**Bilgisayar Uygulamaları Projesi Raporu**

**Ad Soyad:** Alperen ALP

**Numara:** 190508019

**Konu:** Kalp hastalarının bulunduğu bir veri setinin incelenmesi, görselleştirmelerin yapılıp analiz edilmesi, regresyon modeli oluşturulup bir hastanın belirli özelliklerine dayanarak, o kişinin kalp hastalığı olup olmadığının tahmin edilmesi.

# **Çözüm Aşamaları:**

* Veri setindeki özellikleri (features) tespit etme
* Veri setindeki eksik verilerin tespiti ve düzeltilmesi
* Veri setini analiz etme ve görselleştirme
* Veri setini ön işleyerek Kategorik verileri One-Hot Encoding ile numerik veriye çevirme
* Regresyon yöntemlerinden birini kullanarak model oluşturma
* Sonuç aşamasında modelin tahmin başarım oranını hesaplama

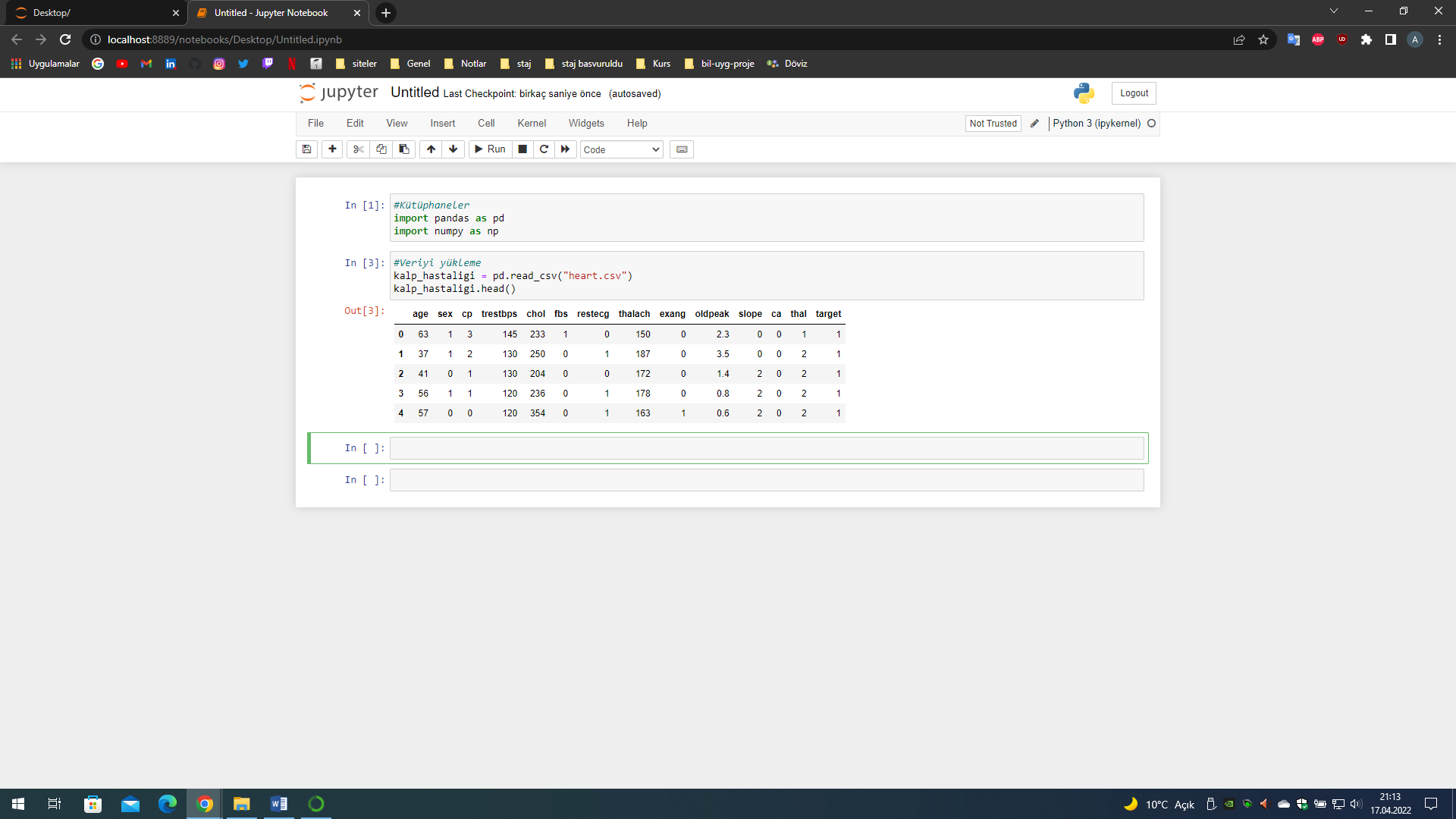
# **Özellikler (Features):** Veri setinde 14 adet özellik bulunmaktadır.

* **age (yaş):** Kişinin yıl cinsinden yaşı
* **sex (cinsiyet):** Kişinin cinsiyeti (1 = erkek, 0 = kadın)
* **cp (göğüs ağrısı tipi):** Değer 0:asemptomatik Değer 1: atipik angina, Değer 2: anjinal olmayan ağrı, Değer 3: tipik angina
* **trestbps (kan basıncı):** Kişinin istirahat kan basıncı (mm Hg)
* **chol (kolesterol):** Kişinin mg/dl cinsinden kolesterol ölçümü
* **fbs (kan şekeri):** Kişinin açlık kan şekeri (> 120 mg/dl, 1=doğru; 0=yanlış)
* **restecg (elektrokardiyografik):** İstirahat elektrokardiyografik ölçüm sonuçları (0= Olası veya kesin sol ventrikül hipertrofisi gösteriyor, 1=normal, 2=ST-T dalga anormalliği var)
* **thalach (max nabız):** Kişinin ulaştığı maksimum kalp atış hızı
* **exang (anjina):** Egzersize bağlı angina (1=evet; 0=hayır)
* **oldpeak (ST depresyon):** Dinlenmeye göre egzersizin neden olduğu ST depresyonu
* **slope (eğim):** Zirve egzersiz ST segmentinin eğimi (0=aşağı eğimli, 1=düz, 2=yukarı eğimli)
* **ca (floroskopi):** Floroskopi ile renklendirilen yer sayısı (0-3)
* **thal (Kalıtsal kan bozukluğu):** Talasemi adı verilen bir kan hastalığı (1=sabit kusur; 2=normal; 3=geri döndürülebilir kusur)
* **target (hedef):** Kalp hastalığı (0=hayır, 1=evet)

**Veri setinin kaynağı:** https://www.kaggle.com/datasets/mragpavank/heart-diseaseuci

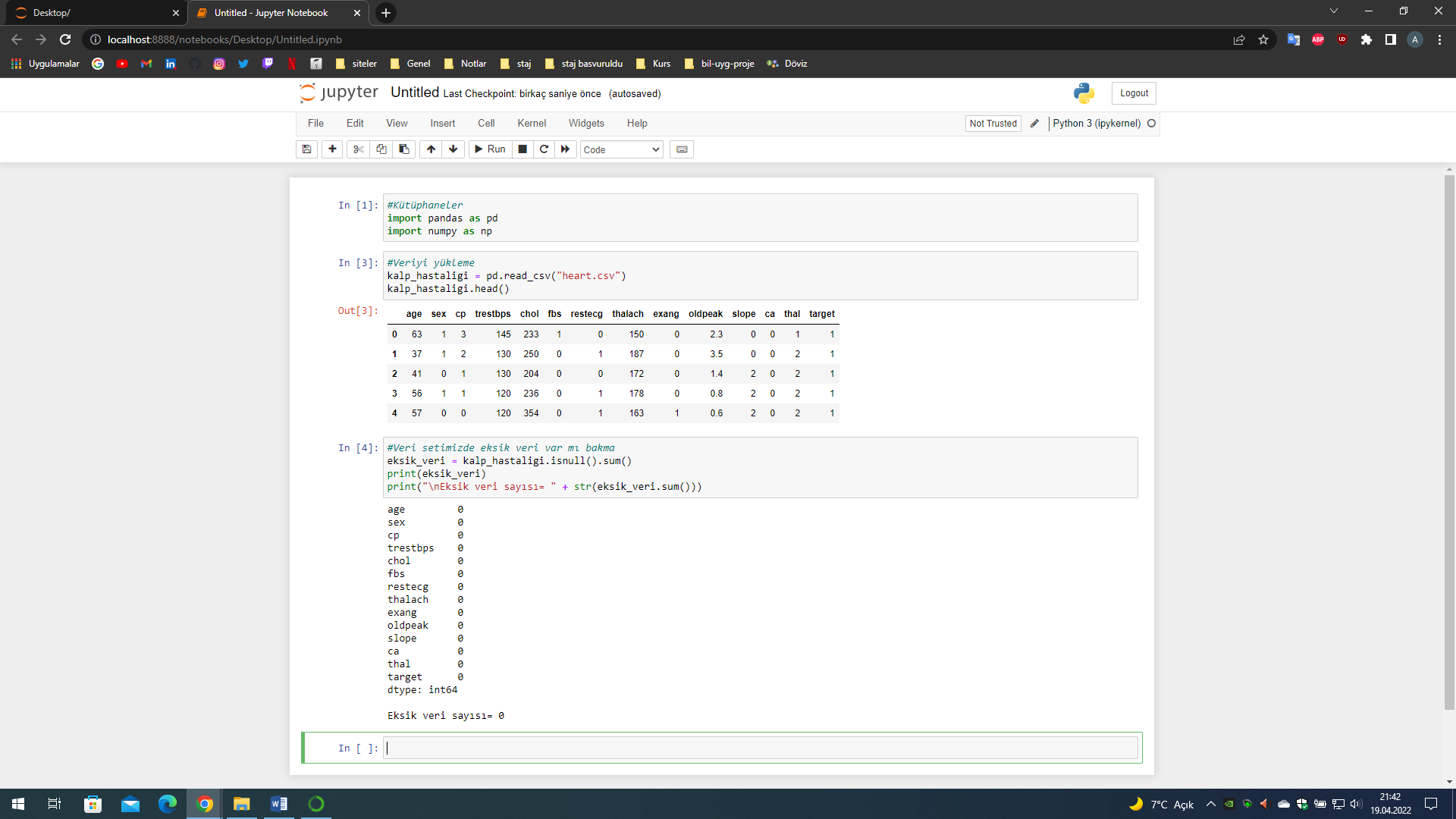
# **Veri Setinin Yüklenmesi:**

* Kalp hastalarımızın bulunduğu veri setimizi pandas kütüphanesini kullanarak projeme yükledim ve ilk 5 hastayı inceledim



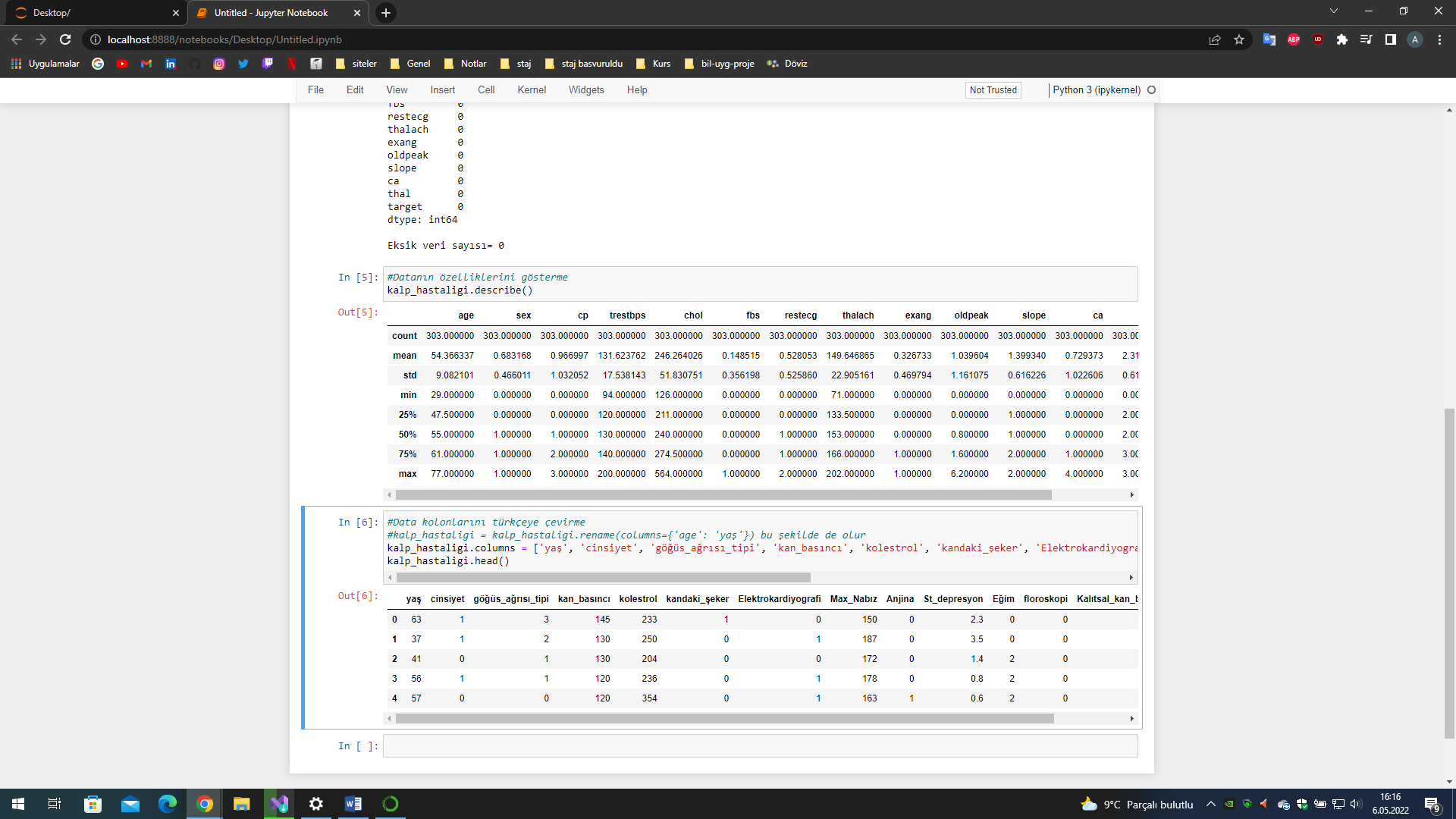
# **Eksik Verilerin Tespiti ve Düzeltilmesi:**

* Veri setimizde eksik veri bulunmamaktadır. Eğer bulunsaydı eksik veri olan satırları droplayabilirdik, sıfır sayısını verebilirdik, özelliğin ortalamasıyla veya özelliğin medyanıyla doldurabilirdik.

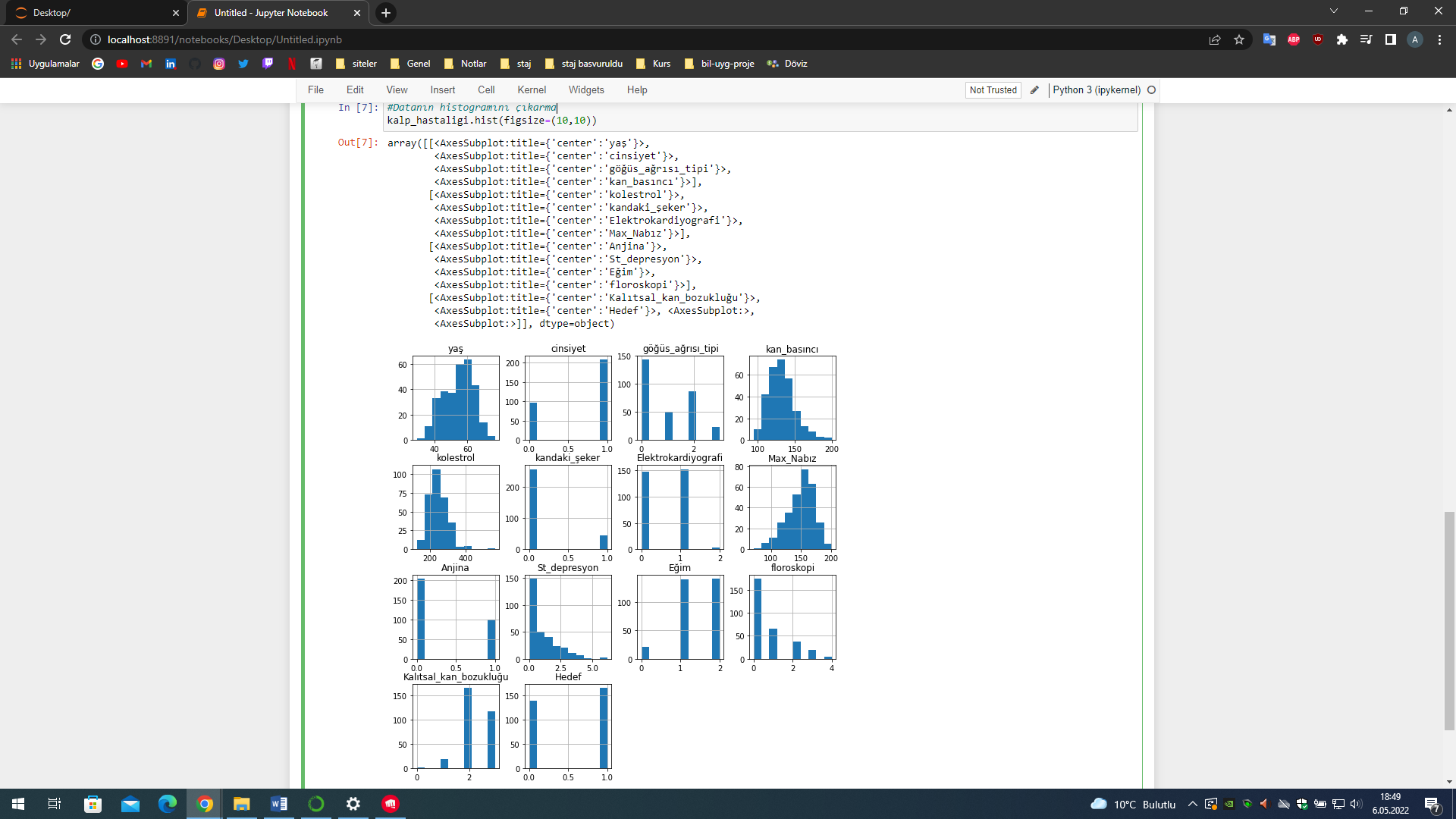


# **Veri Setini Analiz Etme ve Görselleştirme:**

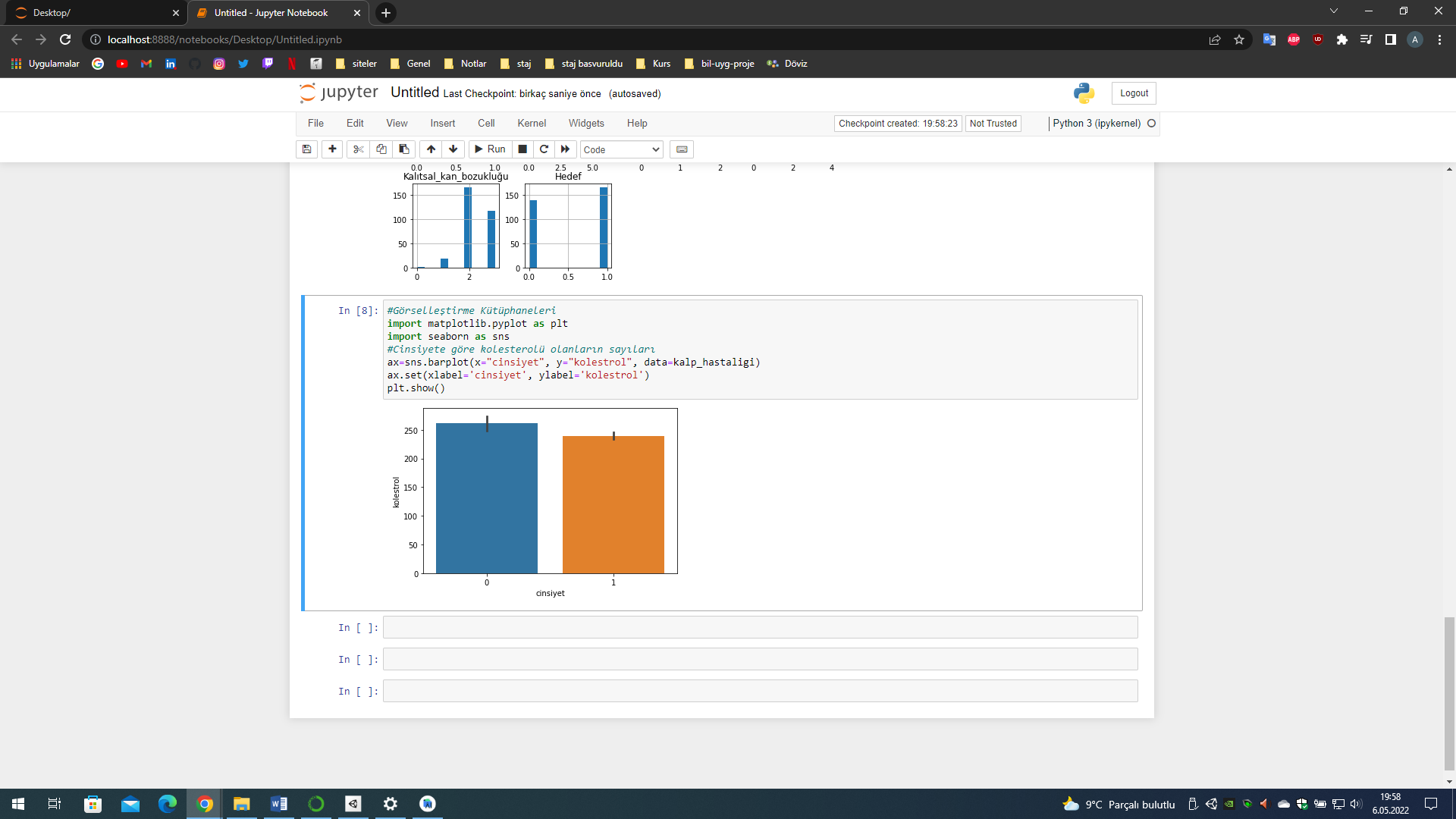
* Describe fonksiyonu ile veri setinin özelliklerinde kaç satır bulunduğunu, medyanını, minumum ve maksimum gibi değerlere baktım. Sonra daha anlaşılır bir görselleştirme olması için veri setinin kolonlarını Türkçeye çevirdim.



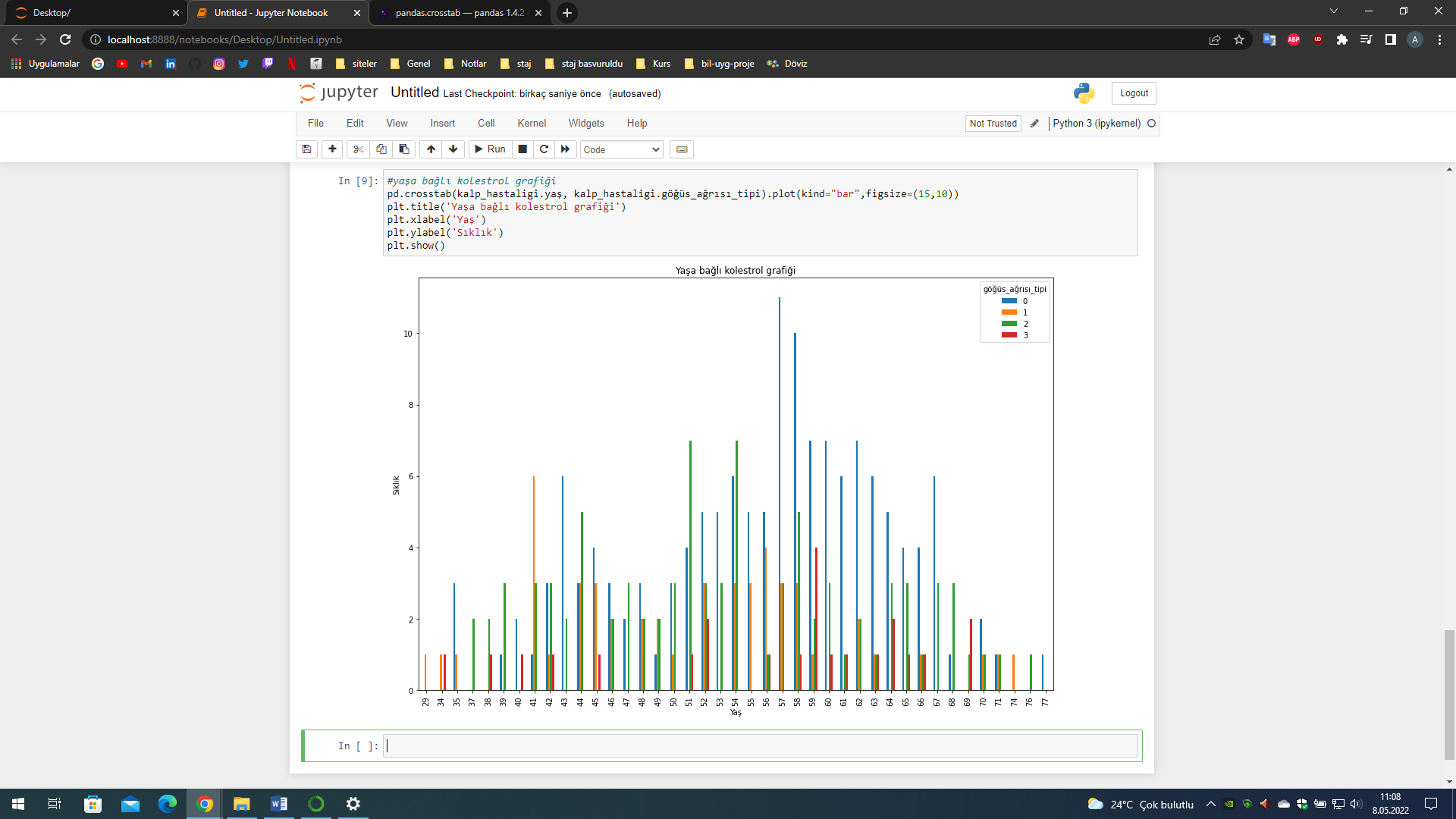
* Veri setinin histogramını çıkararak hangi özelliğin nerelerde yoğun olduğunu buldum ve karşılaştırmasını yaptım.



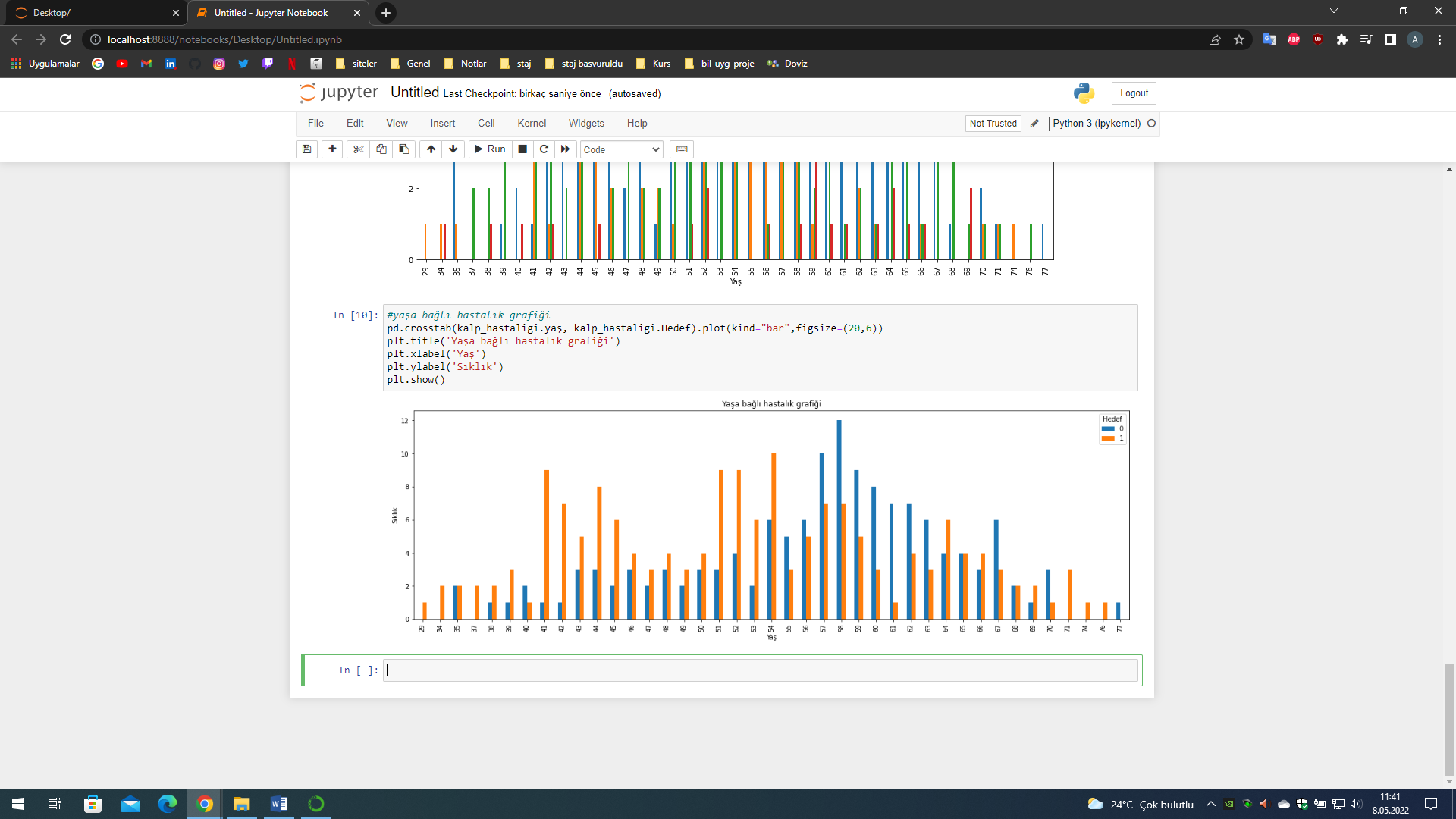
* Matplotlib ve seaborn görselleştirme kütüphaneleri ile görselleştirme yaparak cinsiyete göre kolesterollü olan kişilerin miktarını inceledim



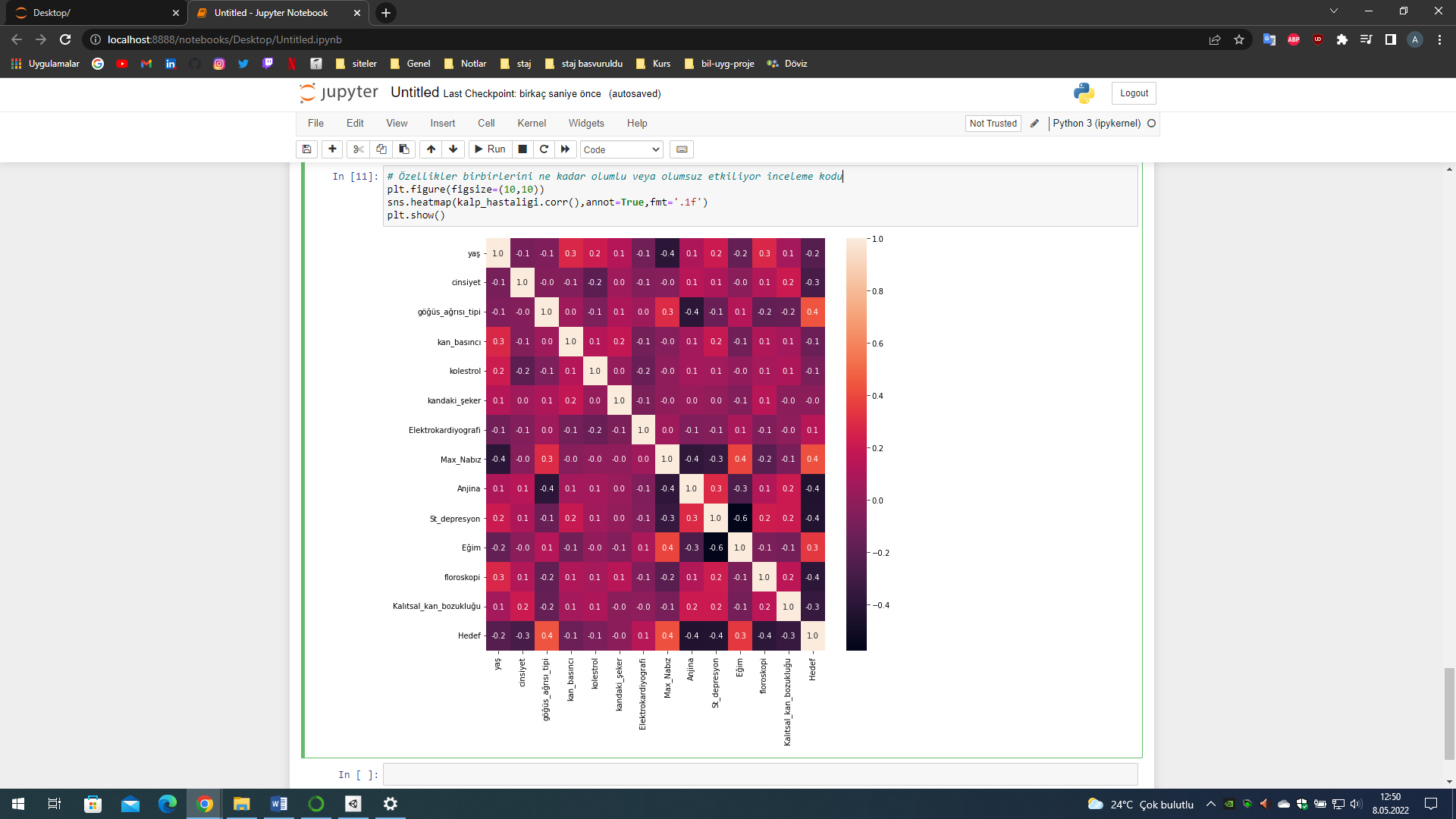
* Pandas’daki crosstab fonksiyonu ile her bir yaşta tekrar eden göğüs ağrısı tiplerinin her birinin miktarını matplotlib ile görselleştirerek inceledim



* Yine aynı şekilde crosstab fonksiyonu ile her yaşın içinde iki adet bulunan hedef yani o yaştakilerin hastalıklı olup olmama durumlarının miktarını görselleştirerek inceledim

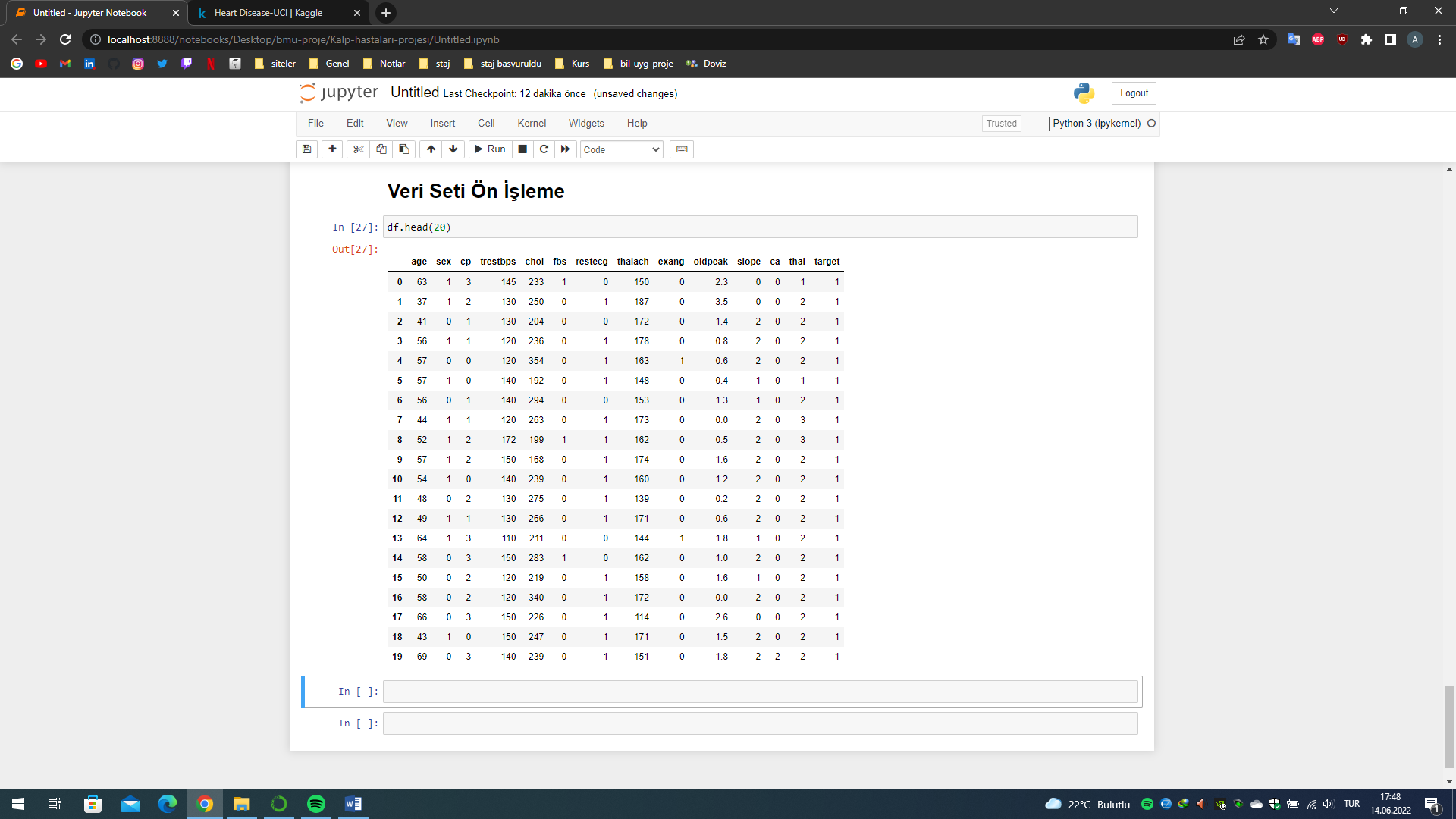


* Özelliklerin birbirleriyle uyumlarının oranını yani her bir özelliğin diğer bir özelliği ne kadar etkilediğinin miktarını seaborn kütüphanesindeki heatmap fonksiyonu ile görselleştirerek inceledim



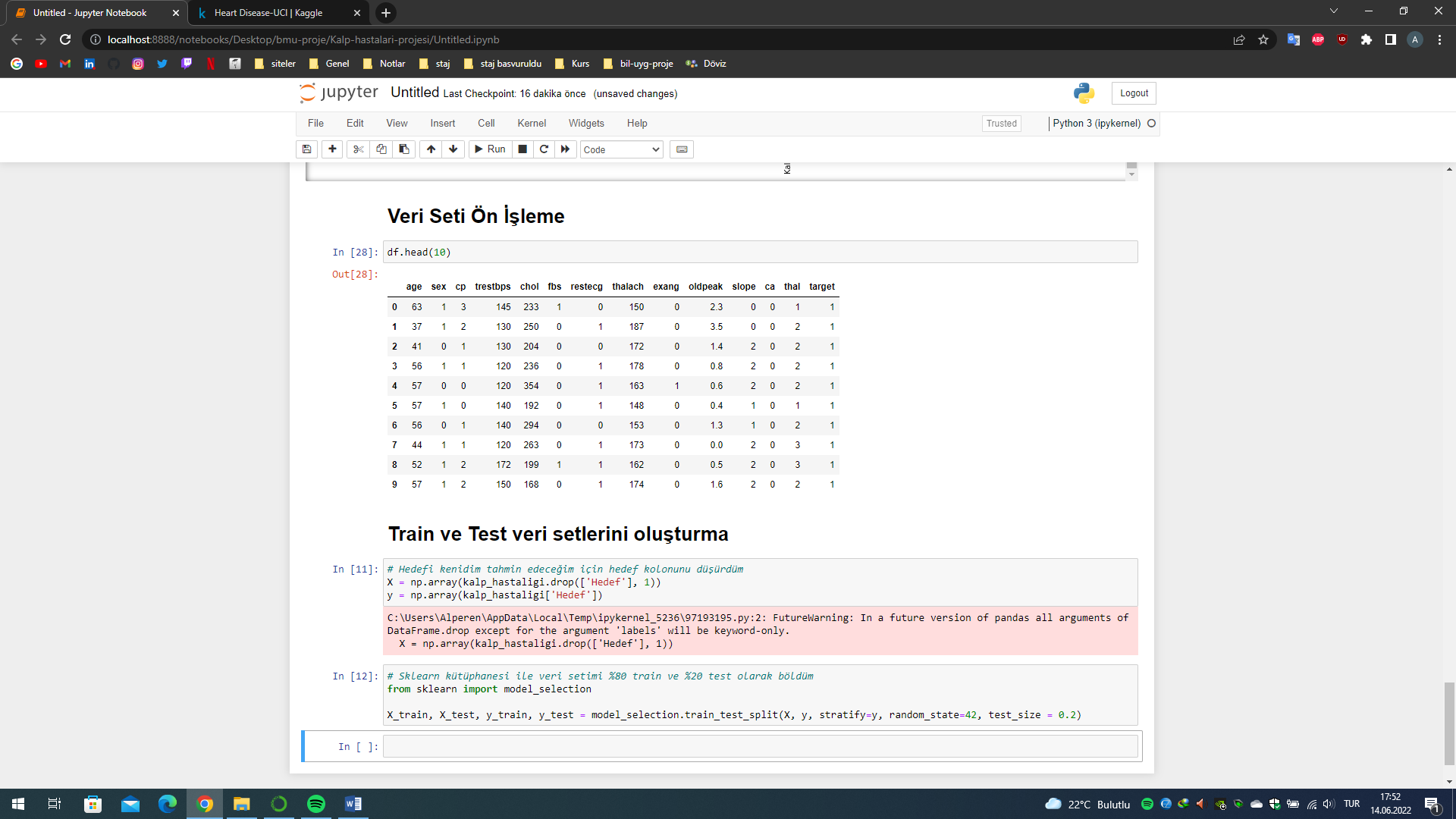
# **Veri Seti Ön işleme:**

* Veri setimde herhangi bir kategorik veri bulunmadığı için kategorik veriden numerik veriye çevirmedim



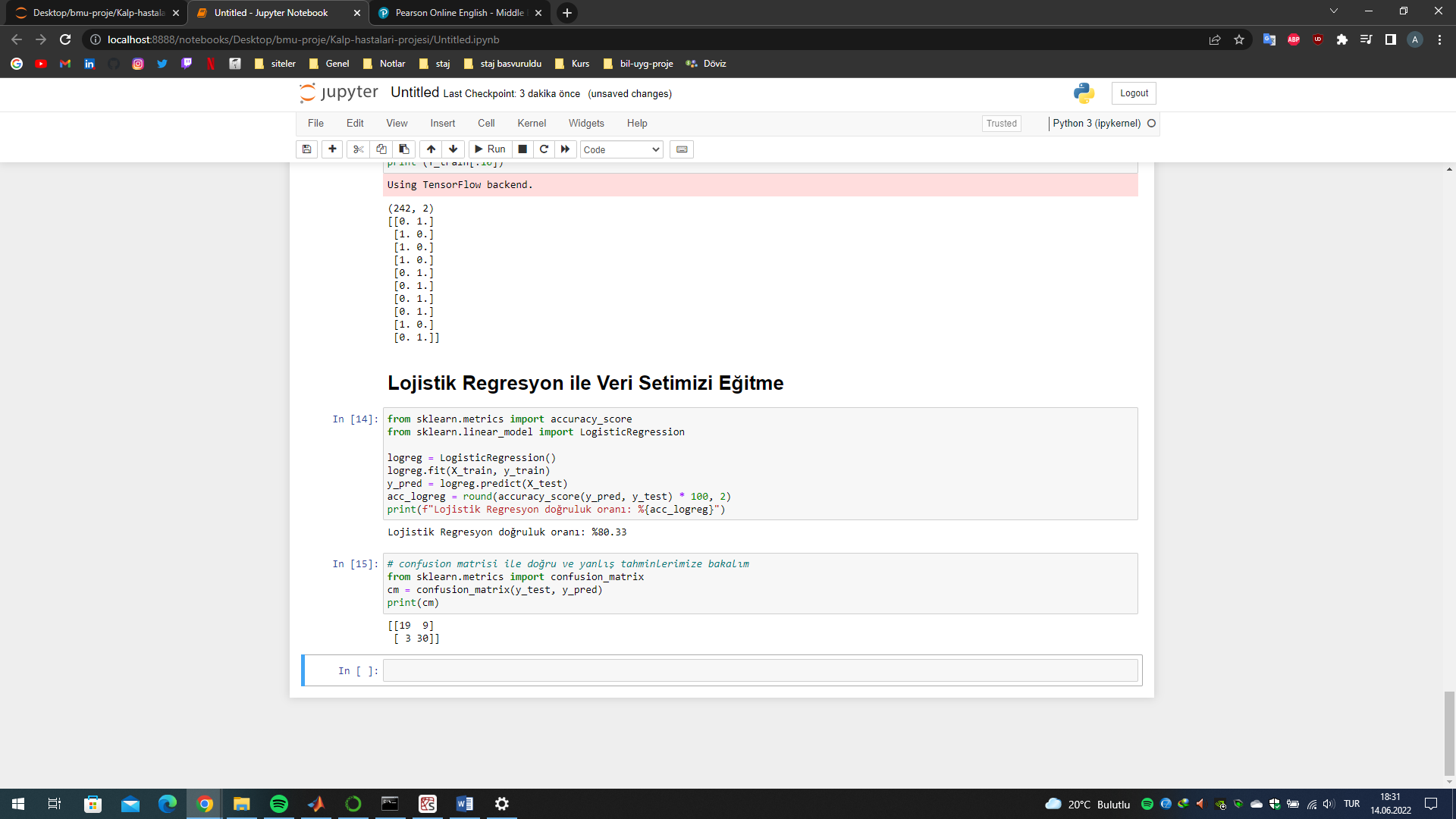
# **Veri Setinin Train ve Test Olarak Bölünmesi:**

* Veri setimizde tahmin edeceğimiz hedef kısmını kendim tahmin edeceğim için hedef kolonunu kaldırdım ve daha sonra testte kullanmak için değişkene atadım
* Sklearn kütüphanesini kullanarak veri setimizi %80’ini Train %20’sini Test için ayırdım



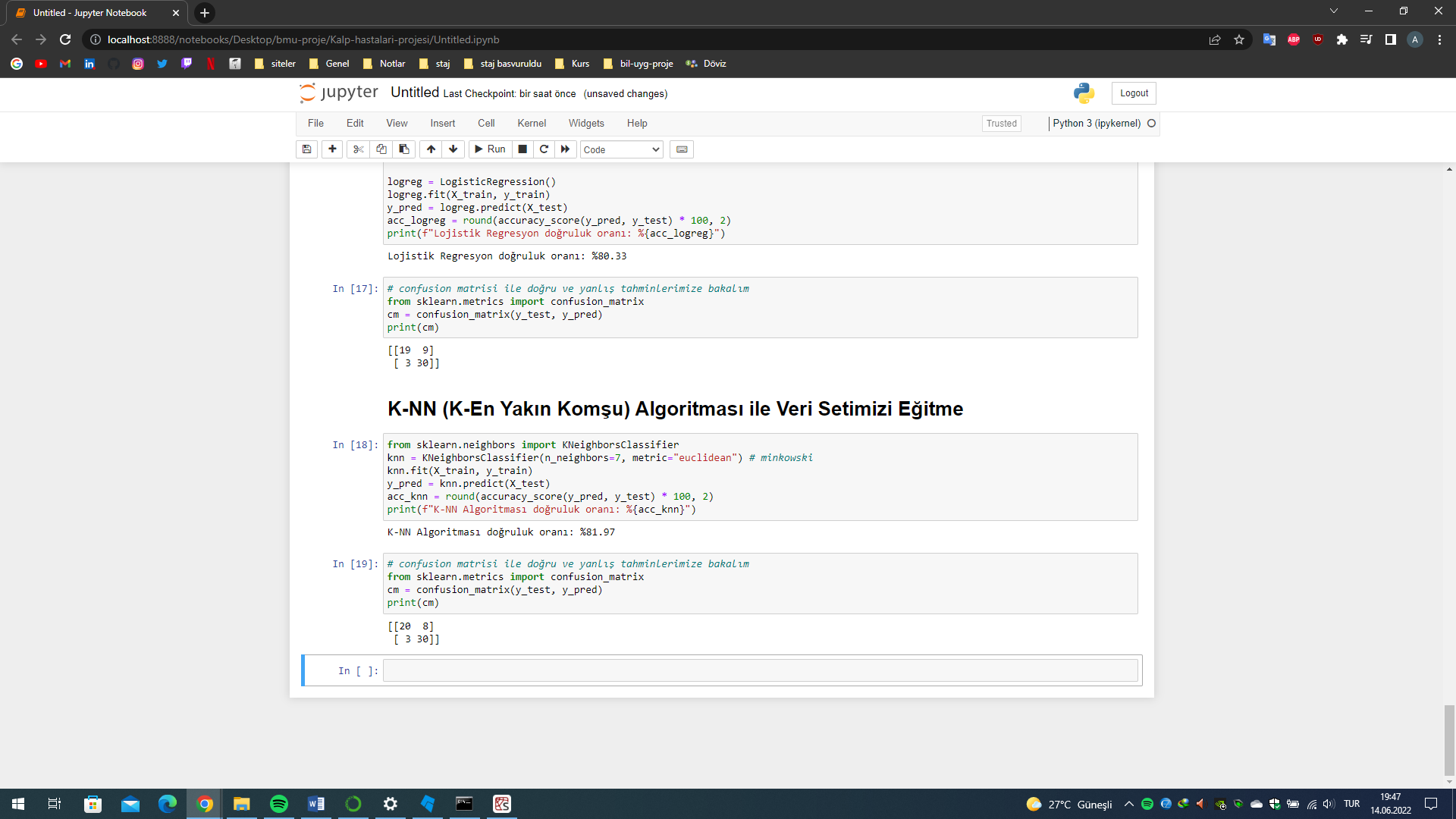
# **Lojistik Regresyon ile Veri Setimizi Eğitme ve Test Etme**

* Sklearn kütüphanesinde lojistik regresyon algoritmasını ve score hesaplamak metrics kütüphanesini import ettim ve hemen veri setimi lojistik regresyon makine öğrenmesi algoritması ile test etmek için fit fonksiyonuna eğitim için böldüğüm x\_train ve y\_train vererek eğitimimi yaptım.
* Lojistik regresyonun tahminleri için predict fonksiyonuna test için ayırdığım X\_test veri setimi verip tahminleri y\_pred değişkenine aktardım. Accuracy\_score fonksiyonu ile gerçek sonuçlarımı karşılaştırıp %80 oranında bir doğruluk buldum.
* Confusion matrisi ile Lojistik Regresyonun doğru ve yanlış tahminlerine baktım. 19TP+30TN doğru 9FP+3FN yanlış tahmin yapmış. Yine accuracy hesaplamak gerekirse 49/61 = 0,8032 üstteki bulduğumuz oranı buluruz.



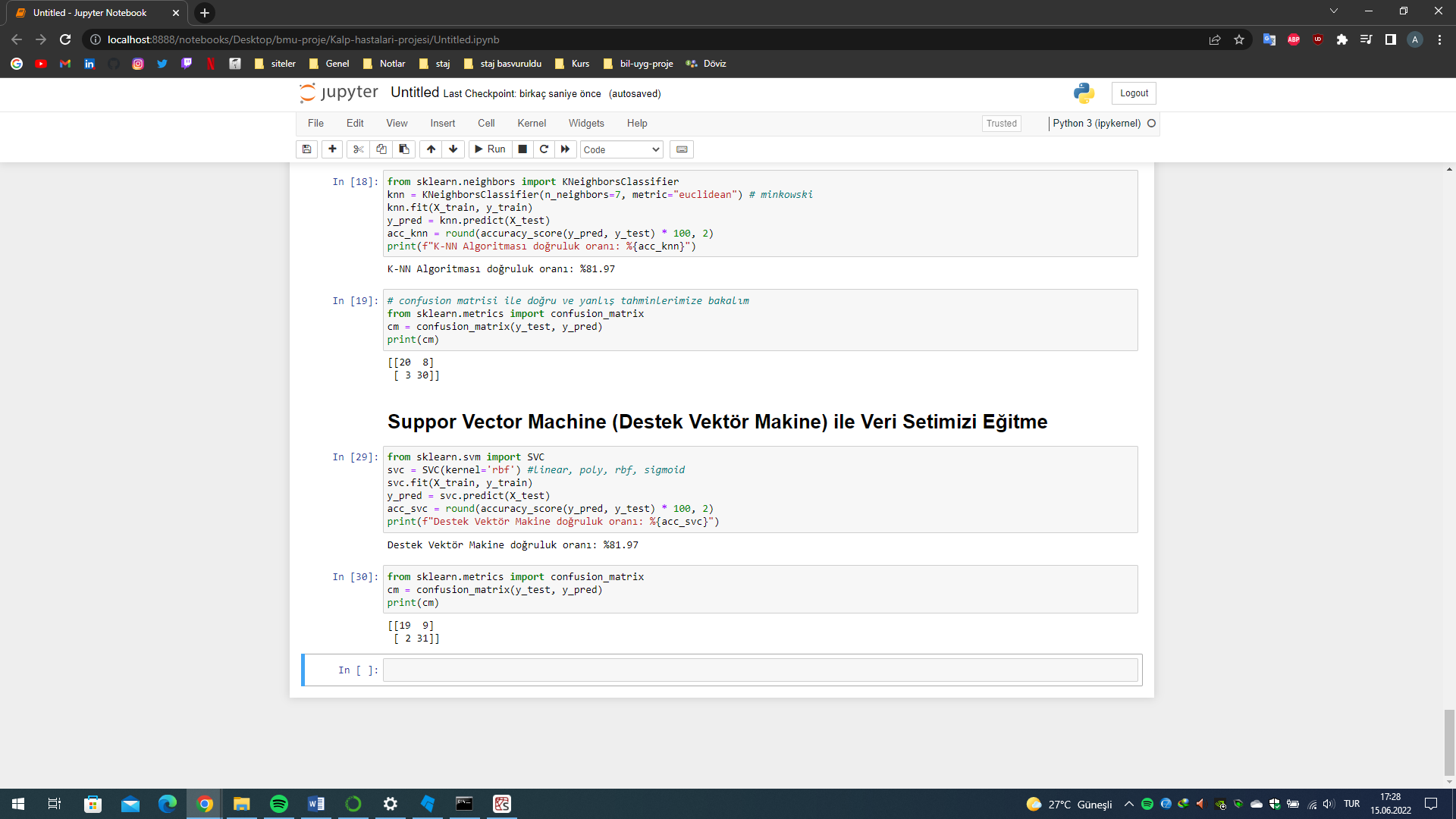
# **K-NN (K-En Yakın Komşular) Algoritması ile Veri Setimizi Eğitme ve Test Etme**

* K-NN Algoritması ile eğittiğim modeli test ettiğimde %81’lik bir accuracy buldum. Metric olarak Öklid yakınlık formülünü kullandım. Farklı konumlarda bulunan fakat aynı sınıfa ait bölgeleri de yakalayabilmesi için komşu sayısını 3 düşündüm ama %80 accuracy yakaladı. 7 seçtiğimde daha iyi sonuç buldum



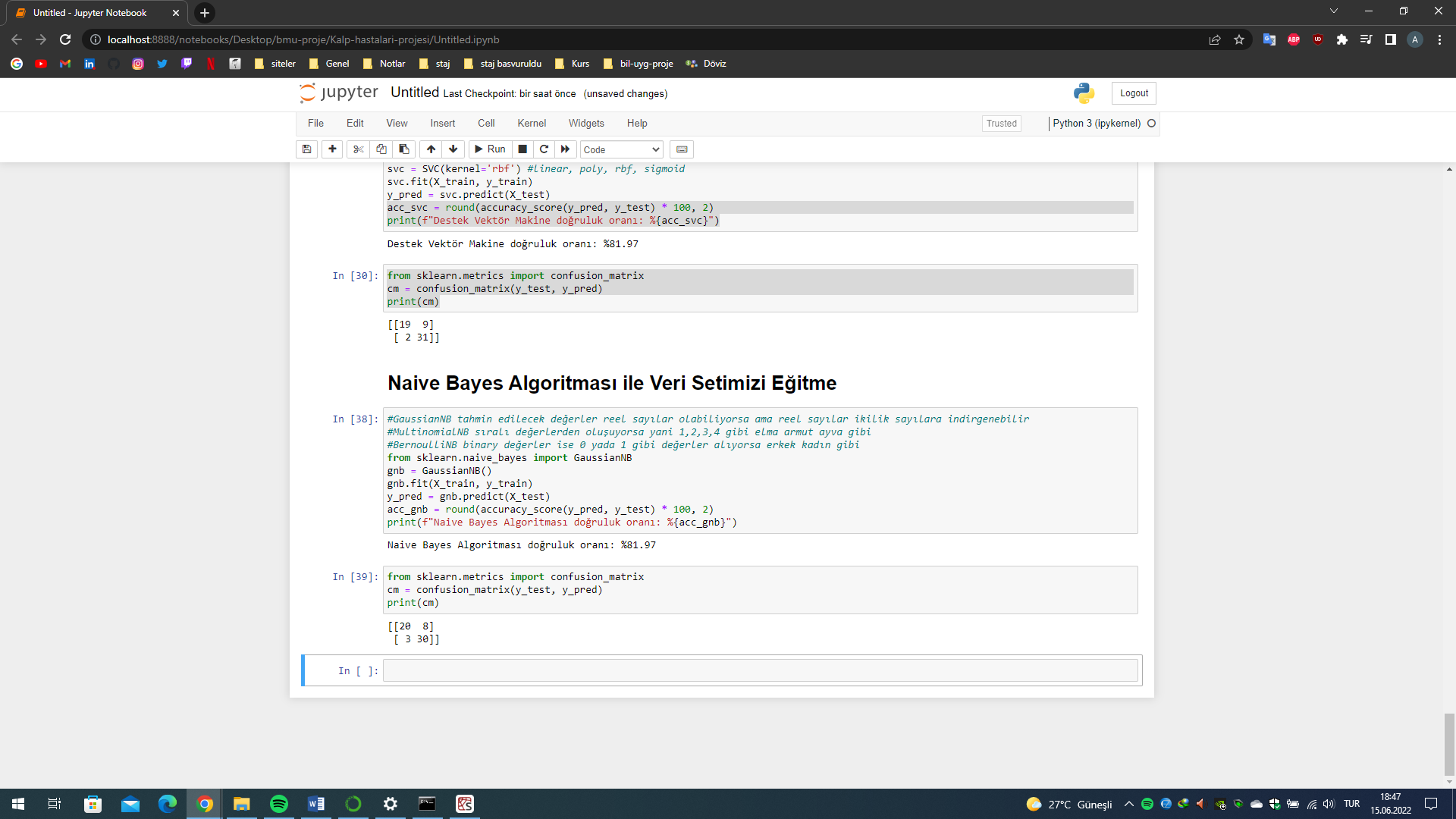
# **Support Vector Machine (Destek Vektör Makine) ile Veri Setimizi Eğitme ve Test Etme**

* Destek Vektör Makine ile eğittiğim modeli test ettiğimde %81’lik bir accuracy buldum. Kernel de en yüksek doğruluk oranını rbf parametresi ile yakaladım



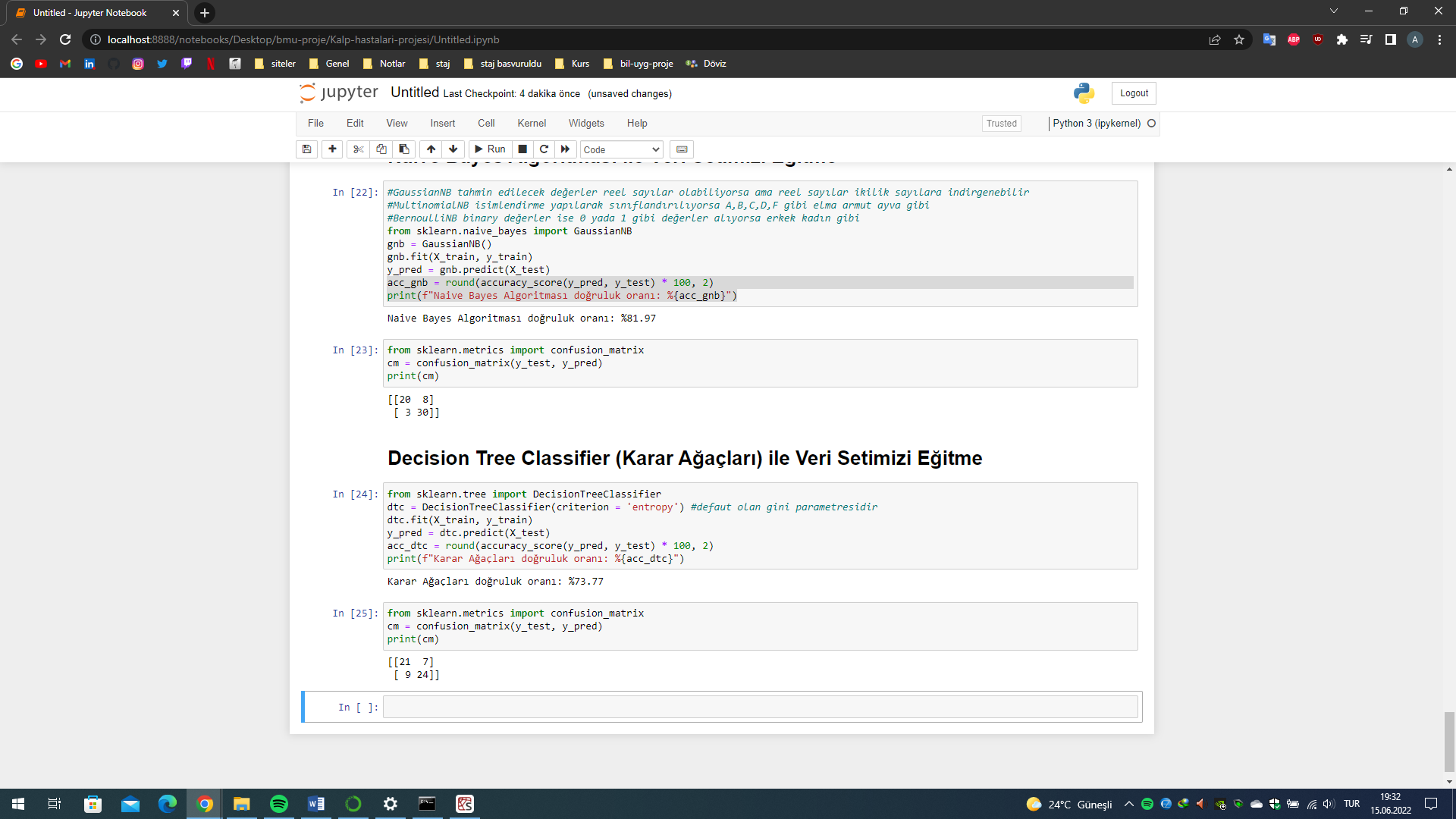
# **Naive Bayes Algoritması ile Veri Setimizi Eğitme ve Test Etme**

* Naive Bayes Algoritmasında Gaussian formülü kullanarak eğitim işlemini yaptım. Eğittiğim modeli test ettiğimde %81’lik bir accuracy yakaladım



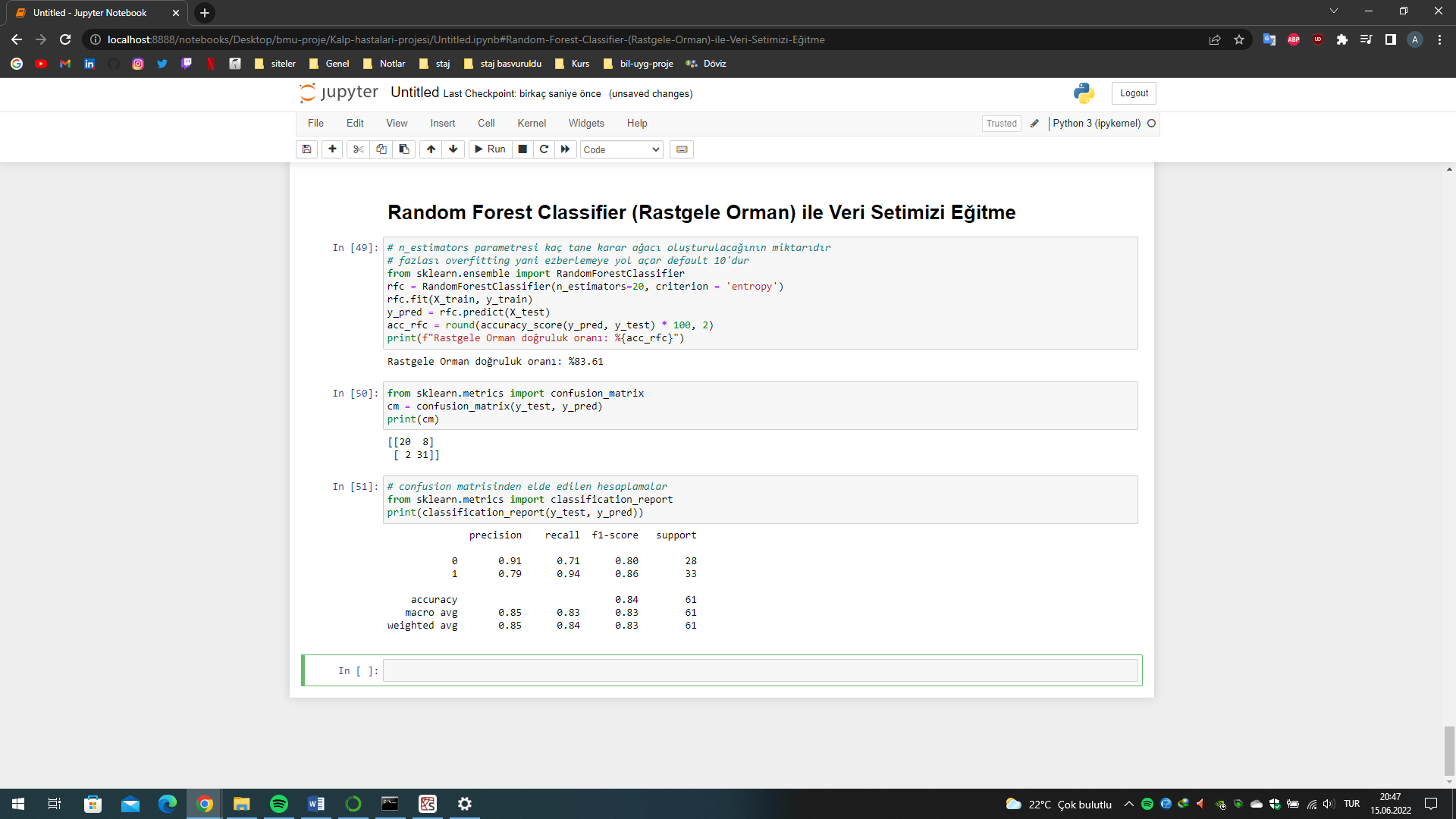
# **Decision Tree Classifier (Karar Ağaçları) ile Veri Setimizi Eğitme ve Test Etme**

* Karar Ağaçları ile sınıflandırmada entropy parametresindeki formülü kullanarak eğitim işlemini yaptım. Eğittiğim modeli test ettiğimde %73’lük bir accuracy yakaladım



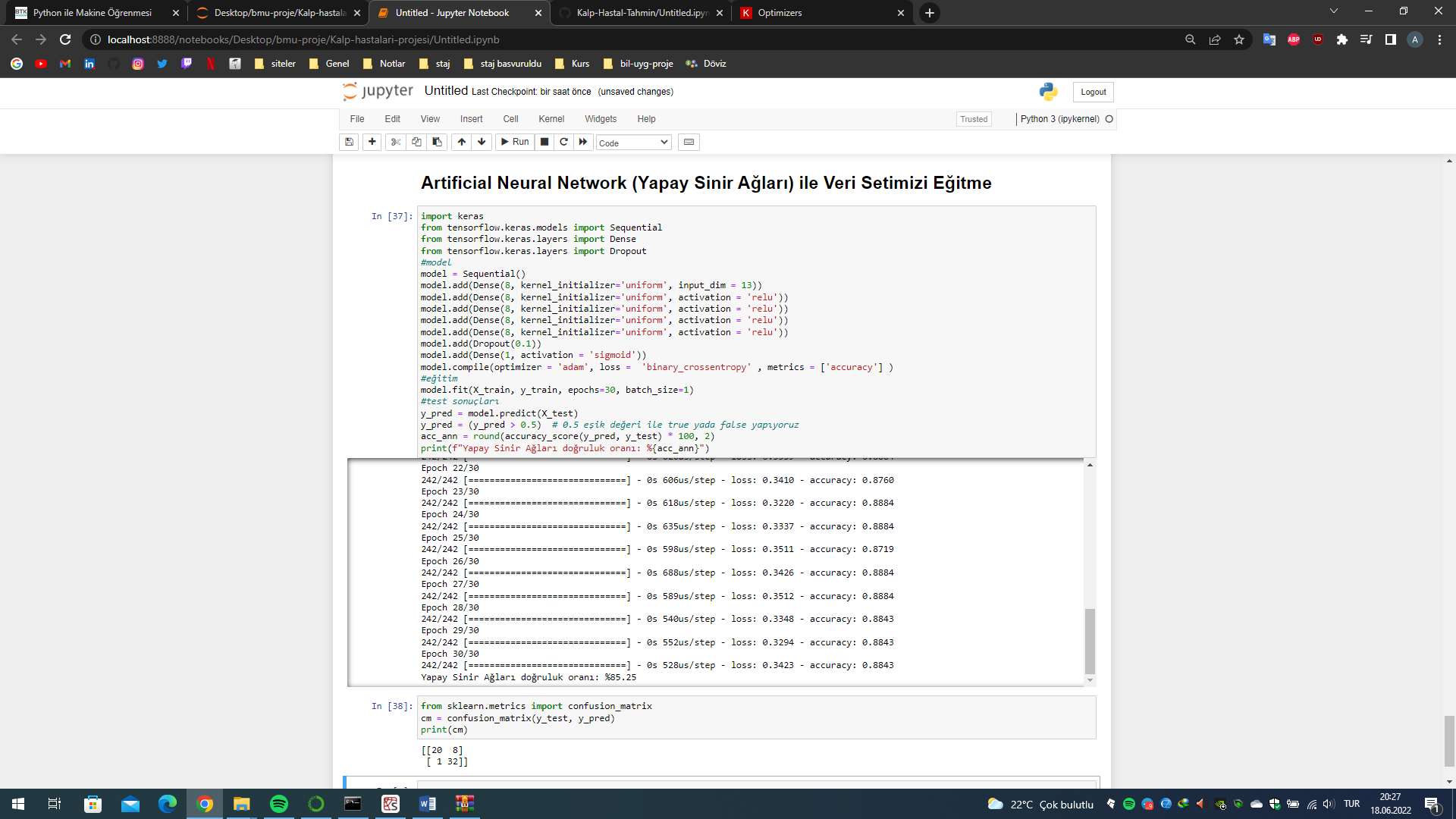
# **Random Forest Classifier (Rastgele Orman) ile Veri Setimizi Eğitme ve Test Etme**

* Rastgele Orman Sınıflandırmada birden fazla karar ağacı kullanarak eğitim işlemi yapılıyor ben n\_estimators parametresiyle 20 adet kullandım fazlası overfitting yani ezberlemeye yol açabilir default olarak 10’dur
* Rastgele Orman sınıflandırmasıyla şu ana kadar en fazla doğruluk oranını yani %83’lük bir oran yakaladım
* Confusion matrisinden elde ettiğimiz bazı hesaplamaları da inceledim Precision(TP/TP+FP), Recall(TP/TP+FN), F1-Score (2\*Prcision\*recall/precision+recall) yani precision(Hassasiyet) ve recall(Duyarlılık) değerlerinin harmonik ortalamasını gösterir



**Artificial Neural Network (Yapay Sinir Ağları) ile Veri Setimizi Eğitme ve Test Etme**

* Keras kütüphanesi ile yapay sinir ağı katmanları oluşturup model oluşturdum ve eğitim yaptım. Yapay sinir ağları sadece 0 ile 1 arasındaki verileri kullandığı için verilerimi train ve test olarak böldükten sonra StandartScaler ile standardizasyon yaptım.
* Modelimde relu(Rectified Linear Unit) aktivasyon fonksiyonunu kullandım. Relu negatif değerleri 0’a pozitif değerleri lineer bir artışla maksimum 1’e kadar değer veriyor. Modelimde 4 adet gizli 1 adet input 1 adet output 1 adet dropout katmanı olacak şekilde 7 adet katmanım var. Katmanlarımda 8 er adet nöron kullandım. İnput katmanındaki İnput\_dim parametresine özellik sayımı girdim. Çıkış katmanında 2 sınıfım olduğu için sigmoid aktivasyon fonksiyonunu kullandım. Sigmoid aktivasyon fonksiyonu, değerleri 0 ya da 1 yaparak sınıflandırmayı yapıyor. Dropout katmanını modelimin ezber yapmamasını sağlamak için kullandım.
* Optimizer olarak adam algoritmasını kullandım. Loss parametresine çıktı olarak 2 sınıfım olduğu için binary\_crossentropy verdim. Birden fazla olsaydı categorical\_crossentropy parametresini verirdim. Sigmoid yerine de softmax aktivasyon fonksiyonunu kullanırdım.
* Eğitim kısmında epoch parametresini 30 verdim. Epoch, modelin kaç defa tekrar yapacağıdır. Batch\_size’ı default bıraktım. Batch\_size ise kaç tane veriyle aktivasyon fonksiyonundaki eşik değerini değiştireleceğinin miktarıdır.
* Tahmin sonuçlarımda 0 ile 1 arasında değerler geliyor. Tahmin sonuçlarımı if koşulu ile True ya da False yapıyorum
* Test yaptığımda %80 ile %85 arasında değişen accuracy yakaladım. Bu oran model üzerinde oynamalar yapılarak arttırılabilir. Mesela katmanlar ve nöronlar arttırılarak veya epoch sayısıyla oynanarak yapılabilir



**Sonuç**

* Veri setimi birden fazla makine öğrenmesi modelim ve yapay sinir ağları modelim ile eğitim yapıp test ettim. En yüksek accuracy oranını %85 ile yapay sinir ağları modelinde yakaladım. Bazı özellikler sıralı şekilde artıyor bu eğitimin yanlış öğrenmesine yol açıyor olabilir. Bu yüzden one hot encoding işlemi yapmaya çalıştım ama hata aldım. Bu işlem de yapılırsa doğruluk oranı arttırılabilir