Diyabet Hastalığı Tespit Sistemi

Uygulama Sunumu

Hakam CHEDO 152120181096

Hüseyin KAYMAK 152120181100

Bengisu ŞAHİN 152120191064

Muhammet Eren SÖME 152120201049

UYGULAMA FAZI

Tasarım aşamasında planlanan diyabet tahmin sistemi, kullanıcı arayüzü, test, yazılım ve veritabanı tasarımlarının uygulama aşamasında nasıl gerçekleştiği incelenmiştir. Uygulama aşamasında ve tasarım aşamasında yapılan kullanıcı arayüz uyulmuştur. Yapılan tasarımına değişikliklerde sistemin yapısının bozulmamasına dikkat edilmiştir.



Geliştirilen Sistemin Sistem Tasarımlarını Karşılanmasının Değerlendirilmesi

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

Tamamlanan gereksinimler:

- Hastalar diyabet verilerini kolayca girebilirler.
- Hastalar girdikleri bilgilerin depolanmasını ve işlenmesini kabul eder.
- Hasta sonucu anında ekranda görebilir aynı zamanda hastanın e-posta adresine de gönderilir.
- Modelin doğruluk oranı şu anda %78'dir ancak daha fazla veri içeren veri seti veya farklı bir makine öğrenmesi modeliyle bu oran artırılabilir.

Kullanıcı ve Sistem Arayüzü Gerçeklemeleri

- Tasarım aşamasında planlandığı üzere, web tabanlı bir uygulama olarak geliştirilmiştir.
- Web uygulama arayüzü olarak streamlit python kütüphanesi kullanılmıştır.
- Web uygulamasının veri tabanı ile haberleşmek için kullandığı servis pyodbe python kütüphanesi ile geliştirilmiştir.



Gerçeklenen Testler

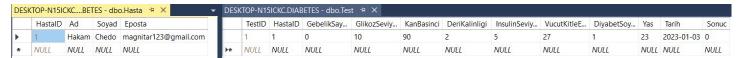
Birim (Unit) Testi

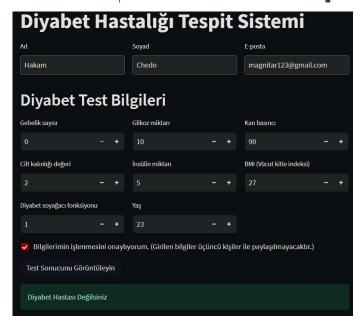
Projede birim testin uygulanan kısım, makine öğrenme modelinin verilen değerlerden hastalık sonucunun tespitidir.

```
import unittest
class TestResult(unittest.TestCase):
   def test(self):
        self.assertTrue(is sick(6,148,72,35,0,33.6,0.627,50))
        self.assertTrue(is sick(1,130,60,30,0,40.3,0.550,45))
def is_sick(Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age):
 input_data = [Pregnancies,Glucose,BloodPressure,SkinThickness,Insulin,BMI,DiabetesPedigreeFunction,Age]
 input data as numby 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
   return True
unittest.main()
```

Entegrasyon Testi

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







Performans Testi

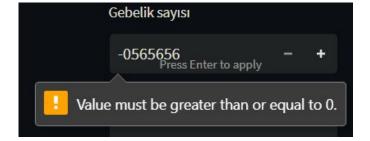
- Modelin kısa sürede (bir saniyeden daha kısa) değerleri sınıflandırıp yanıtladığı tespit edildi.
- Veri tabanı olarak MSSQL kullanıldı ve verilerin aktarılmasında bir sorun yaşanmadı.
- E-mail, Pythondaki SMTP kütüphanesi kullanılarak gönderilmekte ve herhangi bir gecikme yaşanmamaktadır.

Kabul Testleri

Ad, soyad, Eposta bilgileri boş olmamamalıdır.



Girilen değerlerin boş olması, rakamdan başka karakter girilmesi, gereğinden fazla veya az değer girilmemesi engellenmiştir.



Yazılım Gerçeklemeleri

Programlama dili olarak Python kullanılmıştır. Spyder IDE içerisinde modelin eğitim, doğrulanması ve test işlemleri yapılmıştır.

Makine öğrenmesi için Destek Vektör Makinesi yöntemi kullanılmıştır.

Python içindeki sklearn, pandas, numpy ve streamlit kütüphaneleri kullanılmıştır.

Sklearn kütüphanesini makine öğrenme modelini oluşturmak için kullandık.

Pandas kütüphanesini .csv uzantılı veri setimizdeki verileri okumak için kullandık.

Streamlit kütüphanesiyle uygulamamızın sunucu ve istemci taraflarını oluşturduk.

Veri Tabanı Gerçeklemeleri

Veri tabanı Microsoft SQL Server Management üzerinde kurulmuştur.

Veri tabanımızda toplam 2 tablo, 4 saklı yordam ve 1 tetikleyici bulunmaktadır.

Veri tabanı ile uygulamamız arasındaki iletişim Python programlama dilindeki pyodbc kütüphanesiyle bağlantı açılarak sağlanmaktadır.

Sonuç ve Öneriler

Oluşturulan proje sistem gereksinimlerini sağlar biçimde tamamlanmıştır ve sürdürülebilir, kullanıcı dostu bir uygulama olarak kullanılabilmektedir. 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.