



Bilgisayar Mühendisliği Bölümü

Bitirme Projesi

Diyabet Veri Setiyle Makine Öğrenimi

Veli KÖSEOĞLU

190309015

Faruk BULUT

Haziran 2023

Proje Bilgisi

Dosya adı	Diabetes
Toplam Sayfa Sayısı	27

Sürüm	Yazarlar	Yorum	Tarih
İlk taslak			
Son Taslak			
Versiyon 1			
Versiyon 2			

Yayın Onayı

DOÇ. Dr. Faruk BULUT

Ad Soyad

Tez danışmanı

Ad Soyad

Tez koordinatör

Eşek.

Tez Koordinatör
Yardımcısı

HAZIRLIK VE İNCELEME ONAYI

Veli KÖSEOĞLU

Öğrenci Ad Soyad

İÇİNDEKİLER

1. Giriş	1
1.1. Projenin amacı.....	1
1.2. Projenin Kapsamı, Çıktıları ve Tablolar.....	1
1.3. Projenin Hedefleri ve Başarı Kriterleri	4
1.4. Proje Üyelerinin Görev ve Sorumlulukları.....	4
2. Yapılan Çalışma	4
2.1. Makine Öğrenmesi	5
2.1.1. Denetimli Öğrenme	6
2.1.2. Veri Ön İşleme	6
2.1.3. Test-Train	6
2.1.4. KNN	7
2.1.5. Naive Bayes Classifier	7
2.1.6. Support Vector Machines.....	7
2.1.7. Logistic Regression.....	7
2.1.8. Random Forest	7
2.1.9. Decision Trees.....	7
2.1.10. Decision Tree Regression.....	8
2.1.11. Support Vector Regression.....	8
2.1.12. Linear Regression.....	8
2.2. GUI.....	8
3. Önerilen Sistem	8
3.1. Analiz ve Görselleştirme Araçları	8
3.2. Önerilen Sistem Mimarisi	9
3.3. Fonksiyonel Gereksinimler	9
3.4. Fonksiyonel Olmayan Gereksinimler	10
3.4.1. Kalite Gereksinimleri	10
3.4.2. Performans Gereksinimleri	10
3.5. Gerçek Olmayan Gereksinimler	11
3.6. Sistemdeki Modeller	11
3.6.1. Senaryolar	11
3.6.2. Use Case Modelleri	12
3.6.3. Dinamik modeller	13
3.6.4. Sistemin İçeriği	14
3.7. Kullanıcı Arayüzü	19
4. Sonuç	23
5. Referanslar	23
6. Terimler Listesi	24
6.1. Tanımlar	24
6.2. Kısaltmalar	24

1. Giriş

Bu proje, söz konusu diyabet veri setini toplayıp bu veri seti üzerinden analiz etme ve temellerini açıklamayı içermektedir. Bu projede ana konular Makine Öğrenimi, Derin Öğrenme, Yapay Zekâ ve Veri İşleme alanındadır.

Bu projenin temel amacı, son kullanıcılara diyabet hastalığı hakkında algoritmalarla beraber bilgi sunmak ve bunun için bir araç geliştirmektir.

Bu projenin ilk yarısında, web sitelerinden diyabet hastalığıyla alakalı veri setleri araştırılmıştır. Daha sonra uygun görülen veri setiyle alakalı Pandas kütüphanesi kullanılarak veri işleme aşamalarından geçilmiştir ve çeşitli sonuçlar elde edilmiştir.

Bu projenin ikinci yarısı, toplanan ve saklanan diyabet veri setinin veri işleme aşamalarına odaklanılıyor. Burada toplanan veriler üzerinden anlamlı bilgiler çıkarabilme ve algoritmalar üzerinde durulmuştur. Ölçütler ve algoritmalar K Nearest Neighbours, Naive Bayes, Support Vector Machines, Logistic Regression, Random Forest, Decision Trees, Decision Tree Regression, Support Vector Regression, Linear Regression olarak belirlenmiştir. Bu bölümlerle ilgili gerekli açıklamalarda ilerleyen bölümlerde yapılacaktır.

Ve Proje, görsel grafiklerle elde edilen sonuçların gösterilmesiyle sona erer. Bu grafikler ile son kullanıcılar yoğun görselleştirmeler yoluyla toplanan verilerin diyabet hastalığına etkileyebilen özniteliklerin sonuçlarıyla hangisinin daha çok etkileyebileceği ve hangisinin dikkate çeker bir öznitelik olduğunu gözler önüne sererek anlayabileceklerdir.

1.1. Projenin Amacı

Bu projenin temel amacı, son kullanıcılara diyabet hastalığı hakkında bilgi vererek, algoritmalarla beraber alınan sonuçlar hakkında kısa bilgiler vermeyi amaçlamaktadır.

Bütün hastanelerde hastaların memnuniyeti ve iyi bir sağlık hizmeti alabilmeleri hem önemli hem de gerekli bir faktördür. Hastaların, “Diyabet hastası olabilmek olasılığı var mı?”, “Diyabet hastası olma olasılığı varsa nasıl bir diyet uygulamalı” ve “Diyabet hastası olma olasılığı yüksek olan kişisel özelliği nedir?” gibi özellikler hakkında detaylı bilgi alabilmek hem hastanelerin imajını arttırabilir hem de hastaları rahatlatılabilir.

Günümüzde insanlar toplu olarak düşünce ve fikirlerini internet üzerinden paylaşabilmektedirler. Bu noktada elde edilen bilgilerin hem hastalara hem de doktorlara doğru bir şekilde web üzerinden ulaştırılabilirler. Hastaneler daha fazla kar elde etmek, hastalar ise daha ucuz ve etkili bir sağlık hizmeti almak istemektedir. Bu hastaneler daha fazla kar elde etmek ve imajlarını toplumun önünde daha da yükseltmek istiyorlarsa ayrıca hastaların kendilerini tercih etmelerini istiyorlarsa bu tarz bir projeden elde edilen bilgileri edinmeyi düşünmelidirler.

Literatürde yukarıda belirtilen bu tür bir sorun için çalışmalar vardır fakat hem hastaların hem de doktorların ulaşabileceği tam olarak böyle bir çalışma yoktur.

1.2. Projenin Kapsamı, Çıktıları ve Tablolar

Projenin kapsamı, web sitelerinden diyabet veri setini araştırıp ve toparlayıp bu veri seti üzerinden analiz etme ve temellerini açıklamayı içeren bir yazılım geliştirmek ve kullanıcılar için önemli grafikleri görüntülemek üzere verilerden çıkarılan anlamlı çıkarımlar barındıran bir içerik oluşturmaktır.

Bu proje, veri setinden elde edilen analizleri barındıran ve son kullanıcılar için bir yazılım programı ile yan bileşen için bir rapor üretecektir. Yazılım programı, hasta ve doktor odaklı olup dünyanın her yerinden insanların veri analizlerine ve bilgilendirici görsellerine ulaşabileceği bir platform oluşturmayı hedeflemektedir.

Proje, 01.03.2023 tarihinde proje başlıklarının DOÇ. Dr. Faruk BULUT hocamıza duyurulması ile başlamıştır. Bu proje raporunun önceki versiyonunda araştırmaya odaklanılmıştı. Projenin ikinci bölümünde ise projenin uygulanmasına odaklanıldı.

TASK NAME	DURATION	PLANNED START...	PLANNED FINISH ..
Planlama	14 days	23.02.2023	8.03.2023
Proje Konusu Araştır...	6 days	23.02.2023	28.02.2023
Proje Konu Taslaqlar...	1 day	1.03.2023	1.03.2023
Proje Konusunun Se...	1 day	1.03.2023	1.03.2023
Veri Seti Araştırma v...	6 days	2.03.2023	7.03.2023
Veri Setleri Taslaqla...	1 day	8.03.2023	8.03.2023
Veri Setinin Seçilmesi	1 day	8.03.2023	8.03.2023
Analiz	7 days	9.03.2023	15.03.2023
Projenin Amacını ve ...	2 days	9.03.2023	10.03.2023
Veri Seti Hakkında A...	1 day	11.03.2023	11.03.2023
Hangi Makine Öğre...	3 days	12.03.2023	14.03.2023
Tüm Hafta Yapılanl...	1 day	15.03.2023	15.03.2023
Tasarım	3 days	16.03.2023	18.03.2023
Kullanılacak Algorit...	1 day	16.03.2023	16.03.2023
Projenin İçeriğine D...	2 days	17.03.2023	18.03.2023
Üretim	41 days	19.03.2023	28.04.2023
Veri Seti Öğrenimi	3 days	19.03.2023	21.03.2023
Veri Setinin Görselle...	4 days	22.03.2023	25.03.2023
Test-Train Split	3 days	26.03.2023	28.03.2023
K Nearest Neighbou...	3 days	29.03.2023	31.03.2023
Naive Bayes Classifi...	4 days	1.04.2023	4.04.2023
Support Vector Mac...	4 days	5.04.2023	8.04.2023
Logistic Regression	3 days	9.04.2023	11.04.2023
Logistic Regression	3 days	9.04.2023	11.04.2023
Random Forest	4 days	12.04.2023	15.04.2023
Decision Trees	3 days	16.04.2023	18.04.2023
Decision Tree Regre...	3 days	19.04.2023	21.04.2023
Support Vector Reg...	4 days	22.04.2023	25.04.2023
Linear Regression	3 days	26.04.2023	28.04.2023
Test	40 days	20.03.2023	28.04.2023
Veri Seti Öğrenimi T...	2 days	20.03.2023	21.03.2023
Veri Seti Görselleştir...	2 days	24.03.2023	25.03.2023
Test-Train Split Test	2 days	27.03.2023	28.03.2023
K Nearest Neighbou...	2 days	30.03.2023	31.03.2023
Naive Bayes Classifi...	2 days	3.04.2023	4.04.2023
Support Vector Mac...	2 days	7.04.2023	8.04.2023
Logistic Regression ...	2 days	10.04.2023	11.04.2023
Random Forest Test	2 days	10.04.2023	11.04.2023
Decision Trees Test	2 days	17.04.2023	18.04.2023
Decision Tree Regre...	2 days	20.04.2023	21.04.2023
Support Vector Reg...	2 days	24.04.2023	25.04.2023
Linear Regression T...	2 days	27.04.2023	28.04.2023
Bakım	4 days	29.04.2023	2.05.2023
Genel Bakım	2 days	29.04.2023	30.04.2023
Makine Öğrenimi AL...	2 days	1.05.2023	2.05.2023

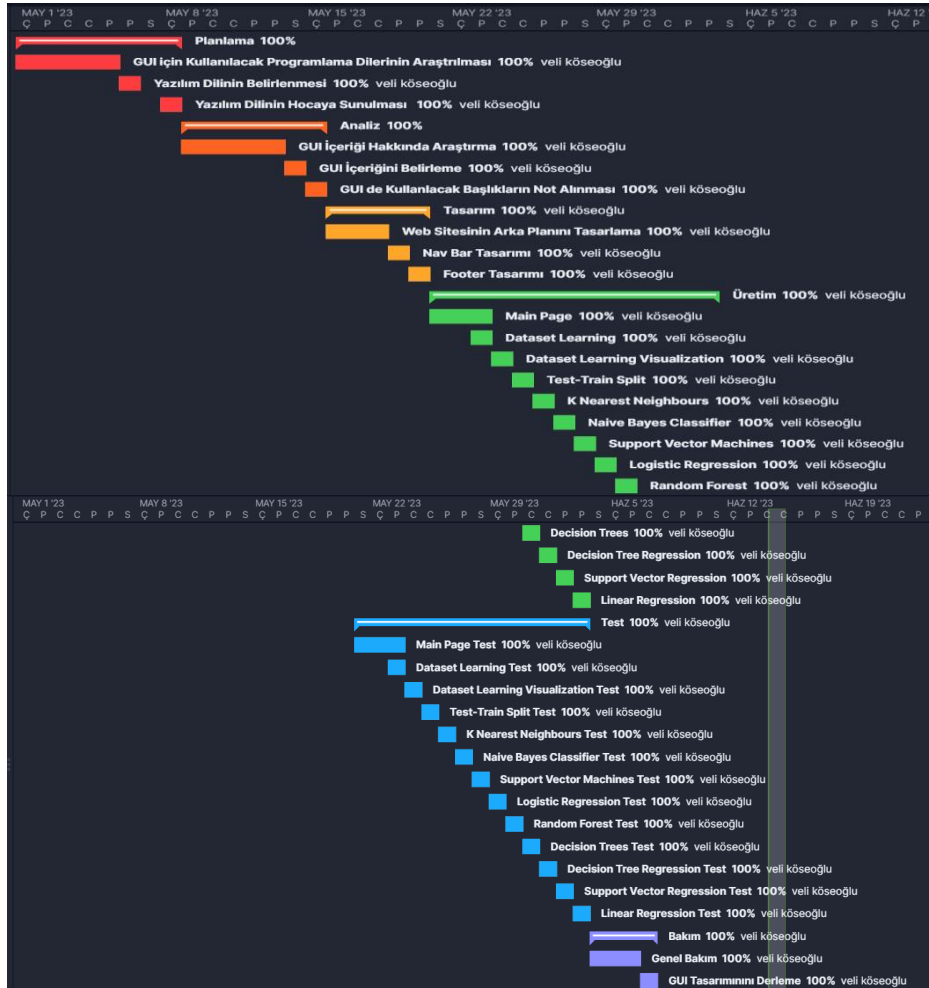
Şekil 1.2. 1- Makine Öğrenmesi Zaman Çizelgesi.



Şekil 1.2. 2 – Makine Öğrenmesi Zaman Çizelgesi.

TASK NAME	DURATION	PLANNED START...	PLANNED FINISH ...	TASK NAME	DURATION	PLANNED START...	PLANNED FINISH ...
Planlama	8 days	3.05.2023	10.05.2023	Decision Tree Regre...	1 day	3.06.2023	3.06.2023
GUI için Kullanılacak...	5 days	3.05.2023	7.05.2023	Support Vector Reg...	1 day	4.06.2023	4.06.2023
Yazılım Dilinin Belirl...	1 day	8.05.2023	8.05.2023	Linear Regression	1 day	5.06.2023	5.06.2023
Yazılım Dilinin Hoca...	1 day	10.05.2023	10.05.2023	Test	14 days	23.05.2023	5.06.2023
Analiz	7 days	11.05.2023	17.05.2023	Main Page Test	3 days	23.05.2023	25.05.2023
GUI İçeriği Hakkında...	5 days	11.05.2023	15.05.2023	Dataset Learning Test	1 day	25.05.2023	25.05.2023
GUI İçeriğini Belirleme	1 day	16.05.2023	16.05.2023	Dataset Learning Vi...	1 day	26.05.2023	26.05.2023
GUI de Kullanılacak ...	1 day	17.05.2023	17.05.2023	Test-Train Split Test	1 day	27.05.2023	27.05.2023
Tasarım	5 days	18.05.2023	22.05.2023	K Nearest Neighbou...	1 day	28.05.2023	28.05.2023
Web Sitesinin Arka ...	3 days	18.05.2023	20.05.2023	Naive Bayes Classifi...	1 day	29.05.2023	29.05.2023
Nav Bar Tasarımı	1 day	21.05.2023	21.05.2023	Support Vector Mac...	1 day	30.05.2023	30.05.2023
Footer Tasarımı	1 day	22.05.2023	22.05.2023	Logistic Regression ...	1 day	31.05.2023	31.05.2023
Üretim	14 days	23.05.2023	5.06.2023	Random Forest Test	1 day	1.06.2023	1.06.2023
Main Page	3 days	23.05.2023	25.05.2023	Decision Trees Test	1 day	2.06.2023	2.06.2023
Dataset Learning	1 day	25.05.2023	25.05.2023	Decision Tree Regre...	1 day	3.06.2023	3.06.2023
Dataset Learning Vi...	1 day	26.05.2023	26.05.2023	Support Vector Reg...	1 day	4.06.2023	4.06.2023
Test-Train Split	1 day	27.05.2023	27.05.2023	Linear Regression T...	1 day	5.06.2023	5.06.2023
K Nearest Neighbou...	1 day	28.05.2023	28.05.2023	Bakım	4 days	6.06.2023	9.06.2023
Naive Bayes Classifi...	1 day	29.05.2023	29.05.2023	Genel Bakım	3 days	6.06.2023	8.06.2023
Support Vector Mac...	1 day	30.05.2023	30.05.2023	GUI Tasarımını Der...	1 day	9.06.2023	9.06.2023
Logistic Regression	1 day	31.05.2023	31.05.2023				
Random Forest	1 day	1.06.2023	1.06.2023				
Decision Trees	1 day	2.06.2023	2.06.2023				

Şekil 1.2. 3- GUI Zaman Çizelgesi.



Şekil 1.2. 4 – GUI Zaman Çizelgesi.

Projenin geliştirilmesi için zaman çizelgeleri yukarıda belirtilmiştir. (Şekil 1.2. 1), (Şekil 1.2. 2), (Şekil 1.2 3), (Şekil 1.2. 4). İlk ikisi makine öğrenme aşamalarının geliştirilmesini gösterir. Diğer iki ise GUI tasarımının geliştirilmesini gösteriyor.

Bu projede Windows işletim sistemi kullanılmaktadır . Ana programlama dili, makine öğrenmesi için oluşturulmuş kullanışlı kütüphaneleri nedeniyle Python olarak belirlenmiştir. Bir web uygulama geliştirme aracı olarak Flask kütüphanesinde Flask çerçevesi kullanıldı. Analiz ve görselleştirmeler Matplotlib ve Seaborn kütüphanesi ile gerçekleştirilmiştir.

1.3. Projenin Hedefleri ve Başarı Kriterleri

Projenin hedefleri ve başarı kriterleri şunlardır:

Aşağıdakileri yapabilen bir makine öğrenmesi programı:

- Programın tüm algoritmalarıyla beraber error vermeden çalışabilmesi.
- Tüm Görsellerin eksiksiz bir şekilde alınabilmesi.

Aşağıdakileri yapabilen bir GUI programı:

- GUI programının hatasız bir şekilde çalışabilmesi.
- Web sayfasında sayfalar arasında geçişin sorunsuz bir şekilde gerçekleşebilmesi.
- Kullanıcının diyabet hastalığıyla alakalı verilere kolayca ulaşabilmesi.

1.4. Proje Üyelerinin Görev ve Sorumlulukları

Tablo 1.4.1 Proje üyeleri, sorumlulukları ve görevleri

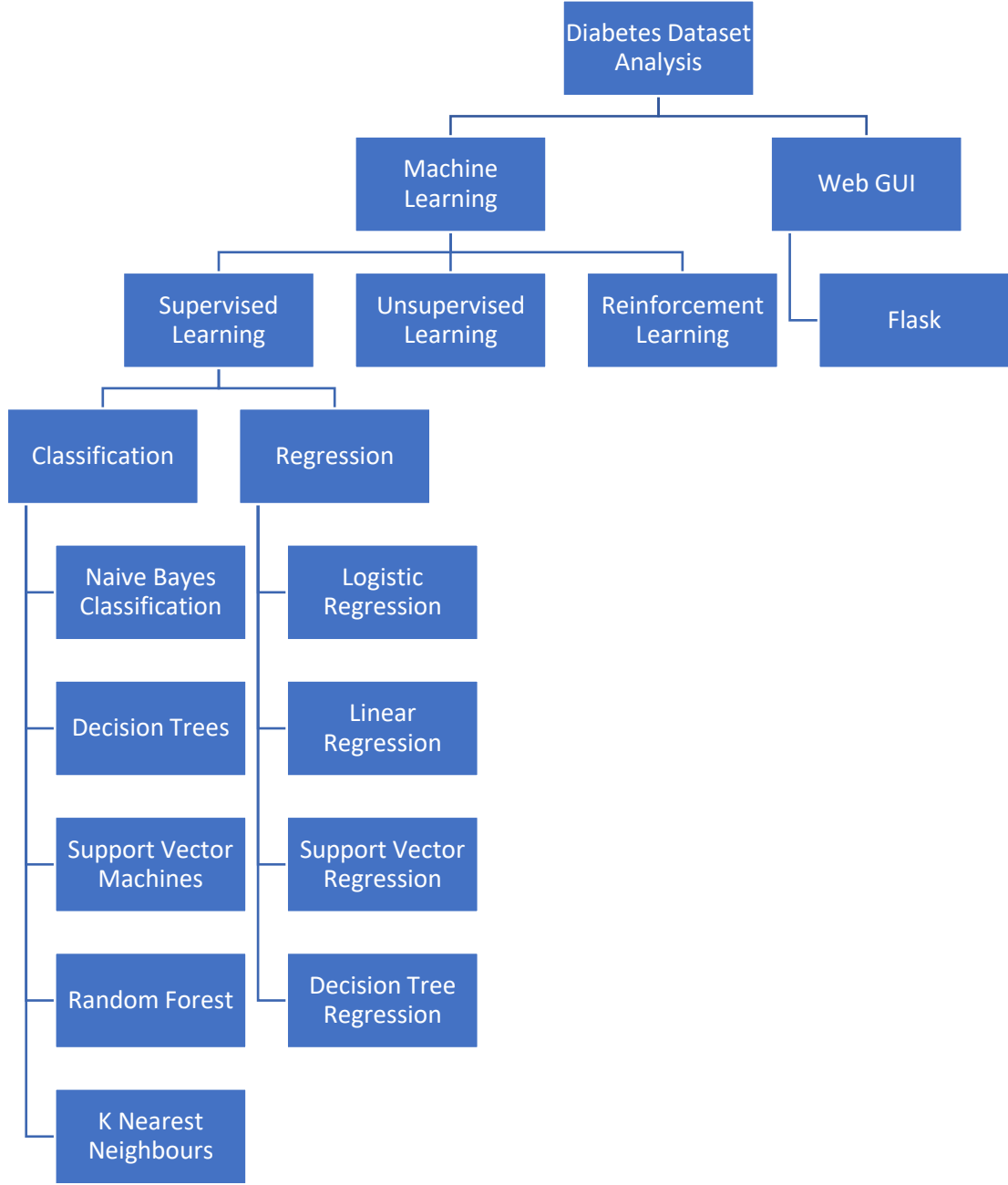
Proje Üyesi	Sorumluluklar ve Görevler
Veli KÖSEOĞLU	Araştırma, diyabet hastalığı analizi için gerekli verileri toplama, toplanan verileri analiz etme, sonuçları görselleştirme ve web sayfası oluşturma.

2. Yapılan Çalışma

Kronik bir hastalık olan diyabet hastalığı, dünya çapında bir salgın olarak ortaya çıkmıştır. Yalnızca gelişmiş tıbbi ekipman olmadan elde edilen tıbbi bilgileri içeren bir dizi veriyi kullanarak diyabet hastalığı için teşhis yardımı sağlamak, hastalığı veya hastalık riskini erken bir aşamada keşfetmek isteyen çok sayıda kişiye yardımcı olabilir. Bu muhtemelen birçok insanın hayatı üzerinde büyük bir olumlu etki yaratabilir.[1] Pima Kızıldereleri Diyabet veri seti üzerindeki deneysel sonuçlar, önerilen yöntemin, önceki çalışmalarda geliştirilen yöntemlerle ilişkili olarak tahmin doğruluğunu önemli ölçüde geliştirdiğini göstermektedir. Hibrit akıllı sistem, tıp pratisyenlerine sağlık uygulamalarında bir karar destek sistemi olarak yardımcı olabilir.[2]

Büyük ölçekli verilerden faydalı bilgilerin keşfedilmesine Veri Madenciliği denir. Veri madenciliği ile veriler arasındaki ilişkileri ortaya çıkarmak ve geleceğe yönelik doğru tahminlerde bulunmak mümkündür.[3] Veri madenciliğinin temel amacı, kurumlardaki karar destek sistemleri için değerli olabilecek verilerin belirli yöntem ve süreçlerden sonra ortaya çıkarılmasıdır. Son on yılda veri madenciliği hemen hemen her alanda çeşitli şekillerde kullanılmaya başlamış ve günümüzde veri madenciliği hemen hemen her türlü disiplin için yararlı olan önemli bir disiplindir.[4]

Basitçe Diyabet hastalığı analizi aşağıdaki grafik ile ifade edilebilir. (Şekil 2.1)



Şekil 2.1- Diyabet Hastalığı Analizi.

2.1. Makine Öğrenmesi

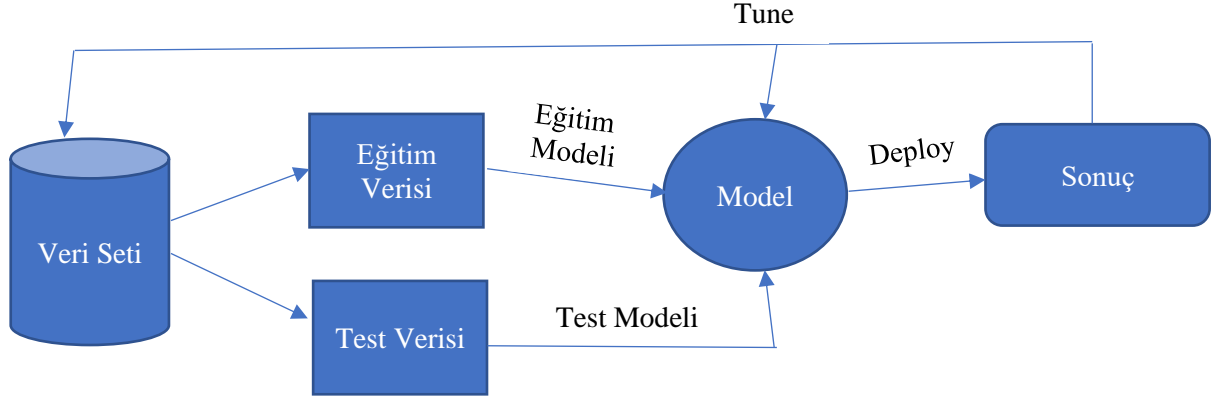
Makine öğreniminde(ML) kullanılacak bir dizi farklı algoritma vardır. Gerekli çıktı, hangisinin kullanılacağına karar veren şeydir. ML algoritmaları karakteristik olarak üç öğrenme türünden birine girer: denetimli, denetimsiz ve yarı denetimli öğrenme. Makine öğrenimi, kullanıcıyla gelişmiş bir deneyim sağlamak için yazılımda yaygın olarak kullanılır.[5]

Makine öğrenimi (ML), makineler verileri nasıl daha verimli kullanacaklarını öğretmek için kullanılır. Bazen verileri görüntüledikten sonra, verilerden çıkarılan bilgileri yorumlayamayız. Bu durumda, makine öğrenimini uygularız. Mevcut veri kümelerinin bolluğu ile makine öğrenimine olan talep artmaktadır.[6]

2.1.1 Denetimli Öğrenme

Denetimli öğrenme, örnek girdi-çıkı çiftlerine dayalı olarak bir girdiyi bir çıktıya eşleyen bir işlevi öğrenmeye yönelik makine öğrenimi görevidir. Eğitim veri kümesi, tahmin edilmesi veya sınıflandırılması gereken çıktı değişkenine sahiptir. Tüm algoritmalar eğitim veri setinden bir tür örüntü öğrenir ve bunları tahmin veya sınıflandırma için test veri setine uygular.[6]

Burada önemli olan denetimli öğrenme algoritmalarından elde edilen doğruluk skorlarıyla karşılaştırma yapılmış ve en iyi skoru veren algoritma keşfedilmiştir.



Şekil 2.1.1 – Denetimli öğrenme çalışma modeli.

2.1.2 Veri Ön İşleme

Bu aşama veri işleme adımlarının olmazsa olmaz en önemli adımlarından biridir. Veri temizleme, dönüştürme ve modele uygun hale getirme işlemleri tüm elde edilmesi beklenen sonuçların büyük bir bölümünü oluşturmaktadır. Önem açısından baktığımızda projeden projeye değişen tek şey modellemeye kullanılan verilerdir. Bu yüzden modellemeye istinaden verinin önemi daha fazladır.[7]

Diyabet veri setiyle ise ilk başta null değerler var mı diye kontrol edildi. Null değer olmadığı için bu aşamadan geçilerek yinelenen verilerin varlığı kontrol edildi. Daha sonra Glukoz, Kan Basıncı, Deri Kalınlığı, İnsülin ve BMI verinde 0 değerler olamayacağından bunlar kontrol edildi fakat hamilelik sayısı kontrol edilmedi çünkü hamilelik sayısının 0 olma durumu mevcuttur. Daha sonra 0 olan değerler ise sütunun ortalama değerleriyle değiştirildi.

2.1.3 Test-Train

Makine öğreniminin başarısını etkileyen en kritik faktör eğitim ve test sürecidir. Araştırmacılar, veri kümelerini train(eğitim) ve test için iki bölüme ayırır. Eğitim ve test miktarı, başarı oranındaki en kritik faktördür. Özellikler ile etiket arasında yüksek bir korelasyon varsa Eğitim-Test seti eşit oranda bölünür. Literatürde kullanılan eğitim-test oranı veri yapısına göre değişmektedir.

Burada yola çıkarak diyabet veri setinde ise bu oran %80'i test ve %20'si train olmak üzere bölünmüştür. X_train 614 satır ve 8 sütundan, X_test 154 satır ve 8 sütun olarak bölümlere ayrılmış ve bunlar üzerinden modeller oluşturulabilecektir.

2.1.4 KNN

KNN algoritması, çoğunluk sınıf etiketinin, özellik uzayındaki en yakın 'K' komşuları arasında yeni bir veri noktasının sınıf etiketini belirlediği bir oylama sistemi olarak düşünülebilmektedir.[10]

Burada bir KNN modeli eğiterek buradan test ve eğitim skorları elde edilmiş ve sırasıyla 0,72 ve 1 olarak elde edilmiştir. Ardından K'nin 1 olduğu modelde doğruluk puanı 0,72 elde edilmiştir. Daha sonra K değerinin en iyi değeri verdiği noktayı bulmak için Çaprazlama doğrulama yöntemiyle 24 olarak elde edilmiştir. Sonrasında başta yapılanlar tekrar K'nin 24 oldu değer için uygulanmıştır. Buradan test ve eğitim skorları elde edilmiş ve sırasıyla 0,81 ve 0,72 olarak elde edilmiş ve doğruluk puanı 0,81 olarak elde edilmiştir.

2.1.5 Naive Bayes Classifier

Naive Bayes, Bayes Teoremine dayalı istatistiksel bir sınıflandırma tekniğidir. En basit denetimli öğrenme algoritmalarından biri olarak bilinir. Naive Bayes sınıflandırıcısı hızlı, doğru ve güvenilir bir algoritmadır. Naive Bayes sınıflandırıcıları, büyük veri kümelerinde yüksek doğruluk ve hıza sahiptir.

Buradan bir Naive Bayes modeli eğiterek doğruluk puanı 0,77 olarak elde edilmiştir.

2.1.6 Support Vector Machines

SVM, lojistik regresyon ve karar ağaçları gibi diğer sınıflandırıcılara kıyasla çok yüksek doğruluk sunar. Doğrusal olmayan giriş alanlarını işlemek için çekirdek numarasıyla bilinir.

Buradan bir SVM modeli eğiterek doğruluk puanı 0,79 olarak elde edilmiştir.

2.1.7 Logistic Regression

Lojistik regresyon, ikili sınıfları tahmin etmek için istatistiksel bir yöntemdir. Sonuç veya hedef değişken doğası gereği ikiye bölünmüştür. İkili, yalnızca iki olası sınıf olduğu anlamına gelir.

Buradan bir Logistic Regression modeli eğiterek doğruluk puanı 0,81 olarak elde edilmiştir.

2.1.8 Random Forest

Rastgele ormanlar, popüler bir denetimli makine öğrenimi algoritmasıdır. Etiketli bir hedef değişkenin olduğu denetimli makine öğrenimi içindir. Regresyon ve sınıflandırma problemlerini çözmek için kullanılabilir. Bir topluluk yöntemidir, yani diğer modellerden tahminleri birleştirir. Topluluktaki daha küçük modellerin her biri bir karar ağacıdır.

Buradan bir Random Forest modeli eğiterek doğruluk puanı 0,81 olarak elde edilmiştir.

2.1.9 Decision Trees

Bir karar ağacı, bir iç düğümün bir özelliği (veya özelliği) temsil ettiği, dalın bir karar kuralını temsil ettiği ve her yaprak düğümün sonucu temsil ettiği akış şeması benzeri bir ağaç yapısıdır.

Bir karar ağacındaki en üstteki düğüm, kök düğüm olarak bilinir. Öznitelik değerine göre bölümlmeyi öğrenir. Ağacı, özyinelemeli bölümleme adı verilen özyinelemeli bir şekilde bölümlere ayırır. Bu akış şeması benzeri yapı, karar vermede size yardımcı olur. İnsan düzeyinde düşünmeyi kolayca taklit eden bir akış şeması diyagramı gibi görselleştirme. Bu nedenle karar ağaçlarının anlaşılması ve yorumlanması kolaydır.

Buradan bir Decision Trees modeli eğiterek doğruluk puanı 0,75 olarak elde edilmiştir.

2.1.10 Decision Tree Regression

Karar ağacı regresyonu, bir nesnenin özelliklerini gözlemler ve anlamlı sürekli çıktılar üretmek için gelecekteki verileri tahmin etmek üzere bir ağacın yapısındaki bir modeli eğitir. Sürekli çıktı, çıktının/sonucun ayrı olmadığı, yani yalnızca ayrı, bilinen bir sayı veya değer dizisiyle temsil edilmediği anlamına gelir.

Burada ise eğitilen modeli puanlayabilmek için rastgele seçilen sayılar üzerinden tahmin oluşturuldu ve sayı 1600 seçildiğinde DTR tahmini 76 , sayı 160 seçildiğinde ise DTR tahmini 54 olarak sonuçlandı.

2.1.11 Support Vector Regression

Destek Vektörü Regresyonu (SVR), regresyon analizi için kullanılan bir tür makine öğrenimi algoritmasıdır. SVR'nin amacı, tahmin hatasını en aza indirirken girdi değişkenleri ile sürekli bir hedef değişken arasındaki ilişkiye yaklaşan bir işlev bulmaktır.

Buradan bir SVR modeli eğiterek 71.93 olarak tahmin sonucu elde edilmiştir.

2.1.12 Linear Regression

Lineer regresyon analizi, bir değişkenin değerini başka bir değişkenin değerine göre tahmin etmek için kullanılır. Tahmin etmek istediğiniz değişkene bağımlı değişken denir. Diğer değişkenin değerini tahmin etmek için kullandığınız değişkene ise bağımsız değişken denir.

Burada ise lineer regresyon için eğitim modeli oluşturularak 0,34 olarak test skoru elde edilmiştir.

2.2. GUI

Flask hakkında yapılan araştırmalarda Flask ile ilgili karşılaştırmalarda ilk olarak Flask ile sosyal ağ uygulaması geliştirilerek başlamış ve Django kullanılarak e-ticaret uygulaması ile sonlandırılmıştır. Arka uç için Python programlama dili, SQLite veritabanı ve ön uç teknolojisi için HTML, JavaScript ve Ajax kullanılmıştır. Proje sonunda her iki uygulama da Heroku adlı bulut platformuna Konuşlandırılmıştır. Karşılaştırma sonucunda Flask'ın en önemli avantajlarının basitlik, esneklik, ince ayarlı kontrol, hızlı ve kolay öğrenme sağlaması olduğu görülmüştür.[8]

Bu yönde etkili olan bu çalışma ile projede GUI tasarımı python ile çalışılabilen öğrenimi kolay ve bir o kadar güçlü minimal bir framework olan Flask tercih edilmiştir.

3. Önerilen Sistem

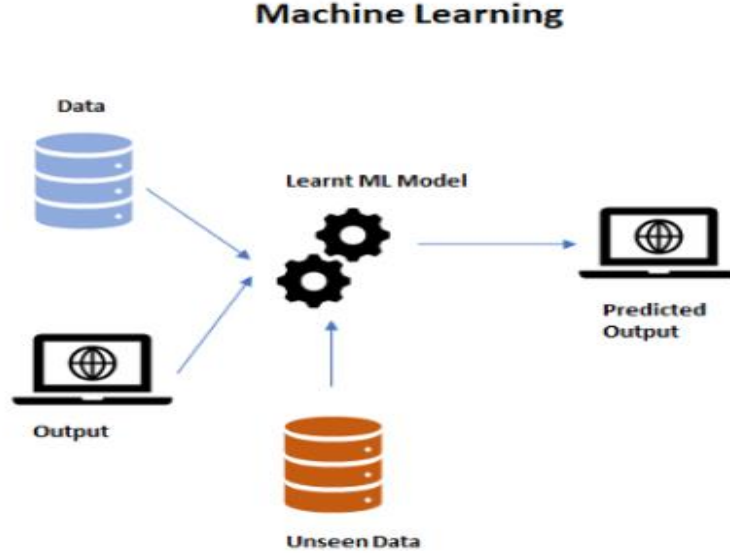
3.1. Analiz ve Görselleştirme Araçları

Web uygulaması geliştirilirken Flask çerçevesi kullanılır. Flask, Python'da yazılmış bir mikro WSGI web çerçevesidir. Flask çerçevesini kullanarak, kullanıcılar karmaşık programlar da oluşturabilirler [9].

Analiz sonuçları Python'da makine öğrenmesi yaparak elde edilmiştir ve elde edilen sonuçlar Flask çerçevesinde içerik olarak oluşturularak kullanıcının kullanımına sunulmuştur.

3.2. Önerilen Sistem Mimarisi

Veri analizi ve web platformları için sistemin 4 ana süreci vardır. Bu süreçler, Veri Ön İşleme, Veri Analizi, Verileri Görselleştirme ve Web'dir. Veri seti yüklendikten sonra veri seti hakkında ön işleme ve veri temizleme işlemleri yapılır daha sonra test ve eğitim veri setleri olarak ayrıldıktan sonra denetimli öğrenme algoritmaları kullanılır. En son olarak buradan elde edilen sonuçları içeren bir Flask çerçevesine sahip web sitesi geliştirildi.



Şekil 3.2 – Makine öğrenimi mimarisi

Veri ön işleme yapmadan önce veri seti CSV dosyası olarak Python dosyası içerisine yüklenmiş olup daha sonrasında veri seti üzerinden bilgileri elde etmek amacıyla bazı işlemlerden geçirilmiştir ve böylece veri seti tanınmış olup ön temizleme gibi işlemlerin yapılabilmesine olanak sağlamış olmuştur.

3.3. Fonksiyonel Gereksinimler

Proje, zamana kadar aşağıdaki gereksinimleri karşılayabilmelidir.

- Proje, web sitesinde kullanıcıya bilgi verir.
 - Web sitesinde sayfalar arasında gezinme ve aranan bilgilere ulaşabilme
- Daha sonra web sitelerinde yayınlanacak analiz sonuçlarının elde edilmesini sağlamak.
 - K Nearest Neighbours
 - Naive Bayes Sınıflandırıcı
 - Support Vector Machines
 - Logistic Regression
 - Random Forest
 - Decision Trees
 - Decision Tree Regression
 - Support Vector Regression
 - Linear Regression
- Nihayet elde edilen analiz sonuçları Flask çerçevesinde yerleştirilerek tasarımı oluşturulur.

3.4. Fonksiyonel Olmayan Gereksinimler

3.4.1. Kalite Gereksinimleri

3.4.1.1 Özel Kalite Gereksinimleri

a) Güvenilirlik

Web uygulamasının sayfalar arasında kesintisiz ve eksiksiz bir şekilde geçişi sağlanması önemlidir. Program bu bağlamda birçok kez test edilmiş ve sistemin güvenilir sonuçlar verdiği kanıtlanmıştır.

b) Kullanılabilirlik

Bu programlar çok kullanıcı dostu olması amaçlanmış fakat şu seviyede maalesef çok kullanıcı dostu değildir. Bu yazılımı kullanmak için kullanıcının bir şeyler bilmesi gerekir. Örneğin, KNN algoritmasının sayfasına giriş yapan bir kullanıcının grafikleri yorumlayabilmesi için makine öğrenmesi alt yapısının sahip olması gerekmektedir.

c) Güvenlik

Web üzerinde yapılan tüm çalışmalarda olabileceği gibi burada da güvenlik açıkları bulunabilir. Fark edilen tüm güvenlik açıkları kaldırıldı.

d) Sürdürülebilirlik

Web sitesinin daha kullanışlı ve daha iyi anlaşılabilir olması için Nav Bar da her algoritma için bir sayfa oluşturulmuş ve kullanıcının istediği bilgilere daha anlaşılabilir bir şekilde ulaşabilmesi amaçlanmıştır.

3.4.2. Performans Gereksinimleri

a) Zamanlama

Yapılan çalışmada Python dosyası açılmadığı sürece web siteye girmek veya sayfayı yenilemek mümkün değildir. Python dosyası açılmadığında zamanlamayı gecikmeli olarak etkilemektedir.

b) Hız

Programın hızı kullanıcının donanımına bağlıdır. Özellikle makine öğreniminde elde edilecek analiz sonuçları için kullanıcının donanımı oldukça önemlidir. Kullanıcının donanımı ne kadar iyiye, program o kadar hızlı çalışacaktır.

c) Hacim

Program dosyası 397 MB alan kaplamaktadır. 400 MB üstü bir alan programın çalışması için yeterlidir.

d) Verim

Program, kullanıcıya elde edilen tüm grafikleri ve doğruluk skorlarını anlaşılabilir ve verimli bir şekilde sunacaktır.

3.5. Gerçek Olmayan Gereksinimler

3.5.1 Fiziksel

Sistemin web sürümünün çalışabilmesi için ilgili dosyanın açılması ve internete ihtiyaç duyulmaktadır. Masaüstünde Windows işletim sistemini çalıştırabilecek bir platforma ihtiyaç duyar. Tüm gereksinimler karşılandıktan sonra herhangi bir web tarayıcısında çalıştırılması yeterli olacaktır.

3.5.2 Yasal

Kazıma yöntemiyle elde edilen verilerin analiz edilmesi ve satılması hukuki açıdan sorun teşkil edebilmektedir. Ayrıca sitelerden veri çekerken robots.txt dosyasına saygı gösterilmelidir. Örneğin, kazımayı yasaklayan web sitelerinden herhangi bir veri alınmamalıdır. Robots.txt dosyasında bundan bahsetmişlerdi.

3.5.3 Kültürel

İnsanların cinsiyeti, cinsel yönelimi, ırkı ve dinine göre çıkarımlarda bulunmak etik olmayacaktır. Kültürel etik için bu projede yorumcuların cinsiyet, din ve ırk ayrımı yapılmayacaktır.

3.5.4 Çevre

Masaüstü ve dizüstü bilgisayarlarda, programın çalışması için Windows işletim sistemi gerekir.

3.5.5 Tasarım ve Uygulama

Sistem, Python, HTML, CSS ve JavaScript programlama dilleri kullanılarak geliştirilmiştir. Ayrıca sistem bu dillere ait kütüphaneler kullanılarak geliştirilmiştir.

3.5.6 Arayüz

Bu projede Python, HTML, CSS ve JavaScript programlama dilleri kullanılmıştır. Bu nedenle, web platformları, python ile uyumlu bir Grafik Kullanıcı Arayüzüne ihtiyaç duyar. Web uygulama platformunda Flask çerçevesi kullanılır. Her türlü web geliştirme kütüphanesi flask ile kullanılabilir.

3.6. Sistemdeki Modeller

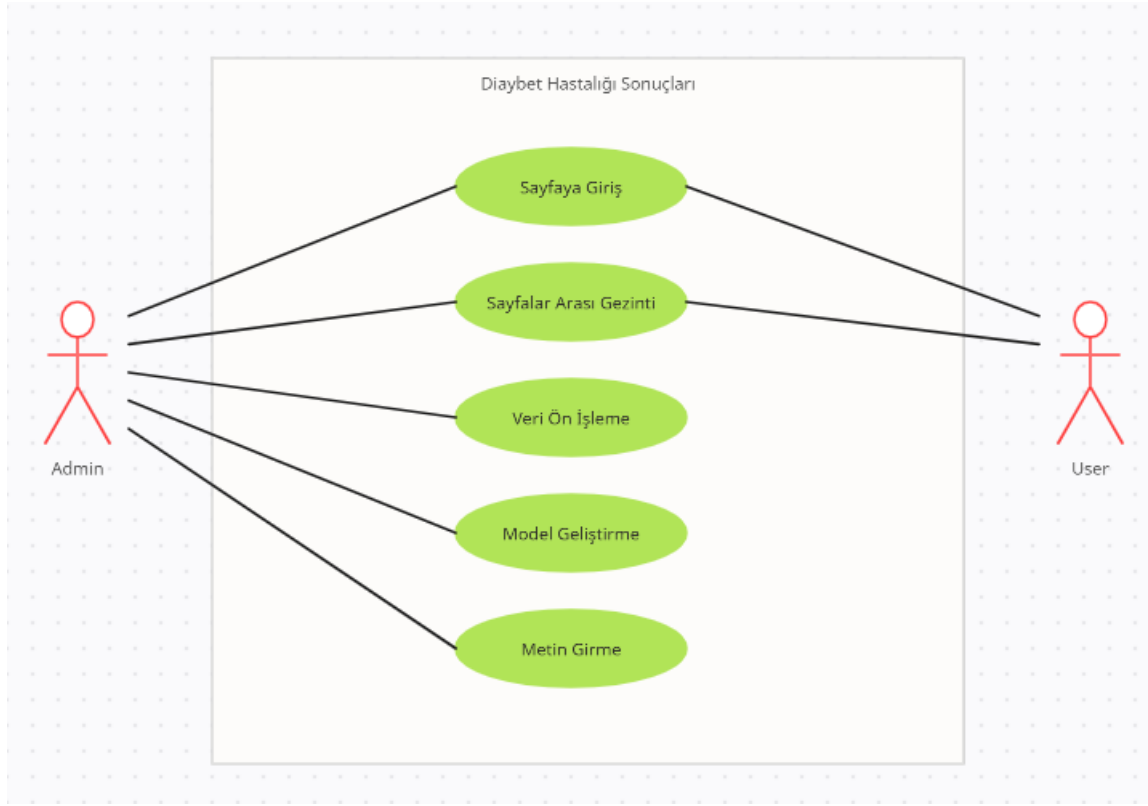
• Senaryolar

Senaryo: Web uygulaması ve masaüstü Platformları için diyabet hastalığı hakkında çıkarımda bulunmak istemesi.

1. Kullanıcı web sitesine girer.
2. Kullanıcı ana sayfadan istediği sonuçlara tıklayarak ulaşabilir.
3. Kullanıcı istediği algoritma sonuçlarına ulaşarak analizlerden çıkarımlar elde edebilir.
4. Kullanıcı analiz edilmiş görselleri veya metinleri dilediği gibi kullanabilir.

•Kullanım Örneği modeli

Programın ana kullanım durumları aşağıda gösterilmiştir



Şekil 3.6.1- Kullanıcının Kullanım Durumu Modeli Sonuç

•Dinamik modeller

•Projenin faaliyet şeması

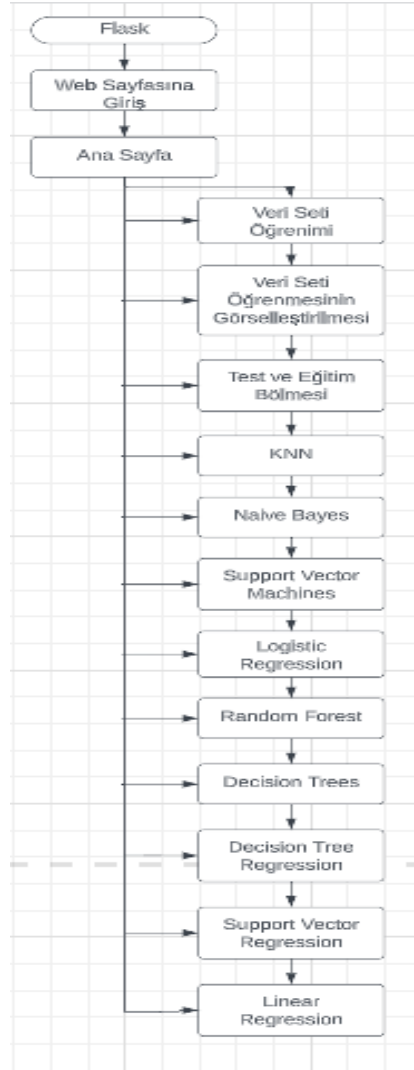
Programın aktivite diyagramı şekil 3.6.2’de olduğu gibidir. Söz konusu diyagram projenin makine öğrenme uygulama platformu için faaliyet diyagramını temsil eder.



Şekil 3.6.2- Projenin Faaliyet Şeması

•Web faaliyet şeması

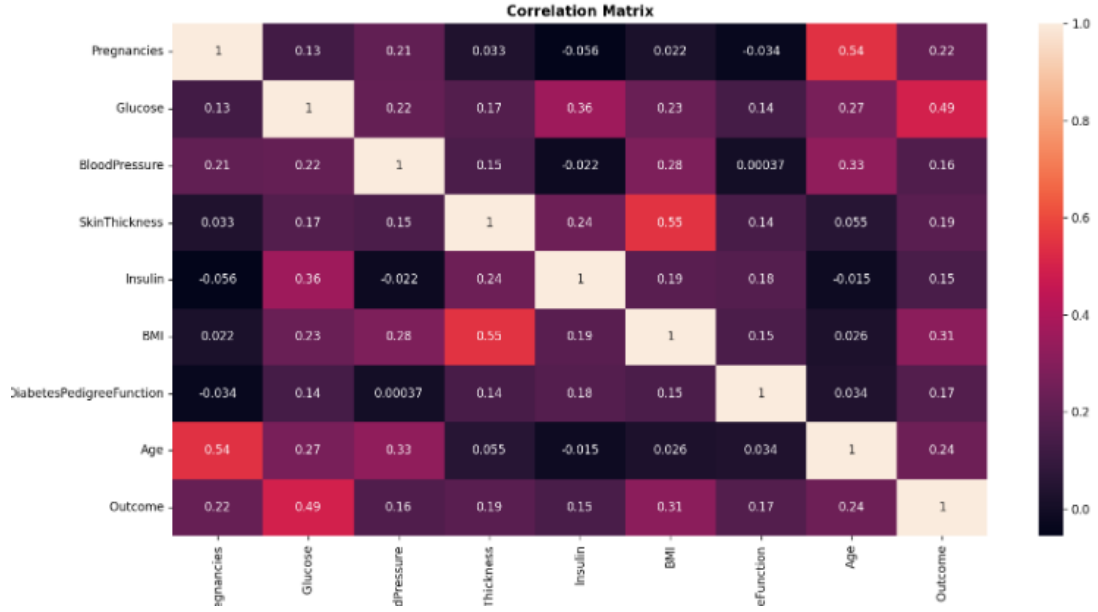
Programın web sitesinde faaliyet şeması aşağıdaki şekil 3.6.3'te gösterilmektedir. Faaliyet şemasında ana sayfaya giriş yapıldığında sayfalar arası faaliyetleri göstermektedir.



Şekil 3.6.3- Proje Web Faaliyet Şeması

3.6.4 Sistemin İçeriği

Sistem de kullanılan veri setinin analizini gerçekleştirdikten sonra denetimli öğrenme algoritmaları kullanılmış ve sonuçlar elde edilmiştir aşağıda da genel bir korelasyon matrisi ile birlikte algoritmalarından elde edilen Karşıtlık matrisleri yer almaktadır



Şekil 3.6.4.1- Korelasyon Matrisi

a. KNN Algoritması:

KNN modeli eğiterek buradan test ve eğitim skorları elde edilmiş ve sırasıyla 0,72 ve 1 olarak elde edilmiştir. Ardından K'nin 1 olduğu modelde doğruluk puanı 0,72 elde edilmiştir. Daha sonra K değerinin en iyi değeri verdiği noktayı bulmak için Çaprazlama doğrulama yöntemiyle 24 olarak elde edilmiştir. Sonrasında başta yapılanlar tekrar K'nin 24 oldu değer için uygulanmıştır. Buradan test ve eğitim skorları elde edilmiş ve sırasıyla 0,81 ve 0,72 olarak elde edilmiş ve doğruluk puanı 0,81 olarak elde edilmiştir.

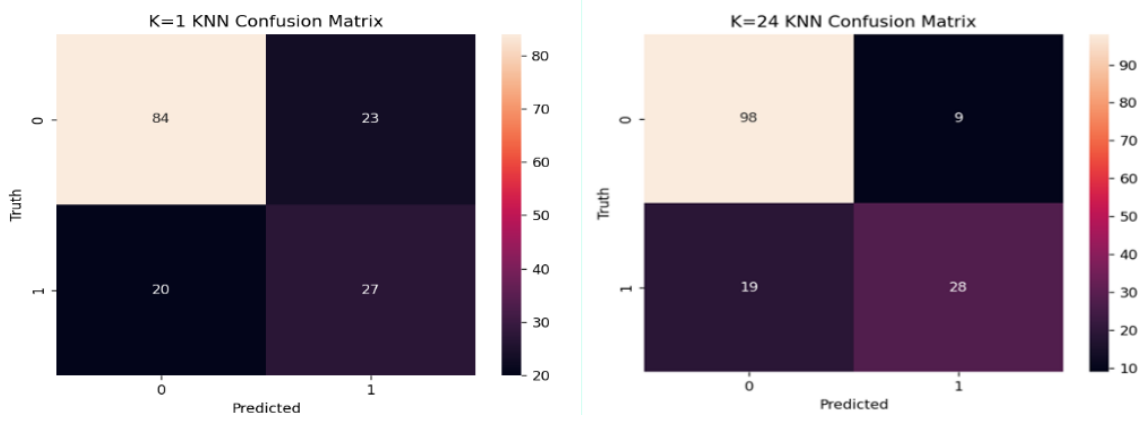
Aşağıda Karşıtlık Matrisleriyle Rapor yer almaktadır.

K=1	precision	recall	f1-score	support
0	0.81	0.79	0.80	107
1	0.54	0.57	0.56	47
accuracy			0.72	154
macro avg	0.67	0.68	0.68	154
weighted avg	0.73	0.72	0.72	154

Şekil 3.6.4.2- KNN K=1 Sınıflandırıcı Raporu

K=24	precision	recall	f1-score	support
0	0.84	0.92	0.88	107
1	0.76	0.60	0.67	47
accuracy			0.82	154
macro avg	0.80	0.76	0.77	154
weighted avg	0.81	0.82	0.81	154

Şekil 3.6.4.3- KNN K=24 Sınıflandırıcı Raporu



Şekil 3.6.4.4- KNN Karşıtlık Matrisi

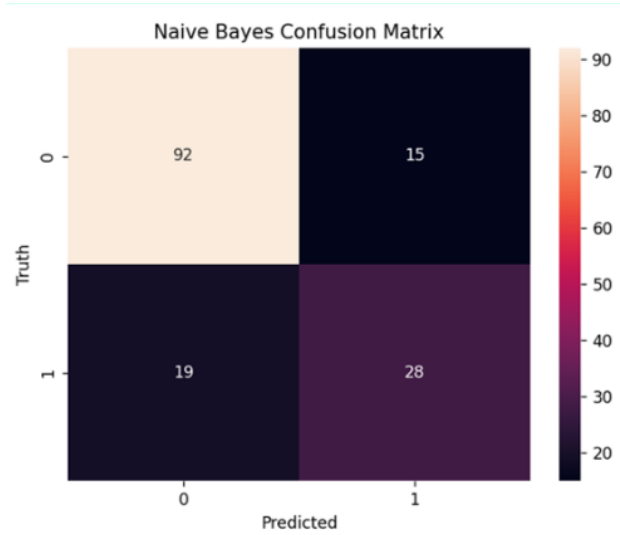
b. Naive Bayes Algoritması:

Naive Bayes modeli eğiterek buradan doğruluk puanı 0,77 olarak elde edilmiştir.

Aşağıda ise Karşıtlık Matrisleriyle Rapor yer almaktadır.

	precision	recall	f1-score	support
0	0.83	0.86	0.84	107
1	0.65	0.60	0.62	47
accuracy			0.78	154
macro avg	0.74	0.73	0.73	154
weighted avg	0.77	0.78	0.78	154

Şekil 3.6.4.5- Naive Bayes Sınıflandırıcı Raporu



Şekil 3.6.4.6- Naive Bayes Karşıtlık Matrisi

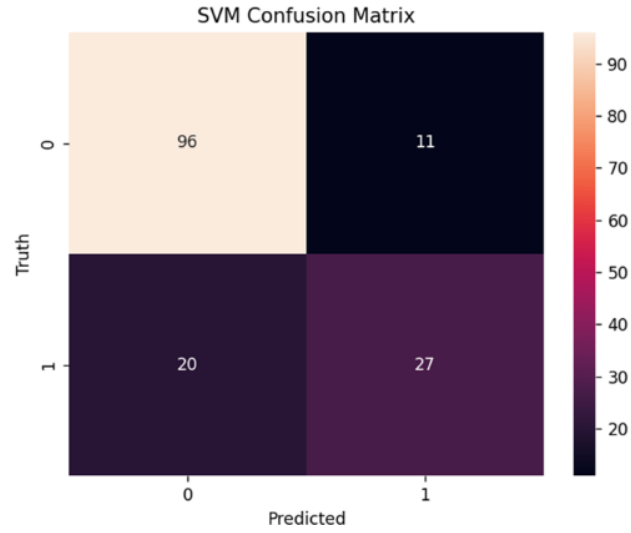
c. Support Vector Machines Algoritması:

Support Vector Machines modeli eğiterek buradan doğruluk puanı 0,79 olarak elde edilmiştir.

Aşağıda ise Karşıtlık Matrisleriyle Rapor yer almaktadır.

	precision	recall	f1-score	support
0	0.83	0.90	0.86	107
1	0.71	0.57	0.64	47
accuracy			0.80	154
macro avg	0.77	0.74	0.75	154
weighted avg	0.79	0.80	0.79	154

Şekil 3.6.4.7- Support Vector Machines Raporu



Şekil 3.6.4.8- Support Vector Machines Karşıtlık Matrisi

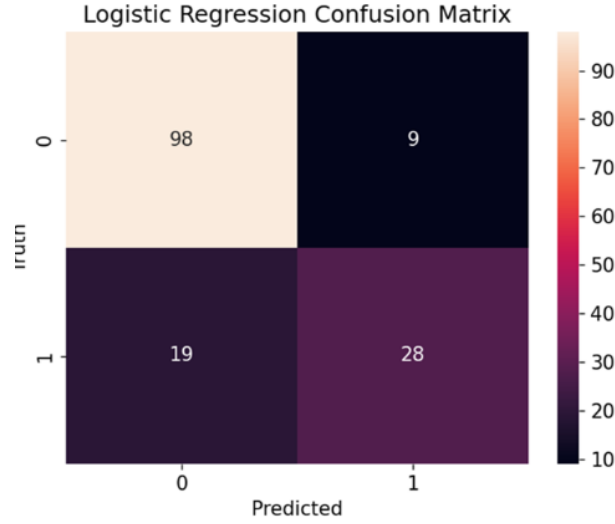
ç. Logistic Regression Algoritması:

Logistic Regression modeli eğiterek buradan doğruluk puanı 0.81 olarak elde edilmiştir.

Aşağıda ise Karşıtlık Matrisleriyle Rapor yer almaktadır.

	precision	recall	f1-score	support
0	0.84	0.92	0.88	107
1	0.76	0.60	0.67	47
accuracy			0.82	154
macro avg	0.80	0.76	0.77	154
weighted avg	0.81	0.82	0.81	154

Şekil 3.6.4.9- Logistic Regression Raporu



Şekil 3.6.4.10- Logistic Regression Karşıtlık Matrisi

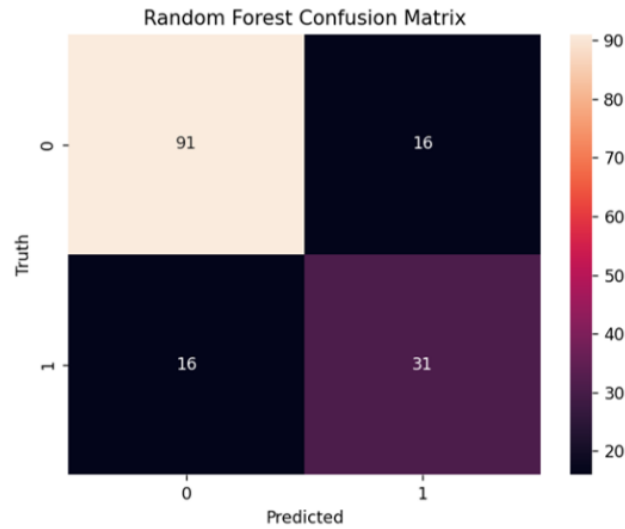
d. Random Forest Algoritması:

Random Forest modeli eğiterek buradan doğruluk puanı 0.81 olarak elde edilmiştir.

Aşağıda ise Karşıtlık Matrisleriyle Rapor yer almaktadır.

	precision	recall	f1-score	support
0	0.86	0.89	0.87	107
1	0.72	0.66	0.69	47
accuracy			0.82	154
macro avg	0.79	0.77	0.78	154
weighted avg	0.81	0.82	0.82	154

Şekil 3.6.4.11- Random Forest Raporu



Şekil 3.6.4.12- Random Forest Karşıtlık Matrisi

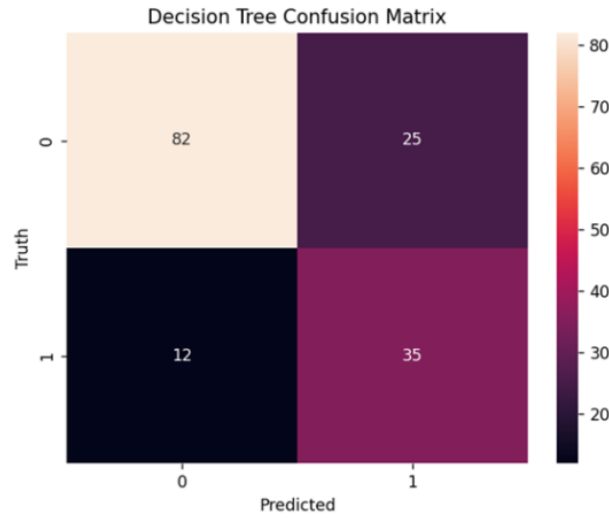
e. Decision Trees Algoritması:

Decision Trees modeli eğiterek buradan doğruluk puanı 0.75 olarak elde edilmiştir.

Aşağıda ise Karşıtlık Matrisleriyle Rapor yer almaktadır

	precision	recall	f1-score	support
0	0.84	0.79	0.82	107
1	0.58	0.66	0.62	47
accuracy			0.75	154
macro avg	0.71	0.73	0.72	154
weighted avg	0.76	0.75	0.76	154

Şekil 3.6.4.13- Decision TreesRaporu



Şekil 3.6.4.14- Decision Trees Karşıtlık Matrisi

f. Decision Trees Regression Algoritması:

Decision Trees Regression modeli eğiterek puanlayabilmek için rastgele seçilen sayılar üzerinden tahmin oluşturuldu ve sayı 1600 seçildiğinde DTR tahmini 76 , sayı 160 seçildiğinde ise DTR tahmini 54 olarak sonuçlandı.

g. Support Vector Regression Algoritması:

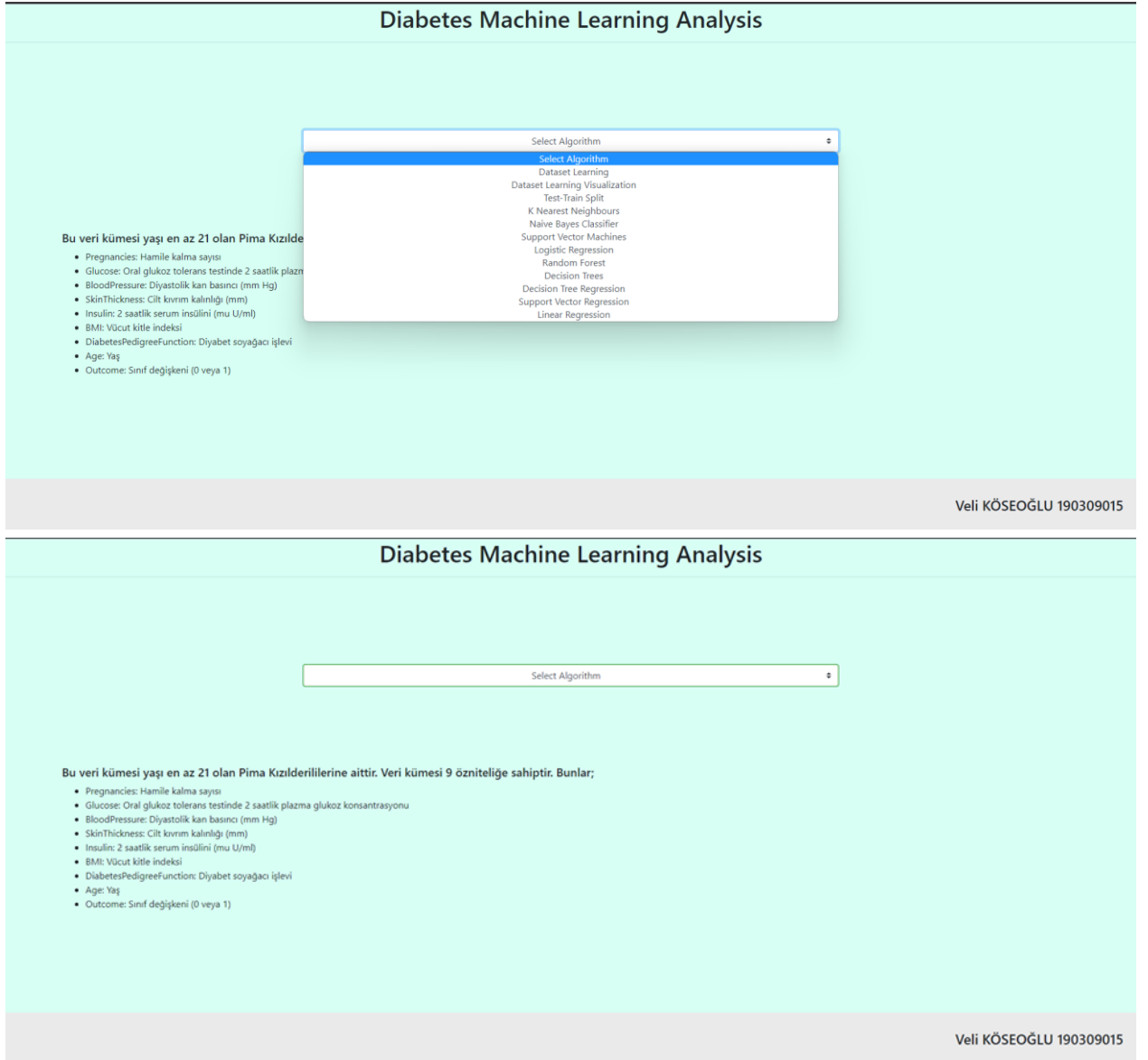
Support Vector Regression modeli buradan bir SVR modeli eğiterek 71.93 olarak tahmin sonucu elde edilmiştir.

ğ. Linear Regression Algoritması:

Linear Regression modeli burada ise lineer regresyon için eğitim modeli oluşturularak 0,34 olarak test skoru elde edilmiştir.

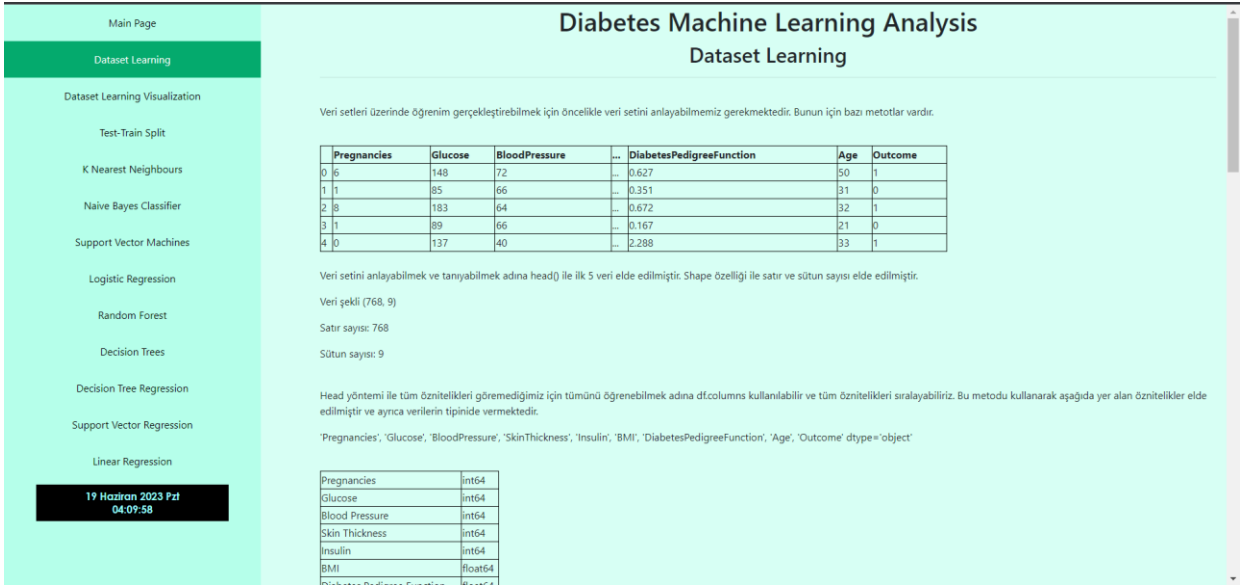
3.7. Kullanıcı arayüzü

Ana sayfada üç bölüm bulunmaktadır. Biri veri seti hakkında bilgi içeren bir bölüm, biri footer kısmı ve diğeri ise option bölümüdür. Buradan istediğimiz içeriğe tıklayarak o sayfayı yönlendiriliriz.

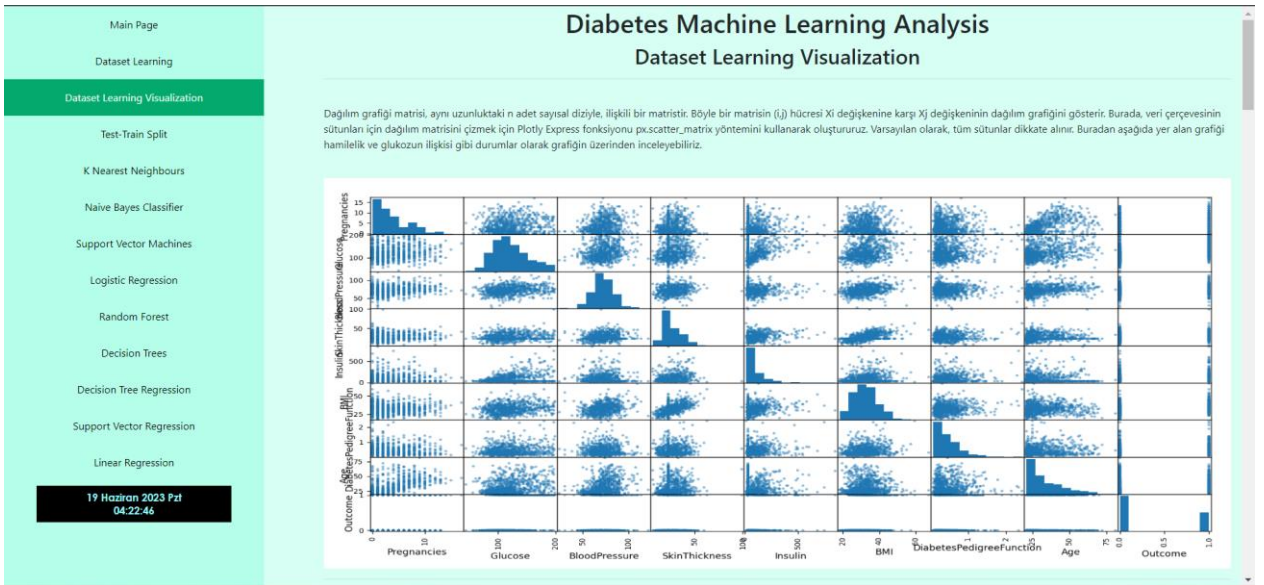


Şekil 3.7.1- Web Uygulamasının Ana Sayfası

Buradan “Dataset Learning” tıklarsak aşağıdaki sayfayı elde edeceğiz. Sırayla tıkladığımız sitlerin görünümü aşağıdaki görüntülerde yer almaktadır.



Şekil 3.7.2- Web Uygulamasının “Data Learning” Sayfası



Şekil 3.7.3- Web Uygulamasının “Data Learning Vizualization” Sayfası

Main Page

Dataset Learning

Dataset Learning Visualization

Test-Train Split

K Nearest Neighbours

Naive Bayes Classifier

Support Vector Machines

Logistic Regression

Random Forest

Decision Trees

Decision Tree Regression

Support Vector Regression

Linear Regression

19 Haziran 2023 Pzt 04:23:47

Diabetes Machine Learning Analysis

Test-Train Split

Bu zamana kadar yapılanlar veri setini anlama ve öğrenmeye dayalıydı. Bundan sonra veri seti üzerinden yeni modeller oluşturarak doğruluk skorları tarzi birimlerde edilmelidir. Bunları başlatılabilmek için öncelikle hangi algoritmaları ve modelleri kullanılabileceğinin analizinin yapılması gerekmektedir. Tüm bu analizler yapıldıktan sonra öncelikle veri setinin ayrıştırılıp test edilmesi, modellerinin iyileştirilmesi gibi süreçlerden geçmesine ihtiyaç duyulmaktadır.

```

graph TD
    ML[Machine Learning] --> SL[Supervised Learning]
    ML --> UL[Unsupervised Learning]
    ML --> RL[Reinforcement Learning]
    SL --> C[Classification]
    SL --> R[Regression]
    UL --> CL[Clustering]
    RL --> DM[Decision Making]
    C --> C_L["• Naive Bayes Classifier  
• Decision Trees  
• Support Vector Machines  
• Random Forest  
• K-Nearest Neighbors"]
    R --> R_L["• Linear Regression  
• Neural Network Regression  
• Support Vector Regression  
• Decision Tree Regression  
• Lasso Regression  
• Ridge Regression"]
    CL --> CL_L["• K-Means Clustering  
• Mean-shift Clustering  
• DBSCAN Clustering  
• Agglomerative Hierarchical Clustering  
• Gaussian Mixture"]
    DM --> DM_L["• Q-Learning  
• R-Learning  
• TD Learning"]
        
```

Yanda makine öğrenmesinde öğrenme çeşitleriyle beraber kullanılan algoritmalar bulunmaktadır. Bu algoritmalar veri setinin uygunluğuna göre makine öğrenmesi çeşitlerinden seçilerek kullanılır. Kendi diyabet veri setimize bakarsak son sütunda yer alan sonuç sütunundan dolayı bizim kullanacağımız makine öğrenmesi çeşidi Supervised Learning(Gözetimli öğrenme)'dir.

Bu projede Gözetimli öğrenme çeşitlerinden sınıflandırma(Classification) ve 3 regresyon(Regression) algoritmaları kullanılmıştır. Fakat bu algoritmaları kullanarak model oluşturmak için veri setini yatağı olarak train ve test veri setleri olarak bölmemiz gerekmektedir.

Train Veri Seti

Train veri seti üzerinde temel modelleme denemeleri yapılarak en doğru makine öğrenmesi algoritması seçilmeye çalışılır. Bu veri seti üzerinden en fazla örneklem alınan gözlemlerden (rows) oluşur.

Validation Veri Seti

Validation bölümü train veri seti içinden seçilir. Train veri seti üzerinde doğru model seçimi yapılarak algoritma belirlenir. Validation bölümünde ise uygulanan model iyileştirilmeye çalışılır. Bunun için hiper parametrik (hyperparameter tuning) uygulamaları denemek en optimum katsayılar/ağırlıklar bulunmaya çalışılır. Özellikle çok büyük veri setleri üzerinde sürekli Train datası üzerinden çalışılmayacağı için küçük bir bölüm alınarak validation olarak tanımlanır.

Şekil 3.7.4- Web Uygulamasının “Test-Train Split” Sayfası

Main Page

Dataset Learning

Dataset Learning Visualization

Test-Train Split

K Nearest Neighbours

Naive Bayes Classifier

Support Vector Machines

Logistic Regression

Random Forest

Decision Trees

Decision Tree Regression

Support Vector Regression

Linear Regression

19 Haziran 2023 Pzt 04:24:06

Diabetes Machine Learning Analysis

K Nearest Neighbours

kNN algoritması, çoğunluk sınıf etiketinin, özellik uzayındaki en yakın 'K' komşuları arasında yeni bir veri noktasının sınıf etiketini belirlediği bir oylama sistemi olarak düşünülebilir. Daha iyi anlayabilmek için bir örnek vermek gerekirse, mesela birkaç yüz kişinin yaşadığı küçük bir köy düşünün ve hangi siyasi partiye oy vereceğinize karar vermeniz. Bunun için en yakın komşunuza gidip hangi siyasi partiyi desteklediğini sorabilirsiniz. En yakın 'K' komşularınızın çoğunluğu A tarafını destekliyorsa, o zaman büyük olasılıkla A partisine de oy verirsiniz. Bu, kNN algoritmasının çalışma şekline benzer; k en yakın komşusu.

-> Daha önce normalleştirerek böldüğümüz veri setini K Nearest Neighbours(KNN-K En Yakın Komşu) algoritmasını kullanmak üzere bir model eğitebiliriz. Böylece veri kümesi üzerinde tahminler yapabiliriz.

-> Model eğitildikten sonra elimizde bulunan k'nin 1 olduğu bir knn modeli ile aşağıdaki eğitim ve test doğruluk sonuçları elde edilmiştir.

- Eğitim doğruluk puanı: 1.000
- Test doğruluk puanı: 0.721

-> K'nin 24 olduğu bir knn modeli ile aşağıdaki eğitim ve test doğruluk sonuçları elde edilmiştir.

- Eğitim doğruluk puanı: 0.772
- Test doğruluk puanı: 0.818

-> Daha sonra bu modeli puanlayabilmek için test veri kümesi üzerinden tahmin oluşturuldu ve aşağıdaki doğruluk skoru elde edildi.

- K=1 KNN Sınıflandırıcı Doğruluk Puanı: 0.7207
- K=24 KNN Sınıflandırıcı Doğruluk Puanı: 0.8181

K=1 KNN Sınıflandırıcı Raporu:

-> KNN sınıflandırıcı algoritmasında k'nin 1 olduğu ve hakkında aşağıda doğruluk, kesinlik, skor vs bildiren bir rapor elde edilmiştir.

	precision	recall	f1-score	support
0	0.81	0.79	0.80	107
1	0.54	0.57	0.56	47
accuracy			0.72	154
macro avg	0.67	0.68	0.68	154
weighted avg	0.73	0.72	0.72	154

Şekil 3.7.4- Web Uygulamasının “K Nearest Neighbours” Sayfası

➔ Sayfalar tıklandığı sürece yukarıda gösterildiği gibi ilerleyecektir ve her birine özel içerikler incelenebilecektir.

4. Sonuç

Sonuç olarak diyabet hastalığı veri seti üzerinden elde edilen sonuçlar gayet dikkat çekici derecede önemlidir. Günümüzde insanlar sağlığını fazlasıyla önemsemektedir. Bu öğrenme modeliyle gelecekte diyabet hastası olma ihtimallerini gözler önüne sererek hasta ve hasta adaylarına yardımcı olabileceği düşünülmektedir.

Bu projede Makine öğrenmesinde denetimli öğrenme modeli algoritmaları kullanılmıştır. Web sitesi tasarımı ise Flask çerçevesi kullanılarak web sitesi oluşumu sağlanmıştır.

Kullanılan algoritmalar arasında en yüksek skoru K'nin 24 olduğu model ve Logistic Regression modeli vermiştir. En düşük skoru ise K'nin 1 olduğu model ile elde edilmiştir.

Web uygulaması bölümü için, Windows işletim sistemleri desteklenmektedir. Bu işletim sisteminde dilediğinde istenen web tarayıcısında çalıştırılabilir.

Diğer projelerin aksine bu projeyle bir Flask çerçeveli web sitesi tasarlanmış olup gelişmeye açık olmasıyla ön plana çıkmaktadır. Ayrıca bu çalışmada taranan makaleler doğrultusunda daha önce çalışılmayan regresyon algoritmaları üzerine çalışılmış olup farklı bir pencereden bakılmaya çalışılmıştır.

5. Referanslar

1. CourtS.SeinE.McCowenC.HackettA.ParkinJ.Children with diabetes mellitus: perception of their behavioural problems by parents and teachersEarly Human Development161988245252
2. KayaerK.Y?ld?r?mT.Medical diagnosis on Pima Indian diabetes using general regression neural networksProceedings of The international Conference on Artificial Neural Networks and Neural information Processing (ICANN/ICONIP)2003181184
3. J. H. Friedman, "Data Mining and Statistics: What's the connection?", Computing science and statistics, vol. 29, no. 1, pp. 3-9, 1998.
4. F. E. Horita, J. P. de Albuquerque, V. Marchezini and E. M. Mendiondo, "Bridging the gap between decision-making and emerging big data sources: An application of a model-based framework to disaster management in Brazil", Decision Support Systems, vol. 97, pp. 12-22, 2017.
5. Bell, Jason. "What is machine learning?." Machine Learning and the City: Applications in Architecture and Urban Design (2022): 207-216.
6. Mahesh, Batta. "Machine learning algorithms-a review." International Journal of Science and Research (IJSR).[Internet] 9 (2020): 381-386.
7. <https://ufukcolak.medium.com/makine-%C3%B6%C4%9Frenmesi-veri-%C3%B6n-i%CC%87%C5%9Fleme-5-58e1ce73c1fb>
8. Ghimire, Devndra. "Comparative study on Python web frameworks: Flask and Django." (2020).
9. Flask, Github , <https://github.com/pallets/flask>
10. <https://www.datacamp.com/tutorial/k-nearest-neighbor-classification-scikit-learn>

6. Terimler Listesi

6.1. Tanımlar

Çerçeve - uygulama geliştirmek için bir platformdur. Geliştiricilerin belirli bir platform için yazılım geliştirebilecekleri bir ortam sağlar.

6.2. Kısaltmalar

GUI	Graphical User Interface (Grafiksel kullanıcı arayüzü)
KNN	K Nearest Neighbours (K En Yakın Komşular)
SVM	Support Vector Machines (Destek Vektör Makineleri)
DTR	Decision Tree Regression (Karar Ağacı Regresyonu)
SVR	Support Vector Regression (Destek Vektör Regresyonu)
CSV	Comma Separated Values (Virgülle Ayrılmış Değerler)