T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

WEB TABANLI BİR DİNAMİK KARAR AĞACI UYGULAMASI

MİNTAZE HACIOĞLU

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Tez Danışmanı: Doç. Dr. İlhan UMUT

EDİRNE-2022

MİNTAZE HACIOĞLU'nun hazırladığı "**WEB TABANLI BİR DİNAMİK KARAR AĞACI UYGULAMASI**" başlıklı bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından **Bilgisayar Mühendisliği** Anabilim Dalında bir **Yüksek lisans tezi** olarak kabul edilmiştir.

Jüri Üyeleri (Ünvan, Ad, Soyad): İmza

Doç. Dr. İlhan UMUT

Dr. Öğretim Üyesi Tarık YERLİKAYA

Dr. Öğretim Üyesi Hayrettin TOYLAN

Tez Savunma Tarihi: 04/01/2022

Bu tezin Yüksek Lisans/Doktora tezi olarak gerekli şartları sağladığını onaylarım.

İmza

Doç. Dr. İlhan UMUT
Tez Danışmanı ……

Trakya Üniversitesi Fen Bilimleri Enstitüsü onayı

Prof. Dr. Hüseyin Rıza Ferhat KARABULUT
Fen Bilimleri Enstitüsü Müdürü

T.Ü.FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ YÜKSEK LİSANS PROGRAMI
DOĞRULUK BEYANI

Trakya Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında, tüm verilerin bilimsel ve akademik kurallar çerçevesinde elde edildiğini, kullanılan verilerde tahrifat yapılmadığını, tezin akademik ve etik kurallara uygun olarak yazıldığını, kullanılan tüm literatür bilgilerinin bilimsel normlara uygun bir şekilde kaynak gösterilerek ilgili tezde yer aldığını ve bu tezin tamamı ya da herhangi bir bölümünün daha önceden Trakya Üniversitesi ya da farklı bir üniversitede tez çalışması olarak sunulmadığını beyan ederim.

.... / .... / ....
Mintaze HACIOĞLU
İmza

Yüksek Lisans Tezi
Web Tabanlı Bir Dinamik Karar Ağacı Uygulaması
T.Ü. Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

Bu tez çalışmasında amaçlanan görsel ve kullanıcı dostu bir karar ağacı ortamı oluşturmaktır. Bu amaç doğrultusunda çalışmaya özel karar ağacı oluşturma ve oluşturulan karar ağacını web ortamında kullanmak için ik farklı uygulama geliştirilmiştir.

Kullanıcının elinde bulunan mevcut verileri bir karar ağacına dönüştürmesi hedeflenmektedir. Oluşturulan karar ağacı kapsamında örnek olarak veri hikayeciliğinin verileri kullanıp hedeflenen karar ağacı oluşturulmuş ve verilerin verileri inceleyen araştırmacı, kullanıcı ya da anlamaya çalışan kişilere daha iyi anlatabilmek adına görselleştirilmesi için doğru grafiksel metodların gösterilmesi ve kullanıcının verisinin hikayesine göre grafiksel metodun kullanılması sağlanmıştır.

Karar ağacı oluşturmak için Delphi dili kullanılarak farklı platformlarda çalışabilen bir uygulama ve javascript, html, css gibi programlama dilleri yardımıyla web tasarımı geliştirilmiştir. Geliştirilen web sayfasının güvenliği ve takibi için üyelik sistemi oluşturulmuştur.

Oluşturulan sistem farklı test araçları kullanılarak test edilmiş ve hatadan büyük ölçüde arındırılmıştır.

Yıl : 2022

Sayfa Sayısı : 55

Anahtar Kelimeler : Dinamik Karar Ağacı, Bilgisayar Yazılımı, Web Tasarımı

Master Thesis
Web based Dynamic Decision Tree an Application
T.U Graduate School of Natural and Applied Sciences
Department of Computer Engineering

ABSTRACT

The aim of this thesis is to create a visual and user-friendly decision tree environment. For this purpose, two different applications have been developed to create a study-specific decision tree and to use the created decision tree in the web environment.

It is aimed to transform the existing data in the hands of the user into a decision tree. Within the scope of the created decision tree, as an example, the targeted decision tree was created by using the data of data storytelling, and in order to better explain the data to the researcher, user or people who are trying to understand the data, it was ensured that the correct graphical methods were shown and the graphical method was used according to the story of the user's data.

Decision tree and a web design was developed with the help of programming languages such as javascript, html, css. A membership system has been created for the security and monitoring of the developed web page.

The created system has been tested using different test tools and has been largely free of errors.

Year : 2022

Number of Pages : 55

Keywords : Dynamic Decision Tree, Computer Software, Web Design

ÖNSÖZ

Tez çalışmam sırasında bilgi ve tecrübesini benimle paylaşan Sayın danışman hocam Doç. Dr İlhan UMUT'a sonsuz teşekkür ve saygılarımı sunuyorum.

İÇİNDEKİLER

ÖZET iv

ABSTRACT v

ÖNSÖZ vi

İÇİNDEKİLER vii

ÇİZELGELER DİZİNİ xi

ŞEKİLLER DİZİNİ xii

BÖLÜM 1 1

GİRİŞ 1

BÖLÜM 2 3

LİTERATÜR ARAŞTIRMASI 3

2.1. Karar Ağacı Nedir? 3

2.2.Karar Ağacı Yapısı 4

2.3. Karar Ağacı Kullanım Alanları Nerelerdir? 5

2.4. Karar Ağacı Nasıl Çalışır? 6

2.5. Kar Ağacı Avantajları Ve Dezavantajları Nelerdir? 7

2.6. Karar Ağacları Temel Kavramları 8

2.6.1 Kök Düğüm 8

2.6.2 Bölme 9

2.6.3 Karar Dügümü 11

2.6.4 Yaprak Düğümü 11

2.6.5. Üst / Alt Düğüm 12

2.7.Karar Ağacı Oluşturma 13
BÖLÜM 3 20
3.1. Kara Ağacı Oluşturma Aracı 20
3.2.Web Sitesi Oluşturulurken Kullanılan Programlama Dili ve Yöntemler 21
3.3 Veri Hikayemiz için Uygun Grafik Yapısını Seçen Karar Ağacı 34
BÖLÜM 4 38
SONUÇ 38
KAYNAKLAR 39
ÖZGEÇMİŞ 44

ÇİZELGELER DİZİNİ

Çizelge 2.1. Harf not sistemi. 10
Çizelge 2.2. Spor salonu üyelik süresine göre terk durumu..... 14
Çizelge 2.3. Cinsiyet, boy, kilo ve beden çizelgesi. 16

ŞEKİLLER DİZİNİ

Şekil 2.1. Ağaç yapısı örneği. 4
Şekil 2.2. Örnek karar ağacı. 9
Şekil 2.3. Çizelge 2.1.'in karar ağacına dönüştürülmesi. 10
Şekil 2.4. Karar ve yaprak düğümleri (Güner, 2008). 11
Şekil 2.5. Yaprak düğümleri (Diagrams, 2005). 12
Şekil 2.6. Üç sınıftan oluşan karar ağacı yapısı (Kavzaoğlu, 2010)..... 13
Şekil 2.7. Karar ağacı örneği..... 14
Şekil 2.8. Boy değişkeninin kök değişken olarak seçme. 17
Şekil 2.9. Cinsiyet değişkenini kök değişken olarak belirleme..... 18
Şekil 3.1. Program görsel arayüzü..... 20
Şekil 3.2. Kayıt ol ekranı. 21
Şekil 3.3. Kayıt-ol.php dosyasının veri tabanı ile yapılan bağlantısını gösterir kod bloğu. 22
Şekil 3.4. Kullanılan veri tabanı. 22
Şekil 3.5. Ders veri tabanı ve üyeler tablosu..... 23
Şekil 3.6. Kayıt olan kullanıcı bilgilerini gösterir veri tabanı örneği. 23
Şekil 3.7. Kayıt e-mail ve şifre kontrol scripti. 24
Şekil 3.8. config.php dosyası kod bloğu. 25
Şekil 3.9. Giriş ekranı – index.php sayfası. 25
Şekil 3.10. index.php giriş kontrol kod bloğu. 26
Şekil 3.11. Karar ağacı oluşturmak için karar ağacı verilerini girdiğimiz ekran. 27
Şekil 3.12. Kullanılan recursive fonksiyon. 29
Şekil 3.13. Kullanılan fonksiyon. 30
Şekil 3.14. showCard fonksiyonu..... 30
Şekil 3.15. Oluşturulan css dosyası. 32

Şekil 3.16. Verinin animated.php dosyası içerisinde karar ağacına dönüştürülmesi örneği. 33
Şekil 3.17. Oluşturulması gereken karar ağacı..... 34
Şekil 3.18. Veri hikayen nedir?..... 35
Şekil 3.19. Birden fazla veri seti var mı?..... 36
Şekil 3.20. Kullanılması gereken uygun grafik..... 36

BÖLÜM 1

GİRİŞ

Günümüzde gelişen bilişim sektörüyle birlikte veriye ulaşma hızımızda artmıştır. Bununla birlikte calışmalarımıza veri sağlayacak ortamlarda hızla gelişmiş ve çalışmalarımızda kullanılmak üzere veri yığınlarını oluşturmuştur.

Bazen elimizde bulunan verilerimizi daha anlaşılır hale getirmek için görşelleştirmeye ihtiyaç duyabiliriz. Çok miktarda veriye sahip olan kullanıcının verilerinin içinden anlamlı sonuçlar üretebilmesi için verilerinin belirli düzende olması gerekmektedir. Aksi taktirde karmaşık olan verilerden anlamlı sonuç üretmek isteyen kullanıcı cok fazla zaman harcayıp tüm verilerini süzememe gibi sorunlarla karşılaşıp doğru sonuçlar alamayabilir. Bunun için verilerini öncelikle belirli özniteliklere göre ayırmalı ve bir agaç yapısına dönüştürmesi gerekmektedir. Karar ağacları ana dügüm, ara dügümler ve yaprak düğümlerden oluşmaktadır.

Bu çalışmada veri hikayeciliği verileri kullanılarak karar ağacı oluşturulmuştur. Veri hikayeciliğinde asıl amaç verileri görselleştirmek olduğunu anlayabiliriz. Tufte ve Schmieg (1985), verilerin görselleştirmek için kullanılan grafiklerin, verilerin hikayesine göre grafik mükemmelliği içermesi gerekliliğini ortaya koymaktadır. Aynı fikirde olan Ryan (2016) görsel açıdan verilerin iyileştirilmesi için boyut, renk ve şekiller yoluyla anlamı daha da görünür hale getirmek, veri hikayesinden cıkan verilmek istenen mesajın net bir şekilde kullanıcıya iletmek olduğunu söylerler. Veri hikayeciliğinin amacı; verilerin, grafiklerin, şekillerin ve anlatımın birleştirerek araştırmacıya daha anlaşılır bir şekilde iletmeyi hedeflemektedir (Dykes, 2015). Veri hikayeciliği anlatımı verilerde nelerden bahsettiği ya da verilerde üzerinde durulan bir özelliğin ya da niteliğin nedenli önemli olduğunu açıklamayı amaçlar. Veri hikaye anlatımı hedef odaklı çalışır yani anlatılmak istenene odaklanır. Bu yüzden verilerimizin hikayelerine göre görselleştirme

yapılması anlatılmak istenilenin açıkca grafik ve şekiller yordamıyla anlatılmaya çalışılması hedeflenmektedir. Veri hikayeciliği uygun bir grafik, görsel veya şekil seçmeyi hedef edinmektedir. Belirli görselleştirme teknikleri belirli veri hikayeleri ve amaçları için daha iyi çalışmaktadır. Örneğin, çizgi grafikler zaman içerisindeki değişiklikleri etkili bir şekilde göstermektedir (Ryan, 2016).

BÖLÜM 2

LİTERATÜR ARAŞTIRMASI

2.1. Karar Ağacı Nedir?

Karar ağacı, akış şemasına benzer bir yapıdadır. Bu ağaç yapısına dahil olan tüm düğümler bir özniteliği temsil eder. Her dal ise bir testin sonucunu temsil ederken, her yaprak düğümü bir sınıf etiketini temsil eder (Sharma & Kumar, 2015).

Bir başka ifadeye göre; karar ağaçları olası sonuçları ve bu sonuçlardan sonra oluşan kararları göstermeye çalışan diyagramlardır. Örneğin; üniversiteye gitmek istiyor olabilirsiniz karar ağaçları üniversiteye gitme ihtimalinizi olası niteliklere göre zamanınızı hangi etkinliklere harcayarak üniversiteye gitme ya da gidememe durumunuzu sonuca ulaştırabilir. Karar ağacı her sınıflandırma ve Regresyon sorunu için kullanılabilen bir tür denetimli öğrenme tekniğidir (Cockerham, 2018).

Karar ağaçları aslında bir veri kümesi üzerinde yapılan çalışmada verilerin daha ayrıntılı ve daha yalın görünebilmesi için bir ağaç şeklinde tasarlanmasını ifade eder. Bu veri kümesinin ağaç şeklinde dizayn edilmesinin beraberinde karar ağaçları aslında bir sınıflandırma örneğidir. Karar ağaçları belirli veri kümelerinin sınıflandırmaya yarar (Patel, 2012).

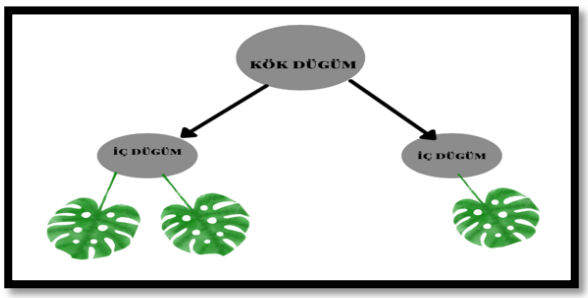
Milyonlarca örnek ve binlerce öznitelik içeren büyük veri kümelerine günümüzde kolay ulaşıldığı için, hızlı karar ağacı geliştirme algoritmalarına ilgi hızla artmaktadır. Karar ağacını oluşturan kök düğüm ve yaprak düğümler aslında bir dizi etiketlenmiş veri setinden oluşmaktadır. Her alt küme için algoritma, bir alt kümedeki tüm örnekler aynı kümeye ait olana kadar özyinelemeli olarak ilerler (Su, 2006).

Karar ağacı aynı zamanda seçim ağacı olarak bilinir. Dalları üzerinde genişleyen ağaç benzeri bir yapı oluşturur ve bir temel düğümle başlar buna kök düğüm denir.

2.2.Karar Ağacı Yapısı

Karar ağaçları karar düğümleri ve yaprak düğümlerinden oluşur. Ağaç formunda görünür ve bir sınıflandırma yöntemidir. Bu yaprak düğümler bir sınıflandırmayı ya da kararı temsil eder. Karar ağaçları üzerinde çalışma yapılan veri setinin küçük parçalara ayrılması ile oluşturulur. Karar ağaçları bir ya da birçok kez dallanma oluşturabilir. Kök düğüm ağacımızdaki ilk düğümümüzdür (Saedsayad, 2021).

Karar düğümü ve Yaprak Düğümü olarak adlandırılan iki düğüm vardır. Karar düğümü karar vermek için kullanılır. Yaprak düğümü ise karar düğümünden sonra gelen bir sonucu ifade eder (Decision Tree Algorithm, 2021).



Şekil 2.1. Ağaç yapısı örneği.

Şekil 2.1'de gösterildiği gibi kök düğüm bizim ağacımızdaki ilk düğümümüzdür diğer düğümler iç düğüm olarak adlandırılır. İç düğümler daha sonra alt düğümleri ayrılabilir. Bir iç düğünde yaprak oluşmuşsa eğer sınıflandırma bitmiş demektedir. Yani o düğümün sınıflandırma işlemi tamamlanmıştır. Diğer düğümlerinde yaprakları oluşuncaya kadar alt düğümlere ayrılabilirler.

2.3. Karar Ağacı Kullanım Alanları Nerelerdir?

Karar ağaçları büyük veri setlerini sınıflandırarak bu veri seti üzerinde çalışan kişiler için anlaşılması kolay bir ağaç görseli oluşturur. Bu ağaç görseli insanların veri setini daha kolay kavramasını sağlar.

Karar ağaçlarının kullanım alanlarına gelecek olursak;

- Bir topluluk üyesinin ekonomik geçimini desteklemek için tele merkezi kullanım kararını verdiği süreci göstermek için etnografı bir karar ağacı modeli geliştirilmesi (Ngwenyama, 1999),
- Sağlık yönteminde veri madenciliği kullanımının karar ağacına uygulanması (Bach, 2008),
- Elektrik enerjisi tüketimini tahmin eden karar ağacı geliştirilmesi (Geoffrey, 2007),
- Tıbbi veri kümelerinde karar ağacı sınıflandırıcılarının performans değerlendirmesi (Lavanya, 2011),
- Akıllı şebekelerde hırsızlık tespiti için karar ağacı geliştirilmesi (Jindal, 2016),
- Sosyal medya kullanımına uygulanan karar ağacı (Geoffrey, 2007),
- Kablosuz ağlar aracılığıyla kapsama restorasyonu kullanımı hakkındaki karar ağaçları (Mutairi, 2007),

Yukarıdaki örneklerden de anlaşılacağı üzere; Karar ağaçları sosyal, psikolojik, sağlık, otomotiv, bilişim gibi birçok alanda kullanılmaktadır. Karar ağaçları genel olarak alternatif seçimler yaparken kullanılır. Yukarıda bahsedilen örnekler günümüzde çalışmaları yapılmış örneklerdir. Aslında karar ağaçları araştırmaya konu olmuş ve bir seçim yapılabilen her hipotez için kullanılabilir. Karar ağacı kullanımı veri yığınının daha kolay anlaşılması ve kullanımı kolay bir ağaç yapısıyla ayrıntılara boğulmuş bir metinsel ifadeden çıkıp görsel bir ağaç yapısı ile araştırmacının sonuca ulaşmasını sağlar.

Yukarıda sayılan alanlara ek olarak karar ağaçları;

İş: İş modellerinin görselleştirilmesinde kullanılır. Ayrıca müşteri ilişkileri yönetiminde ve kredi kartı kullanıcıları için kredi puanlaması için kullanılır.

İzinsiz giriş tespiti: Bir saldırı tespit uzman sistemi için otomatik olarak kurallar oluşturmak için kullanılır.

Enerji modellemesi: Binalar için enerji modellemesi, binalar için önemli oluşturulması gereken yapılardan biridir.

Görüntü işleme,

Tıp: Tıbbi araştırma ve uygulama, karar ağacı tekniklerinin önemli uygulama alanlarıdır. Kalp sesi teşhisi içinde kullanılır.

Endüstri: Üretimde kalite kontrol gibi karar ağacı algoritmasının faydalı olduğu alanlardır.

Akıllı Araçlar: Yolun şerit sınırlarını bulma işi, akıllı araçların geliştirilmesinde önemli bir görevdir.

Uzaktan algılama: Karar ağaçlarıyla örüntü tanıma çalışmaları için güçlü bir uygulama alanıdır.

Web uygulamaları: Büyük internet sitelerinde başarısızlıkları tespit etmek için karar ağacı yaklaşımı kullanılabilmektedir (Patel & Rana, 2014).

2.4. Karar Ağacı Nasıl Çalışır?

Bir karar ağacında veri kümesinin kategorisini tahmin etmek için temel kural olarak algoritma ağacın temel düğümünden yani kök düğümünden başlanır. Kök düğümün özniteliğinin değerleri veri kümesinin özniteliği ile karşılaştırılır. Ve karşılaşmayı destekleyen dal takip edilip bir sonraki dala atlanır. Bu şekilde bütün dallarda öznitelikler karşılaştırılarak bir sonraki dala atlanır. İstenilen sonuç elde edilene kadar bu adımlar takip edilir (Decision Tree Algorithm, 2021).

2.5. Kar Ağacı Avantajları Ve Dezavantajları Nelerdir?

Karar ağacı avantajları;

- Anlaması ve yorumlanması basittir ve ağaçlar görselleştirilebilir.
- Küçük veri hazırlığı gerektirir. Bu hazırlıklar eksik verileri tamamlamak, çıkarmak gibidir.
- Sayısal ve logaritmik verileri işleyebilir.
- Çoklu çıkış gerektiren problemleri çözebilir.
- Veri yığınlarının karar ağaçlarına dönüştürerek şekil üzerinden sonuca ulaşmak daha kolaydır. Metinsel ifadeler arasından sonuca ulaşmamız zaman almaktadır.
- İstatistiksel testler kullanarak bir modeli doğrulamak mümkündür.
- Verilerin üretildiği gerçek model tarafından çıkan sonuç ihtimalleri bir şekilde ihlal edilse bile iyi bir performans göstermektedir.
- Sınıflandırma ve Regresyon için kullanılabilirler (Scikit-Learn, 2007).

Karar ağacı dezavantajları;

- Bir karar ağacının oluştururken kök ve yapraklar belirli öznitelikler göz önünde bulundurularak oluşturulmalıdır. Çalışmış olduğumuz veri yapısının bütün verilerini ağaca uyguladığımız durumlarda aşırı öğrenme gibi sorunlarla karşılaşabiliriz bu durum da oluşturduğumuz ağacımızdan doğru sonuçlar almamızı engeller. Bu tarz durumlardan kaçınmak için verilerimizi bazı yöntemlerle küçültmeye gitmemiz gerekmektedir. Bu yöntemlerden bazıları veriye uygulanan budama yöntemi, yaprak sayısını azaltmak ve ağacın minimum numune sayısının ve ağacın maksimum derinliğinin ayarlanması gerekmektedir.
- Karar ağaçları kararsız olabilir çünkü verilerdeki küçük değişiklikler tamamen farklı bir ağacın üretilmesine neden olabilir.
- Karar ağaçlarında eğer bazı sınıflar baskınsa, önyargılı ağaçlar oluşturulabilir. Bundan dolayı karar ağacı oluşturulmadan önce veri yığınları düzenlenmesi yapılmalıdır.

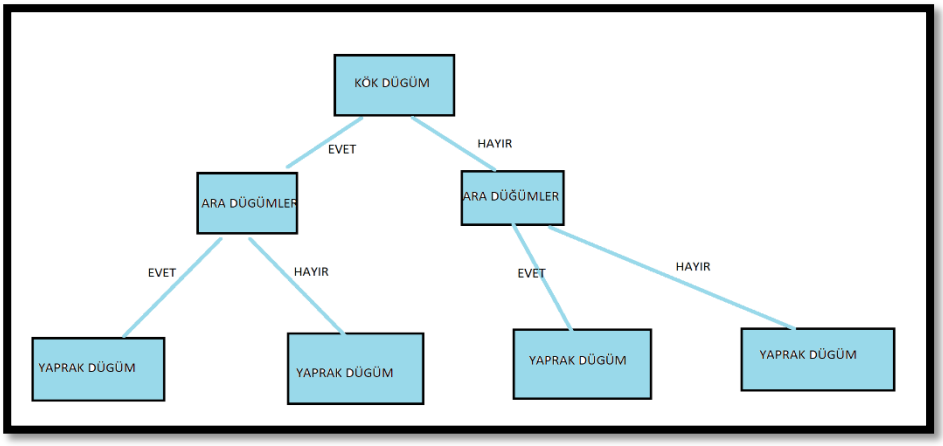
- Zayıf öğrenicilerdir: tek bir karar ağacı normalde büyük tahminlerde bulunamayabilir, bu nedenle daha güçlü topluluk modellerini doğurmak için birden fazla ağaç birleştirilerek ormanlar oluşturmak gerekebilir.
- Karar ağacının matematiksel hesaplaması çoğunlukla daha fazla hafıza gerektirir.
- Karar ağacının matematiksel hesaplaması çoğunlukla daha fazla zaman alır.
- Verilerdeki küçük bir değişiklikler ağaç yapısında büyük değişikliklere neden olabilir.
- Karar ağacı modelinin uzay ve zaman karmaşıklığı modeli nispeten daha yüksektir.
- Karmaşıklık yüksek olduğundan karar ağacı modeli eğitim süresi daha fazladır (Scikit-Learn, 2007; Towardsdatascience, 2020).
- Karar ağaçlarında bölme işlemi sorunu veriler kademeli olarak daha küçük parçalara bölünürse karar ağacımız doğru çıkarım yapamaz.
- Tekrarlama sorunu, özellikler bir karar ağacında tekrar tekrar test ediliyorsa tekrarlama problemi mevcuttur. Bu tekrarlar verileri daha küçük ve daha küçük segmentlere böler, dolayısıyla parçalanma ile sonuçlanır (Patel & Rana, 2014).

2.6. Karar Ağacları Temel Kavramları

Karar ağcı temel kavramları kök düğüm, alt düğümler ve yapraklardan oluşmaktadır.

2.6.1 Kök Düğüm

Karar ağaçlarının ilk hücrelerine kök denir. Her bir yaprak kökteki koşula göre evet ya da hayır olarak sınıflandırılır. Kök düğümü karar düğümü olarak da adlandırılmaktadır (Chakure, 2019).



Şekil 2.2. Örnek karar ağacı.

2.6.2 Bölme

Bölme, karar düğümünü verilen koşullara göre alt düğümlere ayırma işlemidir. Karar ağaçları özyinelemeli bölünme adı verilen bir buluşsal yöntem kullanılarak oluşturulur. Bu yaklaşım aynı zamanda yaygın olarak kullanılan böl ve yönet olarak da bilinir (Chakure, 2019).

Verileri benzer sınıfların daha küçük ve daha küçük alt kümelerine dönüştürür. Tüm veri kümesini temsil eden kök düğümden başlayan algoritma hedef sınıfın en öngörülü olan bir özelliğini seçer. Daha sonra bu özelliğin farklı değerlerinden oluşan gruplara bölünür ve karar ağacın ilk dalları belirlenmiş olur. Algoritma düğümleri bölmeye devam eder.

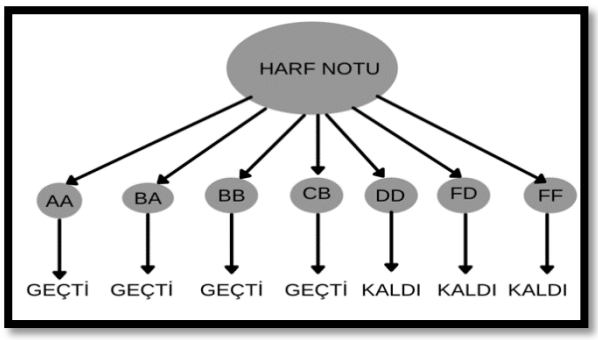
Algoritma alt kümelerdeki verilerin yeterince homojen olduğunu belirlendiğinde süreç durana kadar bu böyle devam eder ya da istenilen bir kriter sağlandığı taktirde durur.

- Düğümdeki örneklerin tümü aynı sınıfa aittir.
- Ağaç, önceden tanımlanmış tanımlanan boyutuna ulaştı (Lantz, 2015).

Çizelge 2.1. Harf not sistemi.

HARF NOTU	100 ÜZERİNDEN DEĞERİ	DURUMU
AA	90-100	GEÇTİ
BA	80-99	GEÇTİ
BB	70-79	GEÇTİ
CB	65-69	GEÇTİ
DD	50-59	KALDI
FD	30-49	KALDI
FF	0-29	KALDI

Yukarıdaki çizelgeyi karar ağacı yapısına dönüştürmek için öncelikle karar ağacından beklentimizin ne olacağı sorusuna cevap bulmamız gerekmektedir. Öğrencinin almış olduğu notlara göre öğrencinin durumunu bizlere gösteren bir karar ağacı oluşturalım.



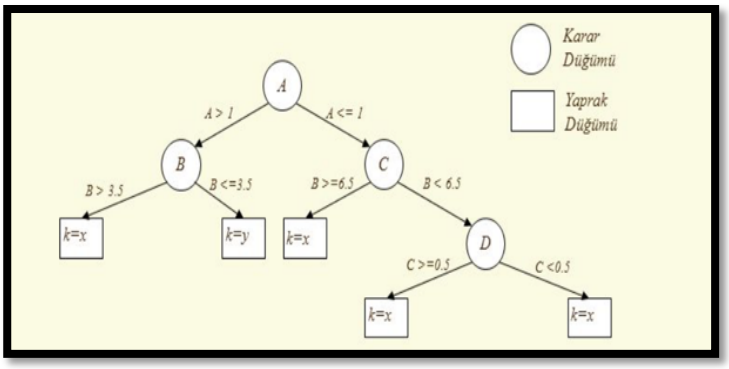
Şekil 2.3. Çizelge 2.1.'in karar ağacına dönüştürülmesi.

Şekil 2.3'te görüldüğü üzere çizelgedeki veriler yedi alt düğüme bölünmüştür. Harf notu kök düğümümüzdür.

2.6.3 Karar Düğümü

Karar ağaçları sınıflandırma veya regresyon modellerini bir ağaç yapısı şeklinde oluşturur. Bir veri kümesini giderek daha küçük alt kümelere ayırırken aynı zamanda ilişkili bir karar ağacı aşamalı olarak geliştirilmiş olur. Bütün bu alt kümelere ayrılan veriler karşımıza karar düğümleri ve yaprak düğümleri olarak ortaya çıkar. Bir karar düğümünün iki veya daha fazla alt dalları olabilir. Kısacası bir alt düğüm başka alt düğümlere bölünürse bu düğümlere karar düğümleri diyoruz (Decision_tree, 2021).

Tam olarak bir karar ağacında karar düğümleri ve yaprak düğümleri vardır. Karar düğümleri herhangi bir karar vermek için kullanılır ve birden çok şubeyi sahipken, yaprak düğümleri bu kararların çıktısıdır ve başka dallar içermez (Ichi, 2020, 2021).



Şekil 2.4. Karar ve yaprak düğümleri (Güner, 2008).

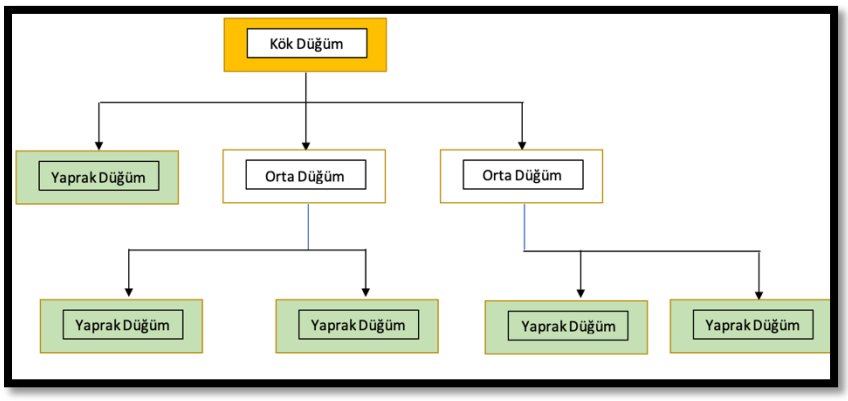
2.6.4 Yaprak Düğümü

Karar ağaçları hem sınıflandırma hem de regresyon için kullanılan parametrik olmayan bir model olarak görülmektedir. Karar ağaçları daha fazla özellik eklendikçe ve bu karar ağaçları doğru bir şekilde oluşturulursa, parametre sayısını artırmayan esnek modeller olduğu veya kategorik bir tahminde bulanabildikleri, bir hayvan türünün belirli bir hayvan türünden olup olmadığının belirlenmesi, ya da sayısal bir tahmin gibi (Towardsdatascience, 2020).

Karar ağaçları birkaç elemandan meydana gelmektedirler bunlar; düğümler: kök düğüm, alt düğümler, karar düğümleri, dallardan ve yapraklardan oluşmaktadırlar. Her bir düğümde, ağacın sınıflandırma aşamasında verileri yerleştirirken bölmek ya da tahmin

yapmak ve belirli bir yolu izlemesini sağlamak için verilerimizi bazı özelliklerine göre dallara ayırmamız gerekmektedir (Koçak, 2020).

Oluştururken farklı özellikleri tekrar tekrar değerlendirerek ve her bir düğümde verileri en iyi bölen özellik kullanarak karar ağaçları oluşturulur. Bunları yaparak en iyi sınıflandırmanın yapılması sağlanmalıdır. Bir karar ağacında yaprak düğümler sınıfları göstermektedir. Bir düğümün yaprak düğümü oluşmuş ise eğer o düğüm için sınıflandırma işleminin bitmiş olduğunu anlayabiliriz. Yaprak düğümler sınıfları gösteren düğümlerdir.



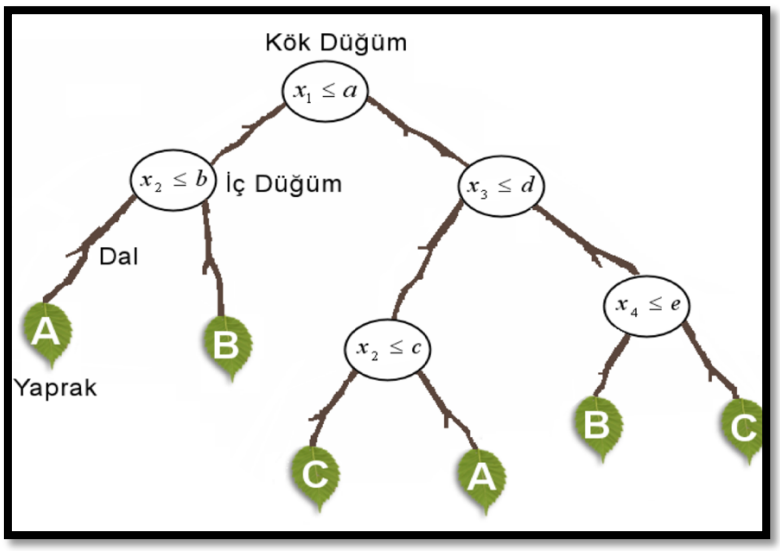
Şekil 2.5. Yaprak düğümleri (Diagrams, 2005).

Yani yaprak düğümler; oluşturulan karar ağaçlarının en son düğümleridir. Yaprak düğümler oluşturulduysa eğer, o karar ağacı için sınıflandırma işlemi yapılmış ve o düğüm için sınıflandırma tamamlanmış demektir.

2.6.5. Üst / Alt Düğüm

Ağacın kök düğümüne ana düğüm, diğer düğümlere alt düğümler denir. Kök düğümü hariç bütün düğümler genel anlamıyla alt düğüm olarak adlandırılır. Bu alt düğümlerin içerisinde yaprak düğümleri de vardır. Yani yaprak düğümleri de alt düğüm olarak adlandırılır.

Kök düğümden sonraki bütün düğümler alt düğüm olarak adlandırılırken kök düğüm ise üst düğüm olarak adlandırılır.



Şekil 2.6. Üç sınıftan oluşan karar ağacı yapısı (Kavzaoğlu, 2010).

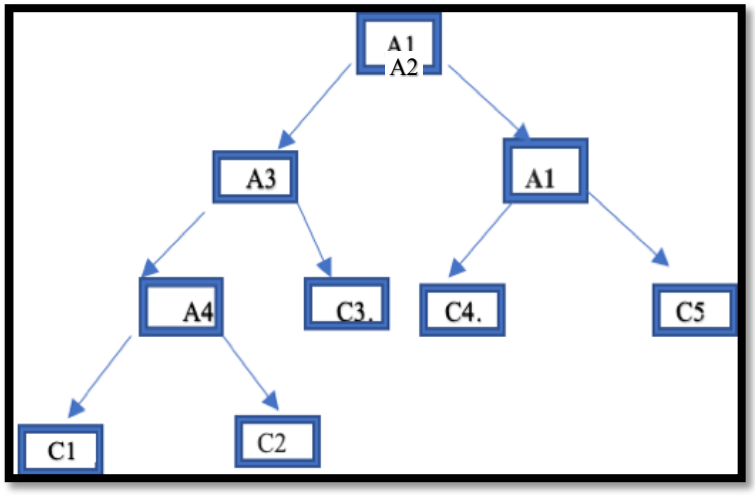
2.7.Karar Ağacı Oluşturma

Karar ağacı şu adımlarla oluşturulur;

1. Her bir düğümü A_i adıyla isimlendirilmiş olsun,
2. Her bir düğüm kendine ait dallara ayrılır ve bu dallar belirli bir sorunun cevabına göre yapraklara iletilir.
3. Ve bu düğümlerden sonra verilen cevaplara göre ağacımızın yaprakları oluşmuştur, oluşan bu yapraklar sınıflandırmanın sonlandığını göstermektedir (Dundan, 2003).

Yukarıda anlatılan adımlara göre bir karar ağacı oluşursa eğer;

Şekil 2.14'de gösterildiği gibi A1, A2, A3, A4 ile ifade edilenler kök düğümleri göstermektedir. A2 ana kök düğümdür, A1, A3, A4 ise alt kök düğümleri ifade eder. A1 düğümünden başlanarak karar vermek isteyen kişiye oluşturulan ağaç belirli sorular sorar ve bu sorulardan aldığı cevaplara göre ağacın dallarında gezinir. C1, C2, C3, C4, C5 ise ağacımızın yaprak düğümlerini ifade eder yani ağacımız bu dallara geldiğinde kararımızı vermiş oluruz. Yaprak düğüme erişmek demek karar verme işleminin tamamlanması demektir.



Şekil 2.7. Karar ağacı örneği

Oluşturmuş olduğumuz karar ağaçları sınıflandırma yapmak için belirli kurallar üretirler ve bu üretilen kurallar sonucunda sınıflandırmayı yapmayı sağlarlar. Bu kuralların yanı sıra bazı kavramları da vardır kaldıraç kavramı bunlardan bir tanesi olup karar ağacı için önem arz etmektedir. Kaldıraç kavramını şöyle ifade edebiliriz; bir düğümdeki kayıtlar için, tüm ağadaki kayıtlara kıyasla ne kadarının hedef sınıfa ait olduğunu gösteren değerdir (Silahtaroğlu, 2016).

Çizelge 2.2. Spor salonu üyelik süresine göre terk durumu.

Üyelik Süresi	**Terk**
1-5	0
6-15	1
6-15	1
1-5	0
6-15	1
6-15	0
1-5	1
1-5	0
1-5	0

Yukarıda da bahsettiğimiz gibi bir karar ağacı oluşturulurken aslında verilerimizin arasında belirli kurallar oluşturuyoruz yukarıdaki Çizelge 2.2'de belirli kurallar oluşur bu kuralları ele alacak olursak eğer;

Kural 1: 1 -5 ile sınırlı olan üyeliklerde terk durumu = 0 olanlar

Kural 2: 6-15 ile sınırlı olan üyelikler ter durumu = 1 olanlar

Kural 1'e kaldıraç uygulandığında: üyelik süresi 1-5 olan kayıt sayısı 5 olmuştur. Fakat kural 1 'de terk durumu sıfır olması gerekmektedir. Bu kayıtlardan üye süresi 1-5 olanlardan yalnızca bir tanesi terk durumu 1'dir. Toplam kayıt sayımız tüm durumlar için 9 tanedir.

(4/5)/(5/9)=1.44

İkinci kural için uygulanan kaldıraç;

(3/4)/(4/9)=1,69

Aslında biz kaldıraç işlemini uygulayarak istenilen kuralın olasılık hesaplarını yapmamızı sağlamıştır. Yani bu formülden şunu öngörebiliriz. Tüm veri kayıtlarının istenilen duruma bölümüyle oluşan ihtimal değeri hesaplanmaktadır.

Burada edinilen bilgilere göre bu kaldıraç sonuçlarını değerlendirecek olursak eğer; örneğimizdeki işletme spor salonuna üye olan hiçbir müşterisini kaybetmek istemez ve kaybetmemek için çeşitli promosyonlar yaparlar. Fakat spor salonumuz bu promosyonları yapmak yerine elinde bulunan verilerin kaldıraç değerlerini hesaplarsa, müşterinin salonu terk etme ihtimalini bularak bu yönde gerekli çalışmalar yapabilirler. Yukarıdaki örnekte kural 2'ye göre yapılan çalışmada müşterinin salonu terk etme ihtimali daha olan müşteriye ulaşma ihtimali 1,69 kat artacaktır.

Karar ağaçları oluşturulurken verimize bir algoritma uygulayıp ağacımız bu algoritma doğrultusunda oluşturabiliriz. Karar ağacı oluştururken kullanılan algoritmalar vardır. Bu algoritmalara göre de karar ağaçlarındaki şekillerde değişiklikler gösterebilmektedir.

Karar ağacımızı oluştururken kullanacağımız özniteliklerimizi doğru ve yerinde seçmemiz gerekmektedir. Ağacımıza uygulayacağımız verimizdeki kök düğümleri de seçerken dikkatli olmalıyız. Yanlış bir kök düğümü seçmek ağacımızın gereksiz yere

dallanmasını ve sonuca ulaşmak için gereksiz zaman kaybına sebep olabilirken bunun yanı sıra ağacımızın aşırı yüklenmesi ve yanlış sınıflandırma değerleri üretmesine sebep olabilmektedir.

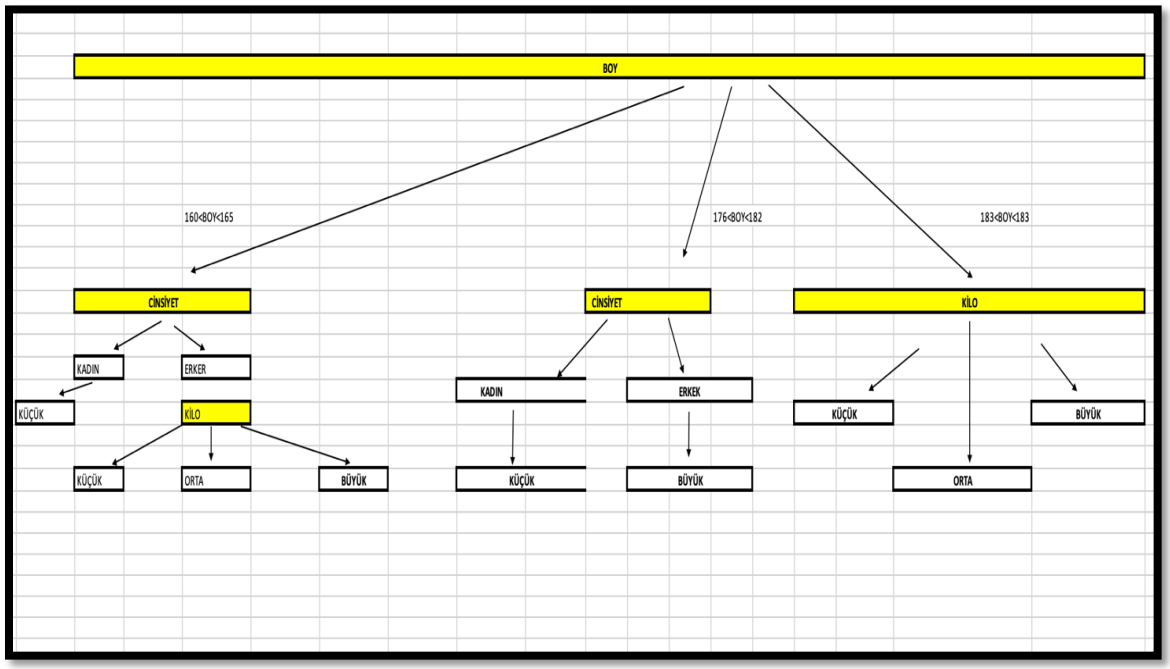
Karar ağacımızı oluştururken veri yığınlarımızın arasından en iyi sınıflandırma sonucunu verecek kök düğümlerimizi belirlememiz gerekmektedir. Bunun yanı sıra en iyi öznitelik değerlerimizi de seçmemiz gerekmektedir. Veri yığınımız ağaca en uygun şekilde yerleştirilmelidir. Oluşturulan her ağaç dalı ağacımızın bir kuralını oluşturmaktadır. Bir örnek ile ağaçta seçilen kök değişkenlerinin ve öznitelik değerlerinin önemini anlatmaya çalışalım.

Çizelge 2.3. Cinsiyet, boy, kilo ve beden çizelgesi.

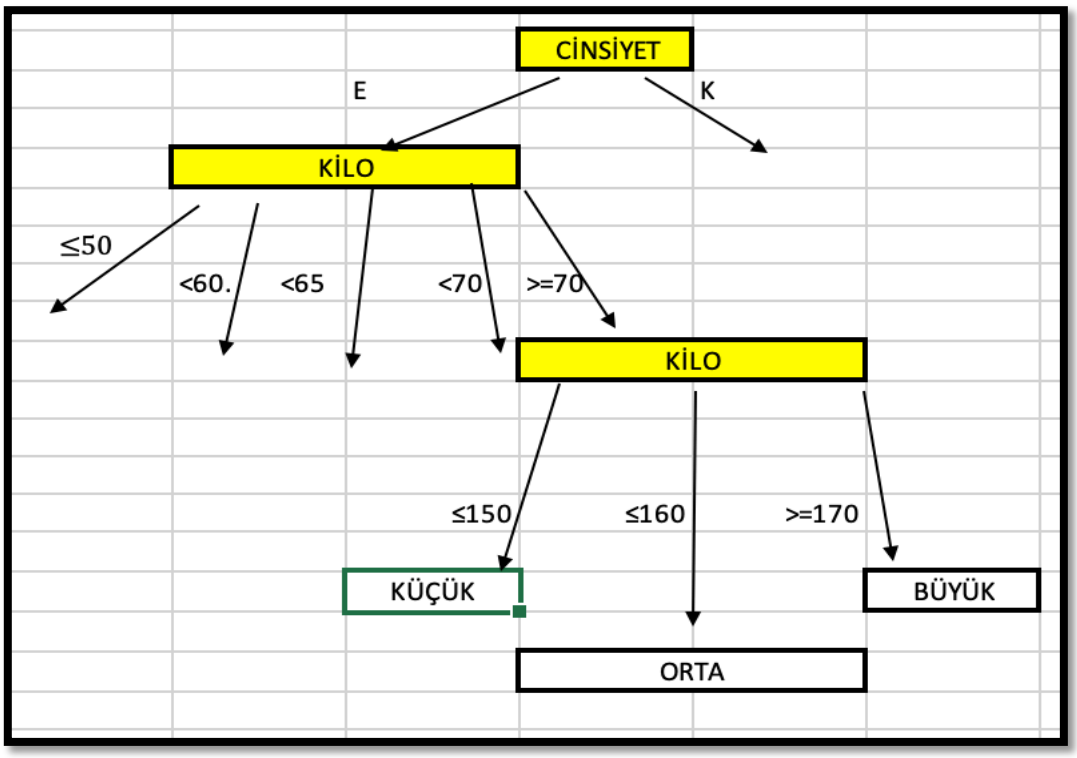
CİNSİYET	**KİLO**	**BOY**	**BEDEN**
K	48	171	ORTA
K	49	151	KÜÇÜK
K	52	158	ORTA
K	56	165	ORTA
E	59	160	KÜÇÜK
K	61	159	ORTA
E	62	162	KÜÇÜK
E	63	174	ORTA
K	68	168	ORTA
K	69	177	BÜYÜK
E	72	170	ORTA
E	74	165	KÜÇÜK
E	85	175	ORTA
E	85	190	BÜYÜK
E	98	190	BÜYÜK

Çizelge 2.3‘te görüldüğü gibi cinsiyet, boy, kilo birer kök değişkenlerdir. Beden ise sınıf değişkenidir. Bu çizelgede kişinin cinsiyet, boy, kilo değerlerine göre hangi beden sınıfına dahil olduğunun bulunması istenmektedir. Aslında burada değişkenlerden bahsetmek gerekirse; beden değişkeni bağımlı değişken tanımına uyarken cinsiyet, boy ve kilo değişkenleri bağımsız değişkenleri ifade etmektedir. Beden değişkeni cinsiyet, boy ve kilo değişkenlerine bağımlı olarak değişmektedir.

Çizelge 2.3‘teki verileri kullanarak bir karar ağacı oluşturmak istenirse eğer buradaki en önemli kriter hangi değişkenin kök düğüm yani ilk düğüm olması gerektiği belirlenmelidir. Kök düğüm belirlendikten sonra ağacın alt dalları aynı hassasiyetle belirlenmelidir. Verilen çizelgedeki verilerden birden fazla ağaç modeli oluşturulabilmektedir. Fakat bize en doğru sınıflandırmayı verecek olan tek bir ağaç modeli olacaktır.



Şekil 2.8. Boy değişkeninin kök değişken olarak seçme.



Şekil 2.9. Cinsiyet değişkenini kök değişken olarak belirleme.

Verilen çizelgeyi karar ağacına dönüştürdüğümüzde birden fazla karar ağacı modeli oluşturabiliriz. Genel anlamda bu ağaçların yapmış olduğu sınıflandırmalar benzer sonuçlar doğrulabilir. Bu durum ise değişken sayısının durumu ile değişiklik göstermektedir. Fazla miktarda değişken olması sınıflandırmanın benzer olmasına olanak tanır. Şekil 2.8 ve Şekil 2.9'da yapılan çalışmada kök değişkenlerinin farklı seçerek ortaya çıkan ağaç modellerini karşılaştırdık. Bu karşılaştırmadan anlaşılacağı üzere farklı seçilen kök değişkenleri ağacın modelini değiştirdiği gibi dallanma ve düğümlerin ayrılma kurallarında da değişiklik göstermektedir.

Yukarıdaki veriler iki tür ağaç örneğinde gördüğümüz gibi ağacın sınıflandırma yaparken dallara ayrılma kriterlerini belirlemek çok önemlidir. Mesela cinsiyet kökünü dallara ayırmak istersek iki dal olarak karşımıza çıkmaktadır. Fakat boy kökünü kullanmak istediğimizde ise birçok dala ayrılabilmektedir ayrıca erkek için ayrı dallar kadın için ayrı dallar olacak şekilde hesaplanır. Bu kararı vermemizde sınıf kategorileri bize yardımcı olacaktır.

Ağaç üzerinde oluşan düğüm ve dal sayıları fazla olması, ağacın alt düğümleri ve dallarına düşen veri sayısının azalmasına sebep olacaktır bu da ağacın sınıflandırma hassasiyetini azaltacaktır (Cabena, 1998).

BÖLÜM 3

3.1. Karar Ağacı Oluşturma Aracı

Uygulama RAD Studio yazılım geliştirme uygulaması kullanılarak Delphi dili ile yazılmıştır. Bu araç dinamik olarak oluşturulacak web sayfanın verilerini görsel olarak oluşturmak için geliştirilmiştir. Programın görsel arayüzü Şekil 3.1' de görüldüğü gibidir. Kişi istediği karar ağacını rahatlıkla oluşturup Şekil 3.11'deki arayüzde girilecek txt şablonu haline getirebilir. Böylece her hangi bir kod veya txt şablonu yazmadan dinamik bir web sitesi oluşturup yayınlauyabilir.



Şekil 3.1. Program görsel arayüzü.

3.2.Web Sitesi Oluşturulurken Kullanılan Programlama Dili ve Yöntemler

Web sitesi oluşturulurken kullanılan programlama dilleri; html, php, javascrpit, css'dir. Kullanıcı oluşturulan karar ağacına bir üye giriş ile erişim sağlamaktadır. Bu erişim için kullanıcının üye olması gerekmektedir. Üyelik bilgilerinin kayıt edilebilmesi ve doğruluğunun sağlanabilmesi için kişi oluşturulan siteye kayıt olma işlemleri için bir veri tabanı kullanılmıştır. Kullanılan veri tabanı işlemleri ise HeidiSQL üzerinde yapılmıştır. Karar ağacı oluşturmadaki temel amacımız kullanıcının belirli değişkenler arasında seçim yapıp bu değişken türüne göre ağaçların dallarında üzerinde gezinmesi hedeflenerek en iyi sınıflandırmanın yapılması sağlanmaya çalışılmıştır. Kullanıcının ağaç üzerinde gezinmesi ve yaprak düğüme ulaşması için bazı sorular ve kullanıcıdan bu sorulara karşılık cevaplar alınıp ağacın dalları üzerinde gezinmesi sağlanmıştır.

Oluşturulan karar ağacına giriş yapabilmek için kullanıcıların web sitesine üye olmaları ve kontrollü bir şekilde girişleri sağlanmıştır. Sayfamıza üye olmak isteyen kullanıcı 'üye ol' butonunu tıklayarak üye olması sağlanmaktadır. Kayıt olma sayfası aşağıdaki görselde görülmektedir.



Şekil 3.2. Kayıt ol ekranı.

Kullanıcı Şekil 3.2'de görüldüğü üzere isim soy isim, e-mail ve bir şifre oluşturarak web sitemize kaydını yapabilmektedir. Kullanıcının kaydı veri tabanımıza işlenmektedir. Kullanıcı kayıt olduktan sonra giriş yap diyerek e-mail ve şifresini girerek karar ağacı oluşturacağımız ekrana girişi sağlanmaktadır.

```php
kayit-ol.php
<?php
    require 'config.php';

    if ( $_POST ) {

        $isim_soyisim = $_POST['isim_soyisim'];
        $email        = $_POST['email'];
        $sifre        = $_POST['sifre'];

        $id = DB::insert("INSERT INTO uyeler(uye_adi,email,sifre) VALUES(?,?,?)",array(
            $isim_soyisim,
            $email,
            $sifre
        ));

        header("Location:index.php?success=1");
    }
?>
```

Şekil 3.3. Kayıt-ol.php dosyasının veri tabanı ile yapılan bağlantısını gösterir kod bloğu.

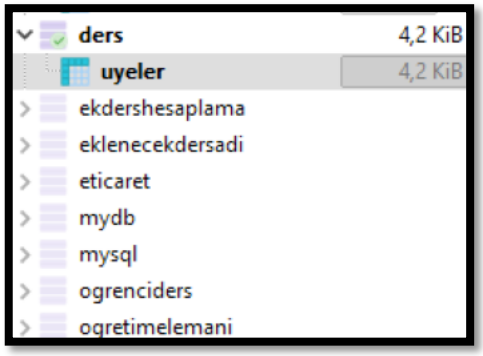
Bu kod bloğunda kullanılan post yöntemi Hyper text transfer protokolü tarafından desteklenen veri kaydetme için izin isteyen bir istemcidir. Bu yöntem sayesinde kullanıcı veri tabanında erişim sağlayıp girilmiş olan bilgileri veri tabanına INSERT yöntemiyle eklememizi sağlıyor.

Veri tabanı aracı olarak ise HeidiSQL kullanılmıştır. HeidiSQL, MySQL'i desteklemektedir. Aynı zamanda SQL Server içinde kullanılan acık kaynaklı bir sunucu bilgisayardan hizmet almayı talep eden diğer bütün client bilgisayarlar için kullanılan bir veri tabanıdır.

MySQL'i içeren HeidiSQL'de bir ders adındı bir veri tabanı ve üyeler adında bir tablo oluşturulmuştur bu veri tabanında id, uye_adi, email ve şifre formlarında alanlar bulunmaktadır. Bu alanlara kayıt olan kullanıcının bilgileri işlenmektedir.



Şekil 3.4. Kullanılan veri tabanı.



Şekil 3.5. Ders veri tabanı ve üyeler tablosu

Şekil 3.5'de kullanılan veri tabanındaki ders isimli veri tabanı ve ders isimli veri tabanı altında oluşturulan üyeler tablosu gösterilmektedir.

id	uye_adi	email	sifre
36	duru besevli	durubesevli@trakya.edu.tr	*******
33	fatma sertkaya	fatmasertkaya@trakya.edu.tr	*******
29	mintaze sertkaya	mintazesertkaya@trakya.edu.tr	*******

Şekil 3.6. Kayıt olan kullanıcı bilgilerini gösterir veri tabanı örneği.

Şekil 3.6'da kayıt olan kullanıcıların veri tabanı tablosundaki bilgilerini göstermektedir.

Kayıt işlemlerini tamamlayan ve kayıt olan kullanıcı giriş yap butonuna tıklayarak giriş yapabilmektedir. Veri tabanından e-mail ve şifre bilgilerinin kontrolü sağlandıktan sonra giriş başarılı olursa, oluşturmak istediği karar ağacını düzenleyeceği ekrana ulaşacaktır.

```php
<?php

    require 'config.php';
    if ( $_POST ) {

        $email = $_POST['email'];
        $sifre = $_POST['sifre'];

        $kontrol = DB::getRow("SELECT * FROM uyeler WHERE email=? AND sifre=?",array(
            $email,
            $sifre
        ));

        /** başarılı ise */
        if ( $kontrol ) {
            $_SESSION['login'] = $kontrol->id;


           header("Location:bayrak/bayrak.php");
            die();
        }
        else
        {
            header("Location:index.php?error=1");
            die();
        }


    }

?>
```

Şekil 3.7. Kayıt e-mail ve şifre kontrol scripti.

Kullanıcının veri tabanındaki e-mail ve şifre karşılaştırılması Şekil 3.7'de gösterilen kod bloğu ile yapılmaktadır.

require 'config.php' ile yazmış olduğumuz kod satırında bulunduğu dosyanın içinde yazılmış olan php dosyasının da çalıştırılması sağlanmış olur, config.php dosyası ile uygulamamızın veri tabanı ile bağlantısını sağlamış oluyoruz. Kod satırı öncelikle config.php dosyasını çalıştırıyor. Config.php dosyamız Şekil 3.8'de gösterilmektedir.

Burada session_start () komutu ile oturum başlatılmış oluyor. Bu dosyada veri tabanımıza vermiş olduğumuz isim veri tabanı kullanıcı giriş adımız, şifremiz ve host adresi olarak localhost oluşturmamız gerekmektedir. Bütün php dosyaları öncelikli olarak localhost'da oluşturulmaya başlanmaktadır.

```php
<?php
/*
*
* Veritabanı bağlantısı için
* gerekli bağlantı bilgilerinin
* bulunduğu ayar dosyası.
*
*/

session_start();

if(isset($_POST["Giriş Yap"]))
{

    $_SESSION["uye_adi"] = $_POST["uye_adi"];
    $_SESSION["sifre"] = $_POST["sifre"];

    header('location: index.php');
}


header('Content-Type: text/html; Charset=UTF-8');
date_default_timezone_set('Europe/Istanbul');

define('MYSQL_HOST',     'localhost');
define('MYSQL_DB',       'ders');
define('MYSQL_USER',     'root');
define('MYSQL_PASS',     '');

include 'db.php';
```

Şekil 3.8. config.php dosyası kod bloğu.

Kayıt olduktan sonra kullanıcı giriş yapabilmek için index.php sayfasına yönlendirilmektedir. index.php sayfasında session açmak için kullanıcının kontrolünü sağlayan kod bloğu Şekil 3.10'da gösterilmektedir. Giriş başarılı ise karar ağacı oluşumu için verileri girdigimiz bayrak.php sayfasına yönlendirilecektir. Eğer kullanıcı adı ya da şifresini yanlış girmiş ise tekrar index.php yani kullanıcı giriş ekranına yönlendirilecektir.

Şekil 3.9'da index.php sayfasının kullanıcı tarafından görülen sayfa gösterilmektedir.



Şekil 3.9. Giriş ekranı – index.php sayfası.

```php
/** başarılı ise */
if ( $kontrol ) {
    $_SESSION['login'] = $kontrol->id;


   header("Location:bayrak/bayrak.php");
    die();
}
else
{
    header("Location:index.php?error=1");
    die();
}
```

Şekil 3.10. index.php giriş kontrol kod bloğu.

Kayıt işlemleri tamamlanan kullanıcı giriş yap butonuna tıklayarak artık web sitemize giriş sağlayabilir. Kullanıcı sisteme giriş yaptıktan sonra Şekil 3.11'de görüldüğü gibi bir texbox karşımıza çıkıyor kullanacı burada istenilen şablonda bir .txt dosyası hazırlayarak herhangi bir veri bloğunu karar ağacına dönüştürebilmektedir.

İstenilen .txt şablonu örneği aşağıdaki gibidir;

```
[
  {
    "answer":"Zaman içindeki değişimini göster",
    "class":"green",
    "message":"Birden fazla veri seti var mı?",
    "decisions":[
      {
        "answer":"Evet",
        "class":"green",
        "message":"Veri kümeleri bir bütün oluşturuyor mu?",
        "decisions":[
          {
```

```
"answer":"Evet",
"class":"green",
"message":"Uygun grafik",
"decisions": [
  {
    "answer":"Yığılmış alan grafiği ",
    "class":"orange"
  }
 ]
},
```



Şekil 3.11. Karar ağacı oluşturmak için karar ağacı verilerini girdiğimiz ekran.

Kodlamış olduğumuz web sitemizde üç ayrı sayfa bulunmaktadır. Bunlar; üye girişinin yapılmış olduğu dosya, karar ağacını oluşturmak için verileri girdiğimiz ekran ve karar ağacını oluşturmuş olduğumuz dosyalar;

Üye girişi için oluşturmuş olduğum dosyalar;

anasayfa.php

config.php

index.php

kayıt-ol.php

style.css, dosyalarından oluşmaktadır.

Bu dosya içerikleri yukarıda anlatılmıştır.

Tasarlanan karar ağacı sayfası animated klasörü altında hazırlanmıştır. Bu klasörlerin altında css, js ve index.php dosyaları bulunmaktadır. Css dosyası ile karar ağacımızın metinlerini ve formatlarını yani görsel acıdan zenginleştirmek için kullanılmıştır. Bir diğer dosyamız olan js içeriğinde ise gerekli fonksiyonların kullanılması sağlanmıştır. Hedeflenen karar ağacı İndex.php içeriğinde oluşturulmuştur.

Karar ağaçları belirli soru ve bu sorulardan aldığı cevaplar doğrultusunda ağacımızın dallarında gezinmesi sağlanmıştır. Verimizin yapısına ve türüne göre kullanıcıya karar ağacında gezinebilmesi için bir ya da daha fazla soru sorulabilmektedir. Bu soruların istenildiği sayıda sorulabilmesi için recursive fonksiyon oluşturularak kullanıcıya soru sormakta ve bu sorulara kullanıcı tarafından çeşitli cevaplar vererek yaprak düğüme ulaşması sağlanmıştır.

Yani recursive fonksiyonlar kelime anlamı olarak özyinelemeli fonksiyonlardır. Bu fonksiyonlar kendine referans alarak çalışır.

Recursive fonksiyonlar şu şekilde çalışmaktadırlar.

```
function X () {
    if (n == 1 ) {
    return n;
    }
       else {
      return n * X (n-1);
}
}
    X (3);
```



```
function renderRecursive(data, elem, id) {
    var container = $('<div></div>')
        .addClass(settings.cardClass)
        .addClass('col-xs-12');
    var message = $('<div></div>').addClass(settings.messageClass).append(data.message);
    container.append(message);

    if (id != null) {
        container.attr('id', id)
    }

    if (typeof data.decisions != "undefined") {
        var decisions = $('<div></div>').addClass('dctree-decisions');
        for(var i=0; data.decisions.length > i; i++) {
            var decision = data.decisions[i];
            var genId = guid();
            var grid = $('<div></div>').addClass('col-md-6');
            var answer = $('<div></div>')
                .addClass("dctree-answer-" + i)
                .append(decision.answer)
                .on('click', function() {
                    getNextCard(this);
                })
                .attr('data-dctree-targetid', genId);
            if (typeof decision.class != "undefined") {
                answer.addClass(decision.class);
            }
            grid.append(answer);
            decisions.append(grid);
            renderRecursive(decision, elem, genId);
        }

        container.append(decisions);
    }


    if (id != 'dctree-first') {
        var controls = $('<div></div>').addClass('dctree-controls col-md-12');
        controls.append($('<h2><a href="javascrip:;" class="dctree-prev">  Geri  </a><a hr
        container.append(controls);
    }

    elem.append(container);
}
```

Şekil 3.12. Kullanılan recursive fonksiyon.

Şekil 3.12'de kullanılan recursive fonksiyon renderRecursive olarak adlandırılmıştır. Bu fonksiyon data, elem ve id adında üç tane değer döndürmektedir. Bu döndürülen değerlerin hepsi birer fonksiyonu ifade etmektedir.

```
77      function getNextCard(elem)
78      {
79          var e = $(elem);
80          currentCard = e.parents('.' + settings.cardClass)[0];
81          prevCard.push(currentCard.id);
82          var nextCard = e.attr('data-dctree-targetid');
83          showCard(nextCard);
84      }
```

Şekil 3.13. Kullanılan fonksiyon.

Şekil 3.13'de tanımlanan fonksiyon bloğu göserilmektedir. Bu fonksiyon kullanıcının ağaçlarının ya da dallarının üzerinde gezinebilmesi için oluşturulmuştur.

```
86      function showCard(id, backward)
87      {
88          var nextCard = $("#" + id);
89
90          if (settings.animation == 'slide') {
91              $(currentCard).slideUp(settings.animationSpeed, function(){
92                  nextCard.slideDown(settings.animationSpeed);
93              });
94          } else if (settings.animation == 'fade') {
95              $(currentCard).fadeOut(settings.animationSpeed, function(){
96                  nextCard.fadeIn(settings.animationSpeed);
97              });
98          } else if (settings.animation == 'slide-left') {
99              var left = {left: "-100%"};
100             var card = $(currentCard);
101
102             if (backward) {
103                 left = {left: "100%"};
104             }
105             card.animate(left, settings.animationSpeed, function(){
106                 card.hide();
107             });
108
109             if (nextCard.css('left') == "-100%" || nextCard.css('left') == "100%") {
110                 left.left = 0;
111                 nextCard.show().animate(left, settings.animationSpeed);
112             } else {
113                 nextCard.fadeIn(settings.animationSpeed);
114             }
115         }
116
117         currentCard = nextCard;
118     }
```

Şekil 3.14. showCard fonksiyonu.

Şekil 3.14'de karar ağacının alt düğümlerinin oluşturulması için ağaç alt düğümlerinin istenilen sayıda olabilmesi için function showCard (id, backward)

fonksiyonu oluşturulmuştur. Card fonksiyonu eklenti arabiriminin farklı 'sayfalarını' temsil eden birden çok kart kullanılarak oluşturulur. Bu fonksiyon sınırsız sayıda, istenilen yapıda ve istenilen dallanma yapısı ve sınıflandırma da bir karar ağacı oluşturmamızı sağlamıştır. Ve bu fonksiyon ile sayfalar arasında ileri ya da geri olacak şekilde gezinim sağlanmıştır.

Aynı zamanda tasarlamış olduğumuz karar ağacı yapısında kullanmış olduğumuz guid fonksiyon ise bize istenilen bir kontrole yani karar ağacı deyimiyle düğümlere art arda tıklama yöntemiyle, oluşturmuş olduğumuz karar ağacında benzersiz bir yapı oluşturmayı sağlamaktadır.

Bu bütün fonksiyonları oluşturduğumuz web sitemizde karar ağacı bloğu kısmında tree.js dosyamızın içerisinde görüntüleyebilmekteyiz.

Sayfa yapısında görselliği yakalamak için oluşturmuş olduğumuz css örnek script'i Şekil 3.15'de gösterilmektedir.

```css

html,body{
background-image: url('http://getwallpapers.com/wallpaper/full/0/3/e/1500495-screen-backgrounds-3840x1200-for-iphone.jp
background-size: cover;
background-repeat: no-repeat;
height: 100%;
font-family: Arial, Helvetica, sans-serif;
}
}
.container{
height: 100%;
align-content: center;
}
.card{
height: 370px;
margin-top: auto;
margin-bottom: auto;
width: 400px;
background-color: rgba(0,0,0,0.5) !important;
}

.social_icon span{
font-size: 60px;
margin-left: 10px;
color: #FFC312;
}
.social_icon span:hover{
color: white;
cursor: pointer;
}
.card-header h3{
color: white;
}
.social_icon{
position: absolute;
right: 20px;
top: -45px;
}
.input-group-prepend span{
width: 50px;
background-color: #FFC312;
color: black;
border:0 !important;
}
input:focus{
outline: 0 0 0 0  !important;
box-shadow: 0 0 0 0 !important;
}
.remember{
color: white;
}
.remember input
{
width: 20px;
height: 20px;
margin-left: 15px;
margin-right: 5px;
}
.login_btn{
color: black;
background-color: #FFC312;
width: 100px;
}
.login_btn:hover{
color: black;
background-color: white;
}
```

Şekil 3.15. Oluşturulan css dosyası.



```
var data = {
    message: "<div style='color:black;'>Senin veri hikayen nedir?</div>",
    decisions: [
            {
                answer: "Zaman içindeki değişimini göster",
                class: "green",
                message: "Birden fazla veri seti var mı?",
                decisions: [
                    {
                        answer: "Evet",
                        class: "green",
                        message: "Veri kümeleri bir bütün oluşturuyor mu?",

                        decisions: [
                    {
                        answer: "Evet",
                        class: "green",
                       message: "Uygun grafik",
                        decisions: [
                    {
                        answer: "Yığılmış alan grafiği ",
                        class: "orange",


                    },

                        ]

                    },
                    {
                        answer: "Hayır",
                        class: "red",
                       message: "Tüm veri kümeleri ortak bir ölçü birimi paylaşıyor mu?",

                       decisions: [
                    {
                        answer: "Evet ",
                        class: "green",
                        message: "Uygun grafik",

                    decisions: [
                    {
                        answer: "Cok satırlı grafik ",
                        class: "orange",


                    },

                        ]
                    },
                    {
                        answer: "Hayır ",
                        class: "red",
                        message: "Uygun grafik",

                    decisions: [
                    {
                        answer: "Kombinasyon çubuğu & çizgi grafik",
                        class: "orange",


                    },

                        ]
                    },
                        ]

                    },

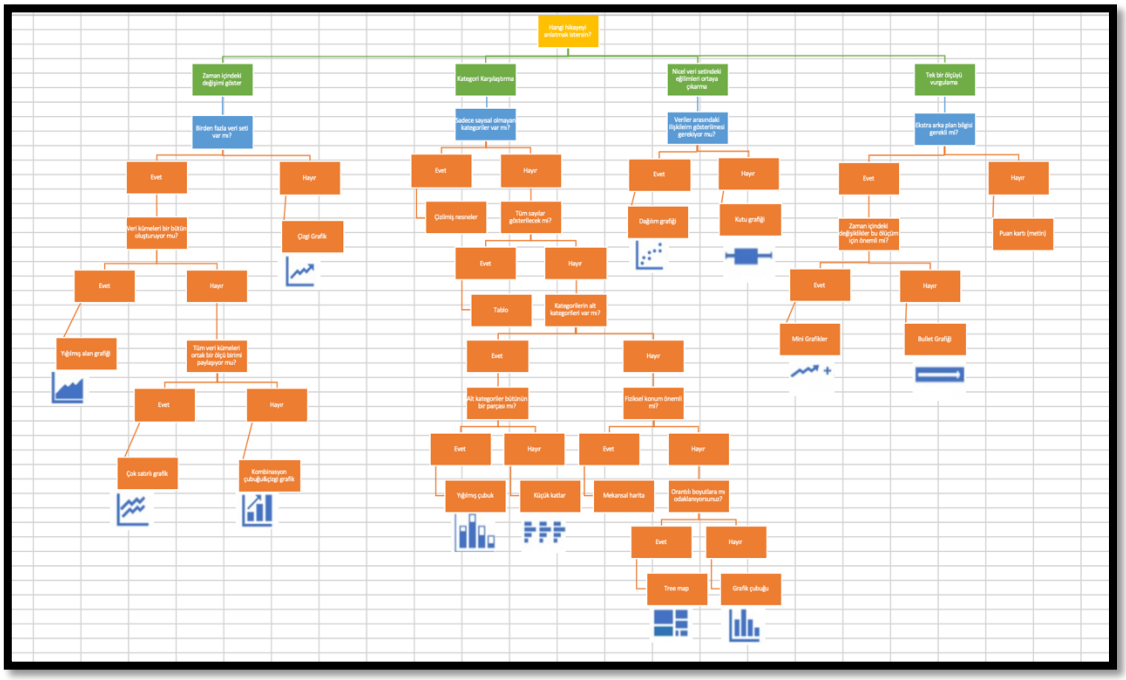
```

Şekil 3.16. Verinin animated.php dosyası içerisinde karar ağacına dönüştürülmesi örneği.

Şekil 3.16'da karar ağacının kod olarak web sayfasına aktarılması için gerekli olan şablonu göstermektedir. Şekil 3.16'da görüldüğü gibi kod bloğunun bu şekilde düzenleyerek sınıflandırmak istenilen veri bloğunun karar ağacına dönüştürülmesi sağlanmaktadır. Fakat kullanıcı istediği taktirde kendi karar ağacını oluşturması sağlanmıştır. Kullanacı karar ağacının oluşturmak için uygun şablonda hazırlamış olduğu veri yığınlarını Şekil 3.11'de gösterildiği gibi text bölümüne yazıp tamam butonuna bastığında istemiş olduğu karar ağacı oluşturulmuş olacaktır.

3.3 Veri Hikayemiz için Uygun Grafik Yapısını Seçen Karar Ağacı

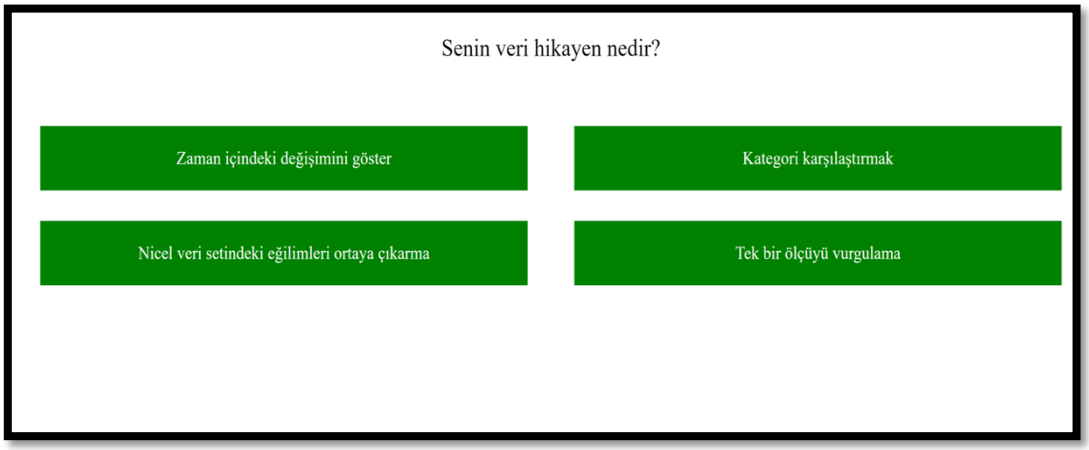
Şekil 3.18'de karar ağacımız veri hikayemizin ne olduğu sorusu ile başlamakta olup, dört adet ana düğümden meydana gelmektedir. Bu ana düğümler soru ve cevaplara göre alt düğümlere alt düğümler ise yaprak düğümlere doğru ilerlemektedir. Yukarıda da anlatıldığı gibi bir karar ağacının yaprak düğümüne ulaşmak demek o veri seti için gerekli soru ve cevaplardan sonra en uygun sınıflandırmanın yapıldığı ve sonuca ulaşıldığını göstermektedir.



Şekil 3.17. Oluşturulması gereken karar ağacı.

Her araştırmacı mutlak olarak ya araştırmanın sonunda ya da araştırmanın başında sahip olduğu verilerle çalışmasını sürdürebilmektedir. Bu veriler yardımıyla birçok sonuca varabilmektedir. Bu sonuçları grafiklere dökmek için bazı yöntemler kullanabilmektedir. Veri hikayeciliği ise tam bu noktada karşımıza çıkmaktadır. Temel olarak veri hikayeciliği araştırmacının mevcutta bulunan verilerinden yola çıkarak verileri daha anlaşılır bir şekilde çıkarım yapmak ve bu verileri inceleyen kişiye daha açık olacak şekilde sunulmasıdır. Araştırmacı birçok konuda veri toplayıp bunlarla ilgili çalışmalar yapabilir hatta çalışmalarının sonucunda veri yığınları elde edebilir. Fakat araştırmacı eldeki verilerin kullanıcıya kendini daha kolay anlatmasını sağlayabilir. Bu yöntem ise grafik ve şemalardan geçmektedir.

Şimdi ise araştırmacının bu grafik ve şemaları hangi ölçülere göre seçeceğine karar vermesi gerekmektedir. Bu tezde temel hedef bir karar dinamik bir karar ağacı oluşturmaktır. Oluşturulan karar ağacımıza örnek oluşturmak maksadıyla veri hikayeciliği verilerini kullanarak bir karar ağacı oluşturulmuştur. Oluşturulan karar ağacı ise bize veri yığınımızın özelliğine göre uygun grafik ya da şema metodunu göstermesidir.



Şekil 3.18. Veri hikayen nedir?

Şekil 3.18'de gösterildiği gibi araştırmacının veri hikayesinin niteliğine göre seçim yapılması istenmektedir. Bu nitelikler;

Zaman içindeki değişimini göstermek istemesi,

Kategorileri karşılaştırmak istemesi,

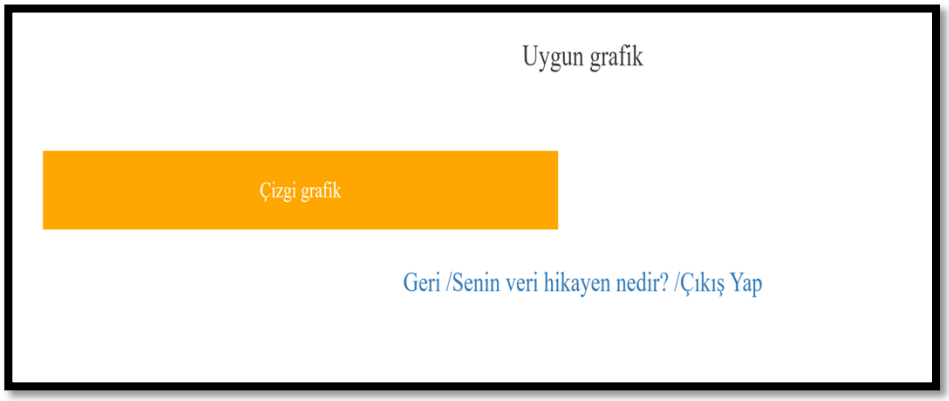
Nicel veri setindeki eğilimleri ortaya çıkarmak istemesi,

Tek bir ölçüye vurgulama yapmak istemesi gibi dört ana bölümden oluşmaktadır.

Zaman içerisinde değişimi göstermek butonunu seçersek eğer, şekil 3.19'da olduğu gibi veri demetinde birden fazla veri seti var mı? Sorusunu yöneltecektir. Bu soruya verilen cevap hayır olduğu taktirde, uygun grafik çiziminin çizgi grafik olduğunu kullanıcıya gösterecektir. Uygun grafik şekil 3.20'de gösterilmektedir.



Şekil 3.19. Birden fazla veri seti var mı?



Şekil 3.20. Kullanılması gereken uygun grafik.

Şekil 3.20'de görüldüğü gibi Birden fazla veri seti var mı? sorusuna evet yanıtını verecek olursak eğer, karar ağacımız veri kümeleri bir bütün oluşturuyor mu? Sorusunu bize yöneltecektir. Bu soru ise evet ya da hayır şeklinde iki alt düğüme ayrılmaktadır. Bu düğümlerden birini seçecek olmamız veri setimizde kullanacağımız grafik türünü belirlemede yardımcı olacaktır. Eğer evet alt düğümünü seçersek, yığılmış alan grafiği kullanarak verilerimizi görselleştirebileceğiz. Hayır alt düğümünü seçersek karar ağacımız tüm veri kümeleri ortak bir ölçü birimi paylaşıyor mu? Sorusunu soracaktır bu

sorudan iki alt düğüm daha ortaya çıkmaktadır bu alt düğümler evet ya da hayır şeklindedir. Evet alt düğümünü seçecek olursak eğer, karar ağacımızın sınıflandırma aşamasını bitirmiş ve bize bir yaprak düğüm olan çok satırlı grafik yaprak düğümünü görüntüleyecektir. Hayır alt düğümünü seçmiş olsaydık eğer karar ağacımız bu sefer yaprak düğüm olarak kombinasyon çubuğu & çizgi grafik sınıfını görüntüleyecektir.

BÖLÜM 4

SONUÇ

Bu tezde oluşturulan dinamik karar ağacına örnek olması maksadıyla veri hikayeciliği verileri kullanılarak bir karar ağacı oluşturulmuştur. Bu karar ağacında seçim yapılması istenilen veri hikayeleri ise dört ana düğümden oluşmaktadır. Bu düğümler, verilerin zaman içindeki değişiminin gösterilmek istenmesi, kategori karşılaştırmasının yapılması, nicel veri setindeki eğilimleri ortaya çıkarmak ve belirli soru ve cevaplarla ağacın dallarında gezinerek verilerimizi dinleyiciye en iyi şekilde anlatabilmek maksadıyla verimizin türüne göre en uygun grafiksel şema ile anlatımı hedeflemektedir. Oluşturulan örnek karar ağacının yapısı Şekil 3.17'de gösterilmektedir.

Karar ağaçları sosyal, psikolojik, sağlık, otomotiv, bilişim gibi birçok alanda kullanılmaktadır. Bu çalışma sonunda farklı disiplinlerde çalışan araştırmacıların verilerini daha görsel ve anlaşılır bir biçimde ifade etmesini sağlayacak faydalı bir uygulama kazandırılmıştır.

Ayrıca çalışma anket firmalarının da anketlerini görselleştirerek daha anlaşılır biçimde ifade etmelerini sağlayabilir. Ayrıca geliştirilen uygulama var olan web sayfalarına birer eklenti olarak ta kullanılabilir.

Sonuç olarak yapılan bu çalışmanın açık kaynak kodlu olarak paylaşılması birçok kullanıcıya fayda sağlayacağı düşünülmektedir.

Gelecekte farklı tasarım şablonları kullanılarak biçimler çeşitlendirilebilir.

KAYNAKLAR

Albahri, A. K., Iwendil, C., Peshar, A., Sujatha, R., Chatterjee J. M., Pasupuleti, M. Pillai, S., & Jo, O. (2020). Role of biological data mining and machine learning techniques in detecting and diagnosing the novel coronavirus (COVID- 19): a systematic review. *J Med Syst*. *44*, 1-11, https://doi.org/10.1007/s10916-020-01582-x

Angulo, J., Velasco-Forero, S., Meyer, F., (2017). *Mathematical Morphology and Its Applications to Signal and Image Processing*. doi: 10. 1007/978-3-319-57240-6

B. R. Patel and K. K. Rana. 2014. A survey on decision tree algorithm for classification. *Journal of Engineering Development and Research*. 1-5.

Bach, M., P., Cosic, D., (2008). Data mining usage in health care: literature survey and decision tree application. *Ekonomski fakultet, Sveučilište u Zagrebu; 2Valicon d.o.o.* 5(1):57-64

Baker, A., Greene, E. (1977). *Story and discourse: narrative structure in fiction and film*. Cornell University Press, Ithaca, N.Y

Bradford, J., Kunz, C., Kohavi, R., Brunk, C., & Brodley, C. (1998). Pruning decision trees with misclassification costs. *Proceedings of the European Conference on Machine Learning*. pp. 131–136

Cabane, P. ve ark. (1998). *Discovering Data Mining: From Concept to Implementation,* Prentice Hall, Upper Saddle River, New Jersey.

Cai, J., (2008). Decision Tree Pruning Using Expert Knowledge. 1st Edn., VDM Publishing, USA., ISBN: 9783836491556, pp: 236.

Cao, Y. Mıao, Q. Lıu, J. Gao, L., (2013). **Advance and Prospects of AdaBoost Algorithm.** Acta Autom. Sin. 39 (6), 745-758.

Chakure, A., (2019). Decision Tree Classification - An introduction to Decision Tree Classifier. 28 Eylül 2021 tarihinde https://medium.com/swlh/decision-tree-classification-de64fc4d5aac adresinden erişildi.

Chau Tu AND Dongil Shin (2009). *A Comparative Study of Medical Data Classification Methods Based on Decision Tree and Bagging Algorithms*. IEEE. P1-5.

Cockerham, R., (December 19, 2018). How to Create a Decision Tree in Word. 28 Eylül 2021 tarihinde https://www.techwalla.com adresinden erişildi.

Decision Tree Algorithm. (2021). 28 Eylül 2021 tarihinde k21academy.com adresinden erişildi.

Decision Tree - Classification. 28 Eylül 2021 tarihinde http://www.saedsayad.com/decision_tree.htm adresinden erişildi.

Diagrams. (2005). 15 Ekim 2021 tarihinde https://www.diagrams.net/ adresinden erişildi.

Doğan, Ö., (2020, Haziran 13). NumPy. 25 Ekim 2021 tarihinde https://teknoloji.org/numpy-kutuphanesi-nedir-nasil-kullanilir/ adresinden erişildi.

Dunham Margaret H., (2003). *Data Mining Introductory and Advanced Topics.*Prentice Hall, Pearson Education Inc. New Jersey. S. 8

Dündar, D. (2011). Hipotez Testi ve Ki-Kare İle Bir Uygulama. *İstanbul Üniversitesi İktisat Fakültesi Mecmuası*, 45 (1-4), Retrieved from https://dergipark.org.tr/tr/pub/iuifm/issue/835/917

Diaz-Perez, F.M., Bethencourt-Cejas, M., (2016). CHAID algorithm as an appropriate analytical method for tourism market semgemtation. J. Destin. Mark. Manag. 5 (3)

Dykes, B. (2015). Data storytelling: What it is and how it can be used to effectively communicate analysis results. *Applied Marketing Analytics, 1*(4), 299–313.

Esposito, F., Malerba, D., & Semeraro, G., (1995). *Simplifying decision trees by Pruning and grafting: New results. In Lavrac, N. & Wrobel, S. (Eds.), Proceedings of the 8th Eupropean Conference on Machine Learning*, Volume 912 of LNAI. Pp. 287-290. Berlin: Spinger.

Frank, E., & Witten, I. (1998). Reduced-error 40runing with significance tests, available at http://citeseer.ist.psu.edu/frank98reducederror.html.

Gadomer, Ł., Sosnowski, Z., A., (2021). Pruning trees in C-fuzzy random forest. *Soft Comput* **25,** 1995–2013 https://doi.org/10.1007/s00500-020-05270-3

Geoffrey, K., F., Kelvin, K., V., (2007). Predicting electricity energy consumption: *A comparison of regression analysis, decision tree and neural networks. ScienceDirect.* pp. 1791-1768

Graphviz. (2021). 25 Ekim 2021 tarihinde https://graphviz.org/ adresinden erişildi.

Güner, S. E., (2008). *Türkçe İçin Derlem Tabanlı Bir Anafor Çözümleme Çalışması.*Trakya Üniversitesi/Fen Bilimleri Enstitüsü, Edirne

Hall, L. O., Bowyer, K. W., Banfield, R. E., Eschrich, S., & Collins, R. (2003). Is error-based pruning redeemable? *International Journal on Artificial Intelligence Tools,* 12(3), pp. 249–264.

https://parsnip.tidymodels.org/reference/boost_tree.html

İchi. (2020-2021). 29 Eylül 2021 tarihinde https://ichi.pro/tr/karar-agaci-uctan-uca-uygulama-191253755224660 adresinden erişildi.

Jindal, A., Kaur, K., Kumar, N., (2016). Decision Tree and SVM-Based Data Analytics for Theft Detection in Smart Grid. *IEEE Transactions on Industrial Informatics*. 1005-1016

KARABAY, A., (2020). Scikit-learn Nedir, https://www.karabayyazilim.com/blog/python/scikit-learn-nedir-2020-02-12-062241. (Erişim Tarihi: 25 Ekim 2021).

Kavzaoğlu, T., Çölkesen, İ., (2010). Karar Ağaçları ile Uydu Görüntülerinin Sınıflandırılması Kocaeli Örneği. *Harita Teknolojileri Elektrik Dergisi*. 36-45 e-ISSN: 1309-3983

Koçak, H. (2020). Çalışan Örgütsel Bağlılıkların Cart Karar Ağacı Algoritması ile Belirlenmesi, *Uluslararası İktisadi ve İdari Bilimler Dergisi,* 6 (2), 66-87

Lantz, B. (2015). Divide and Conquer- Classification Using Decision Trees and Rules. *Packtpub*, pp. 1-9.

Lavanya, D., Rani, U., (2011). Performance Evaluation of Decision Tree Classifiers on Medical Datasets. *International Journal of Computer Applications (0975-8887)*. pp. 2-5

T. Luo, T. Hao, H., Du, W., Wang, Y., and Yin, H. (2011) "Attacks on WebView in the Android system". In: *Proceedings of the 27th Annual Computer Security Applications Conference*. pp. 343–352.

Mansour, Y. (1997). Pessimistic decision tree pruning based on tree size. *In Proceedings 14th International Conference on Machine Learning*. pp. 195-201

Mingers, J., (1989). An empirical comparison of pruning methods for decision tree induction. Machine Learning, 4(2), pp. 227-243.

Mutairi, M., Habib, S., (2007). Coverage Restoration of Wireless Sensor Networks through the Usage of Decision Trees. *A Scientific and Technical Publishing Company*.

Ngwenyama, O. K., Bryson, N. (1999). *Toward Entrepreneurial Behavior in Underserved Communities: An Ethnographic Decision Tree Model of Telecenter Usage.*Doi: 10.1080/02681102.2012.751571

Pandya, R., & Pandya, J. (2015). C5. 0 algorithms to improved decision tree with feature selection and reduced error pruning. *International Journal of Computer Applications*, 117(16).

Patel, N.P., Prajapati, S., & Lakhtaria, K., (2012). Efficient Classification of Data Using Desicion Tree. *Bonfring* Intellectual *Integrity,* pp. 6-11

Patil, D., D., Wadhai, V., M., Gokhale, J., A., (2010). Evaluation of decision tree pruning algorithms forcomplexity and classification accuracy. *International Journal of ComputerApplications*, 11, pp. 23–30.

Pham, B.T., Prakash, I., Singh, S.K., Shirzadi, A., Shahabi, H., Tran, T.-T.-T., Bui, D.T. (2019). *Landslide susceptibility modeling using Reduced Error Pruning Trees and different ensemble techniques: Hybrid machine learning approaches*., *175*, pp. 203–218

Pham, B., T., Jaafari, A., Nguyen-Thoi, T., Van Puhong, T., Nguyen, H., D., (2020). Ensemble machine learning models based on Reduced Error Pruning Tree for prediction of rainfall – induced landslides. *Internotional Journal of Digital Earth.* https://doi.org/10.1080/17538947.2020.1860145

Prodromidis, A., L., Stolfo, S., J., (2001) Cost Complexity-Based Pruning of Ensemble Classifiers. *Knowledge and Information Systems*, vol. 3, no. 4, pp. 449-469

Quinlan, J., R., (1993). C4.5: *Programs for machine learning*. Morgan Kaufmann Publishers San Meteo, California. 35-42

Ryan, L. (2016). *The visual imperative: Creating a visual culture of data discovery*. Amsterdam, Netherlands: Elsevier Science.

Scikit-Learn (2007). 29 Eylül 2021 tarihinde https://scikit-learn.org/stable/modules/tree.html#decision-trees adresinden erişildi.

Sharma, H., Kumar, S. (2015). A Survey on Decision Tree Algorithms of Classification in Data Mining. *International Journal of Science and Research (IJSR)*, pp. 2094-2097.

Silahtaroğlu, G. (2016). Veri Madenciliği (Kavram ve Algoritmaları). İstanbul: Papatya Yay.

Su, J., & Zhang, H. (2006) A fast decision tree learning algorithm. In *Proceedings, the twenty-first national conference on artificial intelligence and the eighteenth innovative applications of artificial intelligence conference*, July 16–20, 2006, Boston, Massachusetts, USA, pp. 500–505

Tosi, S. (2009). *Matplotlib for Python Developers*.Pack Publishing.

Towardsdatascience. (2020). 29 Eylül 2021 tarihinde https://towardsdatascience.com/decision-trees-explained-3ec41632ceb6 adresinden erişildi.

Tufte, E. R., & Schmieg, G. M. (1985). *The visual display of quantitative information* (Vol. 53). American Association of Physics Teachers.

ÖZGEÇMİŞ

29 Ocak 1989 tarihinde Ankara'nın Altındağ ilçesinde doğdum. İlk ve Orta öğrenimimi Karabük Cumhuriyet İlkokulunda, Lise öğrenimimi Karabük Mustafa Yazıcı Lisesinde tamamladım. Selçuk Üniversitesi Kulu Meslek Yüksekokulu ön lisans Bilgisayar Programcılığı bölümünden 2009 yılında mezun oldum, 2010 yılı DGS sınavı sonucunda Trakya Üniversitesi Bilgisayar Mühendisliği bölümüne yerleştim. Trakya Üniversitesi Rektörlüğü Bilgisayar İşletmeni kadrosuna atanarak İdari ve Mali İşler Daire Başkanlığında 01 Temmuz 2011 tarihinde görevime başladım. 2014 yılında görev yeri değişikliği nedeni ile Tunca Meslek Yüksekokulunda görevime başladım ve halen buradaki görevime devam etmekteyim. Bilgisayar Mühendisliği bölümünden 2015 yılında mezun oldum.