T.C. KIRIKKALE ÜNiVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ

DÖNEM PROJESİ 2

Hazırlayan Burak Şencan

Proje Danışmanı Yardımcı Doc Erdal Erdal

Kırıkkale – 2019 Güz

İÇERİK

1) HİPERTANSİYON

1.1 Giriş-Tanım

1.2 Sınıflandırma

1.3 Tedavi

2) Pandas Keras

3) Uygulama Kodları

4) Kaynakça

1) **HİPERTANSİYON**

1.1 Giriş-Tanım

Tekrarlanan ofis olcumlerinde arteriel kan basıncının 140/90 mmHg den daha yuksek olması

hipertansiyon (HT) olarak tanımlanır. Hipertansiyon, surekli kan basıncı yuksekliği ile kendini

gösteren, sistemik bir hastalık olup, ciddi komplikasyonlara neden olması ve toplumda yaygın

olarak gorulmesi nedeniyle onemli bir sağlık problemidir.

Tedavi edilmeyen hipertansiyonun, kalp yetersizliği, koroner kalp hastalığı, hemorajik ve

trombotik inme, bobrek yetersizliği, periferik

arter hastalığı, aort diseksiyonu ve olum oranını artırdığı ortaya konmuştur. Hipertansiyonun

komplikasyonları ve buna bağlı olum oranı, kan basıncı yuksekliği ile doğru orantılı olarak

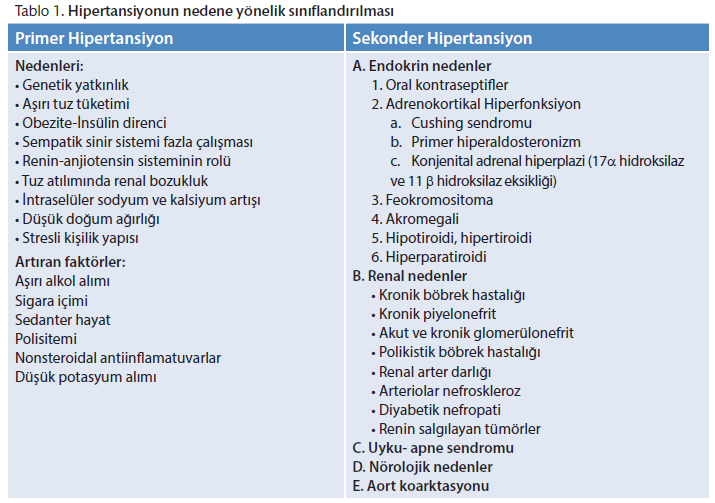
artmaktadır.

Tum hipertansiyon olgularının yaklaşık %80-90’ını oluşturan primer (esansiyel) hipertansiyon,

kesin mekanizması bilinmeyen, herhangi bir ikincil hastalığa bağlı oluşmamış, sistemik arteriyel

kan basıncının surekli yuksekliğidir. Sekonder ise hastalıın nedenlerinin daha belirgin oldugu

durumdur.



Hipertansiyonun belirlenmesinin ve tedavisinin amacı kalp, beyin, damar ve bobrek hastalıkları

riskini ve eşlik eden morbidite ile olum oranını azaltmaktır. Bu nedenle, yuksek riskli bireyleri

belirlemek, izlem ve tedavi ilkelerini sağlamak amacıyla, surekli guncellenen kılavuzların onculluğunde yetişkinler icin kan basıncı sınıfl amaları yapılmaktadır.

1.2) Sınıflandırma

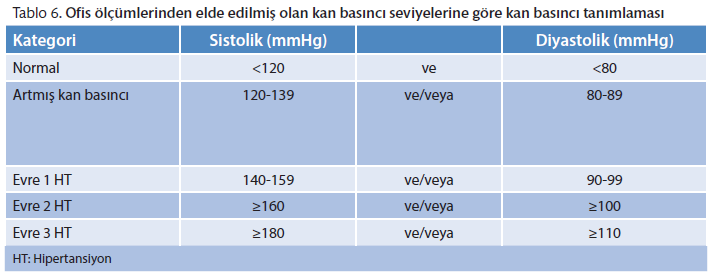
Tekrarlanan olcumlerde, erişkinlerde (>18 yaş) hekim tarafından yapılan standart (ofis) olcum

ile sistolik KB ≥140 mmHg ve/veya diyastolik KB ≥90 mmHg olması hipertansiyon olarak tanımlanır.

120/80 mmHg’ nın altındaki KB değerleri normal olarak kabul edilmiştir. Sistolik KB’

nın 120-129 mmHg ve/veya diyastolik KB’ nın 80-89 mmHg olması artmış KB kategorisinde

değerlendirilmiştir. Değerler Tablo 6’da ozetlenmiştir.



1.3) Tedavi

Hipertansiyon tedavisinde başlıca yaşam bicimi değişikliklerini kapsayan nonfarmakolojik

ve farmakolojik tedavi yontemleri vardır ve birlikte yapılmalıdır.

Hipertansiyonda yaşam bicimi değişiklikleri: Hipertansiyon oluşumunda beslenme alışkanlıkları ve fiziksel aktivitenin yetersizliği gibi cevresel etkenler son derece onemlidir; yaşam bicimi değişiklikleri KB’nı anlamlı olarak azaltır.

Hipertansiyonda farmakolojik tedavi: Herhangi bir KVH risk düzeyiyle 2. ve 3. derece hipertansiyonu bulunan kişilerde, yaşam tarzı değişiklikleriyle eş zamanlı olarak veya bu değişikliklerden birkaç hafta sonra olmak üzere ilaç tedavisine hemen başlanması

2) Python Pandas ,Keras

**Pandas Nedir?**

**Pandas**python programlama dili için **yüksek performanslı**, kullanımı kolay **veri yapıları**ve **veri analiz araçları**sağlayan açık kaynaklı bir BSD lisanslı kütüphanedir. **Csv**ve **text**dosyalarını açmaya ve  içerisinde bulunan verileri okuyarak istenen sonuca  kolayca ulaşmak için kullanılmaktadır. Yani bir excel dosyasını açarak içerisinde bulunan bir sütunu veya satırı seçerek işlemleri yapabiliriz. Numpy kütüphanesinde yapılan verilerin şekillendirilmesi daha detaylı bir biçimde kullanılabilmektedir.

**Time series data**analizinde kullanılmaktadır. Örnek verecek olursak x ve y denklemimiz olduğunu varsayalım. X ekseninin zaman olduğunu varsayalım Y ekseninin ise hız olduğunu düşünecek olursak zamana bağlı veriyi bu şekilde depolayabilir. Hız aralıkları ve değişimlerini kolay bir şekilde bulabiliriz.

Keras Nedir

Keras, derin öğrenme için yazılmış bir python kütüphanesidir. Sembolik işlem temelli ve derin öğrenme için de kullanılan Theano yada Tensorflow kütüphaneleri üzerinden çalışır. GPU yada CPU üzerinde çalışmasını bu temel kütüphaneler üzerinden sağlar. Daha üst düzey bir kütüphane olduğundan Theano yada Tensorflow a göre daha kolay uygulama geliştirebilirsiniz. Oldukça yaygın bir kullanımı vardır

4) Bu uygulamada, kişinin gebelikSayisi','diastolicKanBasinci','vucutKitleIndeksi','yas' bilgisi ile hipertansiyon hastası olup olmadıgını tahmin eden makine öğrenmesi geliştirilmiştir.

Uygulama Kodlari:

import pandas as pd # pandas is a dataframe library

import matplotlib.pyplot as plt # matplotlib.pyplot plots data

import numpy as np

import sklearn

import keras

import tensorflow

from IPython import get\_ipython

get\_ipython().run\_line\_magic('matplotlib', 'inline')

from sklearn import metrics

import matplotlib.pyplot as plt # side-stepping mpl backend

import matplotlib.gridspec as gridspec # subplots

import scipy

from pandas import DataFrame

from keras.models import Sequential

from keras.layers import LSTM

from keras.layers.core import Dropout, Flatten, Activation, Dense

from keras.layers.convolutional import Convolution2D, Convolution1D,MaxPooling1D

#Import models from scikit learn module:

from sklearn.model\_selection import train\_test\_split

from keras.layers import Input, Dense

from keras.optimizers import SGD

from sklearn.preprocessing import Imputer

import pandas as pd

import matplotlib.pyplot as plt

#dosyayı oku

df = pd.read\_csv("C:/Users/burak/Desktop/proje/pima-indians-diabetes.csv")

df.rename(columns={'1':'gebelikSayisi',

'148':'glikozKonsantrasyonu',

'72':'diastolicKanBasinci',

'35':'trisepsKalinligi',

'0':'ikiSaatlikSerumInsulinDegeri',

'33.6':'vucutKitleIndeksi',

'0.627':'aileYakinlikDerecesi',

'50':'yas',

'1.1':'sinif',

},

inplace=True)

df.head()

#özellik kolon isimleri

feature\_col\_names = ['gebelikSayisi','diastolicKanBasinci',

'vucutKitleIndeksi','yas']

#tahmin edilecek sınıf adı

predicted\_class\_names = ['sinif']

X = df[feature\_col\_names].values

y = df[predicted\_class\_names].values

split\_test\_size = 0.05

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=split\_test\_size,random\_state=0)

# test\_size = 0.3 is 30%, 42 is the answer to everything

print("{0:0.2f}% Eğitim setinde".format((float(len(X\_train))/len(df.index)) \* 100))

print("{0:0.2f}% Test setinde".format((float(len(X\_test))/len(df.index)) \* 100))

print("Eğitim setinde bulunan diabet : {0} ({1:0.2f}%)".format(len(y\_train[y\_train[:]== 1]), (float(len(y\_train[y\_train[:] == 1]))/len(y\_train) \* 100.0)))

print("Eğitim setinde bulunan iyi nonDiabet: {0}({1:0.2f}%)".format(len(y\_train[y\_train[:] == 0]), (float(len(y\_train[y\_train[:] ==0]))/len(y\_train) \* 100.0)))

print("")

print("Test setinde bulunan diabet:{0}({1:0.2f}%)".format(len(y\_test[y\_test[:] == 1]),(float(len(y\_test[y\_test[:] == 1]))/len(y\_test) \* 100.0)))

print("Test setinde bulunan nonDiabet:{0}({1:0.2f}%)".format(len(y\_test[y\_test[:]==0]), (float(len(y\_test[y\_test[:] == 0]))/len(y\_test) \* 100.0)))

model = Sequential()

model.add(Dense(4, input\_dim=4, init='uniform', activation='sigmoid')) # 1000neurons

model.add(Dense(256, init='uniform', activation='relu'))

model.add(Dense(128, init='uniform', activation='relu')) # 1 output neuron

model.add(Dense(1, init='uniform', activation='sigmoid')) # 1 output neuron

model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'])

model.fit(X\_train, y\_train, nb\_epoch=64, batch\_size=20, verbose=2) # 150 epoch,10 batch size, verbose = 2

predictions = model.predict(X\_test)

rounded = [np.round(X\_test) for X\_test in predictions]

print("Accuracy:{0:4f}".format(metrics.accuracy\_score(y\_test, rounded)))

from sklearn.metrics import confusion\_matrix,classification\_report

predictions = model.predict\_classes(X\_test)

print(confusion\_matrix(y\_test, predictions))

print(classification\_report(y\_test, predictions))

#predictions = model.predict\_classes(tahmin)

#gebelikSayisi glikozKonsantrasyonu diastolicKanBasinci trisepsKalinligi

#1 85 66 29

tahmin=[ 1. , 100. , 40.2, 37. ]

feature\_try = np.array([tahmin])

result =model.predict\_classes(feature\_try)

if result==0:

print("NO Diabetes")

else:

print("Diabetes")

X\_train[1]

X\_test[1]