

ATATÜRK ÜNİVERSİTESİ

Mühendislik Fakültesi

Bilgisayar Mühendisliği

2021-2022 Yaz Okulu

YAPAY ZEKAYA GİRİŞ

Dr. Öğr. Üyesi Mete YAĞANOĞLU

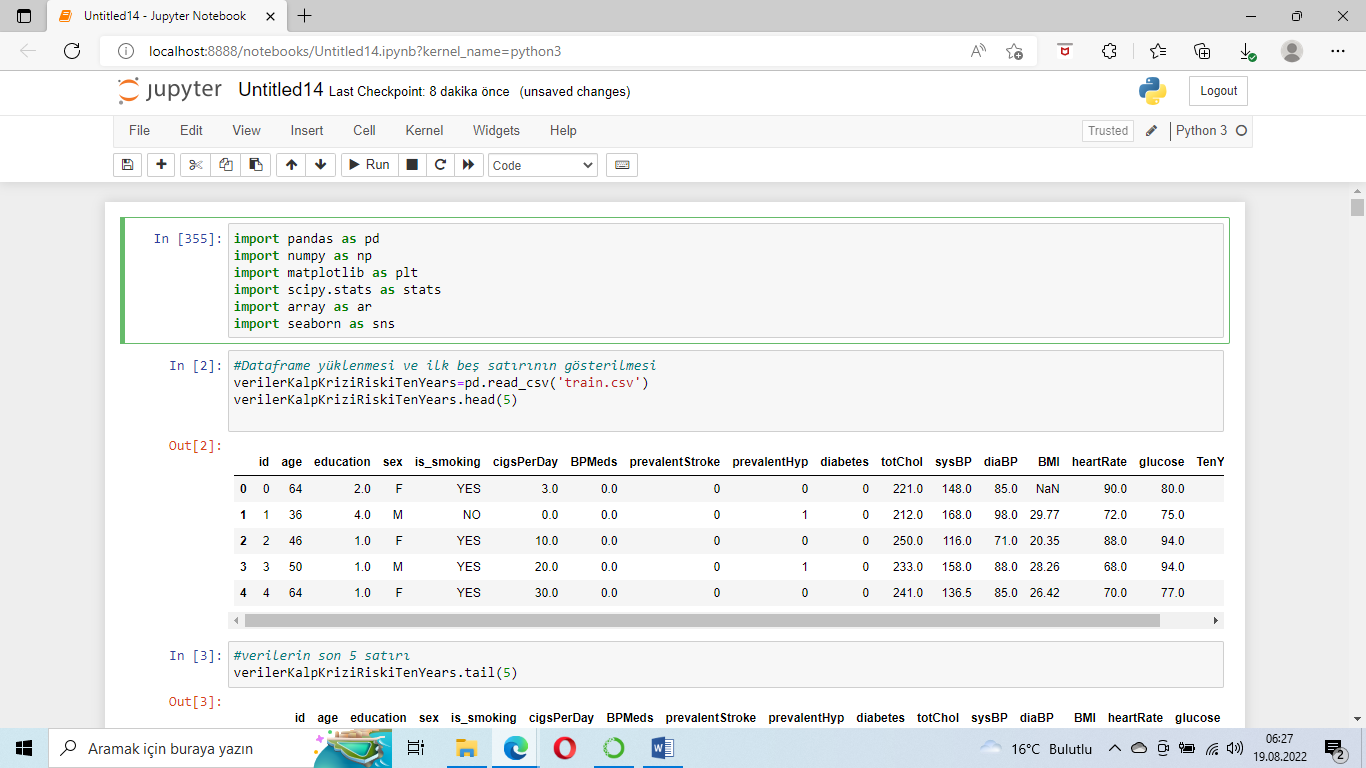
Kişinin On Sene İçerisinde Kalp Krizi Riskini

Hesaplayan Yapay Zeka

Rüveyda Havva SEVİNÇ

170707065

Kütüphane İmport Edilmesi

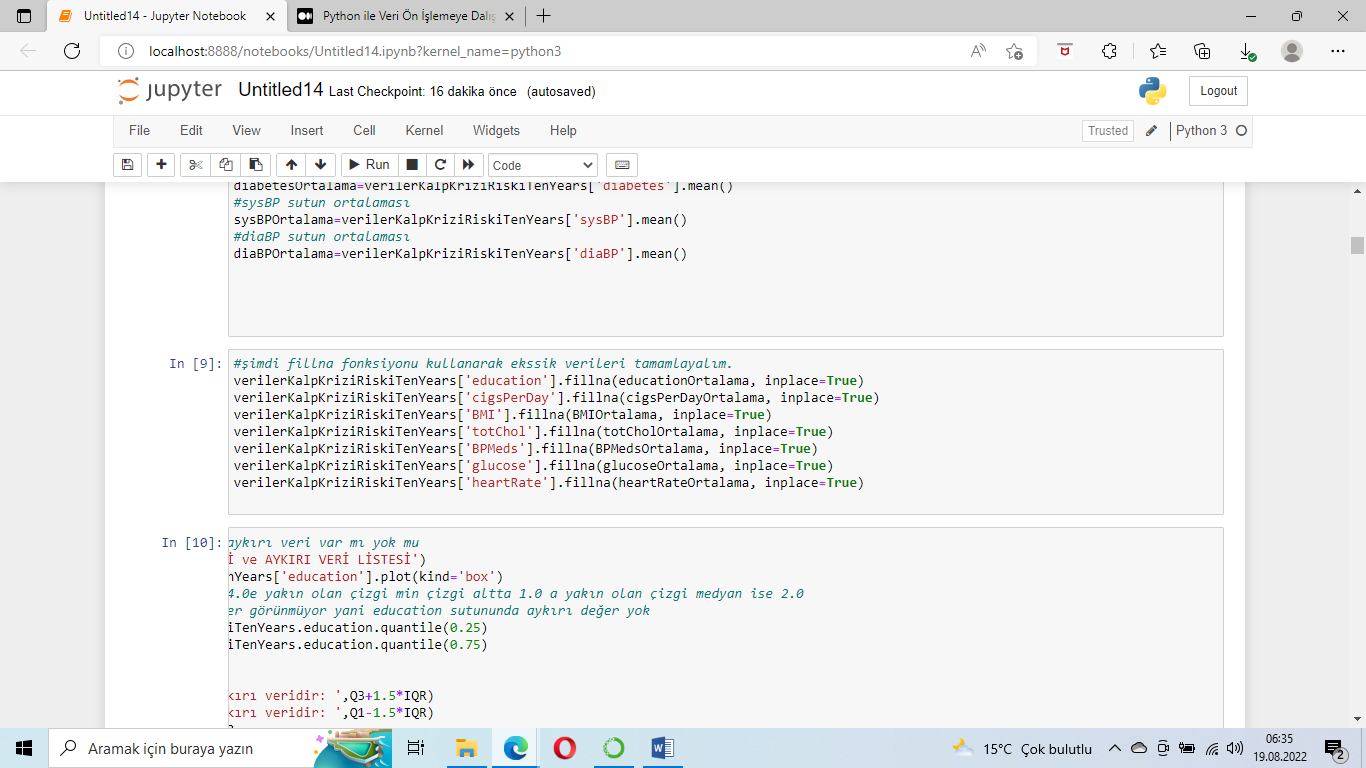


Dataframe işlemleri için Pandas kütüphanesi, arada kullandığım array işlemleri için Numpy kütüphanesini, grafik çizimleri için Matplotlib ve Seaborn kütüphanelerini import ettim.

Sonrasında eğitim verisi olarak kullanacağım csv dosyasını pandas kütüphane işlemlerinden olan **read\_cvs** komutu ile okuyup bir veriçerçevesinde tuttum.

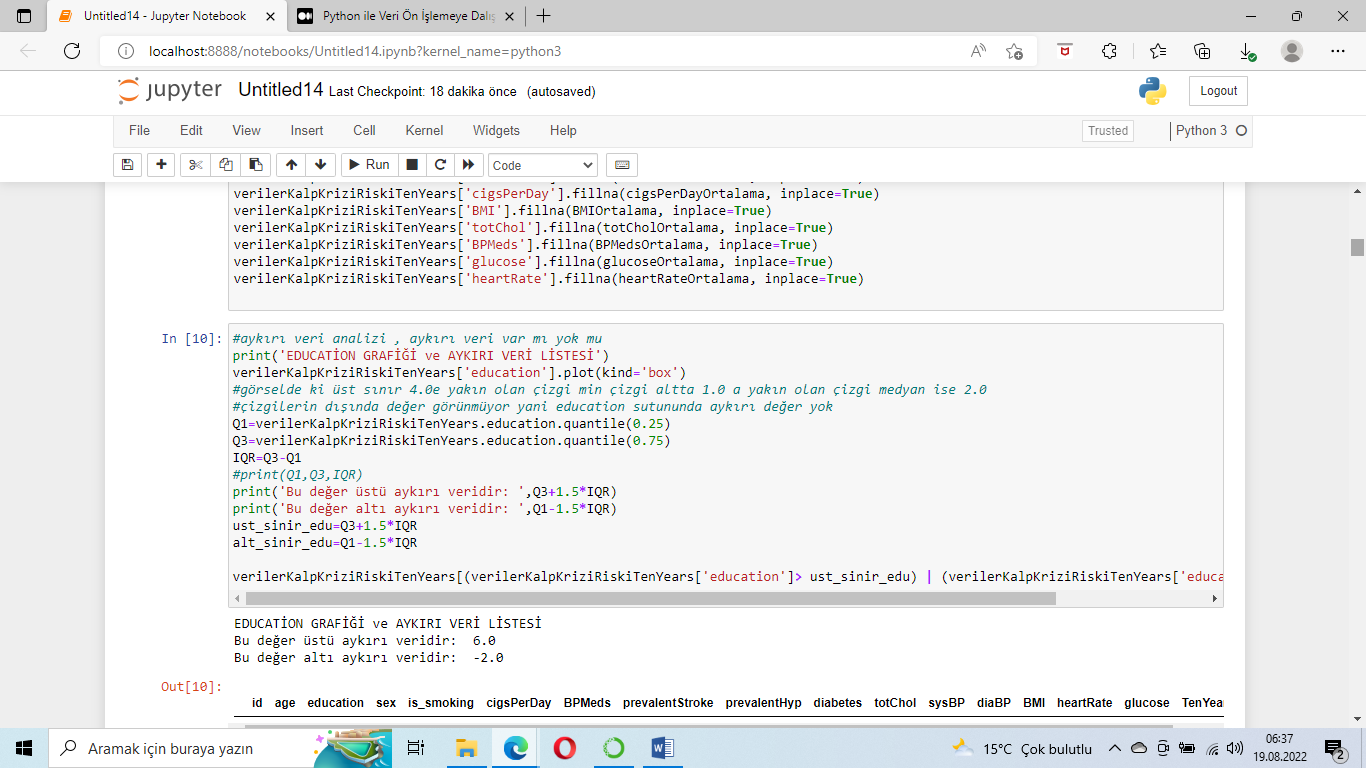
Dataframe’yi kontrol etmek için head,tail komutu ile ilk ve son satırları gösterdim.

Veri Ön İşleme Adımları

Eksik Verilerin Ortalama ile Doldurulması:

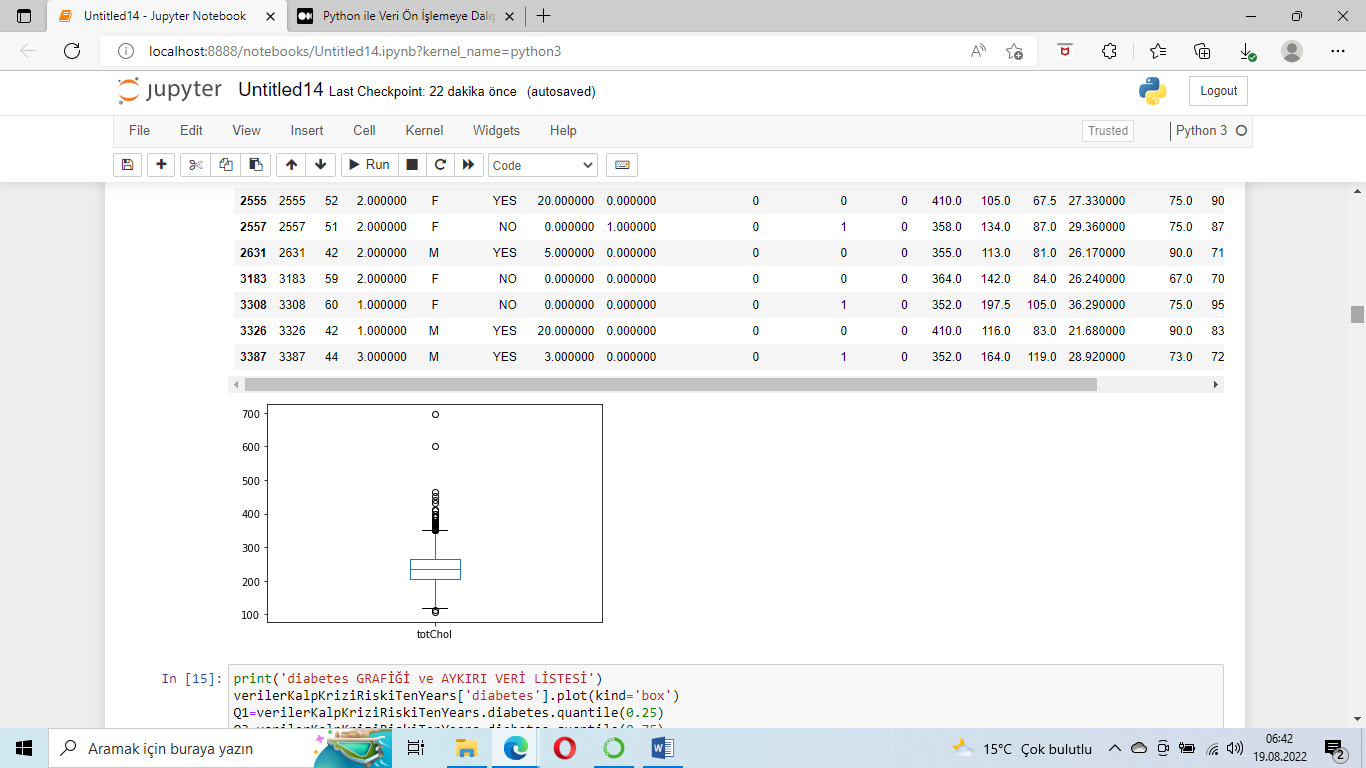
İsnull() komutu kullanarak dataframe de değer olup olmamasına göre false ya da true olarak boolen bir değer döndürdüm.False değer alan index dolu olduğunu gösterir.Sonrasında **isnull().sum()** komutu ile her sutünda toplam kaç değerin eksik olduğunu yazdırdım. Eksik verileri her sutünün ortalamasını bulup, **fillna()** komutu ile tek tek doldurdum. (Diğer bir yöntemi ise Simpleİmpute fonksiyonu ile de NaN değerler belirlenen (mean,std) gibi strategy özelliğinde belirtilene göre doldurulabilir.)

Aykırı Verilerin Gösterilip Ortalama ile Doldurulması



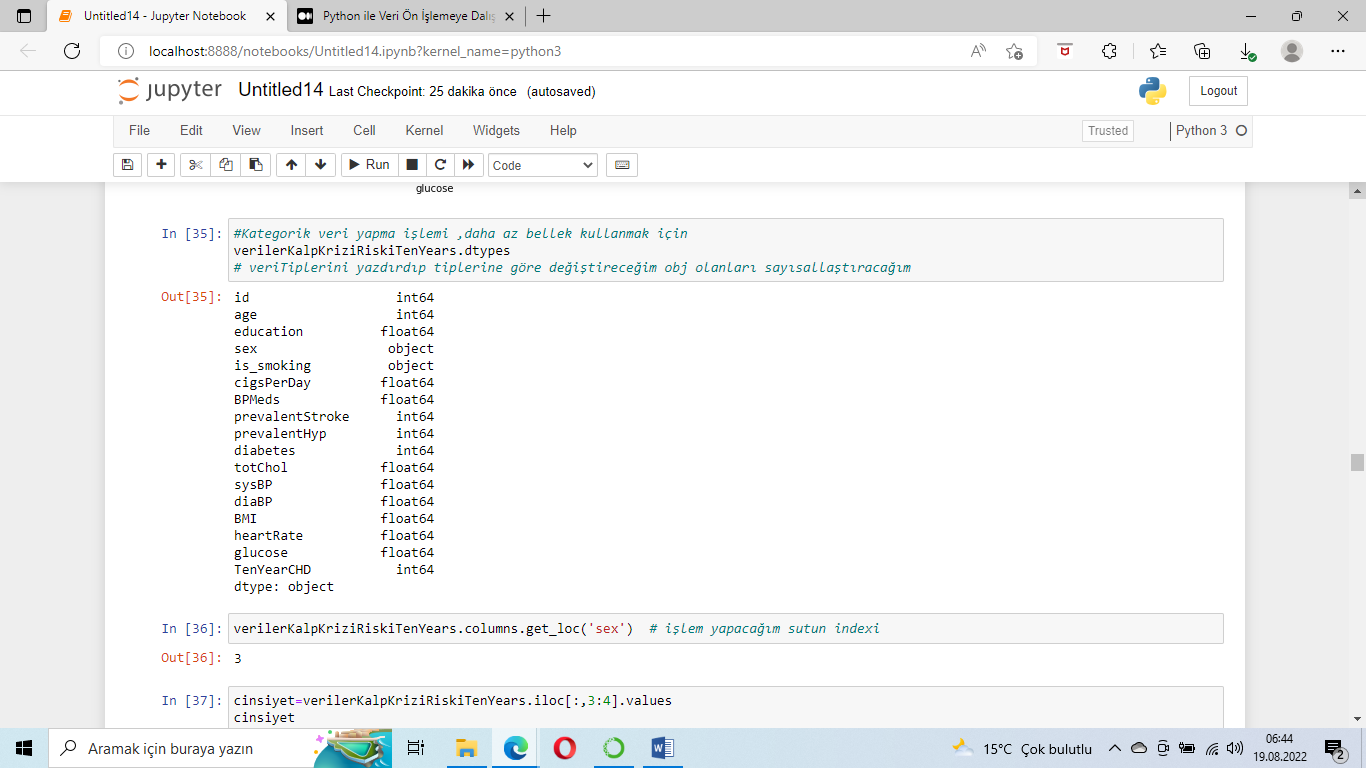
Çeyrekler açıklığı yöntemini kullanarak aykırı değer alt sınır ve üst sınırlar değerlerini buldum.Dataframe de mask işlemi yaparak o alt değer altında ya da üst değer üstünde olan satırların listesini yazıp, görselleştirilmesini **box-plot** ile görselleştirdim.

Aşağıdaki görselde bulunan yuvarlaklar aykırı değerleri gösterir.

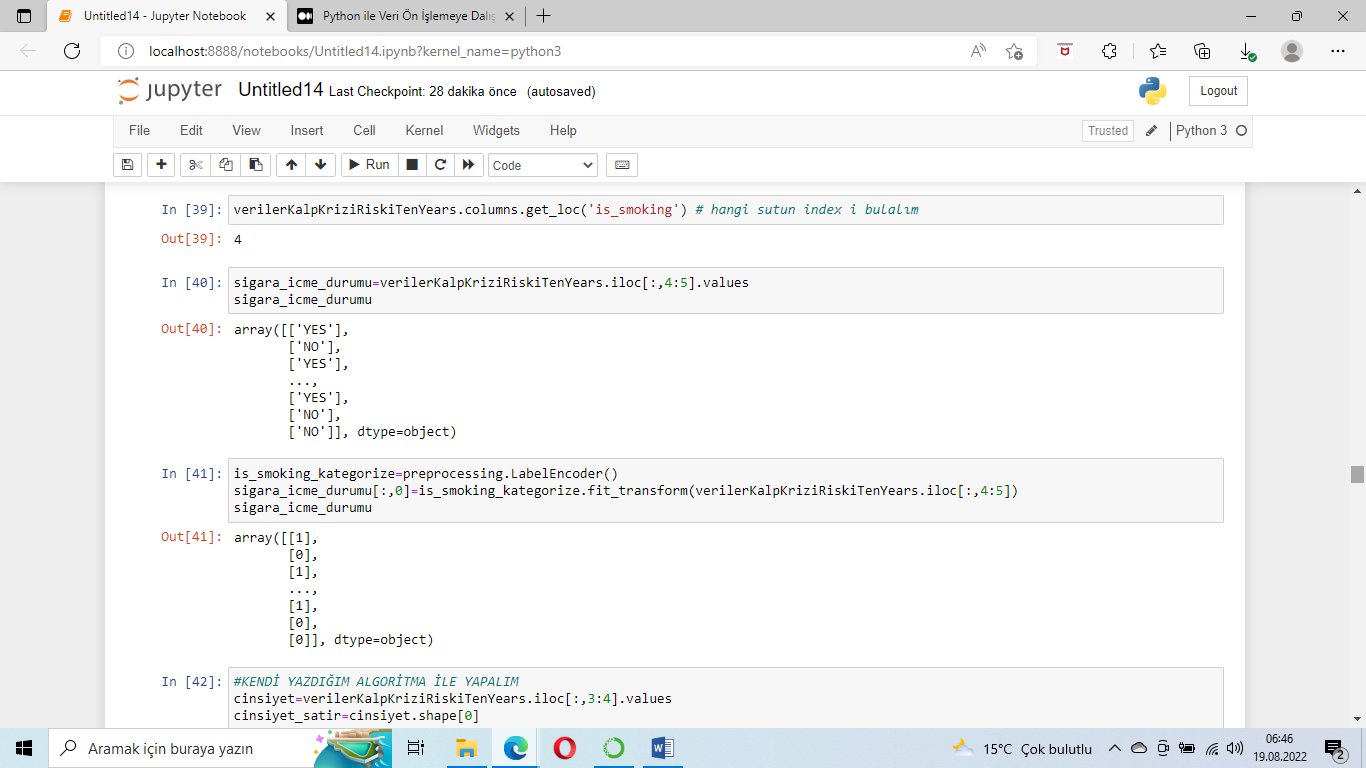


Sonrasında aykırı verileri ortadan kaldırmak için aykırı veri sutünündaki değerleri çeyrek açıklık değerlerine eşitledim (alt sınır ya da üst sınır)

Kategorik Verilerin Sayısallaştırılması

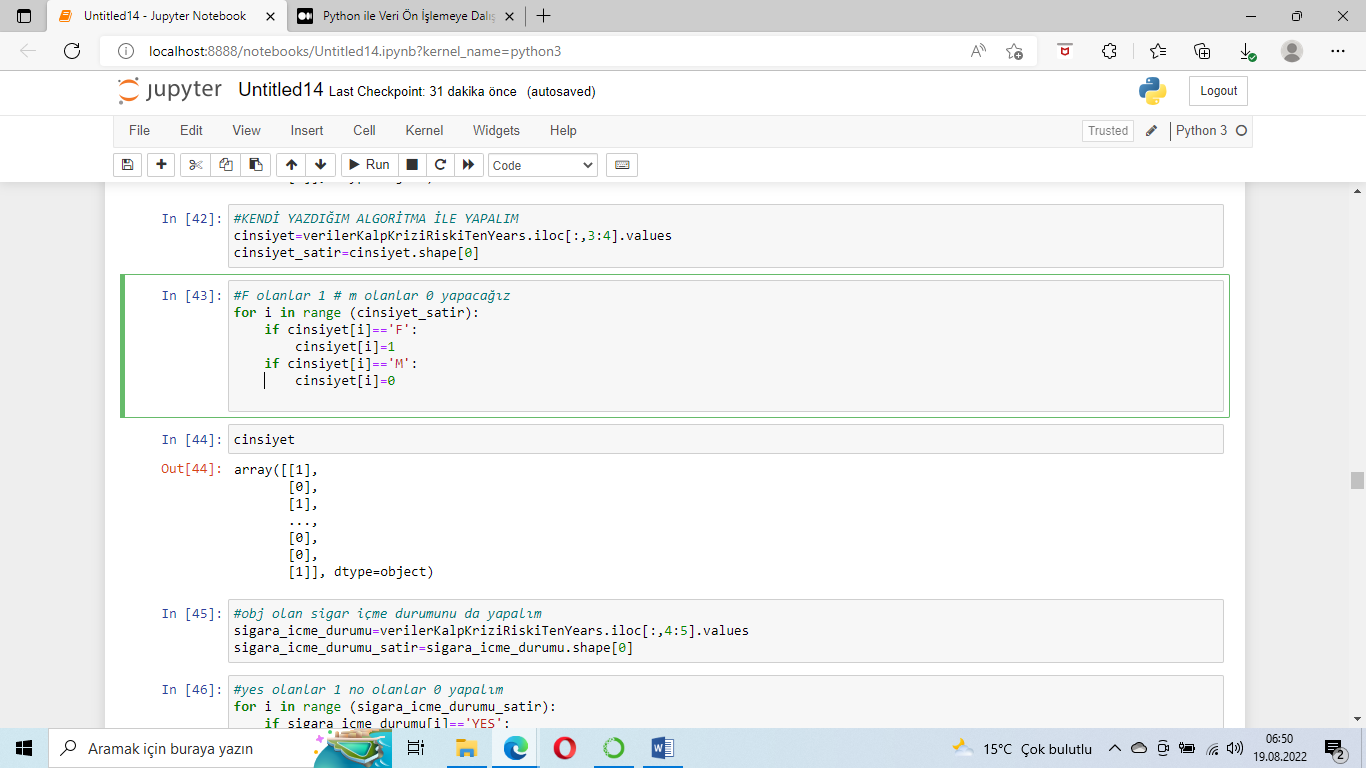
**Dtypes** komutu kullanarak sutünların veri tiplerini gördük

Object türünde olanları sayısallaştırmamız gerekiyor.

**from sklearn import preprocessing** kütüphanesinden işlemi ekledikten sonra Label Encoder işlemi yaptım

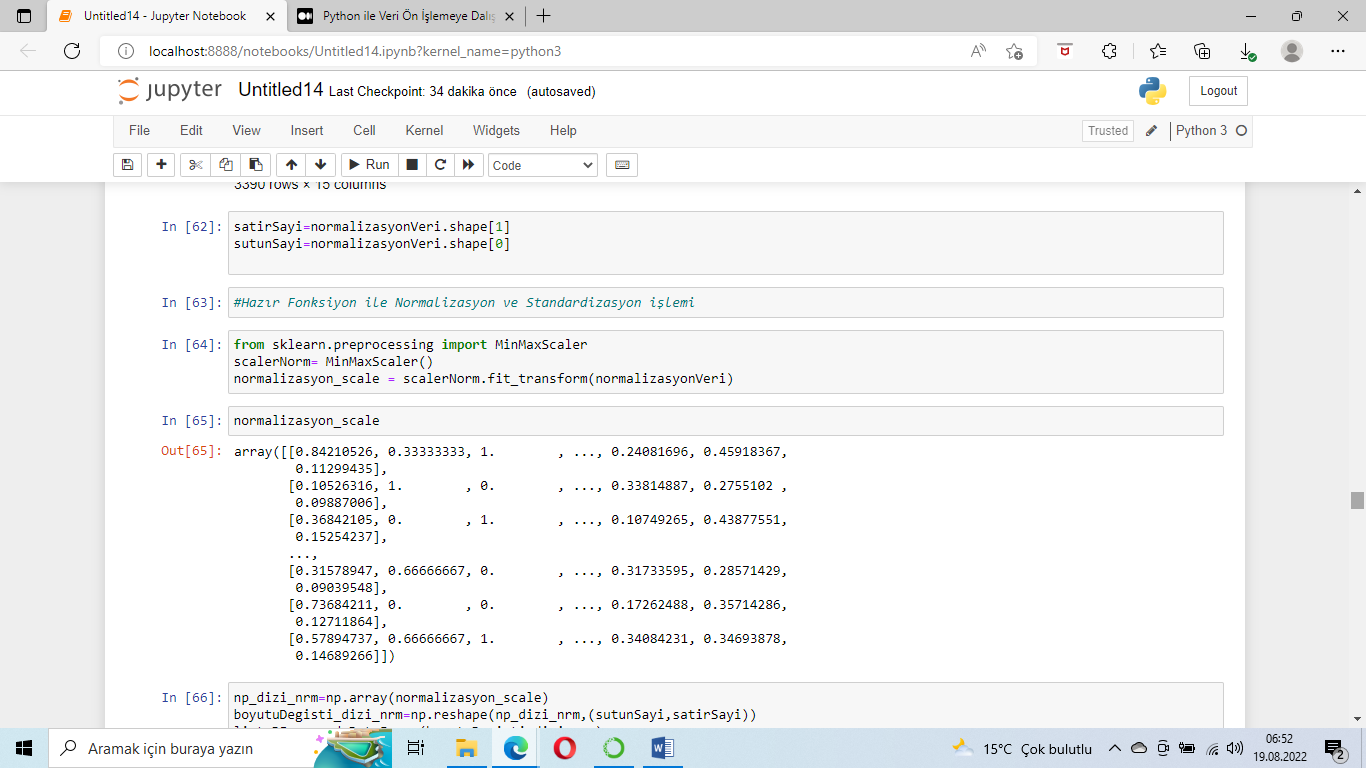
Bunu kendi yazdığım fonksiyona göre

Ekrandaki gibi Label Encoder yerine object typenin olduğu sutünün bütün satırlarını inceleyerek şarta göre index değerini değiştirdim.

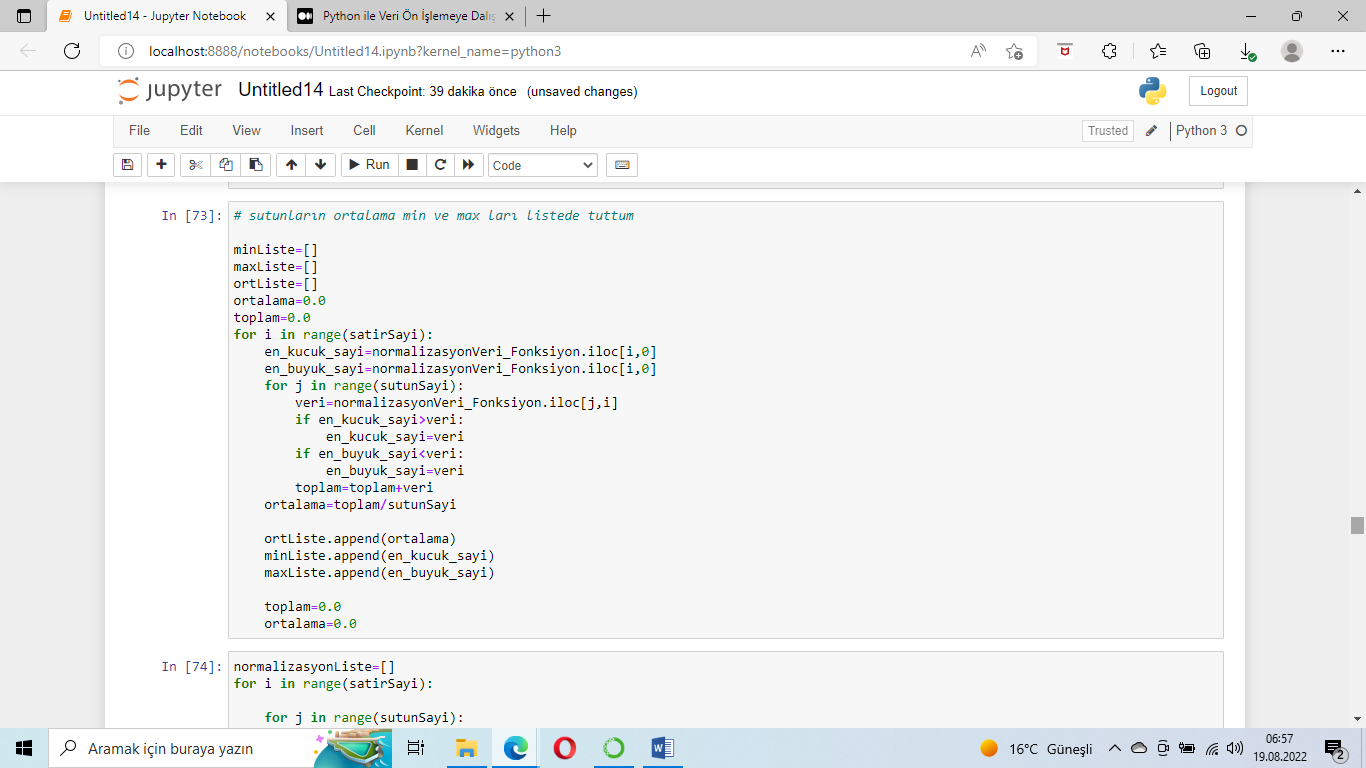
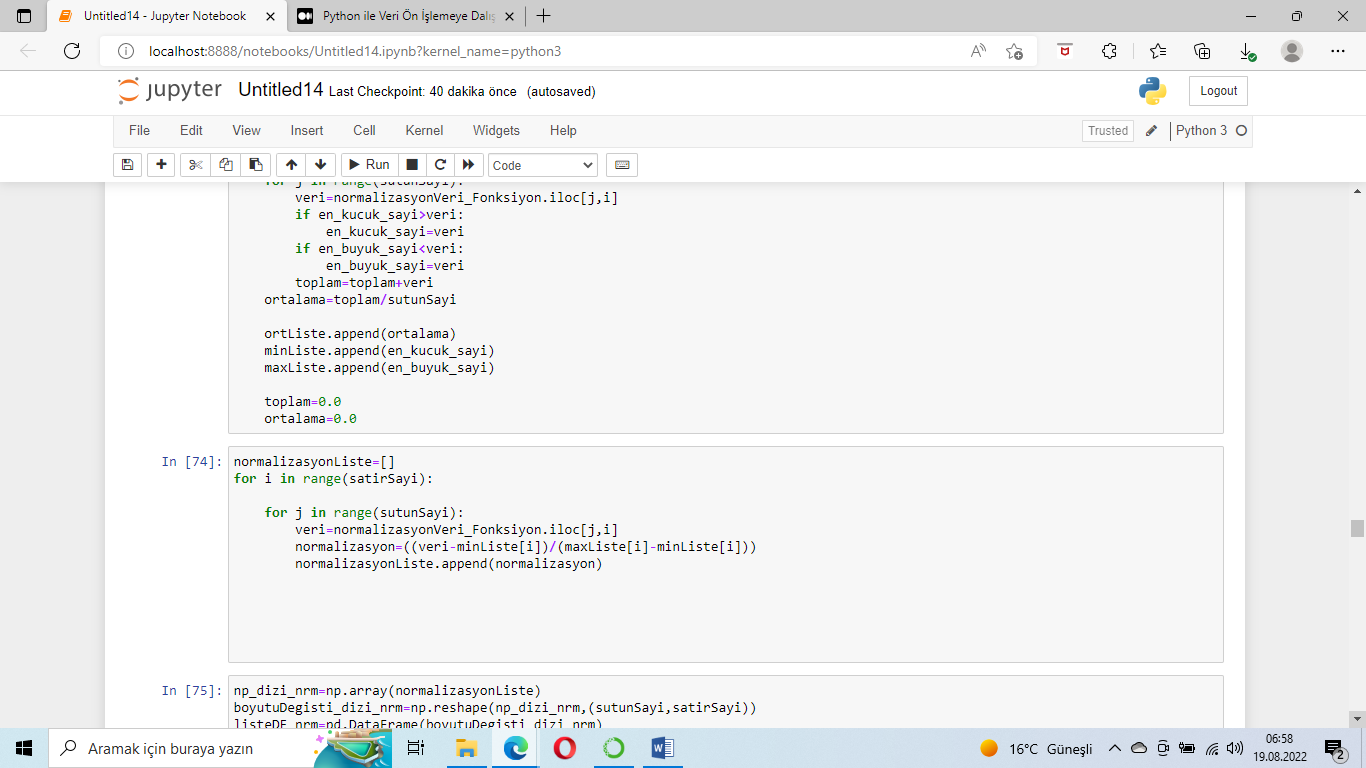


Normalizasyon ve Standardizasyon İşlemi

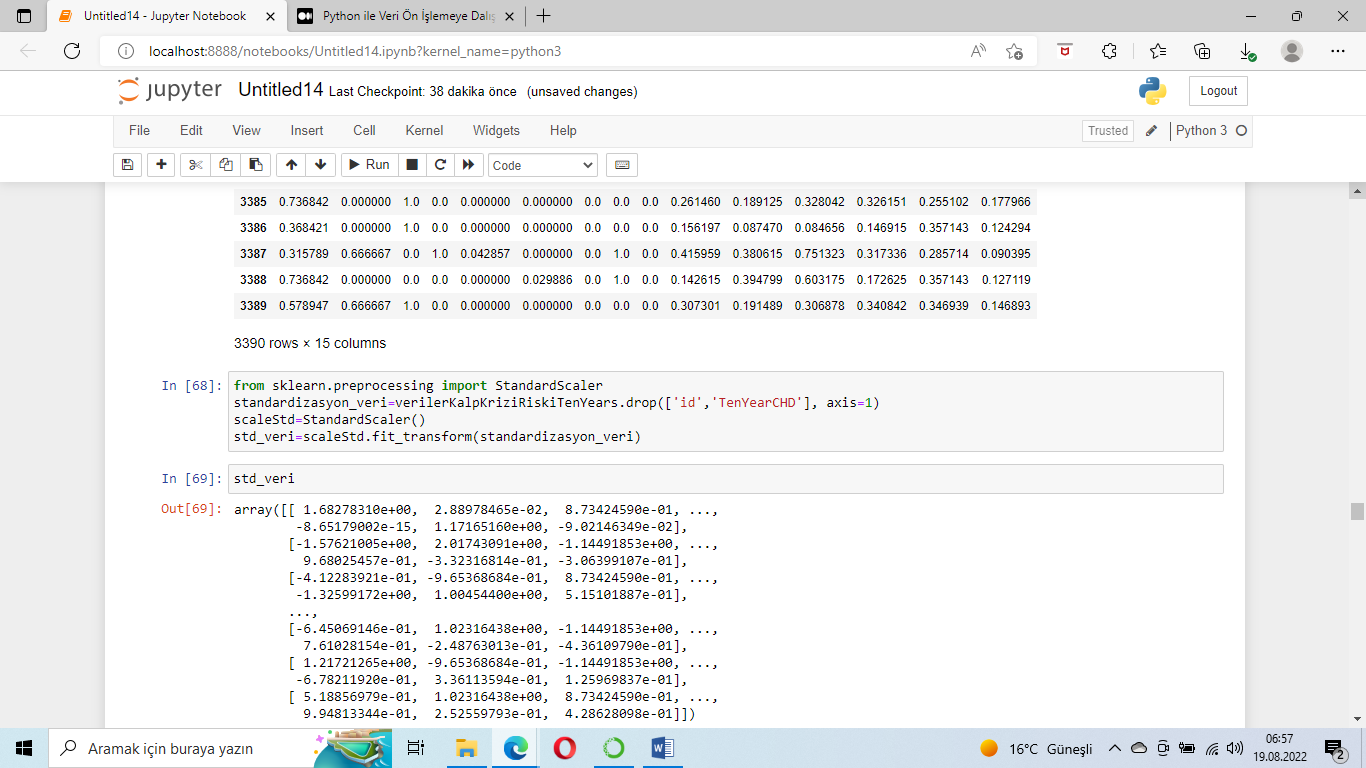
Normalizasyon

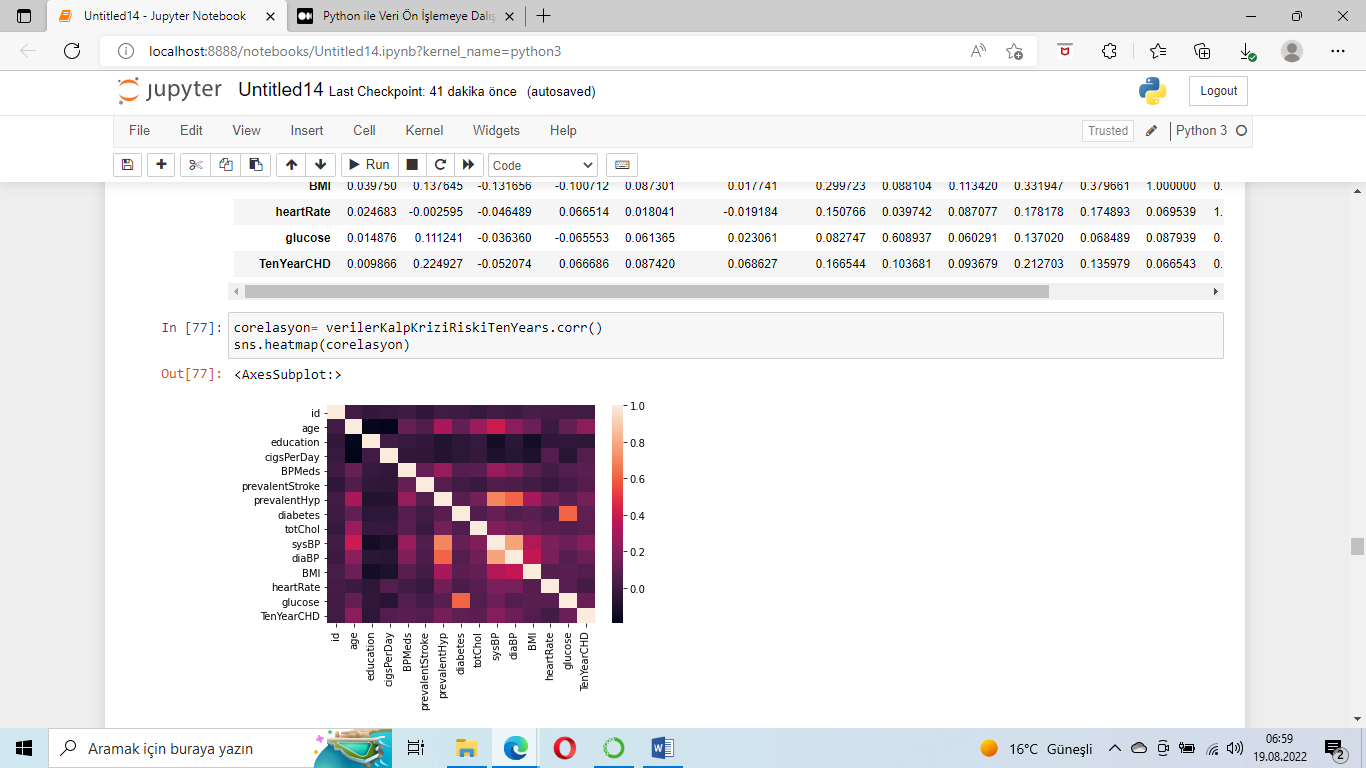


Aynı hazır fonksiyonun işlemini gerçekleştirdim. Normalizasyon işlemi X-Xmin/ Xmax-Xmin şeklinde gerçekleştiği için her sütünün en büyük ve en küçük değerlerini bulup döngü değerlerin hepsini sırasıyla listede tuttum ve .loc komutu kullanarak her değer içinde normalizasyon işlemi gerçekleştirdim. Bunları yaparken list tipinde tuttum.



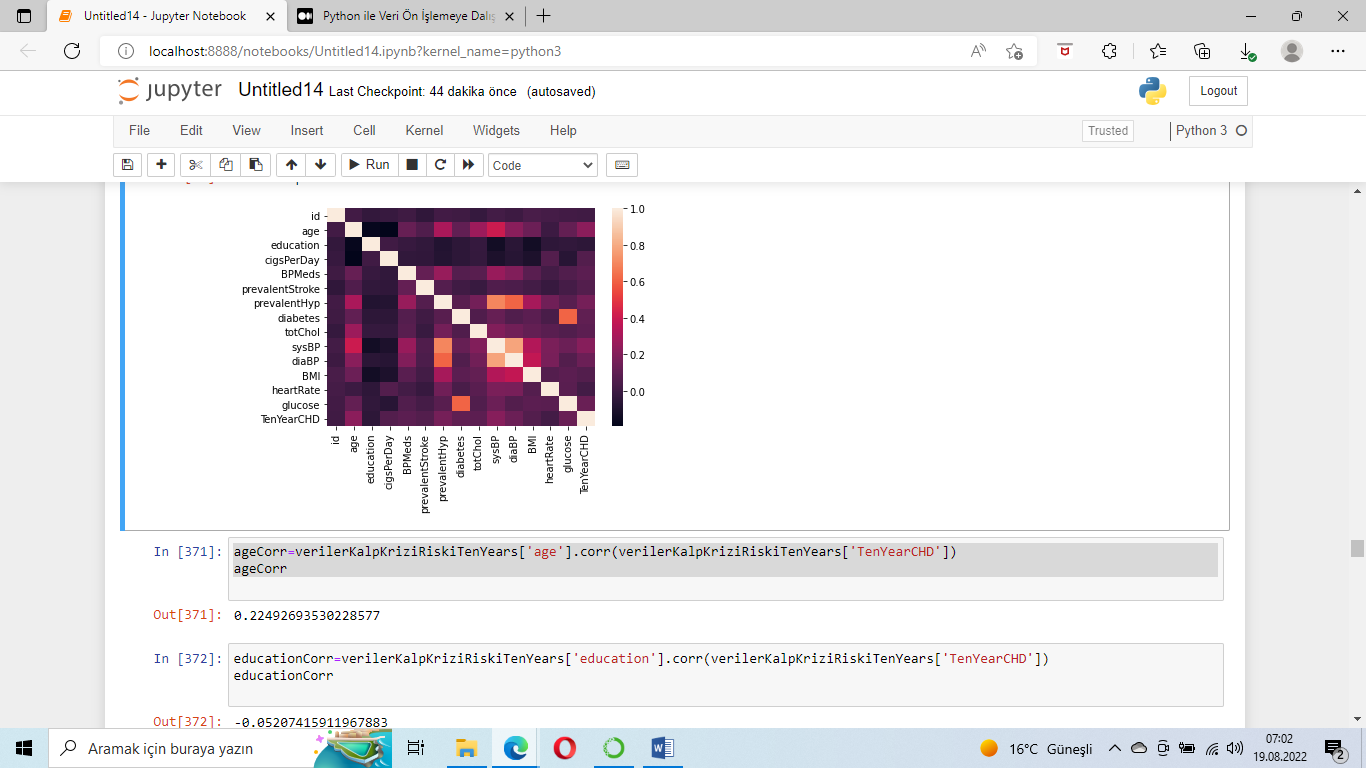
Standardizasyon



Korelasyon

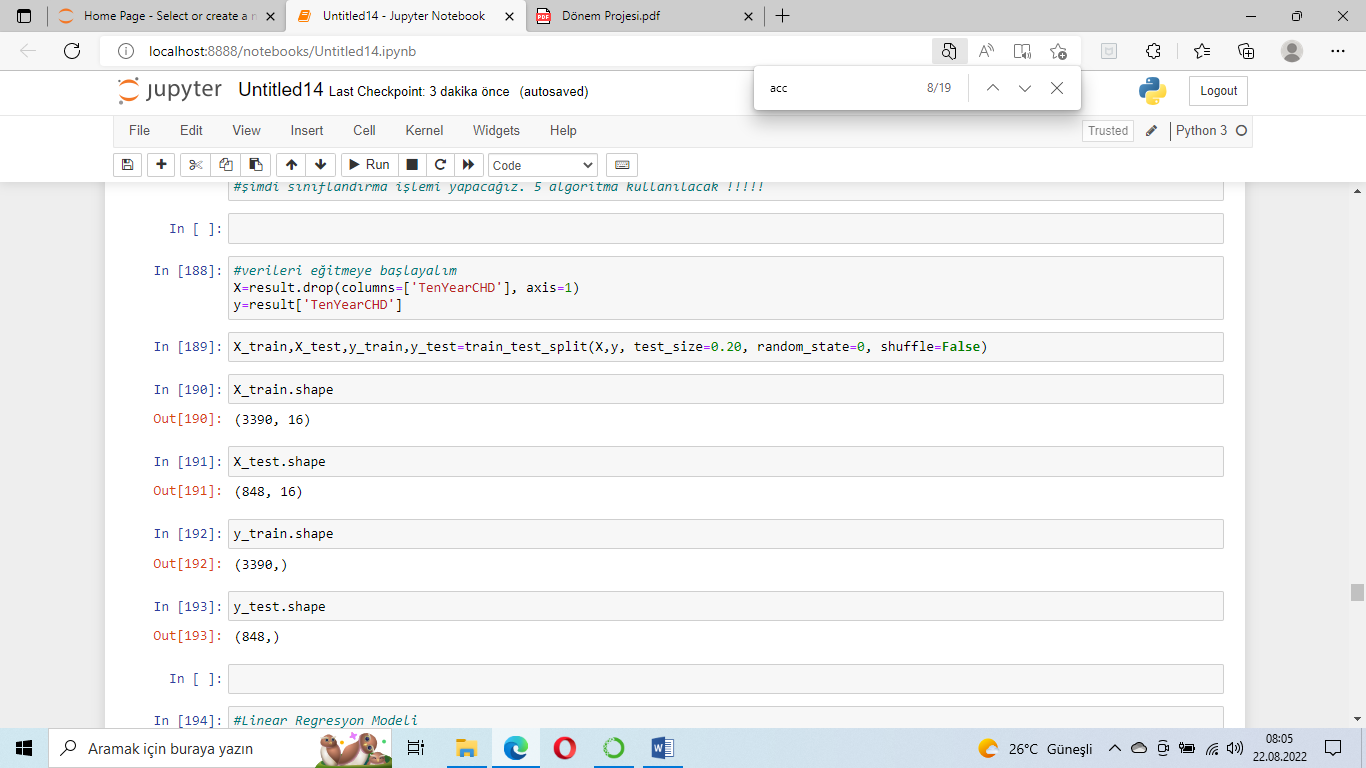
Dataframedeki özelliklerin birbirleri ile ilişkilerinin, birbirlerine etkilerinin ne olduğunu öğrenmek için .corr komutu ile bulup bunları görselleştirdim.

Sonrasında her özelliğin TenYearCHD ile ne kadar alakalı olduğunu görmek için bütün sutünları örneğin;



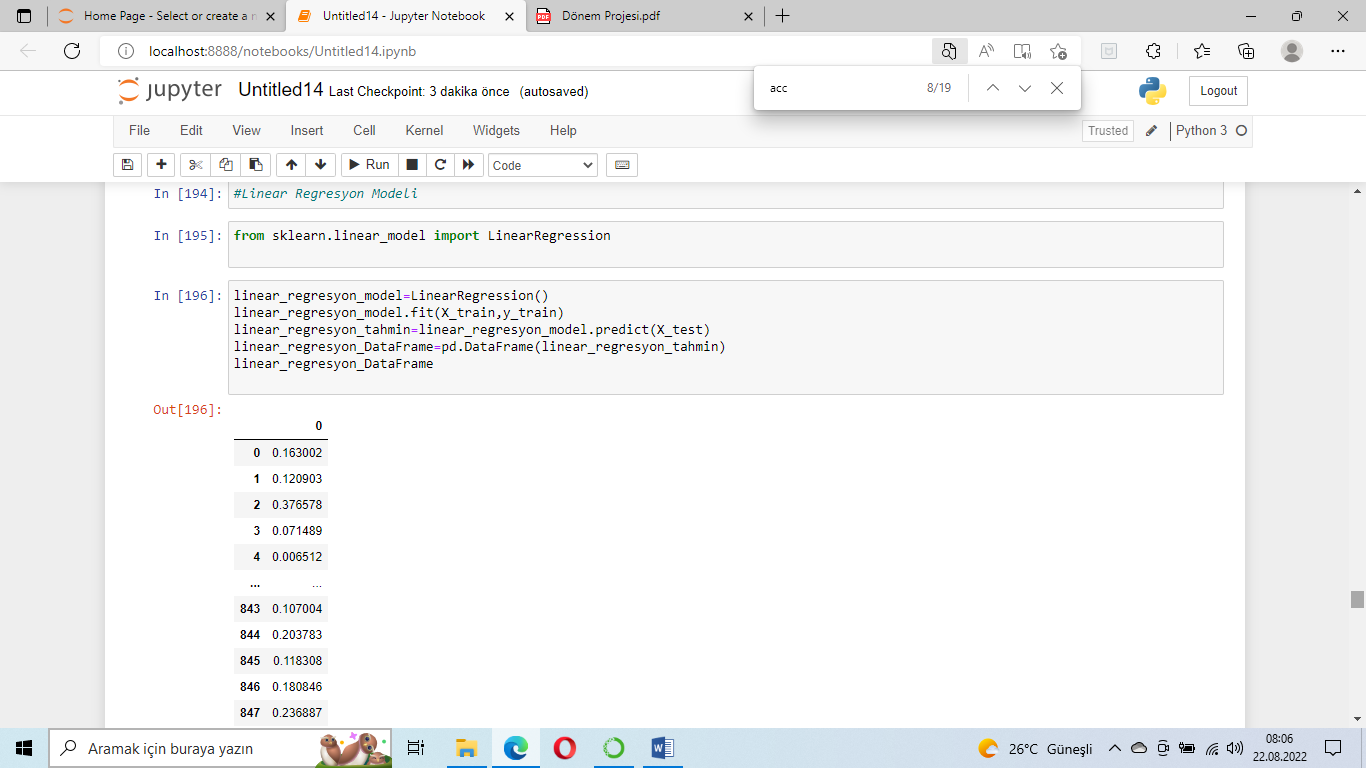
Bunu yapma nedenim hangi özelliklerin kalp krizi riskini daha fazla etkilediğini görmekti. Bulduğum değerleri bir dict te tutup sıralayıp en önemlilerinin hangisi olduğunu gösterdim.

Daha sonra test edeceğim verilerin olduğu datayı okuyup orda da yukarıda yazdığım işlemleri gerçekleştirdim. Sonrasında test verisi ve train verisini birleştirip



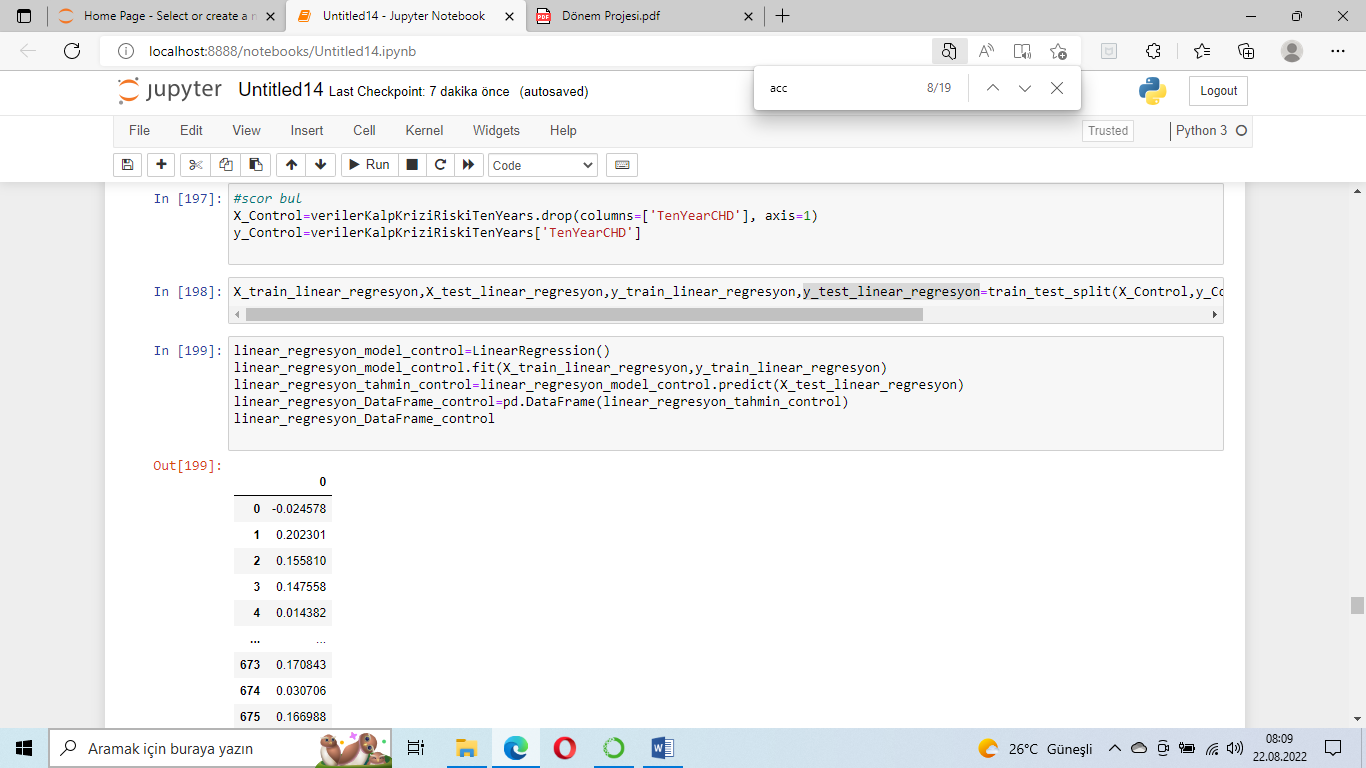
Boyutlarını kontrol ettim shuffle özelliğini false yapma nedenim verileri ayırırken sırasıyla ayırsın.

İlk olarak kullandığım model Linear Regresyon Modeli



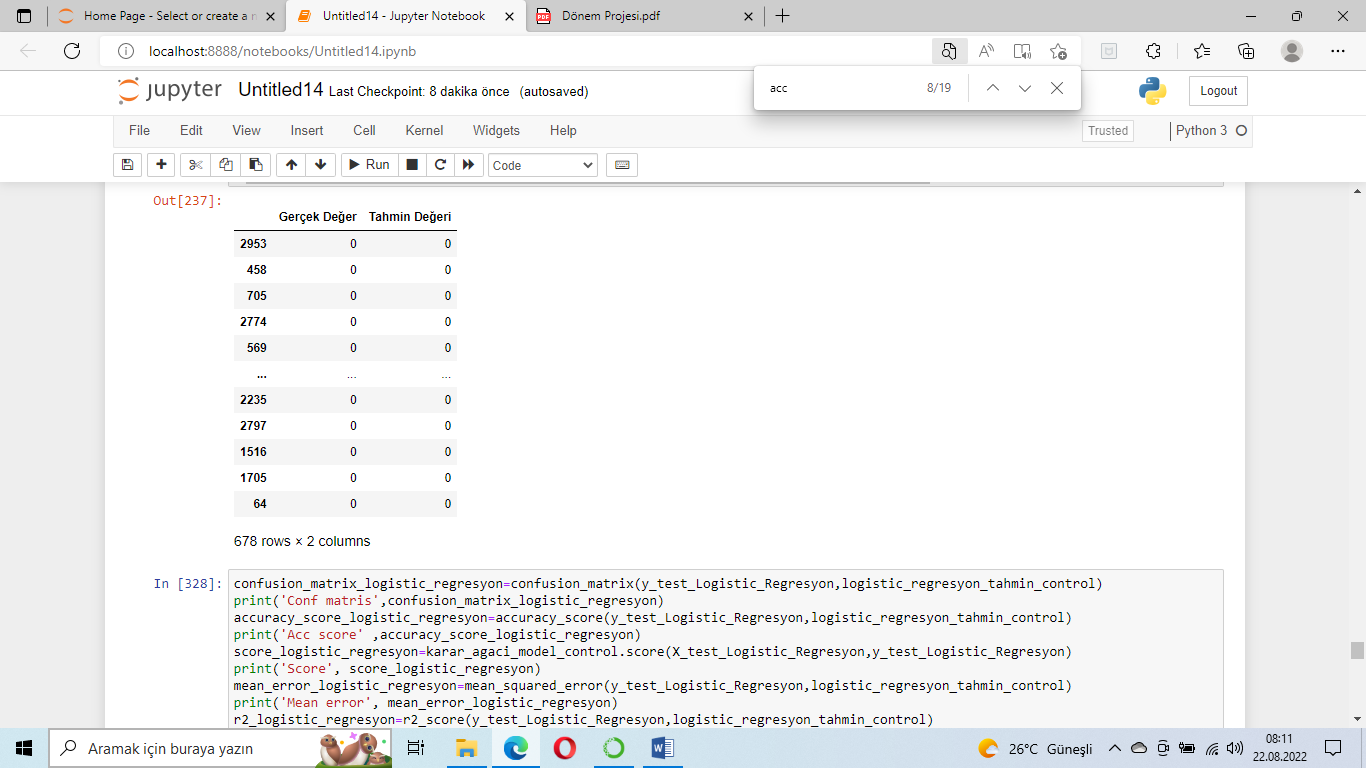
Linear regresyon için gerekli olan kütüphane ve fonksiyonu import ettim. Sonrasında ayırdığım verileri fonksiyona fit fonksiyonu ile eğittim.Daha sonra tahminleri dataframe de tutup yazdırdım.

Bu şekilde acc score,r2 benzeri hesaplamaları bulamayacaktım. Çünkü test verisinde TenYearCHD sütununda değer yoktu.Ama eğitirken kullandığım veriyi tekrar test ve train olarak ayırırsam (çünkü output değerleri var ve kıyaslama yapabilirim.) doğruluk değeri hata değeri gibi hesaplamaları gerçekleştirdim.



Daha sonra aynı işlemleri

1-Linear Regresyon

2-Karar Ağacı

3-Logistic Regresyon

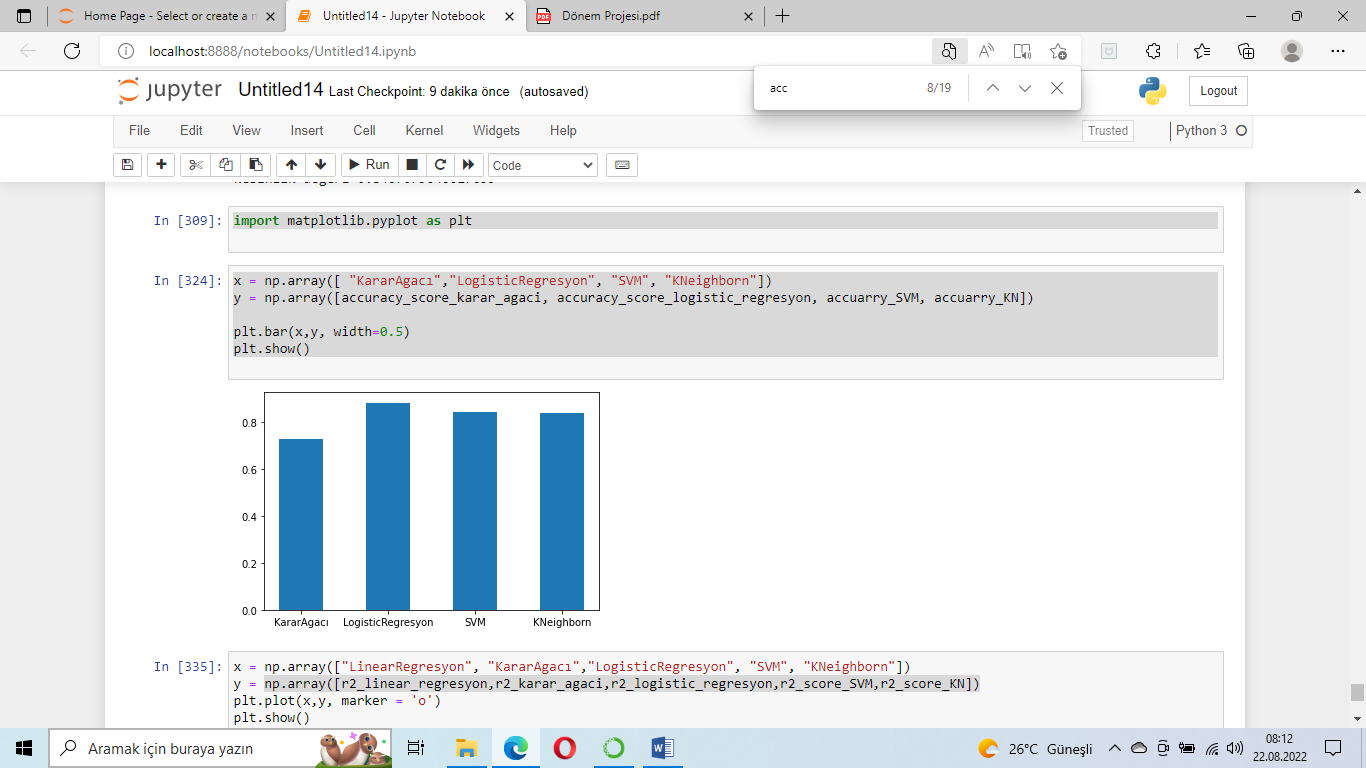
4-Neighborn

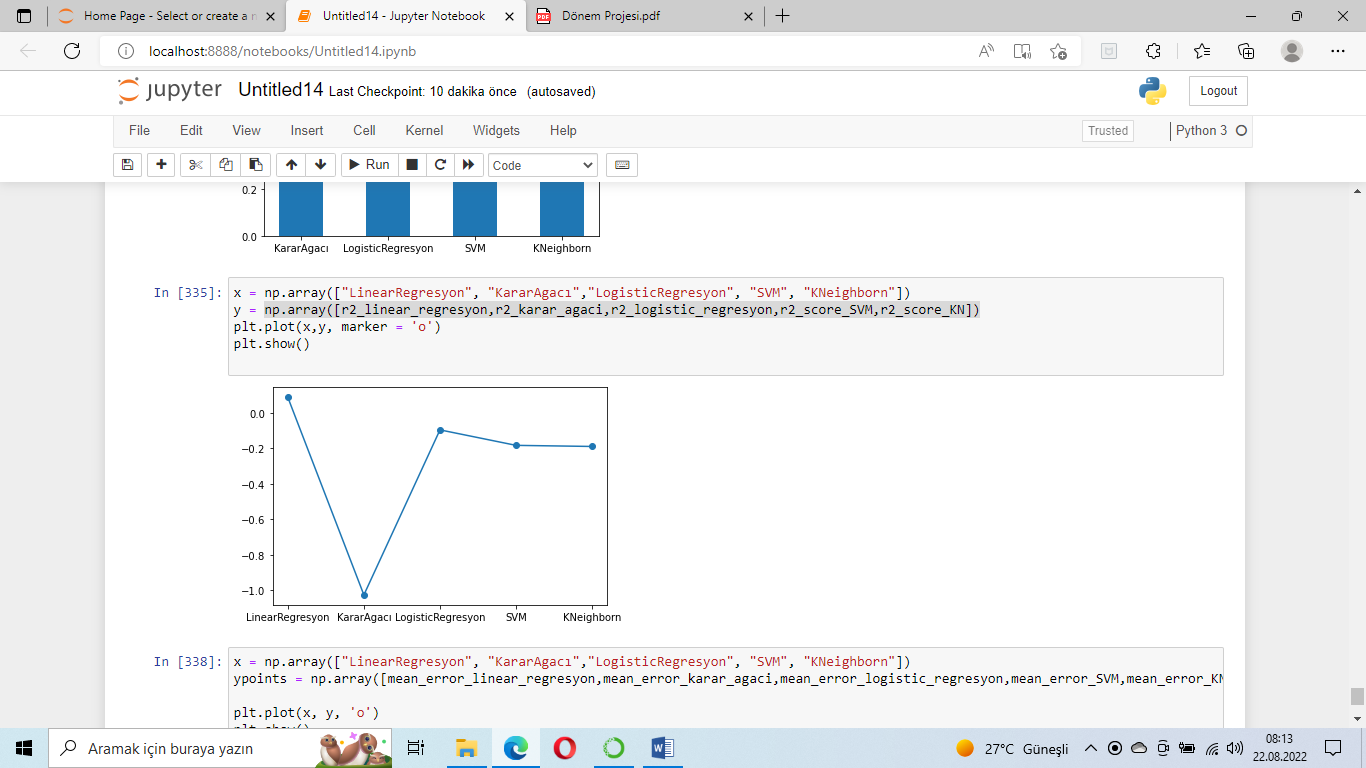
5-SVM

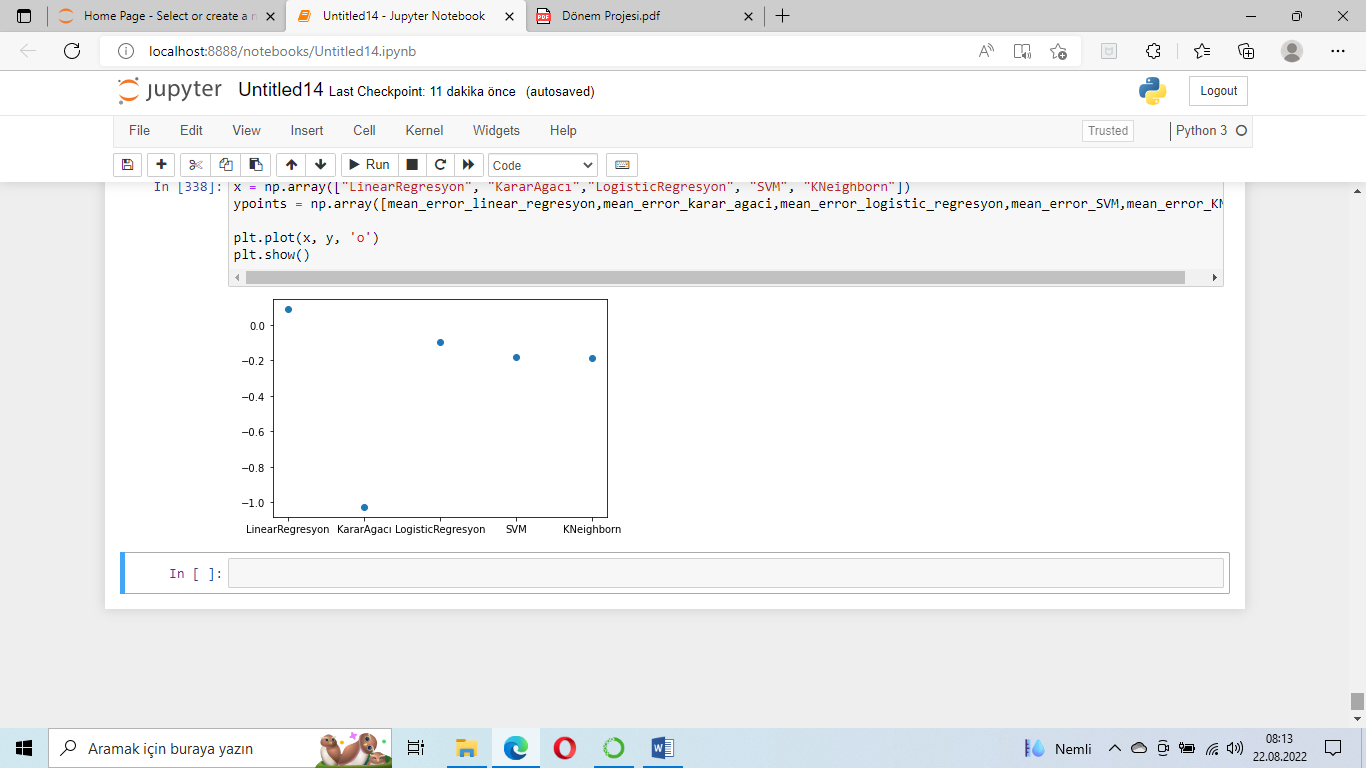
Sınıflandırmaları için gerçekleştirdim.

Tahmin örnek çıktılarından bir örnek

Daha sonra acc skorları, r2 değerleri ve hata değerlerine göre bunları görselleştirdim







Sonuç olarak kullandığım 5 sınıflandırmada bu problem doğruluk değeri en yüksek olan (benim kullandığım 5 i için) Logistic Regresyon oldu.