Bitirme Projesi 1



Ad : Ege Erbilen

Okul No : 160255080

Bölümü : Bilgisayar Mühendisliği

Sınıf : 4.sınıf

**İÇİNDEKİLER**

Proje Amacı …………………………………………………………………….1

Proje Özeti ……………………………………………………………………..1

Proje Kapsamı …………………………………………………………………..1

Kaynaklar …………………….…………………………………………………1

Projede Kullanılan Materyaller …………………………………………………2

Projenin Gereklilikleri ….….……………………………………………………2

Proje Önerileri ...……………...…………………………………………………2

Proje Süreci ……………………………………………………………………..3

Proje Nasıl Kullanılır …...………………………………………………………3

Kaynak Kodlar ………………………………………………………………...18

**Proje Amacı**

Ekran karşısında çok zaman geçiren insanların göz sağlığını ve performansını (konsantrasyon) değerlerini artırmak.

**Proje Özeti**

Kullanıcının kamerasından alınan veriler ile kişiye özel olarak günlük performans grafiği çıkarılır ve bu gerçek veriler ile birlikte kişinin gelecekteki performans grafiğini tahmin eder.

**Proje Kapsamı**

Proje ekran karşısında (bilgisayar) çokça vakit geçiren kişileri kapsamaktadır.

**Proje İçeriği**

Proje kapsamında python kütüphanesi olan numpy, cv2, time, datatime, winsound, pymongo ve matplotlib kütüphanelerinden faydalanılmıştır.

**Kaynaklar**

1-https://stackoverflow.com/questions/6537481/python-making-a-beep-noise

2- https://www.w3schools.com/python/python\_ml\_getting\_started.asp

3- https://www.w3schools.com/python/python\_ml\_mean\_median\_mode.asp

4- <https://www.w3schools.com/python/python_ml_standard_deviation.asp>

5- <https://www.w3schools.com/python/python_ml_percentile.asp>

6- https://www.w3schools.com/python/python\_ml\_data\_distribution.asp

7- https://www.w3schools.com/python/python\_ml\_normal\_data\_distribution.asp

8- https://www.w3schools.com/python/python\_ml\_scatterplot.asp

9- https://www.w3schools.com/python/python\_ml\_linear\_regression.asp

10- https://www.w3schools.com/python/python\_ml\_polynomial\_regression.asp

11- https://www.w3schools.com/python/python\_ml\_multiple\_regression.asp

1

12- <https://www.w3schools.com/python/python_ml_scale.asp>

13- https://www.w3schools.com/python/python\_ml\_train\_test.asp

14- <https://github.com/infoaryan/Eye-blink-detection-game>

15- https://medium.com/datarunner/matplotlibkutuphanesi-1-99087692102b

**Projede Kullanılan Materyaller**

Bilgisayar, internet, bilgisayar kamerası, Win10, Python, Anaconda, OpenCV ve Spyder projenin yapımı için gerekli kaynakların oluşmasına ve yürütülmesine yardımcı olmuştur.

**Sistemin Kullanım Alanları**

Geliştirilen bu yapı tüm şirketler tarafından kullanılabilir bu sayede çalışanlara yönelik daha verimli çalışma saatleri düzenlenebilir bu sayede şirketin verimliliği artırılabilir. Aynı zamanda insanlar bu uygulamayı kişisel olarak da kullanabilirler. Bu program tüm yaştaki kullanıcılar için kullanılabilir.

**Projenin Gereklilikleri**

Win10, Python 3.8.3 versiyonu, ilgi kütüphanelerin ekli olması, kamera, işlemci, ram, ekran kartı, elektrik ve kişinin verileri görmesi için ekran.

**Proje önerileri**

Proje şu anda masaüstü için kullanıma uygundur, geliştirilmiş olan program web için de yazılabilir. Bu sayede güncelleme ve geliştirme işlemleri daha kolay olur kullanıcılar gelen her güncelleme için tekrar indirme yapmak zorunda kalmaz ve kullanıcıların saklama yerlerini işgal etmemiş oluruz. Proje daha küçük bir donanımda yapılabilir ise araç sürücülerinin yorgunluk grafiğini çıkarabiliriz. (Anlık olarak kullanıcıyı uyaran sistemler vardır. https://www.youtube.com/watch?v=DPFoFFd0u6E)

2

**Proje süreci**

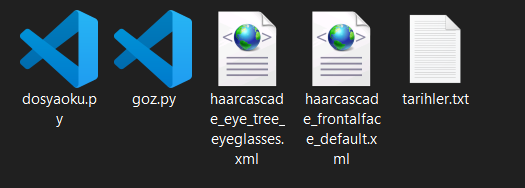
Proje süresince plan odaklı çalışma, problemleri çözebilme ve problemlere farklı bir bakış açısı ile bakma, insanların hayatını nasıl kolaylaştırabilirim ve onlara daha iyi nasıl yardımcı olurum konusunda bana çokça fikir verdi, pes etmemeyi ve farklı bakış açısı ile de problemlerin ortadan kaldırılabileceğini öğrendim. Proje sürecin içerisinde okulda ve okul dışında öğrendiğim bilgileri pekiştirme şansım oldu, python dili içerisinde kendimi şu anda eskiye nazaran daha hakim hissediyorum. Proje içerisinde bazen kafa karışıklığı yaşadım bu kafa karışıklığının sebebi ise 1. olarak veri tabanı için herhangi bir veri tabanı uygulamasını projesini entegre edip etmeme konusunda kararsız kaldım ve biraz araştırma yaptık dan sonra ise txt dosyası kullanmaya karar verdim bu sayede kullanıcı herhangi bir veri tabanı uygulaması kurmak zorunda kalmadı. Dosya boyutu bence hala çok yüksekti ve bu sorunu da günleri kısa tutarak çözmeye karar verdim. Dosya içerisinde ki her karakter (harfler ,sayılar, boşluk vb.) 1 bayttır, eğer dosya içerisinde alt satıra iner isek 2 bayt olur. Tabii bu bilgiden sonra verileri aynı satırda virgül kullanarak yazdırdım fakat veri txt dosyası içeriği çok karışık oldu ben de bu çözümden memnun kalmadım ve her birini alt satıra indirerek bir çözüm elde ettim. 2. olarak ise günleri kod içerisinde string yapıda 7 değişken ile uttum sonradan bu yapının karışık olduğunu düşünüp derslerde öğrendiğim tuple ile çözmeye ulaştım ve kodun anlaşılmasını kolaylaştırmış oldum.

**Proje Nasıl Kullanılır**

Öncelikle Projemizin bulunduğu “goz” klasörünü açarız.



Dosyamızı açtıktan sonra aşağıdaki gibi bir görüntü gelecektir.

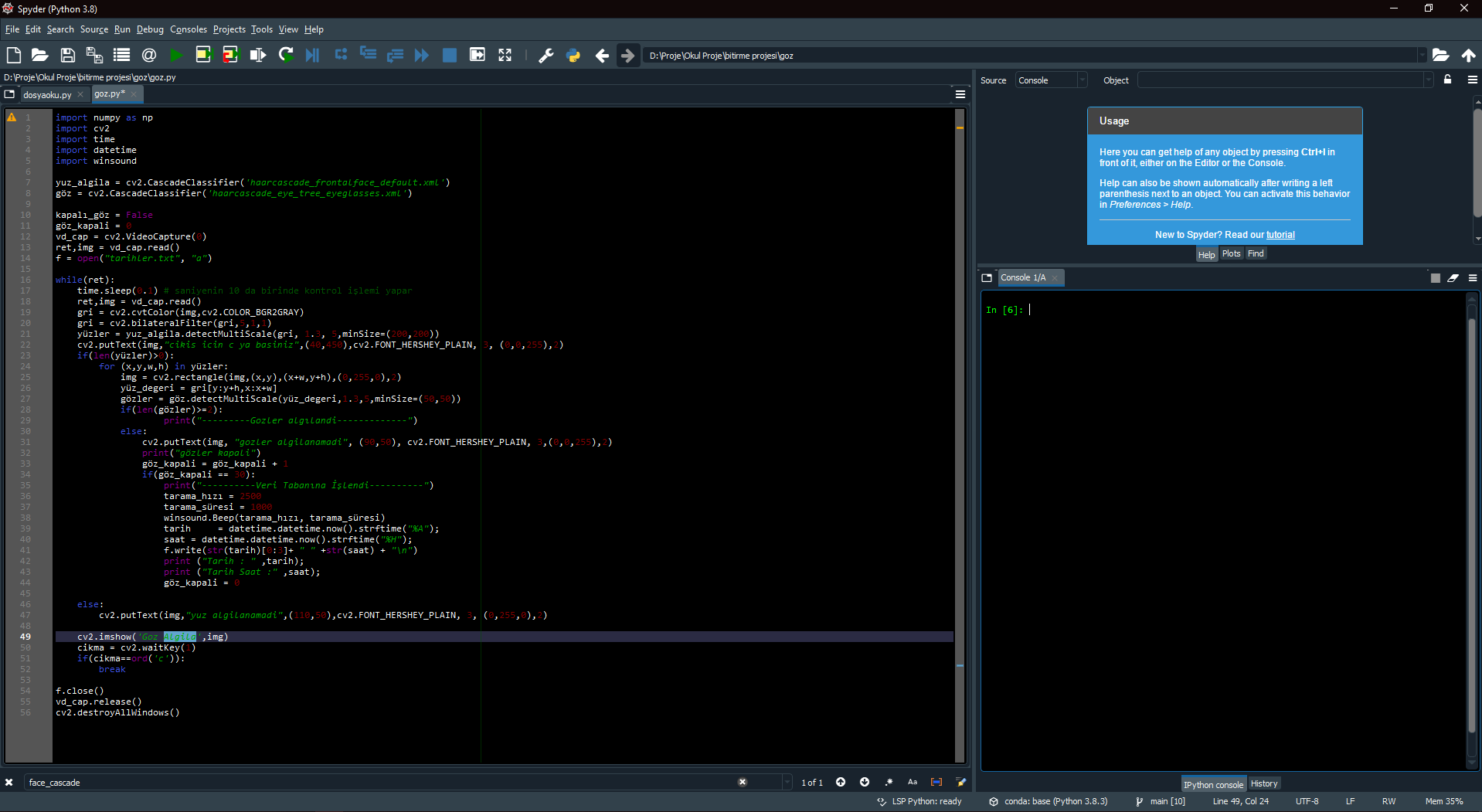


3

Ardından projemizi Anaconda içerisinde bulunan spyderdan açtıktan sonra

C:\Users\Ege\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Ekran Alıntısı.PNG

Burada ki “goz.py” ve “dosyaoku.py” dosyasını spyder içerisine sürükle bırak yaparız ve aşağıdaki gibi bir görüntü karşımıza gelir.



4

“goz.py” dosyamızı açtıktan sonra “F5” tuşuna basarak veya yukarıda bulunan tuşa basarak programımızı başlatırız. Program başlatıldıktan sonra ise



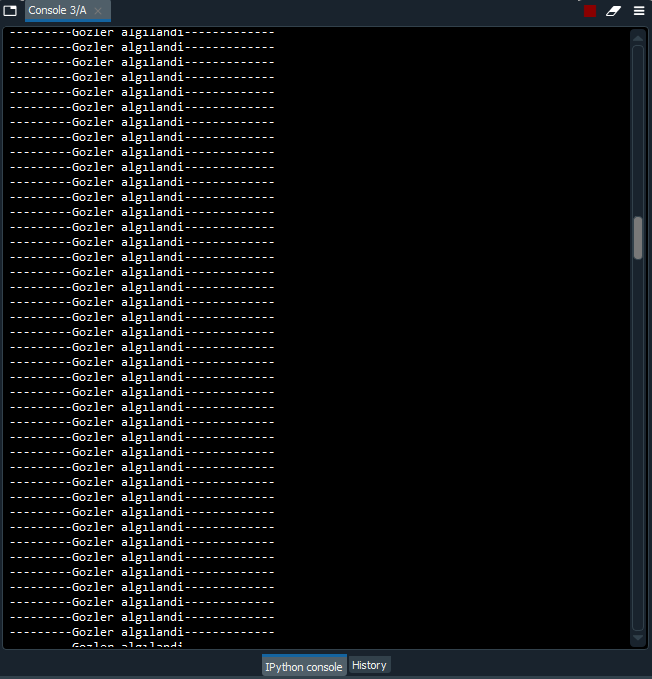
Şeklinde bir ekran gelecektir burada kamerayı elimle kapattığım için “yuz algilanamadi” diye bir yazı görmekteyiz ve biz eğer programımızı kapatmak ister isek “c” tuşuna basmalıyız yukarıdaki çarpı işareti ile kapatırız fakat program bir döngü içerisinde olduğundan kendini tekrar başlatacaktır.

5

Kameramızı açtıktan sonra ise yüzümüz eğer net bir şekilde görünüyorsa programımız bu şekilde çalışacaktır.

Bu şekilde çalışırken sistem içerisindeki konsola da gözlerin algılandığı bilgisini bastıracaktır. Aşağıda ekran çıktısını görebilirsiniz.

6

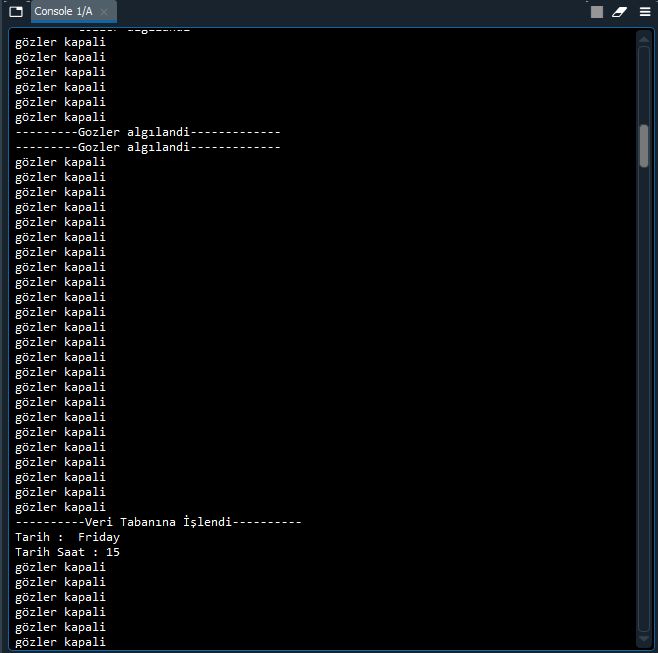


7

Eğer sistem gözleri algılayamaz ise ekranda aşağıdaki gibi bir görüntü göreceğiz.

8

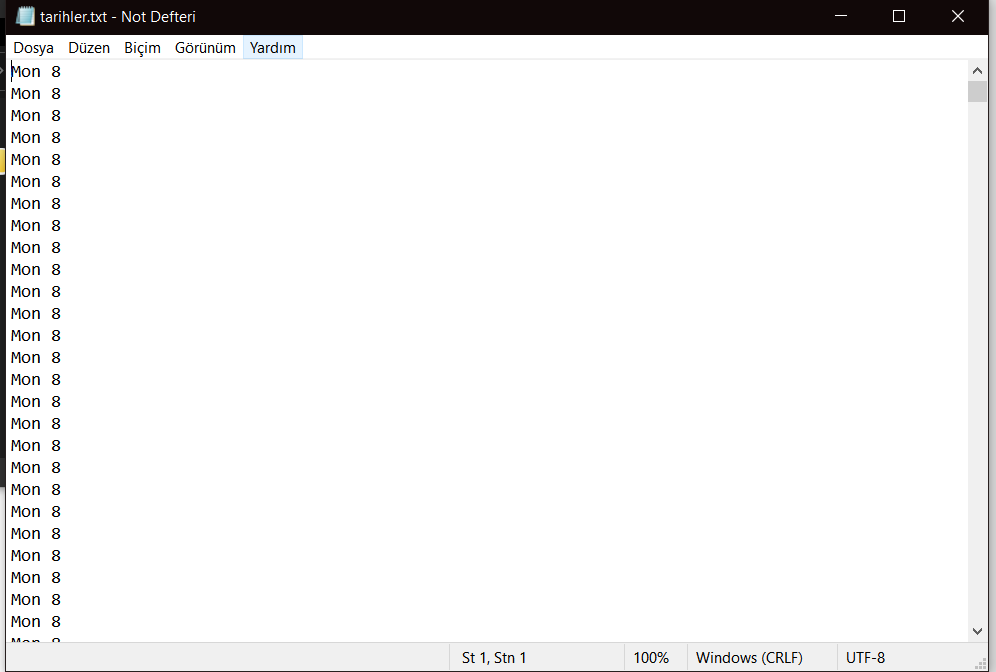
Konsolda ise



Çıktımız bu şekilde olacaktır sistem içerisinde gözleri alığılarsa veya algılayamadığı durumlarda bunu konsola yazdırır, eğer yüz algılanıp gözler algılanamaz ise 5 saniye boyunca sistem “tarih.txt” klasörüne bu veriyi ekler, eğer ki “tarih.txt” yok ise önce oluşturur sonra ekler. 5 saniye sonra aynı zamanda da konsola günü ve saatini bastırır.

9

“tarih.txt” dosyasının içi. Dosyanın içerisinde günleri kısaltarak hafızada daha az yer kaplamasını sağladım her bir karakterin 1 bayt olduğu ve alt satıra geçmek ise 2 bayt olduğunu göz önüne alarak böyle bir sistem yaptım. Peki neden hepsini aynı satıra yazdırıp virgüllerle ayırıp kod içerisinde de bu virgüllere göre ayırıp verileri de okuyabilirdik ama ben burada veri tabanının daha anlaşılır ve kodun daha anlaşılabilir olmasına odaklandım. Eğer sistem için kullanıcılar veriler çok yer kaplıyor der ise yukarıda bahsetmiş olduğum öneriyi sistem için kullanabiliriz hatta “mon 8“ yerine “mon8” şeklinde yazılarak da her gelen veri için 1 bayt tasarruf yapabiliriz.



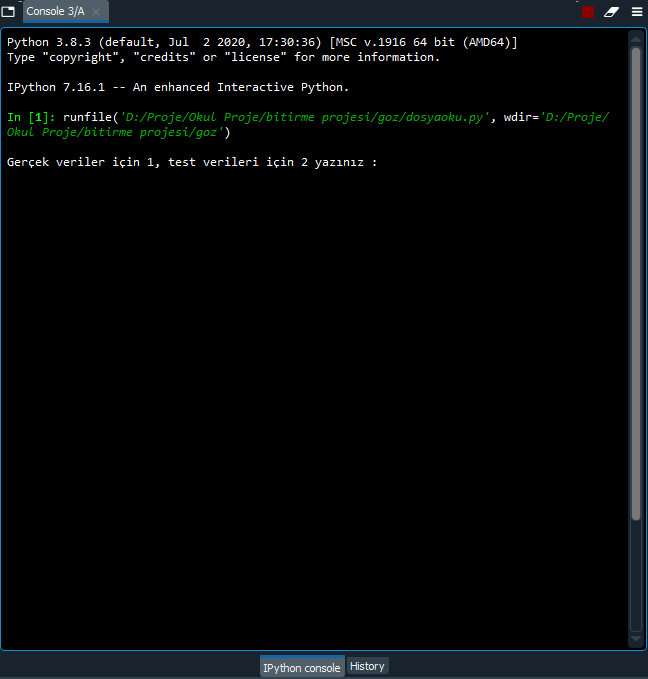
10

Kullanıcı yeterli ışıklandırmaya ve yeterli bir görüntü kalitesine sahip değil ise “Goz Algila” penceresinde ise aşağıda ki çıktıyı görür. Konsola herhangi bir yazı bastırılmaz.

Kullanıcı “c” tuşuna bastıktan sonra program sonlanacaktır ve pencere kapanacaktır.

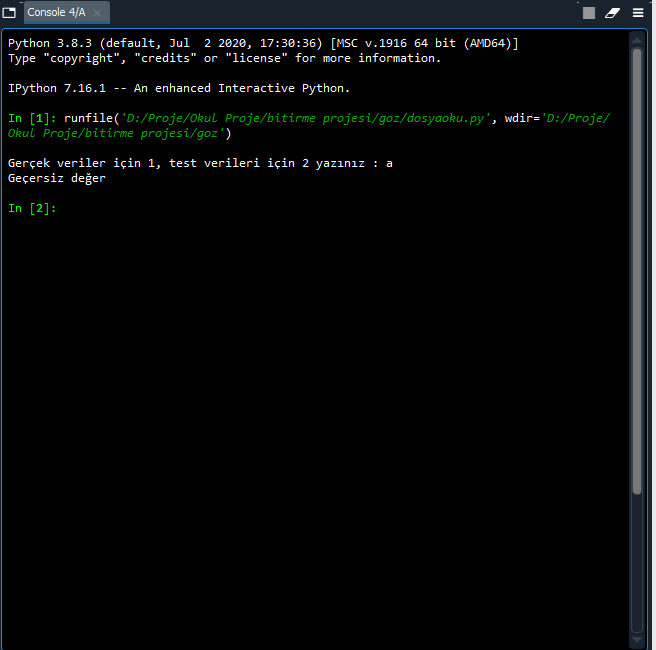
11

Sistem içerisine kaydedilen tüm bu verilerin analizini yapmak için ise “dosyaoku.py” klasörüne geçeriz ve çalıştırırız.



Bu çıktımızı görürüz. Burada eğer 1 yazarsak kişinin gerçek verileri üzerinden işlem yapar, 2 yazarsa önceki verileri işler ve buna bakarak gelecekteki yorgunluk grafiğini tahmin etmeye çalışır. Sisteme başka bir girdi yazmaya çalışırsak (1 ve 2 dışında) .

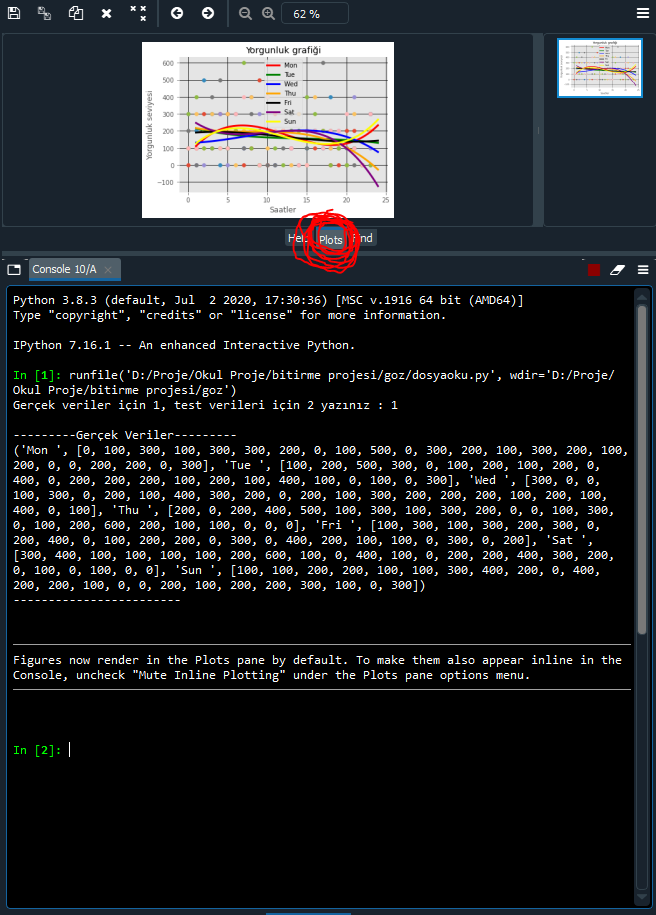
12



Karşımıza yukarıdaki gibi bir çıktı gelecektir.

13

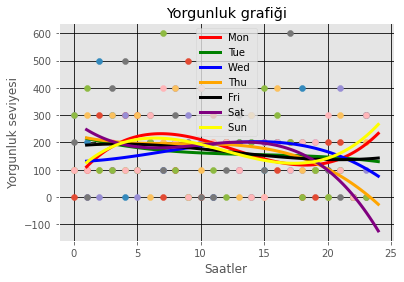
Eğer bu alana 1 yazarsak çıktı aşağıdaki gibi olacaktır.



14

(Plots kısmının yani kırmızı ile işaretlenmiş alanın seçili olduğundan emin olunuz yoksa grafiğimiz gözükmeyecektir.)

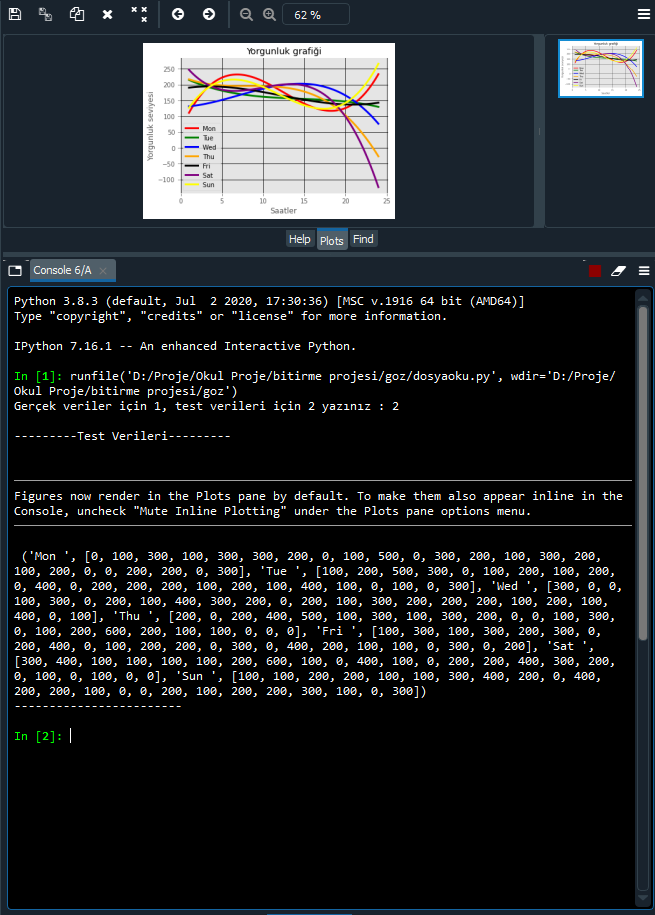
Aşağıda gördüğünüz grafik yukarıda bulunan grafiğin büyütülmüş halidir.



Bu grafik bizim gerçek verilerimiz kullanılarak kişinin yorgunluk seviyesini görmemizi sağlamaktadır.

15

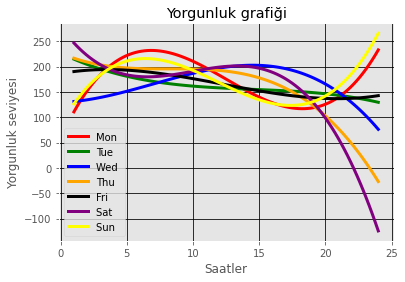
Eğer 2 yazmış olsaydık çıktımız aşağıdaki şekildeki gibi karşımıza gelecektir.



16

(Plots kısmının seçili olduğundan emin olunuz yoksa grafiğimiz gözükmeyecektir.)

Çıkan bu tahmini verilere aşağıdaki büyütülmüş görselden daha rahat bakabilirsiniz.



17

**Kaynak Kodlar**

**dosyaoku.py içerisindeki kodlar**

import pymongo

import numpy

from matplotlib import pyplot as plt

from matplotlib import style

import datetime

knt =input("Gerçek veriler için 1, test verileri için 2 yazınız : ")

f = open("tarihler.txt", "r")

saatler = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23] # saatler

haftanin\_günleri = ("Mon ", [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

                    "Tue ", [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

                    "Wed ", [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

                    "Thu ", [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

                    "Fri ", [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

                    "Sat ", [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

                    "Sun ", [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

                    )

hafta\_günü\_tut = ""

for i in f:

    göz\_kapama\_saati = int(i[-3:-1])

    if(hafta\_günü\_tut != i[0:3]):

        hafta\_günü\_tut = i[0:3]

    for guntk in range(0,14,2):

        if(i == str(haftanin\_günleri[guntk])+str(int(i[-3:-1]))+"\n"):

            haftanin\_günleri[guntk+1][göz\_kapama\_saati-1] = haftanin\_günleri[guntk+1][göz\_kapama\_saati-1] + 1

gstr = False

style.use('ggplot')

renkler = ["red","green","blue","orange","black","purple","yellow"]

for n in range(0,14,2):

    mymodel = numpy.poly1d(numpy.polyfit(saatler, haftanin\_günleri[n+1], 3))

    myline = numpy.linspace(1, 24, 100)

    if(knt == "1"):

        train\_x = saatler[:80]

        train\_y = haftanin\_günleri[n+1][:80] #gerçek veriler

        plt.scatter(train\_x, train\_y)  # gerçek verilerin grafiğini gösterir

    elif(knt == "2"):

        test\_x = saatler[80:]

        test\_y = haftanin\_günleri[n+1][80:] # eğitilmiş veriler

        plt.scatter(test\_x, test\_y)      # eğitilmiş verilerin rafiğini gösterir

    else:

        print("Geçersiz değer")

        gstr = True

        break

    plt.plot(myline,mymodel(myline),renkler[int(n/2)],label=haftanin\_günleri[n], linewidth=3)

if(gstr == False):

    print()

    if(knt == "1"):

        print("---------Gerçek Veriler---------")

        print(haftanin\_günleri)

        print("------------------------")

    else:

        print("---------Test Verileri---------")

    plt.title('Yorgunluk grafiği')

    plt.ylabel('Yorgunluk seviyesi')

    plt.xlabel('Saatler')

    plt.legend()

    plt.grid(True,color='k')

    plt.show()

**goz.py içerisindeki kodlar**

import numpy as np

import cv2

import time

import datetime

import winsound

yuz\_algila = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

göz = cv2.CascadeClassifier('haarcascade\_eye\_tree\_eyeglasses.xml')

kapalı\_göz = False

göz\_kapali = 0

vd\_cap = cv2.VideoCapture(0)

ret,img = vd\_cap.read()

f = open("tarihler.txt", "a")

while(ret):

    time.sleep(0.1) # saniyenin 10 da birinde kontrol işlemi yapar

    ret,img = vd\_cap.read()

    gri = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)

    gri = cv2.bilateralFilter(gri,5,1,1)

    yüzler = yuz\_algila.detectMultiScale(gri, 1.3, 5,minSize=(200,200))

    cv2.putText(img,"cikis icin c ya basiniz",(40,450),cv2.FONT\_HERSHEY\_PLAIN, 3, (0,0,255),2)

    if(len(yüzler)>0):

        for (x,y,w,h) in yüzler:

            img = cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),2)

            yüz\_degeri = gri[y:y+h,x:x+w]

            gözler = göz.detectMultiScale(yüz\_degeri,1.3,5,minSize=(50,50))

            if(len(gözler)>=2):

                    print("---------Gozler algılandi-------------")

            else:

                cv2.putText(img, "gozler algilanamadi", (90,50), cv2.FONT\_HERSHEY\_PLAIN, 3,(0,0,255),2)

                print("gözler kapali")

                göz\_kapali = göz\_kapali + 1

                if(göz\_kapali == 30):

                    print("----------Veri Tabanına İşlendi----------")

                    tarama\_hızı = 2500

                    tarama\_süresi = 1000

                    winsound.Beep(tarama\_hızı, tarama\_süresi)

                    tarih     = datetime.datetime.now().strftime("%A");

                    saat = datetime.datetime.now().strftime("%H");

                    f.write(str(tarih)[0:3]+ " " +str(saat) + "\n")

                    print ("Tarih : " ,tarih);

                    print ("Tarih Saat :" ,saat);

                    göz\_kapali = 0

    else:

        cv2.putText(img,"yuz algilanamadi",(110,50),cv2.FONT\_HERSHEY\_PLAIN, 3, (0,255,0),2)

    cv2.imshow('Goz Algila',img)

    cikma = cv2.waitKey(1)

    if(cikma==ord('c')):

        break

f.close()

vd\_cap.release()

cv2.destroyAllWindows()

21