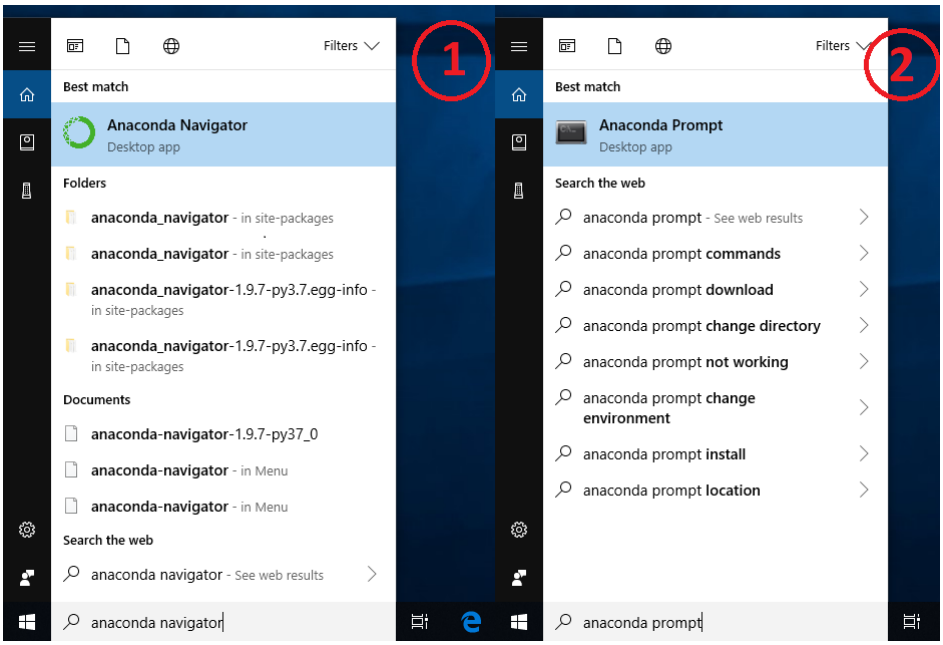
12. Anaconda'yı PyCharm olmadan kurmal için “Next” düğmesine tıklayın.

13. Başarılı bir kurulumdan sonra “Anaconda'yı kurduğunuz için teşekkürler” iletişim kutusunu göreceksiniz:



14. Anaconda.org ve Anaconda'ya nasıl başlayacağınız hakkında daha fazla bilgi edinmek istiyorsanız, "Anaconda Individual Edition Tutorial" ve "Learn More About Anaconda" kutularını işaretleyin. “Finish” düğmesine tıklayın.

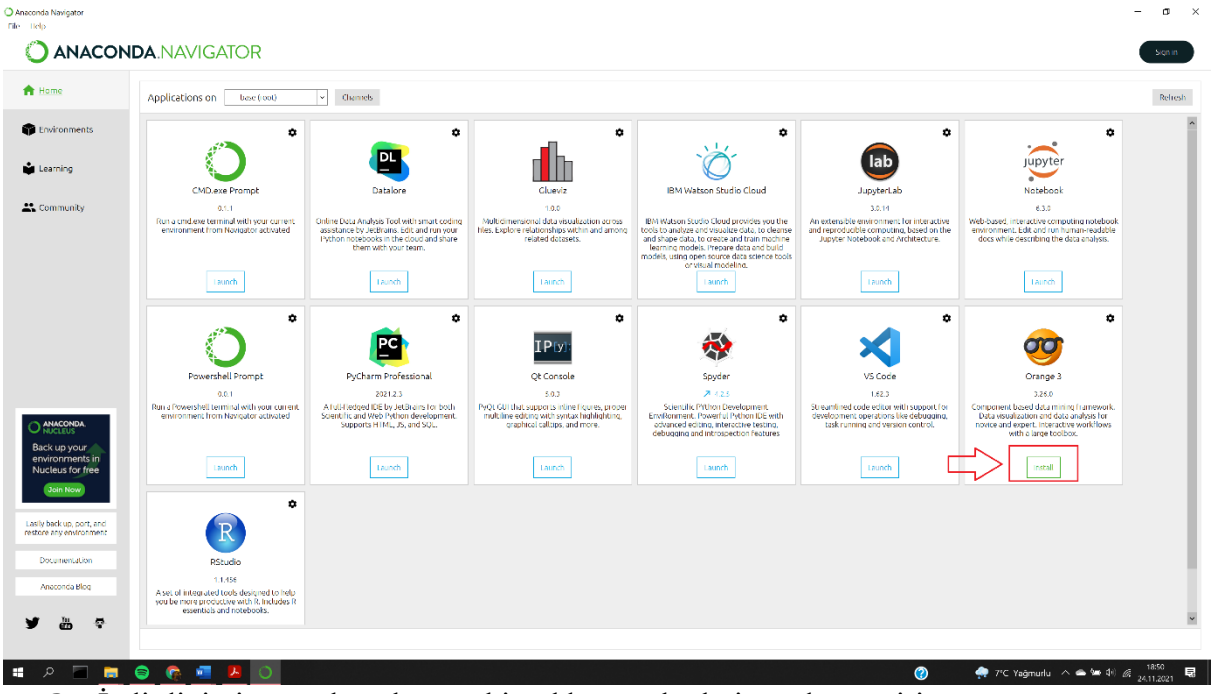
15. Yüklemeyi doğrulamak için Windows arama çubuğunda sırayla “Anaconda Navigator” ve “Anaconda Prompt” uygulamalarını arayalım. Eğer iki uygulama da çalışıyorsa yüklemenin başarılı bir şekilde yapıldığını doğrulayabiliriz.



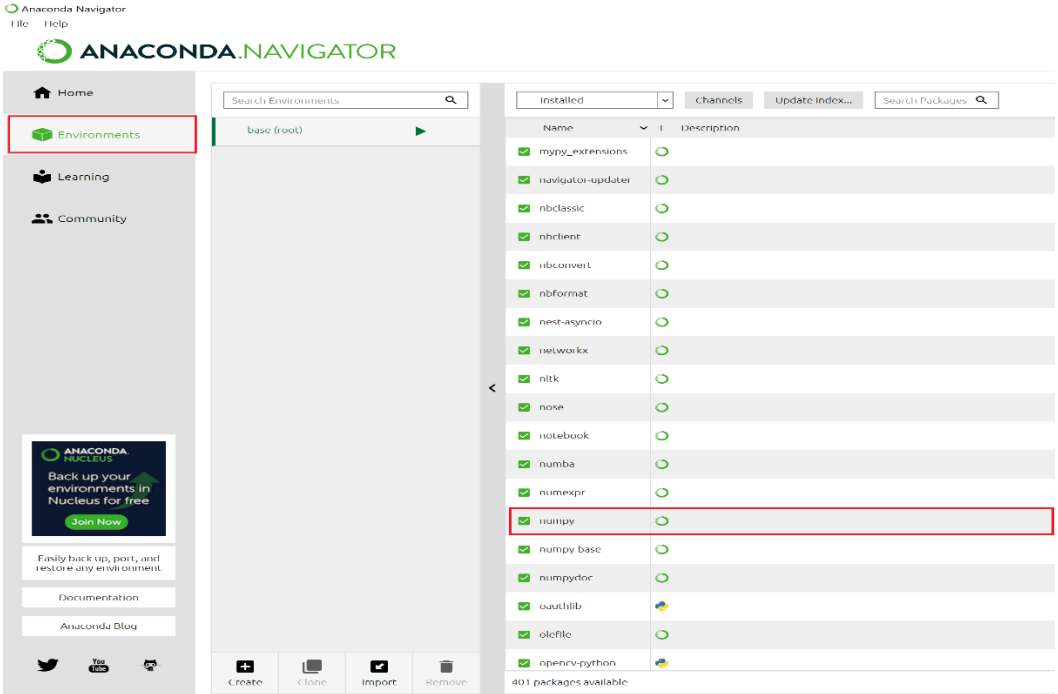
3.2.2 Anaconda Navigator'de Paketlerin Yüklenmesi

1. Anaconda Navigator uygulamasını açın.

2. Karşımıza gelen ekrandan ihtiyacımız olan uygulamaları indirin.



3. İndirdiğimiz uygulamaların sahip olduğu paketleri gözden geçirin.

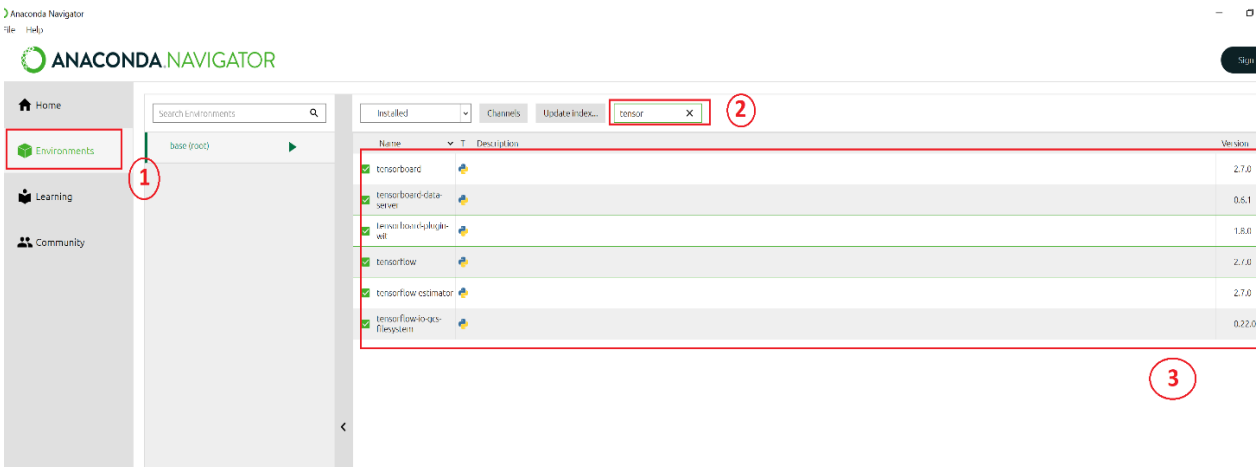


Yukarıdaki yükleme ve kurulum işlemlerini yaptıktan sonra yüklenen paketleri inceledik ve ihtiyacımız olan python yazılım diline ve "Pandas, NumPy, Matplotlib, Seaborn

Scikit-Learn” paketlerine sahip olduğumuzu gördük. Bu durum da bize Anaconda uygulamasının rahatlığını göstermektedir.

3.2.3 Tensorflow 2 ve Keras Paketlerin Yüklenmesi

Anaconda’da yükleme işlemlerini hallettikten sonra eksik kalan Tensorflow 2 ve Keras paketlerini komut isteminde “pip” paketi sayesinde yükledik. Peki “pip” nedir? “Pip” Python için bir paket yöneticisidir. Bu, standart kitaplığın bir parçası olarak dağıtılmayan ek kitaplıkları ve bağımlılıkları kurmanıza ve yönetmenize olanak tanıyan bir araç olduğu anlamına gelir. Tensorflow 2 için anaconda komut istemine “” ve Keras için anaconda komut istemine “” yazıyoruz ve indirmesini bekliyoruz. İndirmenin başarılı olup olmadığını anlamanın iki yolu vardır. İlki anaconda komut istemine “pip list” yazarak gelen listeden paketlerin yüklenip yüklenmediğine bakmaktır. İkinci yol ise daha kolaydır. Anaconda Navigator uygulamasına girip Environments sekmesine tıklıyoruz. Sonra uygulama içindeki arama çubuğuna bulmak istediğimiz paketin simini yazıyoruz. Eğer başarılı şekilde yüklendiyse listede paket ismi çıkar ve “pip” yöntemi ile yüklediğimiz için yanında anaconda işareti yerine python işareti vardır.



3.3 Kullanılacak Materyallerin Geliştirilmesi

3.3.1 Makine Öğrenimi

Makine öğrenmesi, günümüzde nesne tanıma, görüntü işleme, yüz tanıma, sanal gerçeklik, artırılmış gerçeklik, ses tanıma, iris tanıma, pazarlama, sağlık, müşteri hizmetleri, uydu görüntüleri, yer bilimi, sahtekârlık yakalama (fraud) gibi pek çok alanda kullanılan yapay zekânın bir alt ve en geniş koludur.

Temelde makine öğrenmesi gerçekleştiren sistemlerden; öğrendiği bilgiyi saklama, tıpkı insan ve hayvan beyni gibi öğrendiği, sakladığı bilgileri tecrübe haline getirerek yeni durumlara uyarlayıp kullanması beklenir.

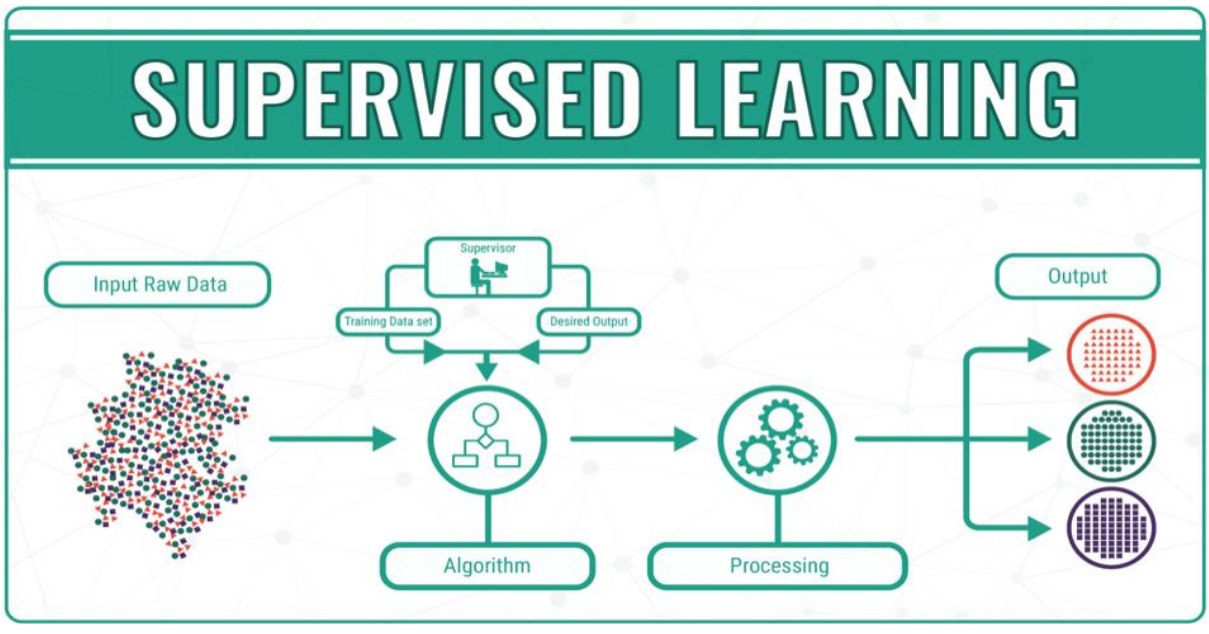
Makine öğrenme diğer adıyla yapay öğrenme; bilgisayarlara daha önceden tanıtılan bilgiyi özümseyerek her olası yeni durum için tekrar programlamaya ihtiyaç duymayan, hazır bulunan verilerden çıkarımlar yapan ve bu çıkarımlarla bilinmeyene dair tahminlerde bunulan yapay zekânın en önemli kullanım alanlarında biridir. Turing makinelerin de insanlar gibi düşünebileceği ve yeni durumlar içi kararlar alabileceği fikrini ilk ortaya atan kişidir (1950). Çalışmalarında makinelerin de tıpkı insanlar gibi düşünüp öğrenebileceğinden bahsederek günümüzde yaygın olarak kullanılan yapay zekâ alanına temel oluşturacak çıkarımlarda bulunmuştur.

Öztemel (2003)'e göre makine öğrenmesi bir diğer ifade ile mekanik öğrenme bilgisayarın bir durumla alakalı elde ettiği bilgileri ve edindiği tecrübeleri kullanarak sonraki zamanlarda karşılaşabileceği yeni bir duruma uyarlayıp karar verebilmesi ve problemlere yeni çözümler getirebilmesidir. Makine öğrenmesi (machine learning) için bazı şartlar bulunmaktadır. Öğrenme sürecinin esası öğrenenin insan ya da makine olmasına bakılmaksızın benzerlikler gösterir. Ayrıca birbiri ile ilişkili dört unsur bulunmaktadır. Bunlar:

• Depolama: Görsel deneylerden, bellekten ve çağrışımlardan yararlanır.
• Soyutlama: Hafızadaki dataları genişletir ve kavramsal öğelere dönüştürür.
• Genelleştirme: Bir önceki adımdaki kavramsal öğeleri kullanır ve genel ifadelere dönüştürür.
• Değerlendirme: Öğrenilen kavramların uygulanabilirliğini test etmek ve tahmin için ön bildirim sistemi kullanır.

3.3.2 Denetimli Öğrenme

Denetimli öğrenme (gözetimli öğrenme); sisteme eğitim veri seti ile test veri setinin yüklenmesiyle başlar. Veri setinde her bir veri için gerekli etiketlenmenin yapılması ve bu sayede girdi veri seti ile çıktı veri seti arasında ilişki kurulması süreçleridir. Bu süreçte, girdiler ve çıktılar arasındaki ilişkiyi tanımlayan bir fonksiyon ya da algoritma öğrenilir. Önceden bilinen çıktılar üstünde sınıflama gerçekleştirildikten sonra sonuçları henüz bilinmeyen veri kümelerinde olası sonuçlar tahmin edilmeye çalışılır. Başka bir deyişle gözetimli öğrenme; bilinen örnekleri yani girdi ve çıktıları kullanarak bir model oluşturup yeni bir girdi verisi üzerinden çıktı değişkenini tahmin etme eylemidir.



4 BULGULAR VE TARTIŞMA

Verileri eğiten ve test verileri üzerindeki doğruluk puanını hesaplayan Python Scikit-Learn'i kullanarak bir ikili sınıflandırma modeli (kimlik avı web sitesi veya değil) için kodlama yapacağız. Kimlik avı web sitesi veri kümesinde bir modeli eğitmek için birden fazla sınıflandırma algoritması kullanmamız iyi olur.

4.1 Veri Seti Hakkında

Kullanacağımız hazır veri seti, hem metin dosyasında hem de model oluşturma için girdi olarak kullanılabilecek aşağıdaki özellikler sahip "csv" dosyasıdır[23]:

1) 11000'den fazla web sitesi URL koleksiyonu. Her örnekte 30 web sitesi parametresi ve onu kimlik avı web sitesi olarak tanımlayan bir sınıf etiketi vardır (1 veya -1).
2) Veri seti aynı zamanda proje kapsamı için bir girdi işlevi görür ve bunun için işlevsel ve işlevsel olmayan gereksinimleri belirlemeye çalışır.
3) Başlık listesi (sütun adları) aşağıdaki gibidir:
   ['UsingIP', 'LongURL', 'ShortURL', 'Symbol@', 'Redirecting//',
   'PrefixSuffix-', 'SubDomains', 'HTTPS', 'DomainRegLen', 'Favicon',
   'NonStdPort', 'HTTPSDomainURL', 'RequestURL', 'AnchorURL',
   'LinksInScriptTags', 'ServerFormHandler', 'InfoEmail', 'AbnormalURL',
   'WebsiteForwarding', 'StatusBarCust', 'DisableRightClick',
   'UsingPopupWindow', 'IframeRedirection', 'AgeofDomain',
   'DNSRecording', 'WebsiteTraffic', 'PageRank', 'GoogleIndex',
   'LinksPointingToPage', 'StatsReport', 'class']

Veri kümesindeki başlıkların alabileceği değerler:

- UsingIP : { -1,1 }
- LongURL : { 1,0,-1 }
- ShortURL : { 1,-1 }
- Symbol@ : { 1,-1 }
- Redirecting// : { -1,1 }
- PrefixSuffix- : { -1,1 }
- SubDomains : { -1,0,1 }
- HTTPS : { -1,1,0 }
- DomainRegLen : { -1,1 }
- Favicon : { 1,-1 }
- NonStdPort : { 1,-1 }
- HTTPSDomainURL : { -1,1 }
- RequestURL : { 1,-1 }
- AnchorURL : { -1,0,1 }

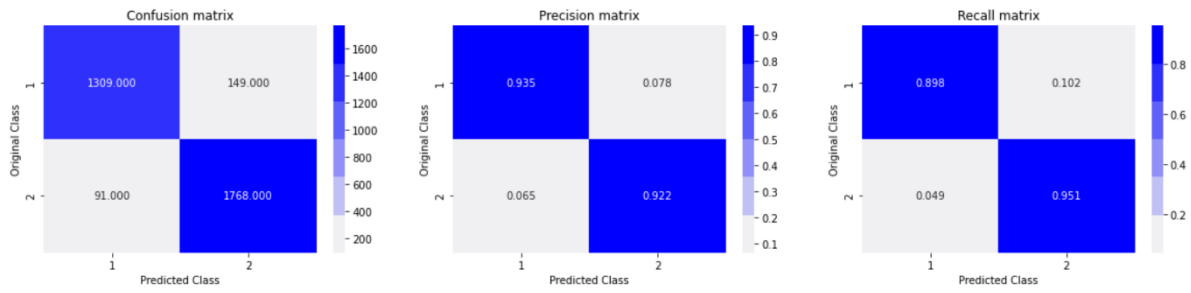
	Index	UsingIP	LongURL	ShortURL	Symbol@	Redirecting//	PrefixSuffix-	SubDomains	HTTPS	DomainRegLen	...	UsingPopupWindow	IframeRedirection	AgeofDomain	DNSRecording	WebsiteTraffic	PageRank	GoogleIndex	LinksPointingToPage	StatsReport	class
0	0	1	1	1	1	1	-1	0	1	-1	...	1	1	-1	-1	0	-1	1	1	1	-1
1	1	1	0	1	1	1	-1	-1	-1	-1	...	1	1	1	-1	1	-1	1	0	-1	-1
2	2	1	0	1	1	1	-1	-1	-1	1	...	1	1	-1	-1	1	-1	1	-1	1	-1
3	3	1	0	-1	1	1	-1	1	1	-1	...	-1	1	-1	-1	0	-1	1	1	1	1
4	4	-1	0	-1	1	-1	-1	1	1	-1	...	1	1	1	1	1	-1	1	-1	-1	1

4.2 Kullanılan Sınıflandırma Algoritmaları

Projemizde 6 adet sınıflandırma algoritması hakkında kısa bilgiler verip “phishing-website-detector” isimli hazır veri seti üzerinde kullanacağız ve bu tezin sonuçlar bölümünde algoritmaların karşılaştırmalarını anlatacağız. Kullanacağımız algoritmalar; Logistic Regression, K-Nearest Neighbour, Decision Tree Classifier, Random Forest Classifier, Support Vector Machine, Adaboost Classifier algoritmalarıdır.

4.2.1 Logistic Regression Algoritması

Lojistik regresyon temelde denetimli bir sınıflandırma algoritmasıdır. Bir sınıflandırma probleminde, hedef değişken (veya çıktı), y, belirli bir özellik(veya girdiler), X için yalnızca ayrık değerler alabilir. Popüler inanışın aksine, lojistik regresyon bir regresyon modelidir. Model, belirli bir veri girişinin “1” olarak numaralandırılmış kategoriye ait olma olasılığını tahmin etmek için bir regresyon modeli oluşturur. Doğrusal regresyon, verilerin doğrusal bir işlevi takip ettiğini varsaydığı gibi, Lojistik regresyon da verileri sigmoid işlevini kullanarak modeller. Algoritmayı veri setimize uyguladığımızda aşağıdaki sonucu elde ediyoruz:

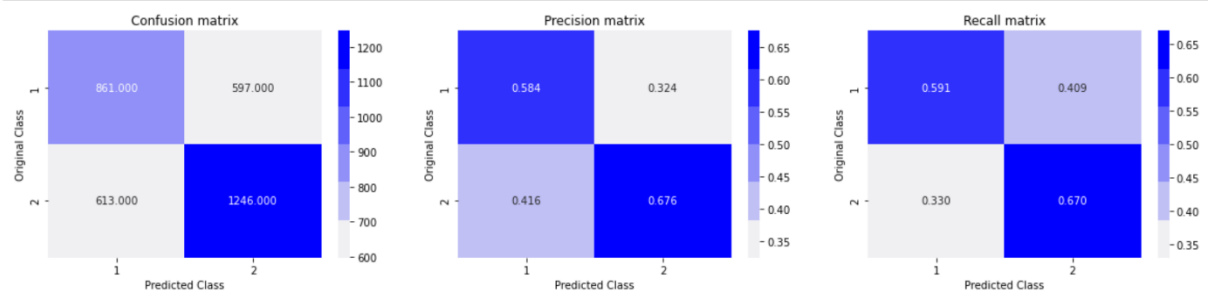


Sınıflandırmadan elde ettiğimiz doğruluk sonucu: `0.9249321676213446`

4.2.2 K-Nearest Neighbour Algoritması

Sınıflandırma ve regresyon için kullanılır. Her iki durumda da girdi, bir veri setindeki en yakın k eğitim örneğinden oluşur. Çıktı, sınıflandırma veya regresyon için k-NN'nin kullanılmasına bağlıdır:

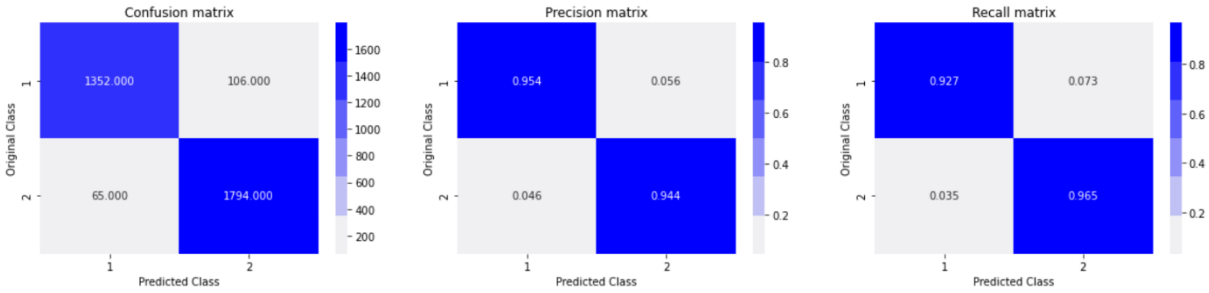
K-NN sınıflandırmasında çıktı bir sınıf üyeliğidir. Bir nesne, komşularının çoğul oyu ile sınıflandırılır ve nesne, en yakın k komşusu arasında en yaygın olan sınıfa atanır (k, pozitif bir tamsayıdır, tipik olarak küçüktür). k = 1 ise, nesne basitçe o en yakın komşunun sınıfına atanır. K-NN regresyonunda çıktı, nesnenin özellik değeridir. Bu değer, en yakın k komşunun değerlerinin ortalamasıdır. Sürekli değişkenler için yaygın olarak kullanılan bir uzaklık ölçüsü Öklid uzaklığıdır. Algoritmayı veri setimize uyguladığımızda aşağıdaki sonucu elde ediyoruz:



Sınıflandırmadan elde ettiğimiz doğruluk sonucu: `0.6343081097377148`

4.2.3 Decision Tree Classifier Algoritması

Decision Tree Classifier, veri madenciliğinde yaygın olarak kullanılan bir yöntemdir. Amaç, birkaç girdi değişkenine dayalı olarak bir hedef değişkenin değerini tahmin eden bir model oluşturmaktır. Bir karar ağacı, örnekleri sınıflandırmak için basit bir temsildir. Bir karar ağacı veya bir sınıflandırma ağacı, her bir dahili (yaprak olmayan) düğümün bir girdi özelliği ile etiketlendiği bir ağaçtır. Bir girdi özelliği ile etiketlenmiş bir düğümden gelen yaylar, hedef özelliğin olası değerlerinin her biri ile etiketlenir veya yay, farklı bir girdi özelliği üzerinde alt bir karar düğümüne yol açar. Ağacın her yaprağı bir sınıfla veya sınıflar üzerinde bir olasılık dağılımıyla etiketlenir; bu, veri kümesinin ağaç tarafından belirli bir sınıfa veya belirli bir olasılık dağılımına (karar ağacı iyiyse) sınıflandırıldığını gösterir. -inşa edilmiş, belirli sınıf alt kümelerine doğru eğridir). Algoritmayı veri setimize uyguladığımızda aşağıdaki sonucu elde ediyoruz:

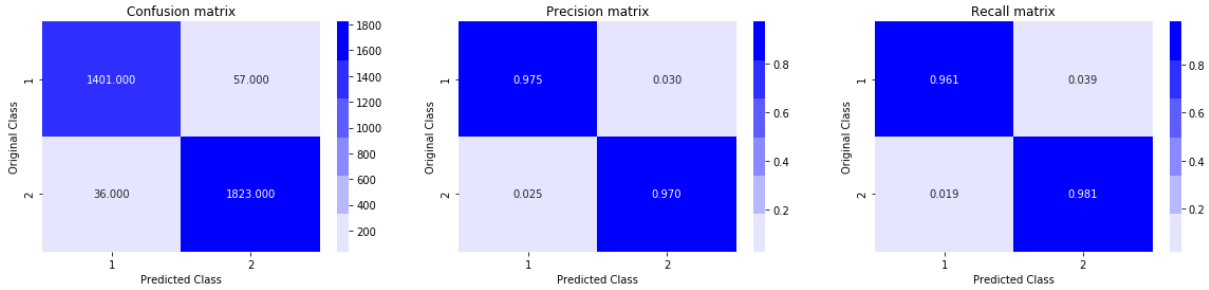


Sınıflandırmadan elde ettiğimiz doğruluk sonucu : `0.9439252336448598`

4.2.4 Random Forest Classifier Algoritması

"Random Forest Classifier" sınıflandırma ve regresyon problemlerinde yaygın olarak kullanılan bir denetimli makine öğrenimi algoritmasıdır. Farklı örnekler üzerinde karar ağaçları oluşturur ve regresyon durumunda sınıflandırma ve ortalama için çoğunluk oyu alır.

"Random Forest Classifier" algoritmasının en önemli özelliklerinden biri, regresyon durumunda olduğu gibi sürekli değişkenleri ve sınıflandırma durumunda olduğu gibi kategorik değişkenleri içeren veri setini işleyebilmesidir.

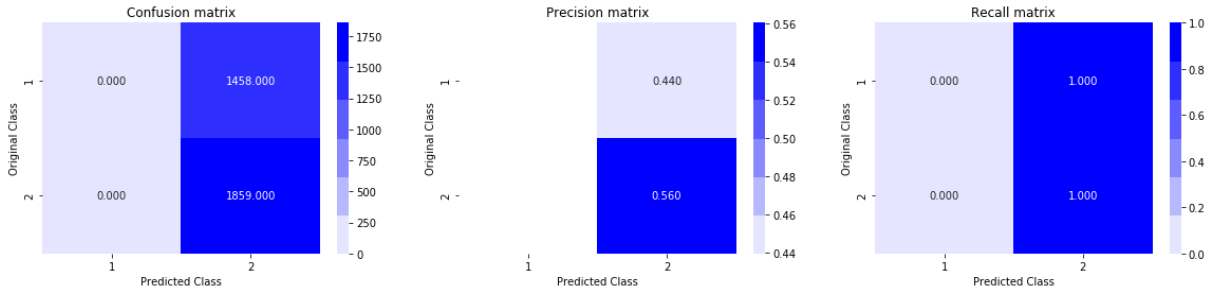


Sınıflandırmadan elde ettiğimiz doğruluk sonucu : 0.9719626168224299

4.2.5 Support Vector Machine Algoritması

" Support Vector Machine " (SVM), hem sınıflandırma hem de regresyon zorlukları için kullanılabilen denetimli bir makine öğrenme algoritmasıdır. SVM algoritmasında, her bir

veri öğesini, her özelliğin değeri belirli bir koordinatın değeri olacak şekilde n-boyutlu uzayda (n, sahip olduğunuz bir dizi özelliktir) bir nokta olarak çizeriz. Ardından, iki sınıfı çok iyi ayıran hiper düzlemi bularak sınıflandırma yapıyoruz.

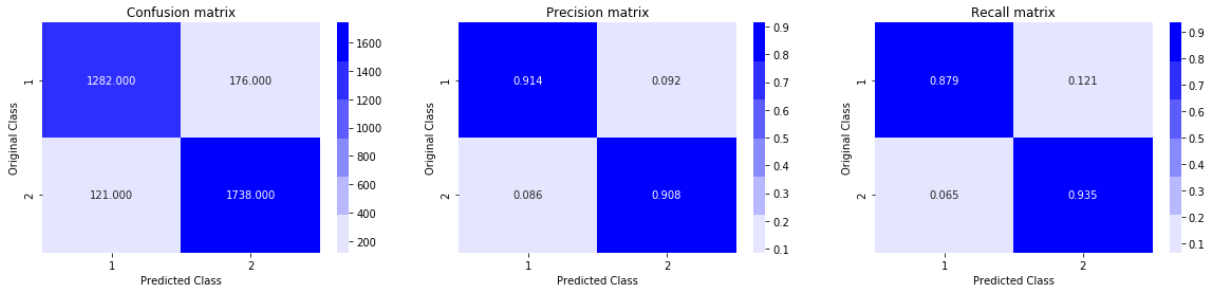


Sınıflandırmadan elde ettiğimiz doğruluk sonucu : 0.5604461863129334

4.2.6 Adaboost Classifier Algoritması

Adaptive Boosting olarak da adlandırılan AdaBoost, Makine Öğreniminde Topluluk Yöntemi(Ensemble Method) olarak kullanılan bir tekniktir. AdaBoost ile kullanılan en yaygın algoritma, tek seviyeli karar ağaçlarıdır, yani sadece bir bölmeli Karar ağaçlarıdır. Bu ağaçlara Karar Kütükleri(Decision Stumps) de denir.

Bu algoritmanın yaptığı, bir model oluşturması ve tüm veri noktalarına eşit ağırlıklar vermesidir. Daha sonra yanlış sınıflandırılmış noktalara daha yüksek ağırlıklar atar. Artık bir sonraki modelde daha yüksek ağırlıklara sahip tüm noktalara daha fazla önem verilmektedir. Düşük bir hata alınana kadar modeller eğitilmeye devam edecektir.



Sınıflandırmadan elde ettiğimiz doğruluk sonucu : 0.9104612601748568

4.3 Algoritmaların Karşılaştırması

Kullandığımız algoritmaların doğruluk sonuçları grafiklerin altında verilmiştir.Bu tabloya göre en başarılı algoritma %97 kesinlik oranı ile “Random Forest Classifier” algoritması oldu. Random forest, birden fazla karar ağacını kullanarak daha uyumlu modeller üreterek daha isabetli sınıflandırma yapmaya çalışan bir sınıflandırma modelidir.Bundan dolayı bizim projemizde daha fazla doğruluk payı katmıştır ve yüzdesi yükselmiştir.

Rassal orman (Random Forest), hiper parametre kestirimi yapılmadan da iyi sonuçlar vermesi hem regresyon hem de sınıflandırma problemlerine uygulanabilir olmasından dolayı popüler makine öğrenmesi modellerinden birirdir. Rassal orman modelinde farklı veri setleri üzerinde eğitim gerçekleştiği için varyans, diğer bir deyişle karar ağaçlarının en büyük problemlerinden olan overfitting azalır. Ayrıca bootstrap yöntemiyle oluşturduğumuz alt-veri kümelerinde outlier bulunma şansını da düşürmüş oluruz.

Rassal orman modelinde farklı veri setleri üzerinde eğitim gerçekleştiği için varyans, diğer bir deyişle karar ağaçlarının en büyük problemlerinden olan overfitting azalır. Ayrıca bootstrap yöntemiyle oluşturduğumuz alt-veri kümelerinde outlier bulunma şansını da düşürmüş oluruz.

Projemizde uyguladığımız adımlar :

Bu çalışma standart bir veri bilimi projesinin en basit hali olarak düşünülebilir.

Veri tiplerini kontrol etme/düzeltme.

- Açıklayıcı veri analizi ve görselleştirme.
- Eksik verileri tahmin etme/veri atama.
- Kategori tipindeki verileri one-hot encoding ile nümerik formata çevirme.
- Veri setini eğitim ve test veri-setlerine ayırma.
- Modeli eğitme ve test verisi üzerinde tahmin yapma.
- Sınıflandırma başarı metriklerine bakma.
- Karar ağacını görselleştirme.
- Modelin yaptığı öznitelik sıralamasını görselleştirme.

5 Sonuç ve Öneriler

Projemizin temel amacını zararlı web sitelerini tespit etme olarak belirledik. Bunun için hazır bir veri seti kullandık. Bu hazır veri setini analiz ederken uygulamamız gereken adımları araştırıp bir analiz çıkardık ve veri analitiği adımlarını uyguladık. Bir "Phishing" saldırısının ne olduğunu ve nasıl uygulandığını tezimizde bir örnekle anlattık. "Phishing" saldırılarında veri tabanı çalınabileceği için veri tabanı ve veri tabanı yönetim sistemlerine değindik.

Kullanacağımız materyal ve metotlarla ilgili araştırma ve okumalar yapıp açıklamalarını yaptık. Projemizde son zamanlarda yapay zeka projelerinde sıklıkla tercih edilen Python programlama dilini kullandık. Anaconda Navigator adlı programı yükledik ve yükleme işlemini tezimizde adım adım anlattık. Derleyici olarak Anaconda Navigator uygulamasının içindeki Jupyter Notebook derleyicisini kullandık. Jupyter Notebook'u seçmemizdeki en önemli etkenler adım adım ilerlemeyi sağlaması ve grafik çıktılarını güzel bir şekilde göstermesidir. Uygulamayı yaparken; veri işlemleri için Pandas, numerik işlemler için Numpy, grafik çizdirme için Matplotlib ve Seaborn, algoritmalar için Scikit-Learn kütüphanelerinden yararlandık. Genel olarak bir kütüphanenin nasıl yükleneceğini de tezimizde anlattık.

Projemizde denetimli makine öğrenimi kullanacağımız için onun hakkında açıklama yaptık. Projede 6 farklı sınıflandırma algoritması kullandık. Bunlar "Logistic Regression" "K-Nearest Neighbour" "Decision Tree Classifier" "Random Forest Classifier" "Support Vector Machine" "Adaboost Classifier" isimli algoritmalardır. Farklı algoritmalar kullanarak hangi algoritmanın ne kadar iyi çalıştığını görmüş olduk. "Logistic Regression" "Decision Tree Classifier" "Random Forest Classifier" "Adaboost Classifier" algoritmalarında yaklaşık olarak %91 ile %97 arasında kesinlik oranı sağlarken "K-Nearest Neighbour" algoritmasında %63 kesinlik oranı, "Support Vector Machine" algoritmasında ise %56 kesinlik oranı elde edebildik. En başarılı algoritma %97 kesinlik oranı ile "Random Forest Classifier" algoritması oldu.

Projemiz makine öğrenimi algoritmaları ile "Phishing" sitelerini algılama üzerinedir. Phishing saldırıları yüzünden çok sayıda insan mağdur olmuştur. Bu proje insanların verilerini çalmaya çalışan web sitelerini makine öğrenimi algoritmaları ile bulmayı amaçlamaktadır.

Proje üzerinde birkaç makine öğrenimi algoritması daha denenebilir. Örneğin DBSCAN (Density Based Spatial Clustering of Applications with Noise) ve GMM(Gaussian Mixture Models) gibi algoritmalar denenebilir. Projemiz uygun hale getirilirse bir web browser eklentisi ya da e-posta sistemi eklentisi olarak kullanılabilir.

KAYNAKÇA

[1] URL 1: https://www.kaspersky.com.tr/resource-center/definitions/web-filter

[2] URL 2: https://www.academia.edu/download/32207596/1112-nitel-arac59ftc4b1rmada-veri-analizi.pdf

[3] URL 3: https://en.wikipedia.org/wiki/Data_analysis

[4] Kitap 1: Sütcü, Cem S. ve Aytekin, Ç. (2018). Veri Bilimi. İstanbul: Paloma Yayınevi.

[5] Kitap 2: Zehra, Alakoç, Burma , Veri tabanı yönetim sistemleri ve SQL/PL-SQL/T-SQL : Seçkin Yayıncılık

[6] URL 6: https://www.projectcubicle.com/what-is-data-analytics-definition-with-examples/

[7] Makale 1: E.S. Mezentseva, Regional Economics: Theory and Practice, 2021, vol. 19, iss. 11, pp. 2086–2106: https://970e22ceb1309b85a3bb416ee262012c4031f684.vetisonline.com/eds/pdfviewer/pdfviewer?vid=0&sid=8f83586f-bdcf-4c72-a127-08124d3392ce%40redis

[8] Kitap 3: Gacovski, Zoran, Internet of Things: Arcler Press

[9] Makale 2: Tuğba ŞEN KÜPELİ, Kurban ÜNLÜÖNEN, Veri Madenciliği ve Turizmde Veri Madenciliği Çalışmaları (Data Mining and Data Mining Studies in Tourism): https://jotags.org/2021/vol9_issue1_article16.pdf

---

[10] URL 9: https://www.computerworld.com/article/2575094/sidebar--the-origins-of-phishing.html

[11] URL 10: https://www.csoonline.com/article/2117843/what-is-phishing-how-this-cyber-attack-works-and-how-to-prevent-it.html

[12] URL 11: https://www.cbsnews.com/news/the-phishing-email-that-hacked-the-account-of-john-podesta/

[13] URL 12: https://techcrunch.com/2016/03/15/prosecutors-find-that-fappening-celebrity-nudes-leak-was-not-apples-fault/

[14] URL 13: https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:online-data-security/xcae6f4a7ff015e7d:cyber-attacks/a/phishing-attacks

[15] URL 14: https://docs.anaconda.com/anaconda/navigator/index.html

[16] URL 15: https://www.python.org/doc/essays/blurb/

[17] URL 16: https://mode.com/python-tutorial/libraries/pandas/

[18] URL 17: https://numpy.org/doc/stable/user/whatisnumpy.html

[19] URL 18: https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/

[20] URL 19: https://www.tutorialspoint.com/scikit_learn/scikit_learn_introduction.htm

[21] Makale 3: Waskom, M. L., (2021). seaborn: statistical data visualization. Journal of Open Source Software : https://joss.theoj.org/papers/10.21105/joss.03021

[22] URL 20: https://keras.io/about/

[23] Veri Seti: https://www.kaggle.com/datasets/eswarchandt/phishing-website-detector