



**T.C.**

**ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ**

**Mühendislik-Mimarlık Fakültesi**

**Bilgisayar Mühendisliği Bölümü**

**2022-2023 Güz Dönemi**

**152116025 - Tasarım Süreçleri (A)**

**Final Raporu**

**Proje Başlığı**

“Diyabet Hastalığı Tespit Sistemi”

**GRUP 14**

**Projeyi Hazırlayanlar:**

Hakam CHEDO - 152120181096

Hüseyin KAYMAK - 152120181100

Bengisu ŞAHİN - 152120191064

Muhammet Eren SÖME - 152120201049

**Danışman**

Öğr. Gör. Dr. Yıldırım ANAGÜN

Dr. Öğr. Üyesi İlker ÖZÇELİK

Ocak 2023

## İÇİNDEKİLER

<b>A. PLANLAMA</b>	<b>3</b>
<b>A.1. Özet ve Anahtar Kelimeler</b>	<b>3</b>
<b>A.2. Bilgi Gereksinim Belirleme, Problemin Tanımlanması</b>	<b>3</b>
A.2.1. Amaç	3
A.2.2. Konu ve Kapsam	4
A.2.3. Literatür Özeti	4
<b>A.3. Fizibilite (Yapılabilirlik) Raporları</b>	<b>5</b>
A.3.1. Kurumsal ve Kültürel Fizibilite	5
A.3.2. Teknolojik Kaynak Fizibilitesi	5
A.3.3. Yasal Fizibilite	6
A.3.4. Finansal Fizibilite	6
A.3.5. Zaman Fizibilitesi	6
<b>A.4. Beklenen Fayda</b>	<b>6</b>
A.4.1. Özgün Değer	6
A.4.2. Yaygın Etki / Katma Değer	7
<b>A.5. Yöntem</b>	<b>7</b>
<b>A.6. Araştırma Olanakları</b>	<b>8</b>
<b>A.7. Çalışma Takvimi</b>	<b>8</b>
A.7.1. İş Zaman Çizelgesi	8
A.7.1.1. GANTT Şeması	8
A.7.2. Kişi – İş Açıklaması	10
<b>B. ANALİZ</b>	<b>10</b>
<b>B.1. Sistem Gereksinimlerini Ortaya Çıkarma Yöntem ve Teknikleri</b>	<b>10</b>
B.1.1. Yüz Yüze Görüşme	10
B.1.2. Yazılı Basılı Belge İnceleme	10
B.1.3. Anket	10
B.1.4. Gözlem	14
B.1.5. Prototip ve Hızlı Uygulama Tasarımı (Rapid Application Design – RAD)	14
B.1.6. Ortak Uygulama Tasarımı (Joint Application Design – JAD)	15
B.1.7. Veri Akış Şemaları	16
B.1.7.1. Kavramsal Veri Akış Şeması	16
B.1.7.2. Mantıksal Veri Akış Şeması	16
B.1.7.3. Fiziksel Veri Akış Şeması	17
B.1.7.4. İş Akış Şeması	18
B.1.8. Olay Tabloları, Durum Formları, İşlevsel Analiz Raporu, İş Akış Şeması	19
B.1.8.1. Olay Tabloları	19
B.1.8.2. Durum Formları	19
B.1.8.3. İşlevsel Analiz Raporu	21
<b>B.2. Sistem Gereksinimleri</b>	<b>21</b>

B.2.1. İşlevsel Gereksinimler	21
B.2.2. Sistem ve Kullanıcı Arayüzleri ile İlgili Gereksinimler	22
B.2.3. Veriyle İlgili Gereksinimler	22
B.2.4. Kullanıcılar ve İnsan Faktörü Gereksinimleri, Güvenlik Gereksinimleri	23
B.2.5. Teknik ve Kaynak Gereksinimleri, Fiziksel Gereksinimler	23
<b>C. TASARIM</b>	<b>24</b>
<b>C.1. Sistem Tasarımları</b>	<b>24</b>
<b>C.2. Kullanıcı ve Sistem Arayüzü Tasarımları</b>	<b>24</b>
C.2.1. Kullanıcı Veri Girişi/Seçim Arayüzleri	25
C.2.2. Raporlama/Bilgilendirme vb Arayüzleri	25
<b>C.3. Veri Tabanı Tasarımı</b>	<b>25</b>
C.3.1. Veri Tabanı İsterler Dökümü	26
C.3.2. E/R Diyagramı	26
C.3.3. Stored Procedures, Triggers, Kısıtların Tasarımı	27
<b>C.4. Yazılım Tasarımı</b>	<b>29</b>
C.4.1. Gereksinime Bağlı Tasarım Kalıpları Seçimi	29
C.4.2. UML Kullanarak Tasarım Diyagramları Oluşturma	29
<b>C.5. Test Tasarımı</b>	<b>30</b>
C.5.1. Gereksinim Analizlerinden Test Hedeflerinin Belirlenmesi	31
C.5.2. Birim Testleri Tasarımı	31
C.5.3. Entegrasyon Testleri Tasarımı	32
C.5.4. Sistem Testleri Tasarımı	32
C.5.5. Kabul Testleri Tasarımı	33
<b>D. UYGULAMA</b>	<b>33</b>
<b>D.1. Geliştirilen Sistemin Sistem Tasarımlarını Karşılanmasının Değerlendirilmesi</b>	<b>34</b>
<b>D.2. Kullanıcı ve Sistem Arayüzü Gerçeklemeleri</b>	<b>34</b>
<b>D.3. Gerçeklenen Testler</b>	<b>35</b>
D.3.1. Fonksiyonel Testler	35
D.3.1.1. Birim (Unit) Testi	35
D.3.1.2. Entegrasyon Testi	35
D.3.2. Performans Testi	37
D.3.3. Kabul Testleri	38
<b>D.4. Yazılım/ Veritabanı/ Donanım Gerçeklemeleri</b>	<b>39</b>
D.4.1. Yazılım Gerçeklemeleri	39
D.4.2. Veri Tabanı Gerçeklemeleri	39
D.4.3. Donanım Gerçeklemeleri	40
<b>E. SONUÇ VE ÖNERİLER</b>	<b>40</b>
<b>Referanslar</b>	<b>41</b>

## A. PLANLAMA

### A.1. Özet ve Anahtar Kelimeler

#### ÖZET

Diyabet (Diabetes Mellitus) kritik hastalıklar arasındadır ve birçok insan bu hastalıktan muzdariptir. Yaş, obezite, egzersiz eksikliği, kalıtsal diyabet, yaşam tarzı, kötü beslenme, yüksek tansiyon vb. şeker hastalığına neden olabilir. Diyabet hastalarının kalp hastalığı, böbrek hastalığı, inme, göz sorunu, sinir zedelenmesi gibi hastalıklara yakalanma riski yüksektir.

Büyük Veri Analitiği, sağlık sektörlerinde önemli bir rol oynamaktadır. Büyük veri analitiğini kullanarak, büyük veri kümelerini incelenebilir ve verilerden bilgi keşfedip sonuçları tahmin etmek için gizli bilgileri, kalıpları bulabilir. Mevcut yöntemde sınıflandırma ve tahmin doğruluğu çok yüksek değildir.

Bu çalışmada, diyabetin daha iyi sınıflandırılması için gebelik sayısı, glikoz, kan basıncı, cilt kalınlığı değeri, insülin, BMI (vücut kitle indeksi), diyabet soyağacı fonksiyonu (diabetes pedigree function), yaş gibi faktörleri içeren bir diyabet tahmin modeli geliştirilecektir.

**Anahtar Kelimeler:** Diyabet Mellitus, Makine Öğrenmesi, Büyük Veri Analizi, Sağlık

### A.2. Bilgi Gereksinim Belirleme, Problemin Tanımlanması

#### A.2.1. Amaç

Mevcut tıbbi tanı yönteminde üç farklı hata türü olabilmektedir.

1. Yanlış negatif tip: Bir hastanın gerçekte zaten diyabet hastası olduğu, ancak test sonuçlarının kişinin diyabet hastası olmadığını söylediği tiptir.

2. Yanlış pozitif tip: Bu tipte hasta gerçekte diyabet hastası değildir, ancak test raporları diyabet hastası olduğunu söyler.

3. Üçüncü tip: Vakanın teşhis edilemediği sınıflandırılmaz tiptir. Bu, geçmiş verilerden yetersiz bilgi çıkarımı nedeniyle olur ve bir hasta sınıflandırılmamış bir tipte

tahmin edilebilir. Ancak gerçekte hasta ya diyabetik kategoride ya da diyabetik olmayan kategoride olacağını tahmin etmelidir.

Teşhisteki bu tür hatalar gereksiz tedavilere veya gerektiğinde hiç tedavi yapılmamasına neden olabilir. Bu etkinin zararlarını önlemek veya azaltmak için makine öğrenimi algoritması ve veri madenciliği teknikleri kullanılarak doğru sonuçlar verecek ve insan emeğini azaltacak bir sistem oluşturulmasına ihtiyaç vardır.

### **A.2.2. Konu ve Kapsam**

Projemiz konu ve kapsam itibariyle günümüzde çok yaygın olarak rastlanan diyabet hastalığının erken evre teşhisinin konulmasında rol almaktadır. Diyabet, tedavisi olmayan, yaygın ve ölümcül bir hastalıktır. Erken tedavi sayesinde diyabetin etkilerini azaltmak ve hastaların hayat standartlarını arttırmak mümkün olsa da çoğunlukla teşhis konulması yıllar sürebilen bir süreçtir. Projemiz bu süreçte tasarlanan makine öğrenmesi modeli ile kullanıcı girdilerine göre diyabet hastalığı tahmini yapacak ve kullanıcıyı gerekli önlemler alması için bilgilendirecektir. Modelimizin eğitim ve test aşamaları örnek veri seti üzerinden gerçekleştirilecektir.

### **A.2.3. Literatür Özeti**

Diyabet WHO ‘nun (Dünya Sağlık Örgütü/World Health Organization) tanımına göre, vücudumuzda pankreas adlı salgı bezinin yeterli miktarda insülin hormonu üretememesi ya da ürettiği insülin hormonunun etkili bir şekilde kullanılamaması durumunda gelişen ve ömür boyu süren ölümcül bir hastalıktır. Hastalığın belirtileri diğer birçok hastalıkla benzer olabilmektedir. Diyabet tek başına kişi için ölümcül bir risk ortaya koyduğu gibi retinopati (gözün damarları ve retinanın bozuklukları), nöropati (çevresel sinirlerde görülen bozukluklar), diyabetik nefropati (böbreklerdeki kılcal damar bozukluğuna bağlı bozukluklar), beyin damarlarında ve koroner damarlarda tıkanma, bacak ve ayak damarlarında daralma ve tıkanma gibi diğer hastalıklara da yol açabilir. Bu sebeple erken tanı çok büyük önem arz etmektedir. Aynı zamanda teşhisin konulmasında çeşitli faktörler rol aldığı için doktor tarafından diyabet teşhisi konulması karmaşıktır ve hata payı yüksektir. Bu sebeple literatürde hekime yardımcı yapay zeka sistemleri bulunmaktadır. Örneğin Amelec Viloriaa, Yaneth Herazo-Beltranb, Danelys Cabrerac ve Omar Bonerge Pinedad‘ ın “Diabetes Diagnostic Prediction Using Vector Support Machines” adlı çalışmalarında

hastalarda belirtilen faktörlere dayalı olarak diyabet tanısını tahmin etmek için bir “support vector machine” (destek vektör makinaları) kullanıldı. Kolombiyalı hastalardan %99.2 doğrulukla oluşturulan bu model kişinin girdilerine göre diyabeti olmayan , diyabete yatkınlığı olan ve diyabet hastalığına sahip olan olmak üzere üç çıktı üretilmesini sağladı.

Biz de projemizde hekimin erken tanı koyabilmesine yardımcı faktör olacak şekilde hastanın glikoz seviyesini, hamilelik sayısını, kan basınç değerini, insülin seviyesini, vücut kitle indeksini, ailesindeki diyabet soyağacı yatkınlığını, yaşını, kişinin cildindeki kolajen miktarını (deri kalınlığı) girdi olarak alan bir web uygulaması ile hastanın diyabet olup olmadığını gözetimli öğrenme yöntemlerinden biri olan destek vektör makinesi yöntemini kullanarak tahminde bulunacağız ve erken tanı koymada hekime ve tabikide hastaya yardımcı olacağız. Bunun için örnek veri setini eğiteceğiz ve test edeceğiz.

### **A.3. Fizibilite (Yapılabilirlik) Raporları**

#### **A.3.1. Kurumsal ve Kültürel Fizibilite**

Proje çalışanları söz konusu “Diyabet Hastalığı Tespit Sistemi” projesini gerçeklemeye teknik açıdan yeterlidir. Kullanılacak teknolojilere ekip üyeleri hakimdir. Ekip üyelerinin iletişim becerileri projenin gerçekleşmesi için uygun seviyededir. Bu uygulamayı kullanacak olan kişilerin temel düzeyde bilgisayar kullanımını bilmeleri (fare, klavye vb) yeterlidir.

#### **A.3.2. Teknolojik Kaynak Fizibilitesi**

Projede kullanılacak teknolojiler aşağıda verildiği gibidir. Bu teknolojiler ile projeyi geliştirmek mümkündür.

- Python
- MSSQL
- Streamlit Python Kütüphanesi
- Sklearn Python Kütüphanesi
- Pyodbc Python Kütüphanesi
- Spyder IDE

### **A.3.3. Yasal Fizibilite**

Diyabet hastalığı tespit sisteminde hastalardan aldığımız bilgilerin içinde kişisel bilgi olarak Ad, Soyad ve E-posta bilgileri bulunmaktadır. Testi yaptırmak isteyen hastalar verilerinin depolanacağını ve işleneceğini kabul etmiş olurlar. Bu onayı test bilgilerini göndermek için kullanılan butonun üstünde beliren onay kutucuğunun işaretlenmesiyle alıyoruz. “Diyabet Hastalığı Tespit Sistemi” programımızda veriler Kişisel Verilerin Korunması Kanunu’na uygun şekilde toplanır, işlenir ve depolanır. Makine öğrenmesinde algoritmamız için kullandığımız veri seti ise verilerinin toplanmasını ve depolanmasını onaylayan anonim hastalardan oluşmaktadır.

### **A.3.4. Finansal Fizibilite**

Diyabet hastalığı tespit sistemi, maliyetin düşük olduğu bir projedir.

Donanım ve yazılım gereksinimleri karşılandıktan sonra, sistemimizin kullanıcısının herhangi bir ek masraf için para harcamasına gerek kalmaz. Sistem aşağıdaki açılardan ekonomik olarak uygulanabilir olacaktır:

- Fazladan kağıt işini azaltacaktır. Dolayısıyla maliyet düşecektir.
- Geleneksel süreçlerde boşa harcanan zamanı azaltacaktır.

### **A.3.5. Zaman Fizibilitesi**

Diyabet Hastalığı Tespit Sistemi projesi çok zaman almayan bir proje. Dolayısıyla belirtilen zamanda tamamlanabilir. Bu projenin en çok zaman alan kısmı bilgi toplamak ve planlama yapmaktır. Bu bölümü tamamladıktan sonra gerisi daha az zaman alacaktır.

## **A.4. Beklenen Fayda**

### **A.4.1. Özgün Değer**

Çağın hastalıkları arasında en ön sıralarda yer alan diyabet (şeker hastalığı), ölümcül birçok hastalığın oluşumunda birinci sırada rol oynayan ve dünyanın her yerinde çok yaygın olarak görülen bir hastalık türüdür. Uluslararası Diyabet Federasyonu'nun sağlamış olduğu istatistiki verilere göre her 11 yetişkinden biri diyabet hastalığına sahip olmakla birlikte her 6 saniyede 1 birey diyabet kaynaklı sorunlar nedeniyle hayatını kaybetmektedir. Derin

öğrenmede kullanacağımız veri seti için internet ortamında bulduğumuz hazır gerçek hasta verileri kullanılacaktır. Uygulamayı kullanan hastalara sisteme girilen kan değerleri ve yaşam bilgileri doğrultusunda hastaya diyabet teşhisi konulabilecektir. Bu teşhis sonucunda hastanın vermiş olduğu e-posta adresine test sonucunu, test bilgilerini ve hastalık hakkında kapsamlı bilgi alacağı bir internet sitesi linkini içeren bir mail gönderilecektir.

#### **A.4.2. Yaygın Etki / Katma Değer**

Diyabet hastalığında erken ve doğru teşhis önemlidir çünkü ölümcül birçok hastalığın oluşumunda büyük rol oynamaktadır. Gerçek hasta verilerinden elde edeceğimiz veri setini Makine Öğrenmesi ile birleştirerek erken, hızlı ve doğru tanı koymayı hedefliyoruz. Böylece diyabet hastalarımız diyetisyenleri tarafından hazırlanacak özel beslenme programıyla şeker hastalığını kontrol altına alacaktır. Uygulamamız sayesinde yanlış pozitif, yanlış negatif ve geç tanı durumlarının önüne geçerek maddi açıdan katma değer sağlayacaktır. Bunun yanında yanlış tanıları azaltarak gerçekte diyabet hastası olmayıp tedavilere başlayan hastaların psikolojik sıkıntı çekmesi gibi durumların önüne geçmeyi hedeflemektedir.

#### **A.5. Yöntem**

Diyabet hastalığı tahmin sistemi projemizde, diyabet hastalığını tahmin etmek için destek vektör makine sınıflandırıcısını kullanıyoruz. Gebe kalma sayısı, Plazma glukoz konsantrasyonu, 2 saatlik oral glukoz tolerans testi, Diyastolik kan basıncı (mmHg), Triceps deri kıvrım kalınlığı (mm), 2 Saatlik serum insülini (mu U/ml), Vücut kitle indeksi (BMI) (kg cinsinden ağırlık/(m cinsinden boy)<sup>2</sup>), Diyabet soyağacı fonksiyonu, Yaş (yıl) ve sonuç sınıfı değişkeni (0 veya 1) içeren diyabet veri kümesi verileri eğitmek için kullanıyoruz.

Destek vektör makinesi denetimli bir öğrenme sistemidir ve sınıflandırma ve regresyon problemlerinde kullanılır. Destek vektör makinesi, daha az hesaplama gücüyle kayda değer doğruluk ürettiği için birçok kişi tarafından son derece tercih edilmektedir. Daha çok sınıflandırma problemlerinde kullanılır. Denetimli, denetimsiz ve takviyeli öğrenme olmak üzere üç tür öğrenmemiz var. Bir destek vektör makinesi, hiper düzlemi bölerek resmi olarak tanımlanan seçici bir sınıflandırıcıdır.

Etiketli eğitim verileri verildiğinde, algoritma yeni örnekleri sınıflandıran en iyi hiper düzlemi verir. İki boyutlu uzayda, bu hiperdüzlem, bir düzlemi, her sınıfın her iki tarafta yer







### A.7.2. Kişi – İş Açıklaması

Hakam Chedo	Hüseyin Kaymak	Bengisu Şahin	Muhammet Eren Söme
<ul style="list-style-type: none"> <li>● Problemin tanımlaması</li> <li>● Literatür araştırması</li> <li>● Sistem gereksinimlerinin belirlenmesi</li> <li>● Kullanıcı arayüzü oluşturma</li> <li>● Web uygulamasının test edilmesi</li> <li>● Final rapor ve sunumun hazırlanması</li> <li>● Entegrasyonlar</li> </ul>	<ul style="list-style-type: none"> <li>● Problemin tanımlaması</li> <li>● Literatür araştırması</li> <li>● Sistem gereksinimlerinin belirlenmesi</li> <li>● Veri setinden model oluşturma</li> <li>● Web uygulamasının yapılması</li> <li>● Final rapor ve sunumun hazırlanması</li> <li>● Entegrasyonlar</li> </ul>	<ul style="list-style-type: none"> <li>● Problemin tanımlaması</li> <li>● Literatür araştırması</li> <li>● Sistem gereksinimlerinin belirlenmesi</li> <li>● Veri setinden model oluşturma</li> <li>● Kullanıcı arayüzü oluşturma</li> <li>● Final rapor ve sunumun hazırlanması</li> </ul>	<ul style="list-style-type: none"> <li>● Problemin tanımlaması</li> <li>● Literatür araştırması</li> <li>● Sistem gereksinimlerinin belirlenmesi</li> <li>● Modelin sunucuda çalıştırılması</li> <li>● Web uygulamasının test edilmesi</li> <li>● Final rapor ve sunumun hazırlanması</li> <li>● Veri Tabanı Oluşturma</li> </ul>

## B. ANALİZ

### B.1. Sistem Gereksinimlerini Ortaya Çıkarma Yöntem ve Teknikleri

#### B.1.1. Yüz Yüze Görüşme

Projenin başlangıç aşamasından projenin bitimine kadar izlenecek yolun belirlenmesi için proje danışmanı Öğr. Gör. Dr. Yıldray ANAGÜN ile görüşülmüştür. Yapılan görüşmeler doğrultusunda projede ilerleme kaydedilmiştir.

#### B.1.2. Yazılı Basılı Belge İnceleme

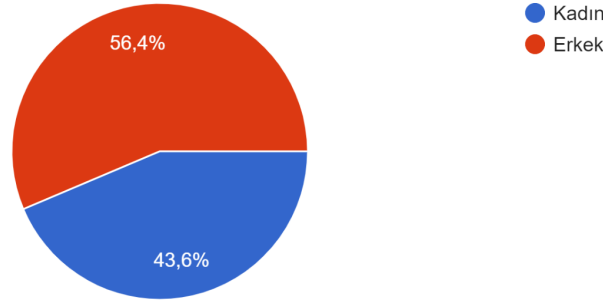
Projenin planlama aşamasında itibaren benzer makaleler incelenmiş, K-Nearest Neighbors (KNN), Random Forest (RF), Support Vector Machine (SVM), Artificial Neural Network (ANN) ve Decision Tree (DT) gibi sınıflandırma algoritmaları değerlendirilmiştir.

#### B.1.3. Anket

Anket çalışmasında kişilerin diyabet hastalığıyla ilgili bir şikayetlerinin olup olmadığı ve hastalığa sahip olup olmadığıyla ilgili görüşlerine yer verilmiştir. Bu anket çalışmasıyla

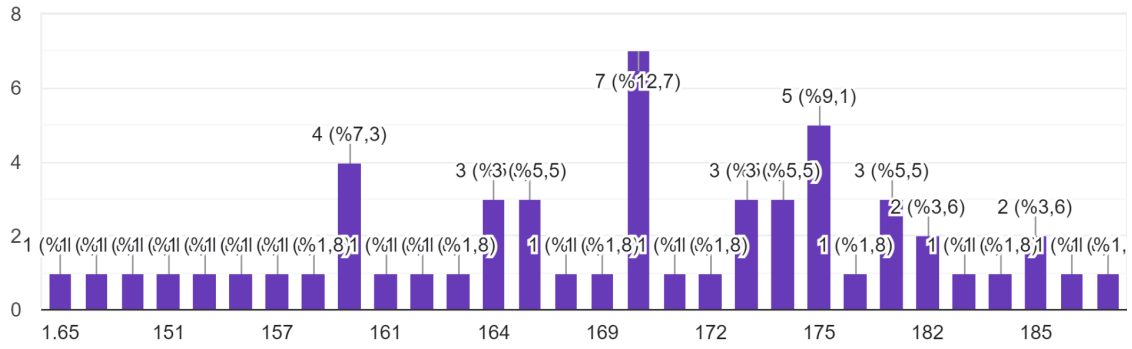
diyabet hastalığının farkındalığını ölçmeyi amaçladık. Anket sonucunda kişilerin diyabet hastalığındaki önemli faktörlere yatkınlığını ve diyabet hastası olan kişilerin ankette sorulan sorular sonucunda sahip olduğu özellikleri tespit etmiş olduk. Anket 55 kişi üzerinde yapılmıştır. Katılımcıların %92,7 si diyabet hastası değilken geri kalan %7,3 lük kısmı diyabet hastalığına sahiptir.

**Soru:** Cinsiyetiniz nedir ?



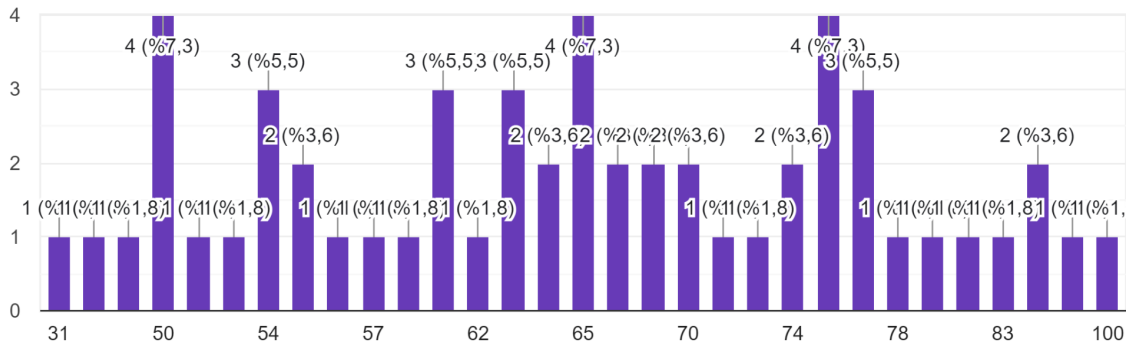
Şekil 1. Anket Soru 1 ve Sonuçları

**Soru:** Boyunuz nedir? (cm)



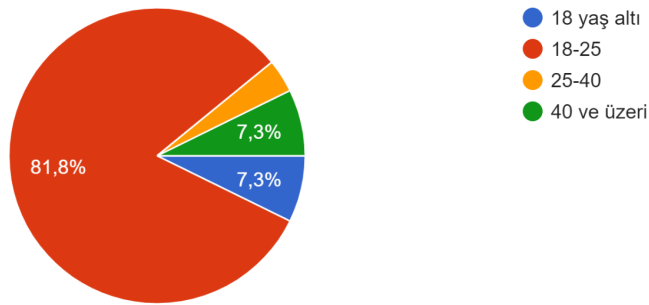
Şekil 2. Anket Soru 2 ve Sonuçları

**Soru:** Kilonuz nedir? (kg)



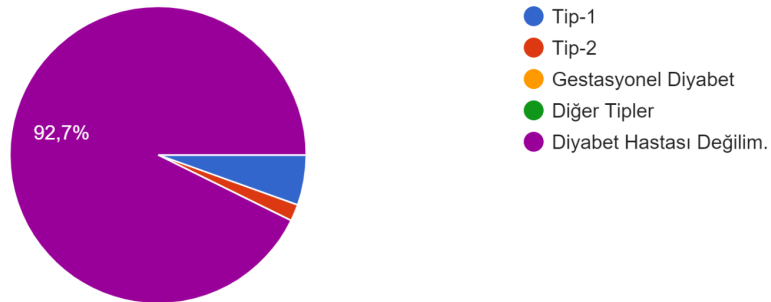
Şekil 3. Anket Soru 3 ve Sonuçları

**Soru:** Kaç yaşındasınız?



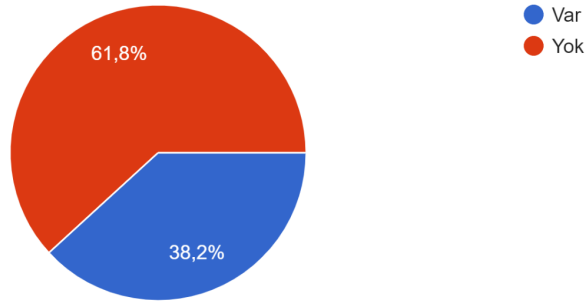
Şekil 4. Anket Soru 1 ve Sonuçları

**Soru:** Hangi tip diyabete sahipsiniz ?



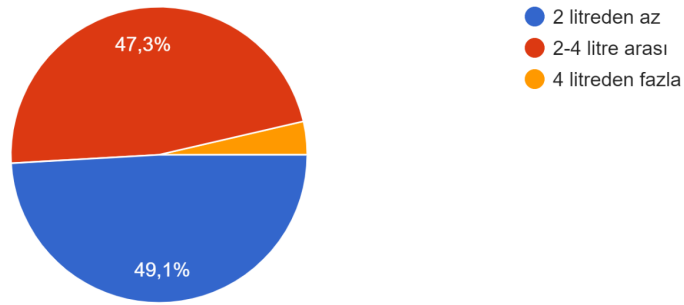
Şekil 5. Anket Soru 5 ve Sonuçları

**Soru:** Ailenizde diyabet hastası var mı?



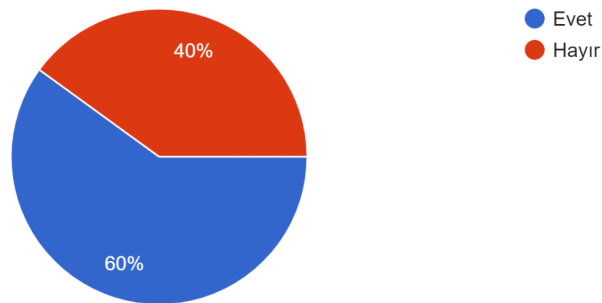
*Şekil 6. Anket Soru 6 ve Sonuçları*

**Soru:** Günde kaç litre su içiyorsunuz?



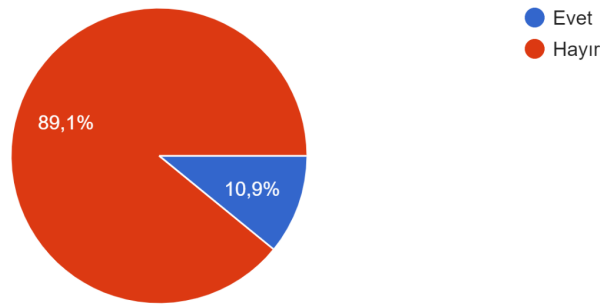
*Şekil 7. Anket Soru 7 ve Sonuçları*

**Soru:** Fiziksel olarak aktif misiniz?



*Şekil 8. Anket Soru 8 ve Sonuçları*

**Soru:** Size hiç yüksek tansiyon tanısı konuldu mu?



*Şekil 9. Anket Soru 9 ve Sonuçları*

#### **B.1.4. Gözlem**

Proje kapsamında yapılan literatür taraması ve araştırmalar sonucunda bu projenin gerçekleştirilebileceği, ayrıca hem hastalar hem de doktorlar tarafından kullanılabileceği verilerle kanıtlanmıştır. Projemiz ilk aşamada diyabet hastalığı teşhisinde uygulanmasına rağmen uygun veri seti oluşturulması halinde diğer hastalıklarda da başarı ile uygulanabilir olduğu sonucuna varılmıştır. Bununla birlikte proje ilk olarak sadece belirli bir konu üzerinde uygulanabilmesine rağmen daha sonraki süreçte hasta veya doktorlardan gelecek isteklere göre gelişmesi mümkün olabilecektir.

#### **B.1.5. Prototip ve Hızlı Uygulama Tasarımı (Rapid Application Design – RAD)**

Projemizin web uygulaması kısmında kullandığımız örnek veri setini test etmek ve oluşturduğumuz veri modelini ölçmek için hızlı bir şekilde makine öğrenmesi modeli geliştirilmiştir. Hazırlanan modelin eğitimi ve testi Python programlama dili ile yazılmıştır. Modelin test aşamasında ise eğitim aşamasında kullanılan verilerden farklı olarak hasta kişinin girdileri için örnek girdiler oluşturulmuş ve modele verilmiştir. Bunun sonucunda modelin diyabet hastalığını tespiti başarılı bir şekilde sonuçlanmıştır. Test aşaması sonucunda elde edilen çıktı ile modelin kullanılabilirliği doğrulanmıştır. Kullanıcı arayüzü bir web uygulaması olacağından basit bir tasarım ve deneme çalışması yapılmıştır. Modelin çıktısı bu deneme çalışması ile prototip olarak oluşturulmuştur. Kullanacak olan hekim ve hastalara

prototip sunularak konu hakkındaki fikirleri alınmıştır. Şekil 10'da oluşturulan prototip çalışması görülmektedir. Modelin sonucunda elde edilen başarı oranının projenin ilerleyen aşamalarında artırılması hedeflenmektedir.

### B.1.6. Ortak Uygulama Tasarımı (Joint Application Design – JAD)

Projeyi hazırlayanlar da kullanıcı olduğu için sistemin gereksinimleri belirlenirken hem geliştirici hem de kullanıcı olarak sistem çözümlenmeye çalışılmıştır. Prototip ve Hızlı Uygulama Tasarımı kapsamında Ortak Uygulama Tasarımı gerçekleştirilmiştir. Bu aşamada, Şekil 10'de gösterilen örnek diyabet hastalığını tespit etmek için Makine öğrenmesi kullanılarak yapılan girdi verilerinin çıktısı ve Şekil 11'de gösterilen web uygulamasındaki diyabet hastalığı tespit sisteminin çıktısı verilmektedir.

```
Input:(5, 166, 72, 19, 175, 25.8, 0.587, 51)
Output: The person is diabetic
```

Şekil 10. Makine öğrenmesi kullanılarak diyabet hastalığını tespit edilen çıktısı

**Diyabet Hastalığı Tespit Sistemi**

Ad: Hakam Soyad: Chedo E-posta: magnitar123@gmail.com

**Diyabet Test Bilgileri**

Gebelik sayısı: 0 Glikoz miktarı: 10 Kan basıncı: 90

Cilt kalınlığı değeri: 2 İnsülin miktarı: 5 BMI (Vücut kitle indeksi): 27

Diyabet soyağacı fonksiyonu: 1 Yaş: 23

☒ Bilgilerimin işlenmesini onaylıyorum. (Girilen bilgiler üçüncü kişiler ile paylaşılmayacaktır.)

Test Sonucunu Görüntüleyin

**Diyabet Hastası Değilsiniz**

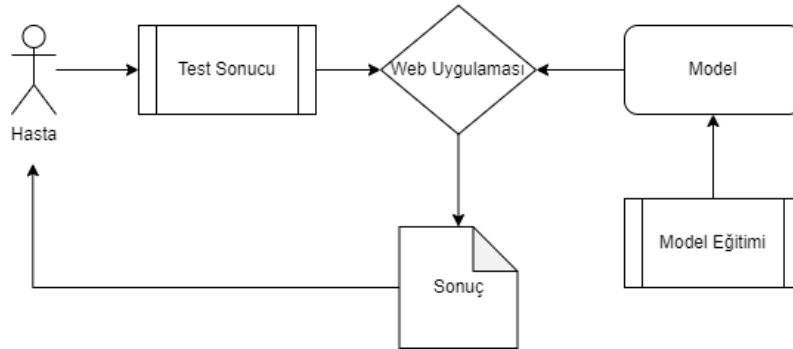
Şekil 11. Web uygulamasındaki diyabet hastalığı tespit sisteminin örnek çıktısı



### B.1.7. Veri Akış Şemaları

#### B.1.7.1. Kavramsal Veri Akış Şeması

Kavramsal veri akış şemasında hasta, test sonuçlarını web uygulamasına girerek geliştirdiğimiz makine öğrenim modeli ile sonuç elde etme süreci gösterilmektedir.

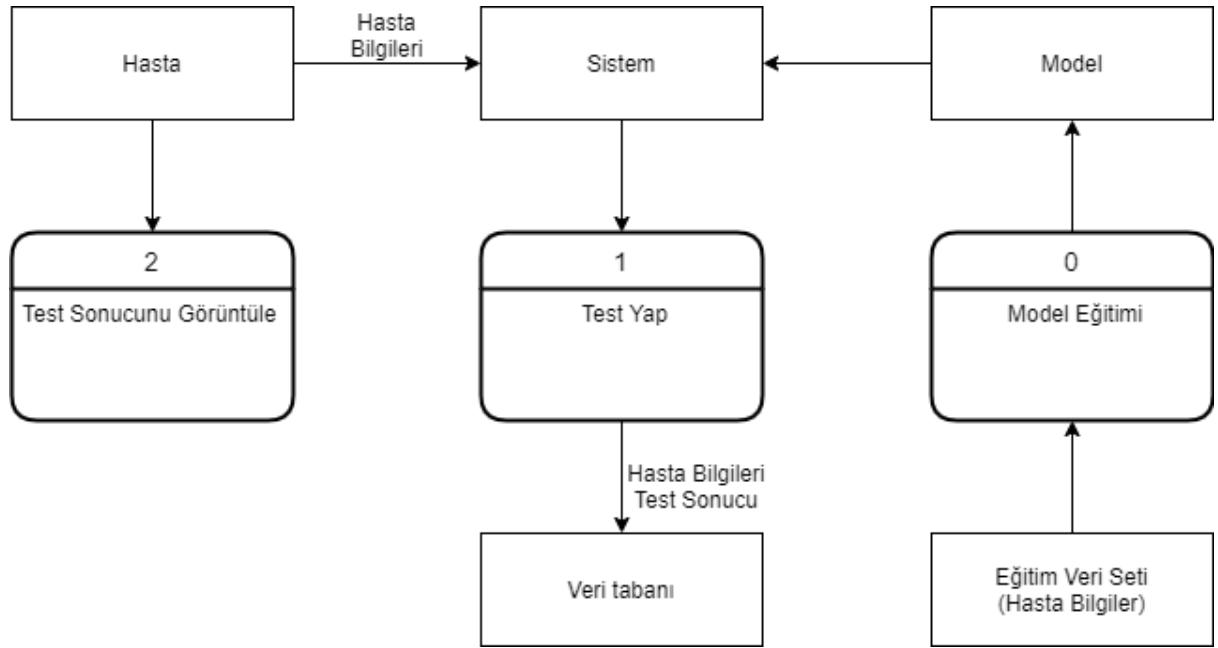


Şekil 12. Kavramsal Veri Akış Şeması

#### B.1.7.2. Mantıksal Veri Akış Şeması

Mantıksal veri akış şemasında, sistem boyunca veri akışlarının gösterilmesi hedeflenmiştir. Şekil 13’te verilen mantıksal veri akış şemasında 3 varlık ve 3 işlemden oluşmaktadır. Sistemdeki varlıklar ve işlemler arasındaki veri akışı maddeler halinde aşağıda açıklanmıştır.

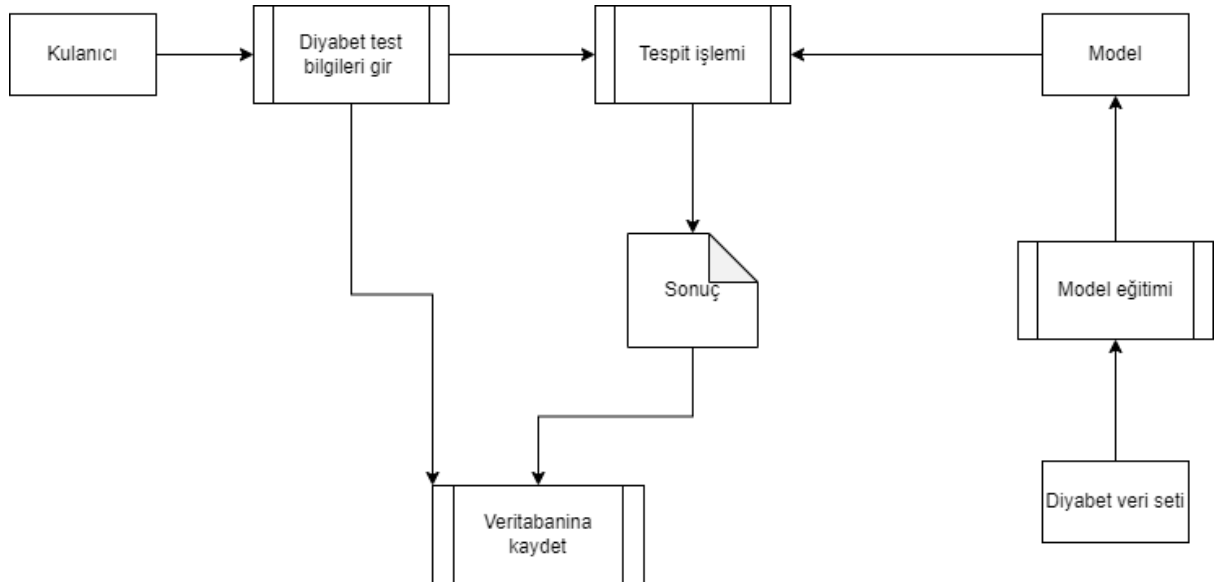
- Model isimli varlık hastalığı teşhis eden makine öğrenmesi modelidir ve “Model Eğitimi (0)” işlemi sonucunda oluşur. Bu işlemin girdisi “Eğitim Veri Seti” varlığıdır ve modeli eğitmek için bu veri seti kullanılır. Veri setinin içinde hasta bilgileri ve test sonuçları bulunmaktadır.
- Sistem isimli varlık web uygulamasını temsil etmektedir.
- Sistem “Test Yap (1)” işlemi ile “Hasta” varlığından aldığı “Hasta Bilgileri” ile test yapmaktadır ve sonucu yazdırmaktadır. Hasta bilgileri ve sonuçlar veri tabanına kaydedilmektedir. Veritabanı
- Hasta isimli varlık uygulamayı kullanan hastaları temsil etmektedir.
- Hasta, test işlemi için bilgilerini sisteme girer.
- Hasta testin sonucunda sağladığı e-posta adresine test bilgilerini içeren bir mail alacaktır.



Şekil 13. Mantıksal Veri Akış Şeması

### B.1.7.3. Fiziksel Veri Akış Şeması

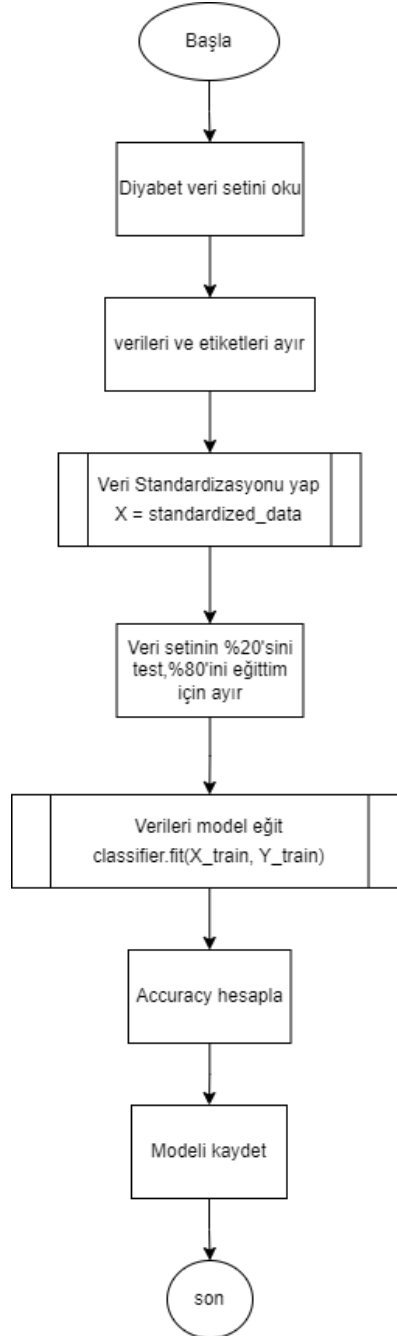
Proje kapsamında verilerin yer değiştirdiği fiziksel ortamlar ve hangi fiziksel yollarla iletilip kaydedildiğini gösteren fiziksel veri akış şeması Şekil 14'de verilmiştir. Kullanıcı diyabet test bilgilerini girip tespit işlemine geçiyor.



Şekil 14. Fiziksel Veri Akış Şeması

#### B.1.7.4. İş Akış Şeması

Diyabet tespit yapan makine öğrenme modelinin eğitim aşamasının akış şeması Şekil 15’de gösterilmektedir. İlk olarak diyabet veri seti okunup verilerini ve etiketleri ayrılır. Veriler veri standardizasyonu yapılır. Veri setinin %20’si test ve %80’i eğitim verisi olmak üzere ikiye ayrılır. Eğitilecek veriler destek vektörünün eğitimi makine sınıflandırıcı kullanarak eğitilir. Sonunda eğitilen veriler veri set modelini olarak kaydedilecek.



Şeki 15. Makine öğrenme modeli İş Akış Şeması

## B.1.8. Olay Tabloları, Durum Formları, İşlevsel Analiz Raporu, İş Akış Şeması

### B.1.8.1. Olay Tabloları

Olay	İstemci	Tetikleyici	İşlem	Yanıt	Hedef
Model oluşturulması	Uygulama	Eğitim Veri Seti	Modelin eğitim setine göre oluşturulması	Model	Uygulama
Hasta veya hekimden hastanın sağlık bilgilerinin alınması	Uygulama	Hasta sağlık bilgileri	Hastadan sağlık bilgilerinin alınması	Uygulama	Model
Hastalık tespiti sınıflandırılması	Uygulama	Hasta sağlık bilgileri	Modele göre hasta girdilerinin sınıflandırılması	Model	Uygulama
Hastanın girdilerine yanıt gönderilmesi	Hasta, Hekim	Girdilere yanıt	Hastalık tespitinin hastaya iletilmesi	Uygulama	Hasta
Test sonucunun hastaya e-posta yoluyla gönderilmesi	Uygulama	Test	Sonucun ve değerlerin hastaya iletilmesi	Uygulama	Hasta

### B.1.8.2. Durum Formları

<b>Durum İsmi:</b> Model oluşturulması			<b>İşlem No:</b> 1
<b>Tanım:</b> Eğitim veri seti kullanılarak model oluşturulur.			
<b>Tetikleyici:</b> Eğitim veri seti			
<b>Tetikleyici Türü:</b> İç kullanıcı   Dış kullanıcı   Başka bir işlem   Başka bir sistem			
<b>Girdi İsmi</b>	<b>Kaynak</b>	<b>Çıktı</b>	<b>Hedef</b>
Eğitim veri seti	Eğitim veri seti	Model	Uygulama

<b>Durum İsmi:</b> Hasta veya hekimden hastanın sağlık bilgilerinin alınması			<b>İşlem No:</b> 2
<b>Tanım:</b> Test sonuçlarından sağlık bilgilerinin alınması			
<b>Tetikleyici:</b> Hasta sağlık bilgileri			
<b>Tetikleyici Türü:</b> İç kullanıcı   Dış kullanıcı   Başka bir işlem   Başka bir sistem			
<b>Girdi İsmi</b>	<b>Kaynak</b>	<b>Çıktı</b>	<b>Hedef</b>
Test sonuçları	Test sonuçları	Test sonuçları	Model

<b>Durum İsmi:</b> Hastalık tespiti sınıflandırılması			<b>İşlem No:</b> 3
<b>Tanım:</b> Modele göre hasta girdilerinin sınıflandırılması			
<b>Tetikleyici:</b> Hasta sağlık bilgileri			
<b>Tetikleyici Türü:</b> İç kullanıcı   Dış kullanıcı   Başka bir işlem   Başka bir sistem			
<b>Girdi İsmi</b>	<b>Kaynak</b>	<b>Çıktı</b>	<b>Hedef</b>
Test sonuçları	Test sonuçları	Hastalık sonucu	Uygulama

<b>Durum İsmi:</b> Hastanın girdilerine yanıt gönderilmesi			<b>İşlem No:</b> 4
<b>Tanım:</b> Hastalık tespitinin hastaya iletilmesi			
<b>Tetikleyici:</b> Hasta sağlık bilgileri			
<b>Tetikleyici Türü:</b> İç kullanıcı   Dış kullanıcı   Başka bir işlem   Başka bir sistem			
<b>Girdi İsmi</b>	<b>Kaynak</b>	<b>Çıktı</b>	<b>Hedef</b>
Test sonuçları	Analiz sonucu	Hastalık sonucu	Hasta

### B.1.8.3. İşlevsel Analiz Raporu

**Proje Kodu:** DPSML

**Proje Adı:** Makine Öğrenmesi ile Diyabet Hastalığının Teşhisi

**Hazırlayanlar:** Bengisu ŞAHİN, Hakam CHEDO, Hüseyin KAYMAK, Muhammet Eren SÖME

**Başlangıç Tarihi:** 17.10.2022

**Son Değiştirme Tarihi:** 14.12.2022

**Versiyon:** 1.1

**İlgili Kişi:** Öğr. Gör. Dr. Yıldırım ANAGÜN

**Görevi:** Proje Danışmanı

**Mevcut Sistemlere Etkisi:** Diyabet hastalığı şüphesi olan hastalara hızlı test yaparak hastalığın erken teşhisini sağlama.

**Kırtasiye ve Sarf Malzemesi İhtiyacı:** Herhangi bir kırtasiye ve sarf malzemesi ihtiyacı bulunmamaktadır.

**Dönüştürme:** Hasta bilgilerinin ve şikayetlerinin işlendikten sonra veri seti genişletilerek sistemi etkilemeyecek şekilde performans arttırılacaktır.

**Donanım:** Projenin oluşturulması ve test aşamasında şu anki ekipmanlar yeterlidir. Ancak gerçek zamanlı çalışmalarda daha iyi sistemlere gerek duyulabilir.

**Maliyetler:** Lisanslama açısından bir maliyet bulunmamaktadır. Gerçek zamanlı çalıştırmada donanım, hosting ve domain masraflarından dolayı maliyet artar.

## B.2. Sistem Gereksinimleri

### B.2.1. İşlevsel Gereksinimler

GEREKSİNİM ID	AÇIKLAMA
EDD (Enter Diyabet Data)	Kullanıcı, hastaya ait glikoz seviyesini hamilelik sayısını, kan basıncı değerini, insülin seviyesini, vücut kitle indeksini, ailesindeki diyabet soyağacı yatkınlığını, yaşını, kişinin cildindeki kolajen miktarı (deri kalınlığı) bilgilerini girerek diyabet hastalığını tespit edebilir
GDI (Get Diyabet Disease Information )	Kullanıcı, diyabet hastalığını tespit ettikten sonra kullanıcının e-mailine gönderilen

	sonuç ile diyabet hastalığının bilgileri bir link tıklayarak öğrenilir
--	--

### B.2.2. Sistem ve Kullanıcı Arayüzleri ile İlgili Gereksinimler

- Sistemde bir web uygulaması, makine öğrenmesi modelinin çalıştığı sunucu yer almaktadır. Web uygulaması, oluşturulan makine öğrenmesi modeli ile iletişim kurmalıdır.
- Web uygulaması kullanıcının (hasta veya hekim) girdiği hastaya ait glikoz seviyesini, hamilelik sayısını, kan basıncı değerini, insülin seviyesini, vücut kitle indeksini, ailesindeki diyabet soyağacı yatkınlığını, yaşını, kişinin cildindeki kolajen miktarı (deri kalınlığı) bilgilerine ihtiyaç duyar ve bu bilgileri kullanıcıdan alır.
- Kullanıcı girdileri sonucu elde edilen sonuç aynı arayüzde belirgin ve anlaşılır şekilde kullanıcıya gösterilecektir.
- Kullanıcılar geliştirilecek web uygulamasını kullanmak için bir web tarayıcısı kullanmalıdırlar.
- Kullanıcılara sade ve anlaşılır bir arayüz sunularak, kullanıcı dostu bir web uygulaması geliştirilmesi planlanmaktadır.

### B.2.3. Veriyle İlgili Gereksinimler

- Örnek test sonuçlarını içeren veri seti csv dosyasında saklanacaktır.
- Kullanıcıların uygulamaya girdiği değerler veritabanına kaydedilecektir.
- Veritabanına eklenen yeni veriler, istenildiğinde modelin geliştirilmesi için kullanılabilir.
- Diyabet hastalığı tespitinde doğru sonuca ulaşmak için gereken veriler:
  - Ad
  - Soyad
  - E-mail
  - Gebelik sayısı
  - Glikoz miktarı
  - Kan basıncı
  - Cilt kalınlığı değeri
  - İnsülin miktarı
  - BMI (Vücut kitle indeksi)

- Diyabet soyağacı fonksiyonu (Diabetes pedigree function)
- Yaş

#### **B.2.4. Kullanıcılar ve İnsan Faktörü Gereksinimleri, Güvenlik Gereksinimleri**

- Sistemde tek tip kullanıcı (Hasta, Hekim vb.) olacaktır. Kullanıcı, bilgilerini sisteme giren ve hastalık sonucunu öğrenen kişilerdir.
- Kullanıcıların uygulamayı kullanabilmeleri için bir web tarayıcısına ihtiyacı vardır. Bu nedenle kullanıcıların bir web tarayıcısı kullanmayı bilmeleri gerekmektedir.
- Kullanıcıların uygulamayı kullanmaları için herhangi bir eğitime ihtiyacı yoktur.
- Kullanıcılardan ad, soyad ve e-posta gibi kişisel bilgiler ve test için sağlık bilgileri istenmektedir. Uygulamayı kullanacak her kullanıcı bu bilgilerin toplanmasını, depolanmasını ve işlenmesini onaylamış olur.
- Sistemde şifre gereksinimi yoktur. Testler herhangi bir hesap aracılığıyla olmayacaktır.

#### **B.2.5. Teknik ve Kaynak Gereksinimleri, Fiziksel Gereksinimler**

- Web uygulamasını kullanabilmek için web tarayıcı yazılımı bulunan cihazlara ihtiyaç duyulmaktadır.
- Destek vektör makine sınıflandırıcı gibi yüksek doğruluğa sahip bir makine öğrenme modeli kullanılmıştır.
- Modeli geliştirmek için Python programlama dili kullanılmıştır.
- Makine öğrenme yöntemlerinde kullanılan işlemleri hızlandırmak adına sklearn, numpy, pandas gibi hazır kütüphaneler kullanılmıştır.
- Model veri setinin sayısının artırılması ile geliştirilmeye açıktır.
- Web uygulamasının geliştirilmesi için Python'un streamlit hazır kütüphanesi kullanılmıştır. Geliştirme sırasında karşılaşılabilecek problemlere göre başka teknolojiler de kullanılabilir.



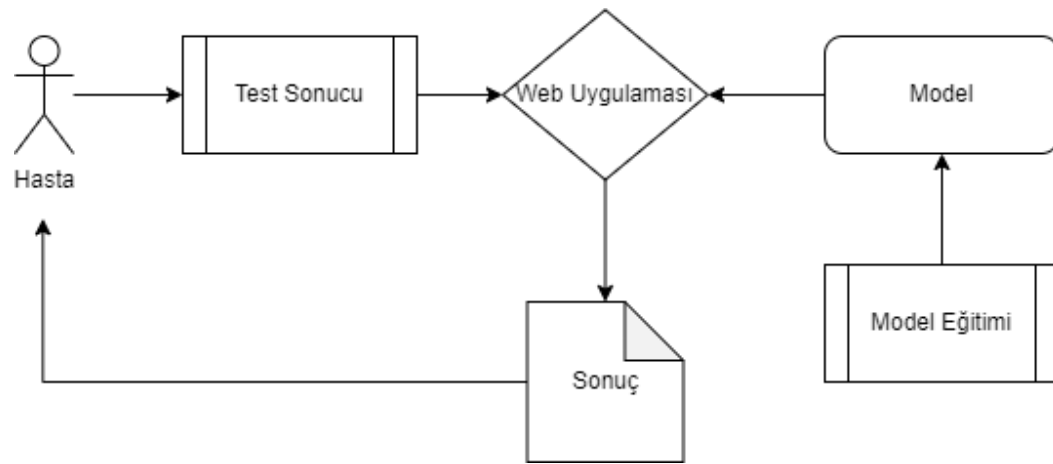
## C. TASARIM

### C.1. Sistem Tasarımları

Makine öğrenimi modelinin eğitimini ve testini gerçekleştirmek için The N. Inst. of Diabetes & Diges. & Kidney Dis' den diyabet veri kümesi kullanıyoruz.

Python, diyabet veri setini eğitmek ve diyabet tahmini web uygulama sistemi tasarlamak için bir programlama dilidir.

sklearn python kütüphanesi, diyabet veri setini taramak için kullanılır ve streamlit python kütüphanesi, diyabet tahmini web uygulama sistemi tasarlamak için kullanılır.



Şekil 16. Sistem Tasarımları

### C.2. Kullanıcı ve Sistem Arayüzü Tasarımları

Kullanıcı dostu olması için tek sayfalık bir web sistemi oluşturduk. Kullanıcılar, kullanıcı verilerini girebilecek ve test sonuçlarını hemen görüntüleyebilecek.

The screenshot shows the 'Diyabet Hastalığı Tespit Sistemi' (Diabetes Disease Detection System) interface. It features a dark theme with white text. The form includes input fields for personal information (Ad, Soyad, E-posta) and diabetes-related data (Gebelik sayısı, Glukoz miktarı, Kan basıncı, Cilt kalınlığı değeri, İnsulin miktarı, BMI, Diyabet soyağacı fonksiyonu, Yaş). A 'Diabetes Test Result' button is present, and the output shows 'şeker hastası değilsiniz' (you are not a diabetic patient) in a green box.

Şekil 17. Kullanıcı ve Sistem Arayüzü Tasarımları

### C.2.1. Kullanıcı Veri Girişi/Seçim Arayüzleri

Bu kısımda kullanıcı kullanıcı adını soyadını ve diyabet hastalığı bilgilerini girebilir.

Şekil 18. Kullanıcı Veri Girişi/Seçim Arayüzleri

### C.2.2. Raporlama/Bilgilendirme vb Arayüzleri

Değerler girdikten sonra kullanıcı test result butonuna basıp sonucu görebilir. Ayrıca test sonucu ve girilen değerler E-posta olarak kullanıcıya gönderilecektir.

Şekil 19. Raporlama/Bilgilendirme vb Arayüzleri

## C.3. Veri Tabanı Tasarımı

Projemiz bir web uygulamasıdır ve kullanılacak verilerin tamamı online bir veri tabanında tutulacaktır. Web Hosting firmasından alınan bir sunucu üzerine MSSQL kurularak veri tabanı oluşturulacaktır. Web uygulamasının veri tabanı ile iletişimi bir service ile gerçekleştirilecektir. Bu servis Python programlama dili ile veri tabanına erişim sağlamaktadır.

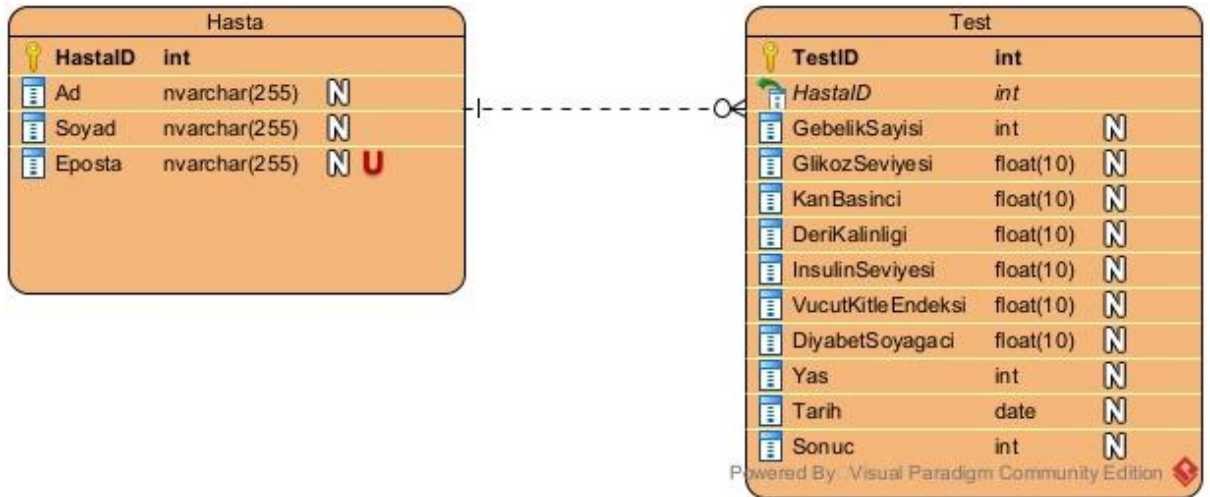
### C.3.1. Veri Tabanı İsterler Dökümü

- Uygulamada tek tip kullanıcı vardır ve bu da hastaları temsil etmektedir.
- Her kullanıcıdan diyabet testi için gerekli olan hastalık bilgileri istenmektedir.
- Kullanıcıların testleri ayrı bir tabloda hastalık bilgileri, tarihi ve sonucuyla birlikte tutulur.
- Bir kullanıcı birden fazla defa test yaptırabilir.
- Kullanıcı sistemden silindiği zaman yaptırdığı testlere ve diyabet bilgilerine erişilemez.

### C.3.2. E/R Diyagramı

Şekil 20’de veri tabanımızın E/R Diyagramı bulunmaktadır. Bu diyagram Visual Paradigm programı ile çizilmiştir. Projede kullanılmak üzere oluşturulan veri tabanında toplam 2 tane tablo bulunmaktadır. Bu tablolar sırasıyla: Hasta ve Test tablolarıdır. Sistemimiz şifresiz üyelik ile çalıştığından dolayı üyelerin bilgileri Hasta tablosunda saklanmaktadır. Hasta tablosunda HastaID, Ad, Soyad, E-posta bilgileri tutulmaktadır.

Üyelik işlemini tamamlayan hastalar test için istenen Hastalık Bilgilerini dolduracaktır. Bu bilgiler Test tablosunda tutulacaktır. Test tablosunda TestID, HastaID, gebelik sayısı, glikoz seviyesi, kan basıncı, deri kalınlığı, insülin seviyesi, vücut kitle endeksi, diyabet soyağacı, yaş, tarih ve sonuç bilgileri tutulmaktadır.



Şekil 20. E/R Diyagramı

### C.3.3. Stored Procedures, Triggers, Kısıtların Tasarımı

Kısıtlarımız Primary Key, Foreign Key ve Unique kısıtlarıdır.

**Primary Key:** Primary Key kısıtı tablodaki her kaydın eşsiz olmasını sağlar. Primary Key kısıtına sahip sütunlar boş bırakılamaz (Not Nullable). Bir tablonun yalnızca bir adet Primary Key kısıtı olabilir ancak bu kısıta tek kolon dahil edilebildiği gibi birden fazla kolon da birlikte dahil edilebilir. Bu kısıt Hasta tablosunda HastaID için bulunmaktadır. Böylece her hastaya eşsiz bir id vererek hastaların birbirinden farklı olmasını sağlar. Aynı kısıt Test tablosundaki TestID alanı için de geçerlidir.

**Foreign Key:** Foreign Key kısıtı bir tablodaki kolonda bulunan verinin ilişkili başka bir tablodaki kolondaki verilerle aynı olmasını ister. Bu sayede iki tablo arasında referans bütünlüğü sağlanmış olur. Test tablosunda bulunan HastaID alanı Foreign Key kısıtına girer, böylece test tablosundan çekilen verilerin hangi hastayla ilişkili olduğu anlaşılır.

**Unique:** Bu kısıt, tanımlandığı kolonun tüm satırlarındaki verilerin birbirinden farklı olmasını gerektirir. Böylece benzersiz verilerle oluşturulmuş bir veri kolonu elde edilebilir. Hasta tablosunda bulunan e-posta alanı için bu kısıt geçerlidir. Böylece veri tabanına eklenen her hastanın e-postaları eşsiz olmak zorundadır. Bir e-posta adresi üzerinde birden fazla hasta kaydı bulunamaz.

**Stored Procedure:** Saklı yordamlar belli bir işlevi yerine getirmek için kullanılan kod parçacıklarıdır. Uygulamamızda toplam 4 adet saklı yordam bulunmaktadır. Bunlar:

- **HastaEkle:** Verilen bilgilere göre Hasta tablosuna yeni hasta kaydı açar.
- **TestEkle:** Verilen bilgilere göre Test tablosuna yeni test kaydı açar.
- **HastaTestGetir:** E-postası belirtilen hastanın bütün test sonuçları yeniden eskiye göre sıralandırarak getirir.
- **BelirtilenSonucTestGetir:** Belirtilen test sonucuyla (1 = pozitif, 0 = negatif) uyuşan bütün testleri yeniden eskiye göre sıralandırarak getirir.

BelirtilenSonucTestGetir saklı yordamımızda girdi olarak test sonucu alınıyor ve çıktı olarak test sonuçları istenen şekilde olan hastaların; TestID, Ad-Soyad, Yaş ve Test Tarihi ile Sonucu döndürülüyor. Döndürülen liste alfabetik ve kronolojik olarak sıralandırılıyor.

```
--BelirtilenSonucTestGetir--
CREATE PROCEDURE BelirtilenSonucTestGetir @sonuc int
AS
BEGIN
SELECT t.TestID as 'Test Numarası', h.Ad + ' ' + h.Soyad as 'Ad Soyad', t.Tarih as 'Test Tarihi', t.Sonuc as 'Test Sonucu'
FROM Hasta h, Test t
WHERE t.HastaID = h.HastaID AND t.Sonuc = @sonuc
ORDER BY 'Ad Soyad', t.Tarih DESC
END
```

Şekil 21. BelirtilenSonucTestGetir Saklı Yordamı

	Test Numarası	Ad Soyad	Test Tarihi	Test Sonucu
1	6	Eren Söme	2022-12-31	1
2	1	Hüseyin Kaymak	2022-12-31	1
3	4	Hüseyin Kaymak	2022-12-21	1

Şekil 22. BelirtilenSonucTestGetir çıktısı

**Trigger:** Tetikleyici (Trigger) yapısı, ilişkisel veri tabanı yönetim sistemlerinde, bir tabloda belirli olaylar meydana geldiğinde veya gelmeden önce otomatik olarak çalışan özel bir saklı yordam türüdür. Bizim uygulamamızda Test tablosuna bir satır eklendiğinde tetikleyicimiz, tarih sütununun boş bırakılıp bırakılmadığını kontrol eder. Eğer tarih sütunu boş ise güncel zamana eşitler.

```
CREATE TRIGGER Test_Tarih
ON dbo.Test
AFTER INSERT
AS
begin
UPDATE dbo.Test
SET Tarih = GETDATE()
WHERE Tarih IS NULL;
end
```

Şekil 23. Test\_Tarih Tetikleyicisi

## C.4. Yazılım Tasarımı

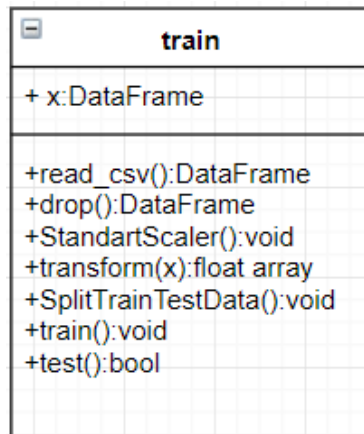
### C.4.1. Gereksinime Bağlı Tasarım Kalıpları Seçimi

Geliştirilecek uygulama için herhangi bir tasarım kalıbı kullanılmasına ihtiyaç duyulmamaktadır.

### C.4.2. UML Kullanarak Tasarım Diyagramları Oluşturma

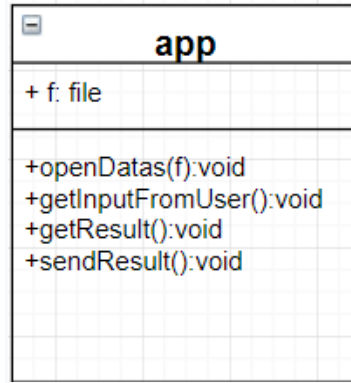
Model eğitim fonksiyonel UML diyagramındaki train sınıfı makine öğrenmesi modelini eğitmek için kullanılır. Şekilde verilen UML diyagramı bu sınıf ile model arasındaki aşağıdaki işlemler gerçekleştirilmektedir.

- .csv uzantılı dosyadaki veri seti okunur.
- Veri setinde ön işleme yapılır.
- Model eğitilir.
- Model test edilir.



Şekil 24. Model eğitim UML diyagramı

Aşağıda verilen şekilde kullanıcı tarafında çalışan programda bulunan dört tane fonksiyon verilmiştir. openFile fonksiyonu veri setini okur. getInputFromUser fonksiyonu kullanıcının ara yüzden girdiği bilgileri alır. getResult fonksiyonu eğitilen makine öğrenmesi modeline göre kullanıcının girdiği bilgilere göre bir tahmin üretir. showResult fonksiyonu getResult sonucunda elde edilen sonucu kullanıcı arayüzünde kullanıcıya gösterir.



Şekil 25. Uygulama UML diyagramı

## C.5. Test Tasarımı

Yazılım testi, belirli gereksinimleri karşılayıp karşılamadığını belirlemek ve herhangi bir kusuru belirlemek için bir yazılım uygulamasını veya sistemini değerlendirme sürecidir. Yazılımın yüksek kalitede ve kullanım amacına uygun olmasını sağlamaya yardımcı olduğu için yazılım geliştirme sürecinde önemli bir faaliyettir. Etkili yazılım testi, yazılım uygulamalarının kalitesini ve güvenilirliğini sağlamak için gereklidir. Zamandan ve kaynaklardan tasarruf sağlayabilen geliştirme sürecinin erken aşamalarında kusurları tespit edip düzeltmeye yardımcı olur.

Aşağıdakiler de dahil olmak üzere farklı yazılım testi türleri vardır:

**Unit Testing (Birim testi):** Bu, yazılımın bireysel birimlerine veya bileşenlerine odaklanan bir test türüdür. Her birimin amaçlandığı gibi çalışmasını sağlamak için genellikle geliştirme sürecinin bir parçası olarak geliştiriciler tarafından yapılır.

**Integration Testing (Entegrasyon testi):** Bu, yazılımın farklı bileşenlerinin veya birimlerinin entegrasyonunu doğrulamaya odaklanan bir test türüdür. Farklı bileşenlerin amaçlandığı gibi birlikte çalışmasını sağlamak için yapılır.

**System Testing (Sistem testi):** Bu, tüm yazılım uygulamasına veya sistemine odaklanan bir test türüdür. Yazılımın kullanım amacına uygun olması ve belirtilen gereksinimleri karşılaması için yapılır.

**Acceptance Testing (Kabul testi):** Bu, yazılımın yayına hazır olup olmadığını belirlemek için yapılan bir test türüdür. Genellikle yazılımın son kullanıcıları veya müşterileri tarafından yapılır.

### C.5.1. Gereksinim Analizlerinden Test Hedeflerinin Belirlenmesi

Modelin yeni veri eklenmesinden sonra yeniden eğitime tabi tutulur. Doğruluk değerinin %78 ve üzerinde olması hedeflenmektedir. Doğruluk oranının düşmesi durumunda veri setini kontrol etme ve sinir ağı yapısında değişiklik yapma gibi işlemler gerçekleştirilecektir.

### C.5.2. Birim Testleri Tasarımı

Birim testi, bir sistemde mantıksal olarak izole edilebilecek en küçük kod parçası olan bir birimi test etmenin bir yoludur. Projede birim testin uygulanacağı kısım makine öğrenme modelinin verilen değerlerden hastalık sonucunun tespiti.

```
import unittest
class TestResult(unittest.TestCase):
    def test(self):
        self.assertTrue(is_sick(6,148,72,35,0,33.6,0.627,50))

def is_sick(Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age):
    input_data = [Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age]
    input_data_as_numpy_array = np.asarray(input_data)

    # reshape the array as we are predicting for one instance
    input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
    print(input_data_reshaped)
    # standardize the input data
    std_data = scaler.transform(input_data_reshaped)
    print(std_data)
    prediction = trained_model.predict(std_data)
    print(prediction)
    if (prediction[0] == 0):
        return False
    else:
        return True

unittest.main()
```

Şekil 26. Birim Testleri Tasarımı

```
-----
Ran 1 test in 0.002s

OK
[[ 6.   148.   72.   35.    0.   33.6   0.627  50.  ]]
[[ 0.64602007  0.85464203  0.14982594  0.90469953 -0.68957614  0.20596027
   0.47274601  1.43277032]]
[1]
```

Şekil 27. Birim Testleri sonucu



### **C.5.3. Entegrasyon Testleri Tasarımı**

Yazılım geliřtirmede entegrasyon testi, bir sistemin veya uygulamanın farklı bölümlerinin beklendiđi gibi birlikte çalışıp çalışmadığını kontrol eden bir test türüdür. Bu testler genellikle bireysel bileşenler test edildikten sonra gerçekleştirilir ve farklı bileşenlerin beklenen şekilde entegre edilebilmesini ve birlikte çalışabilmesini sağlamayı amaçlar. Entegrasyon testleri, farklı modüllerin veya alt sistemlerin entegrasyonunun test edilmesini veya bir uygulamanın harici sistemler veya hizmetlerle entegrasyonunun test edilmesini içerebilir.

Web uygulaması, bir python kütüphanesi olan streamlit ile geliştirilmiştir. Kullanıcıdan alınan veriler veritabanına kaydedilmektedir. Test sonucu uygulama üzerinden gösterilmekte ve ayrıca mail olarak tüm bilgiler gönderilmektedir.

### **C.5.4. Sistem Testleri Tasarımı**

Sistem testi, belirtilen gereksinimleri karşıladığını ve amaçlandığı gibi çalıştığını doğrulamak için tüm sistemin veya uygulamanın test edilmesini içeren bir tür yazılım testidir. Sistem testi genellikle entegrasyon testinden sonra gerçekleştirilir ve sistemin bir bütün olarak düzgün çalışmasını ve kullanıcıların ihtiyaçlarını karşılamasını sağlamayı amaçlar.

Sistem testi, sistemin performansının, güvenilirliğinin, güvenliğinin ve diğer işlevsel olmayan gereksinimlerinin test edilmesini içerebilir. Ayrıca, sistemin kullanıcı arayüzünün ve kullanıcı deneyiminin test edilmesinin yanı sıra sistemin diğer sistemler veya harici hizmetlerle entegrasyonunun test edilmesini de içerebilir.

- Sistem performansını etkileyecek gereksiz işlemlerden kaçınılmıştır.
- Uygulamanın hastalık tespitindeki doğruluk değeri %78 dir.
- Uygulamaya girilen veriler üçüncü kişilerle paylaşılmaz.

```

Accuracy Score

[ ] # accuracy score on the training data
    x_train_prediction = classifier.predict(X_train)
    training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

[ ] print('Accuracy score of the training data : ', training_data_accuracy)

    Accuracy score of the training data :  0.7866449511400652

[ ] # accuracy score on the test data
    x_test_prediction = classifier.predict(X_test)
    test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

[ ] print('Accuracy score of the test data : ', test_data_accuracy)

    Accuracy score of the test data :  0.7727272727272727

```

*Şekil 28. Sistem Testleri Tasarımı*

### C.5.5. Kabul Testleri Tasarımı

Sisteme gerekli değerler girildiğinde, sistem doğru sonuç vermektedir. Ancak bu doğru sonuca ulaşmak için aşağıdaki kurallara uygun veri girişi yapılmalıdır.

- Ad, Soyad, E-Posta bilgileri boş olmamalıdır.
- Gebelik sayısı, Glikoz miktarı, Kan basıncı, Cilt kalınlığı değeri, İnsülin miktarı, BMI, Diyabet soyağacı fonksiyonu, Yaş gibi test sonuçlarının boş olmaması ve negatif değer girilmemesi gerekir.
- Test sonuçlarına rakamdan başka karakter girilmemesi gerekir.

## D. UYGULAMA

Bu fazda tasarım aşamasında planlanan sistem, kullanıcı arayüzü, test, yazılım ve veri tabanı tasarımlarının uygulama aşamasında nasıl gerçekleştiği incelenmiştir. Uygulama aşamasında ve tasarım aşamasında yapılan kullanıcı arayüz tasarımına uyulmuştur. Veritabanında verilerin saklanması değişiklikler yapılmıştır. Yapılan bu değişiklikler, tasarım aşamasında yapılması planlanan sistemin bütünlüğünü bozmayacak şekilde gerçekleşmiştir. Proje planlama fazında ana hatlarıyla belirlendiği şekliyle gerçekleştirilmiştir.

## D.1. Geliştirilen Sistemin Sistem Tasarımlarını Karşılamanın Değerlendirilmesi

Diyabet Tespit Sistemi projemiz sistem gereksinimlerini sağlar şekilde tamamlanmıştır ve gelecekte daha da geliştirilebilir.

Tamamlanan gereksinimleri:

- Hastalar diyabet verilerini kolayca girebilirler.
- Hastalar girdikleri bilgilerin paylaşılmasını kabul edebilirler.
- Hasta sonucu anında ekranda görebilir ve hastanın e-posta adresine gönderilir.
- Tahminin doğruluğunu oranı şu anda %78'dir, ancak daha fazla veri seti ile artırılabilir.

## D.2. Kullanıcı ve Sistem Arayüzü Gerçeklemeleri

Kullanıcı arayüzü, projenin tasarım aşamasında planlandığı üzere, web tabanlı bir uygulama olarak geliştirilmiştir. Web uygulama arayüzü olarak streamlit python kütüphanesi kullanılmıştır. Diyabet tespit sistemi web uygulaması henüz bir prototip olduğundan, çevrimiçi sunucuda yayınlamak için bir sunucu kiralamaya gerek yoktur.

Web uygulamasının veri tabanı ile haberleşmek için kullandığı servis pyodbc python kütüphanesi ile geliştirilmiştir. MSSQL ve veritabanı yerel sunucuda tüm kullanıcı bilgileri tutulmuştur.

Şekil 29'daki arayüz ile hasta, bilgilerini girerek test sonucunu görebilir.

Şekil 29. Uygulama Arayüzü

## D.3. Gerçeklenen Testler

### D.3.1. Fonksiyonel Testler

#### D.3.1.1. Birim (Unit) Testi

Projede birim testin uygulanan kısım, makine öğrenme modelinin verilen değerlerden hastalık sonucunun tespitidir. Model farklı değerler ile test edilmiştir ve testler başarıyla sonuçlanmıştır.

```

73 import unittest
74 class TestResult(unittest.TestCase):
75
76     def test(self):
77         self.assertTrue(is_sick(6,148,72,35,0,33.6,0.627,50))
78         self.assertTrue(is_sick(1,130,60,30,0,40.3,0.550,45))
79
80
81
82 def is_sick(Pregnancies,Glucose,BloodPressure,SkinThickness,Insulin,BMI,DiabetesPedigreeFunction,Age):
83     input_data = [Pregnancies,Glucose,BloodPressure,SkinThickness,Insulin,BMI,DiabetesPedigreeFunction,Age]
84     input_data_as_numpy_array = np.asarray(input_data)
85
86     # reshape the array as we are predicting for one instance
87     input_data_resaped = input_data_as_numpy_array.reshape(1,-1)
88     print(input_data_resaped)
89     # standardize the input data
90     std_data = scaler.transform(input_data_resaped)
91     print(std_data)
92     prediction = trained_model.predict(std_data)
93     print(prediction)
94     if (prediction[0] == 0):
95         return False
96     else:
97         return True
98
99
100 unittest.main()

```

Şekil 30. Birim (Unit) Testi

```

-----
Ran 1 test in 0.007s

OK
[[ 1.    93.    70.    31.     0.    30.4    0.315 23.   ]]
[[-0.84488505 -0.8730192  0.04624525  0.65635768 -0.69289057 -0.20212881
  -0.47378505 -0.87137393]]
[0]
The person is not diabetic
0
[[ 6.   148.    72.    35.     0.    33.6    0.627 50.   ]]
[[ 0.63994726  0.84832379  0.14964075  0.90726993 -0.69289057  0.20401277
  0.46849198  1.4259954  ]]
[1]
[[ 1.   130.    60.    30.     0.    40.3    0.55  45.   ]]
[[-0.84488505  0.28497518 -0.47073225  0.59362962 -0.69289057  1.05437172
  0.23594284  1.00055664]]
[1]

```

Şekil 31. Birim (Unit) Testi Sonucu

#### D.3.1.2. Entegrasyon Testi

Entegrasyon testinde uygulamanın model, veritabanı ve E-mail ile olan iletişimi test edildi.

- Uygulamaya girilen değerlerden doğru sonuç çıkmış, modelin doğru sonuç verdiği test edilmiştir. Sonuç “Diyabet Hastasıdır” veya “Diyabet Hastası Değilsiniz” şeklinde gösterilmiştir.

## Diyabet Hastalığı Tespit Sistemi

Ad  
Hakam

Soyad  
Chedo

E-posta  
magnitar123@gmail.com

### Diyabet Test Bilgileri

Gebelik sayısı  
0 - +

Glikoz miktarı  
10 - +

Kan basıncı  
90 - +

Cilt kalınlığı değeri  
2 - +

İnsülin miktarı  
5 - +

BMI (Vücut kitle indeksi)  
27 - +

Diyabet soyağacı fonksiyonu  
1 - +

Yaş  
23 - +

☒ Bilgilerimin işlenmesini onaylıyorum. (Girilen bilgiler üçüncü kişiler ile paylaşılmayacaktır.)

Test Sonucunu Görüntüleyin

Diyabet Hastası Değilsiniz

Şekil 32. Entegrasyon Testi Arayüz Sistemi

- Uygulamaya girilen değerler veri tabanına başarılı bir şekilde aktarılmıştır.

HastaID	Ad	Soyad	Eposta
1	Hakam	Chedo	magnitar123@gmail.com
* NULL	NULL	NULL	NULL

TestID	HastaID	GebelikSay...	GlikozSeviy...	KanBasinci	DeriKalinaligi	InsulinSeviy...	VucutKitleE...	DiyabetSoy...	Yas	Tarih	Sonuc
1	1	0	10	90	2	5	27	1	23	2023-01-03	0
** NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Şekil 33. Entegrasyon Testi Veritabanı

- Uygulamaya girilen değerler e-mail olarak kullanıcılara iletilmektedir.



diabetespredictionn@gmail.com

to me ▼

Sayın Hakam Chedo

2023-01-03 tarihinde yaptırmış olduğunuz diyabet testinizin sonucu negatiftir.

Bize gönderdiğiniz tüm bilgiler:

[E-posta:magnitar123@gmail.com](mailto:E-posta:magnitar123@gmail.com)

Gebelik Sayısı:0

Glikoz Seviyesi:10.0

Kan Basıncı:100.0

Deri Kalınlığı:1.0

Insulin Seviyesi:23.0

Vücut Kitle Endeksi:27.0

Diyabet Soyağacı:1.0

Yaş:23

Diyabet hastalığı hakkında bilgi almak için şu <https://www.turkdiab.org/diyabet-hakkinda-hersey.asp?lang=TR&id=46> ulaşabilirsiniz

### Şekil 34. Entegrasyon Testi E-mail

#### D.3.2. Performans Testi

Proje geliştirme aşamasında model local bilgisayarlarda çalıştırıldı ve model az sayıda sonuç isteğini yanıtladı ancak uygulama çok sayıda kullanıcı tarafından kullanıldığında çok sayıda isteğe cevap veren sunucular kullanılmalıdır.

- Modelin kısa sürede (bir saniyeden daha kısa) değerleri sınıflandırıp yanıtladığı tespit edildi.
- Veri tabanı olarak MSSQL kullanıldı ve verilerin aktarılmasında bir sorun yaşanmadı.
- E-mail, Pythondaki SMTP kütüphanesi kullanılarak gönderilmekte ve herhangi bir gecikme yaşanmamaktadır.

### D.3.3. Kabul Testleri

Doğru sonuca ulaşmak için aşağıdaki kurallara uygun veri girişi yapılmalıdır.

- Ad, soyad, Eposta bilgileri boş olmamalıdır.

Şekil 35. Kabul Testi Yanlış Girdi

- Gebelik sayısı, Glikoz miktarı, Kan basıncı, Cilt kalınlığı değeri, İnsülin miktarı, BMI, Diyabet soyağacı fonksiyonu, Yaş gibi test sonuçlarının boş olması, rakamdan başka karakter girilmesi, gereğinden fazla veya az değer girilmemesi engellenmiştir.

Şekil 36. Kabul Testi Yanlış Değer Girişi

## **D.4. Yazılım/ Veritabanı/ Donanım Gerçeklemeleri**

### **D.4.1. Yazılım Gerçeklemeleri**

Projemiz makine öğrenmesi modeli, veri tabanı servisi ve web uygulamasından oluşmaktadır. Spyder IDE kullanılarak modelin eğitimi, doğrulaması ve test işlemleri gerçekleştirilmiştir. Makine öğrenmesi için Destek Vektör Makinesi yöntemi kullanılmıştır. Ayrıca makine öğrenmesi uygulamaları için geliştirilmiş ve açık kaynaklı kütüphanelerden olan scikit-learn kütüphanesi de kullanılmıştır. Bu kütüphaneye ek olarak pandas, numpy ve pickle kütüphanelerinden de faydalanılmıştır. Uygulamamızın sunucu ve istemci tarafları için streamlit kütüphanesinden faydalanılmıştır.

Makine öğrenmesi modeli, internet üzerinde açık kaynak olarak paylaşılan bir veri seti ile eğitildi. Bu veri seti; hastaya ait hamilelik, glikoz miktarı, kan basıncı, yaş, deri kalınlığı, diyabet soyağacı fonksiyonu, insülin miktarı, vücut kitle indeksi ve test sonuç bilgilerini tutan 769 satırlık csv. uzantılı dosyadır.

Kullanılan email.message modülü ile kullanıcının e-posta adresine SMTP protokolü ile yaptırdığı son testin sonucu ve test için girdiği bilgilerin bulunduğu bir e-posta gönderilmektedir. Bu gönderim tanımlanan SendEmail metodu içinde yapılmaktadır. E-posta gönderimi projeye ait diabetespredictionn@gmail.com gmail adresi ile yapılmaktadır.

Veri setinin %80'i model eğitimi, %20'si ise test için kullanılmıştır. Kullandığımız modelin hastalık tespitindeki doğruluk oranı %78'dir.

Geliştirilen uygulama yerel sunucunun 8501 portunda çalışmaktadır. Kullanıcının gönderdiği bilgiler MSSQL üzerindeki veri tabanına kaydedilmektedir. Bu işlem Python'da bulunan pyodbc kütüphanesiyle veri tabanına açılan bağlantı sayesinde gerçekleşmektedir.

### **D.4.2. Veri Tabanı Gerçeklemeleri**

Projemiz bir web uygulamasıdır ve depolanacak verilerin tamamı online bir veri tabanında tutulacaktır. Server Hosting firmasından alınan bir sunucu üzerine MSSQL kurularak veri tabanı oluşturulacaktır. Bu veri tabanı tasarım aşamasında planlanan veri tabanı tasarımına bağlı kalınarak oluşturulacaktır. Veri tabanımızda toplam 2 tablo, 4 saklı



yordam ve 1 tetikleyici bulunmaktadır. Web uygulamasının veri tabanı ile iletişimi bir service ile gerçekleşir. Bu servis Python programlama dili ile veri tabanına erişim sağlamaktadır. Projenin şu anki aşamasında veri tabanı online sunucu yerine yerel sunucudur. Oluşturulan veri tabanı tasarım aşamasında planlandığı şekilde oluşturulmuştur.

#### **D.4.3. Donanım Gerçeklemeleri**

Sistemimizde donanım gerçekleştirilmesi bulunmamaktadır.

### **E. SONUÇ VE ÖNERİLER**

Oluşturulan proje sistem gereksinimlerini sağlar biçimde tamamlanmıştır ve sürdürülebilir, kullanıcı dostu bir uygulama olarak kullanılabilir. Projede şeker hastalığının tespitine dair makine öğrenmesi modeli ile çalışan bir web uygulaması geliştirilmiştir. Test sonucunu kullanıcı arayüzüne göstermek dışında kullanıcının mail adresine test bilgilerini ve test sonucu mail olarak gönderen bir yapı oluşturulmuştur. Bu şekilde kullanıcının test bilgilerini ve test sonucuna istediği zaman ulaşabilme imkanı verilmiştir. Yapılan bu proje ile diyabet hastalığı için farkındalık yaratmak, sağlık sistemine yardımcı olmak amaçlanmış ve başarıyla uygulanmıştır. Veri setini güncelleyerek veya farklı bir makine öğrenmesi modeli kullanarak doğruluk oranı artırılabilir.

## Referanslar

- [1] <https://search.trdizin.gov.tr/yayin/detay/315597/>
- [2] <https://www.medicalpark.com.tr/seker-hastaligi-diyabet-nedir/hg-1703>
- [3] <https://medium.com/@aysincetin/makine-%C3%B6%C4%9Frenmesi-y%C3%B6ntemleriyle-diyabet-hastal%C4%B1%C4%9F%C4%B1-te%C5%9Fhisi-a7b18ba3a0fb>
- [4] <https://www.ijraset.com/research-paper/diabetes-prediction-using-ml>
- [5] <https://realpython.com/python-testing/>
- [6] <https://www.guru99.com/software-testing-introduction-importance.html>
- [7] [https://en.wikipedia.org/wiki/Software\\_testing](https://en.wikipedia.org/wiki/Software_testing)
- [8] <https://www.kaggle.com/datasets/mathchi/diabetes-data-set>