**TOGU Bilgisayar Mühendisliği**

**BM3002 Veri Madenciliğine Giriş**

**2023-2024 Proje Ödevi**

**Zeynep Ceren MEMİŞ**

**213908068**

**Konu :**

Yapılan çalışmada yapay sinir ağı kullanılarak bir insanın diyabet hastalığının olup olmadığının tespit edilmesi amaçlanmıştır.

**Gerekli Genel Bilgiler:**

**Şeker (Diyabet) Hastalığı :**

Diyabetes Mellitus (DM); insülin hormonunun yokluğu, yetersizliği veya doku düzeyinde etkisizliği nedeniyle oluşan hiperglisemi (kan şekeri yüksekliği), dislipidemi (kan lipid düzeylerinin bozukluğu), glukozüri (idrara glukoz kaçışı) ve bunlara eşlik eden birçok klinik ve biyokimyasal bulgu ile seyreden sistemik kronik bir metabolizma hastalığıdır.

**Diyabetin Sınıflandırılması :**

Diyabetli hastaların %5-10’u Tip 1 DM, %90-95’ini ise Tip 2 DM oluşturmaktadır.

**Tip-1 Diyabet:**

Tip 1 DM Tip 1 DM herhangi bir yaşta ortaya çıkabilirken genellikle çocuklarda ve genç erişkinlerde oluşur. Pankreas β hücreleri insülin salgılayamaz ve hasta kişiler, kanlarındaki glikoz seviyesini kontrol etmek için her gün insülin enjeksiyonu 3 yapmak zorundadırlar.

**Tip-2 Diyabet:**

Tip 2 DM’ li kişilerde, insülin üretimi azdır veya onu yeterince kullanamamaktadırlar. Genellikle hastaların insülin gereksinimleri yoktur. Genellikle, diyetlerini kontrol ederek, düzenli egzersiz yaparak, ağızdan ilaç ve bazen de insülin alarak kanlarındaki glikozu kontrol edebilirler.

**Risk Faktörleri:**

Birçok yetişkin, hastalık tanısı konulmadan birkaç yıl önce diyabet olmuş olabilir. Tanı konduğunda birçoğunda, diyabet hastalığı komplikasyonlar gelişmeye başlamıştır. Görme azalması, böbrek yetmezliği, kalp hastalığı, felç ve sinir hasarı vb.

**Tip-1 Diyabet İçin Risk Faktörleri:**

Neden oluştuğu tam olarak saptanamamıştır. Fakat genellikle genetik ve çevresel olduğun düşünülmektedir.

**Tip-2 Diyabet İçin Risk Faktörleri:**

Yaş, Şişmanlık, Diyabete İlişkin aile hikayesi, Fiziksel tembellik, Bozulmuş Glikoz Toleransı, Irk/Etnik Özellikler, Hamilelik sırasında diyabet, Hipertansiyon varlığı, Kolesterol değerleri , Polikistik Over sendromu …

Veri setinde bu bilgilere dayanarak maximum doğru sonuç için 768 kişinin yaş, aile diyabet geçmişi , vki ( vücut kitle indeksi ), kan basıncı, kandaki glikoz değeri, kandaki insülin miktarı ve gebelik değerlerine göre bilgileri girdi olarak kullanılmıştır. Çıktı olaraksa diyabet (1) veya diyabet değil (0 ) çıktısı vermektedir.

Bilgileri kullanılan 768 kişinin 500’ü sağlıklı ( diyabetli değil), 268’i ise hasta (diyabetli ) insanlardır.

* Gebelikler: Hamile kalma sayısı
* Glikoz: Oral glukoz tolerans testinde 2 saat plazma glukoz konsantrasyonu
* Kan Basıncı: Diyastolik kan basıncı (mm Hg)
* Cilt Kalınlığı: Triceps cilt kıvrım kalınlığı (mm)
* İnsülin: 2 Saatlik serum insülini (mu U / ml)
* BMI: Vücut kitle indeksi (kg cinsinden ağırlık/(m cinsinden boy)^2)
* Aile Diyabet Geçmişi: Diyabet soyağacı işlevi
* Yaş: Yaş (yıl)
* Sonuç: Sınıf değişkeni (0 veya 1)

**Yapay Sinir Ağı Tasarımı :**

Veri setinin rastgele (random) seçilen %20’lik kısmı test geri kalan %80’lik kısmı ise eğitim verisi olarak kullanılmıştır.

**Kodlar :**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| In [1]:  import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  import seaborn as sns  #veri setini yükleme  In [2]:  veriseti = pd.read\_csv('C:/Users/ASUS/OneDrive/Masaüstü/diabetes.csv)  #Veri kümesini, boyutlarını, özelliklerini ve istatistiksel özetini görüntüleme  In [3]:  veriseti.head()  Out[3]:   |  | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigreeFunction | Age | Outcome | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | 6 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | 1 | | 1 | 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | 0 | | 2 | 8 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | 1 | | 3 | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 | | 4 | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |   In [4]:  veriseti.shape  Out[4]:  (768, 9)  In [5]:  veriseti.info()  <class 'pandas.core.frame.DataFrame'>  RangeIndex: 768 entries, 0 to 767  Data columns (total 9 columns):  # Column Non-Null Count Dtype  --- ------ -------------- -----  0 Pregnancies 768 non-null int64  1 Glucose 768 non-null int64  2 BloodPressure 768 non-null int64  3 SkinThickness 768 non-null int64  4 Insulin 768 non-null int64  5 BMI 768 non-null float64  6 DiabetesPedigreeFunction 768 non-null float64  7 Age 768 non-null int64  8 Outcome 768 non-null int64  dtypes: float64(2), int64(7)  memory usage: 54.1 KB  In [6]:  veriseti.describe().T  Out[6]:   |  | count | mean | std | min | 25% | 50% | 75% | max | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Pregnancies | 768.0 | 3.845052 | 3.369578 | 0.000 | 1.00000 | 3.0000 | 6.00000 | 17.00 | | Glucose | 768.0 | 120.894531 | 31.972618 | 0.000 | 99.00000 | 117.0000 | 140.25000 | 199.00 | | BloodPressure | 768.0 | 69.105469 | 19.355807 | 0.000 | 62.00000 | 72.0000 | 80.00000 | 122.00 | | SkinThickness | 768.0 | 20.536458 | 15.952218 | 0.000 | 0.00000 | 23.0000 | 32.00000 | 99.00 | | Insulin | 768.0 | 79.799479 | 115.244002 | 0.000 | 0.00000 | 30.5000 | 127.25000 | 846.00 | | BMI | 768.0 | 31.992578 | 7.884160 | 0.000 | 27.30000 | 32.0000 | 36.60000 | 67.10 | | DiabetesPedigreeFunction | 768.0 | 0.471876 | 0.331329 | 0.078 | 0.24375 | 0.3725 | 0.62625 | 2.42 | | Age | 768.0 | 33.240885 | 11.760232 | 21.000 | 24.00000 | 29.0000 | 41.00000 | 81.00 | | Outcome | 768.0 | 0.348958 | 0.476951 | 0.000 | 0.00000 | 0.0000 | 1.00000 | 1.00 |   In [7]:    #veri setindeki boş değerleri algılama  veriseti.isnull().sum()  Out[7]:  Pregnancies 0  Glucose 0  BloodPressure 0  SkinThickness 0  Insulin 0  BMI 0  DiabetesPedigreeFunction 0  Age 0  Outcome 0  dtype: int64  Data Visualization  #Veri Görüntüleme  #çıktı verisi görüntüleme  In [8]:  sns.countplot(x = 'Outcome',data = veriseti)  Out[8]:  <AxesSubplot:xlabel='Outcome', ylabel='count'>  ekran görüntüsü, dikdörtgen, kare, tasarım içeren bir resim  Açıklama otomatik olarak oluşturuldu  In [9]:  In [10]:  *# Korelasyon Grafiği*  sns.heatmap(veriseti.corr(), annot = True)  plt.show()  metin, ekran görüntüsü, ekran, görüntüleme içeren bir resim  Açıklama otomatik olarak oluşturuldu  In [11]:  *# Verilerin İşlenmesi*  *#Sıfır değerlerinin NaN ile değiştirme*  Yeni\_veriseti = veriseti  Yeni\_veriseti[["Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI"]] = yeni\_veriseti[["Glucose", "BloodPressure", "SkinThickness", "Insulin", "BMI"]].replace(0, np.NaN)  In [12]:  *# NaN değerlerinin sayısı (ölçülmeyen verilerin Tespitini sağlar.)*  Yeni\_dataseti.isnull().sum()  Out[12]:  Pregnancies 0  Glucose 5  BloodPressure 35  SkinThickness 227  Insulin 374  BMI 11  DiabetesPedigreeFunction 0  Age 0  Outcome 0  dtype: int64  In [13]:  *# NaN değerleri ortalama değer ile değiştirme*  yeni\_dataseti["Glucose"].fillna(yeni\_veriseti["Glucose"].mean(), inplace = True)  yeni\_veriseti["BloodPressure"].fillna(yeni\_veriseti["BloodPressure"].mean(), inplace = True)  yeni\_veriseti["SkinThickness"].fillna(yeni\_veriseti["SkinThickness"].mean(), inplace = True)  yeni\_veriseti[" ["Insulin"].fillna(yeni\_veriseti[" ["Insulin"].mean(), inplace = True)  yeni\_veriseti[" ["BMI"].fillna(yeni\_veriseti[" ["BMI"].mean(), inplace = True)  In [14]:  #NaN değerlerin sayısı kontrolü  yeni\_veriseti[".isnull().sum()  Out[14]:  Pregnancies 0  Glucose 0  BloodPressure 0  SkinThickness 0  Insulin 0  BMI 0  DiabetesPedigreeFunction 0  Age 0  Outcome 0  dtype: int64  Logistic Regression  In [15]:  #Lojistik Regresyon  y = yeni\_veriseti["'Outcome']  X = yeni\_veriseti[".drop('Outcome', axis=1)  In [16]:  *# Veri setini ikiye ayırma*  from sklearn.model\_selection import train\_test\_split  X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, y, test\_size = 0.20, random\_state = 42, stratify = yeni\_veriseti['Outcome'] )  In [17]:  *# Test veri seti ve eğitim veri setini oluşturma*  from sklearn.linear\_model import LogisticRegression  model = LogisticRegression()  model.fit(X\_train, Y\_train)  y\_predict = model.predict(X\_test)  /opt/conda/lib/python3.7/site-packages/sklearn/linear\_model/\_logistic.py:818: ConvergenceWarning: lbfgs failed to converge (status=1):  STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.  Increase the number of iterations (max\_iter) or scale the data as shown in:  https://scikit-learn.org/stable/modules/preprocessing.html  Please also refer to the documentation for alternative solver options:  https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression  extra\_warning\_msg=\_LOGISTIC\_SOLVER\_CONVERGENCE\_MSG,  In [18]:  y\_predict  Out[18]:  array([1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1,  0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0,  0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0,  1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0,  1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1,  0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0])  In [19]:  *# Karmaşıklık Matrisi*  from sklearn.metrics import confusion\_matrix  cm = confusion\_matrix(Y\_test, y\_predict)  cm  Out[19]:  array([[82, 18],  [27, 27]])  In [20]:  *# Karmaşıklık Matrisi Korelasyon grafiği*  sns.heatmap(pd.DataFrame(cm), annot=True)  Out[20]:  <AxesSubplot:>  ekran görüntüsü, metin, tasarım içeren bir resim  Açıklama otomatik olarak oluşturuldu  In [21]:  from sklearn.metrics import accuracy\_score  In [22]:  accuracy =accuracy\_score(Y\_test, y\_predict)  accuracy  Out[22]:  0.7077922077922078  In [23]:  #Örnek: Kişinin şeker hastası olup olmadığını bazı rastgele değerler kullanarak kontrol etme  y\_predict = model.predict([[1,148,72,35,79.799,33.6,0.627,50]])  print(y\_predict)  if y\_predict==1:  print("Diabetic")  else:  print("Non Diabetic")  [1]  Diabetic |