Yiğit Mert Dönmez - 1421221016

Veri Bilimine Giriş Dersi Final Projesi

1. Proje Özeti
   1. Veri Seti Özeti

Bu veri seti 23 tür mantar türüne karşılık gelen varsayımsal örneklerin açıklamalarını içerir. Başlangıçta, veri kümesinin çıktıları vardır ve yenilebilir veya zehirli olarak tanımlanır. Ancak, veri kümesine kümeleme algoritmaları uygulayacağımız için çıktı sütununu atarız.

* 1. Çalışmanın Amacı

Belirtildiği gibi, kümeleme algoritmaları veri kümesine uygulanacaktır. Burada amaç veri seti içerisinde bulunan değerlerin kaç farklı şekilde ve düzgün bir şekilde kümelenebildiğinin tespitinin yapılmasıdır.

* 1. Analizin Ana Bulguları

Veri seti 6 farklı kümeye bölündüğünde en güzel ve doğru şekilde bölünebilmiş olmaktadır. Bunun yanı sıra k means algoritması agglomerative algoritmasından daha iyi bir şekilde kümeleme yaptığı tespit edilmiştir.

* 1. Proje kapsamında yararlanılan kaynaklar ve yararlanılan siteler

<https://www.kaggle.com/uciml/mushroom-classification>

<https://www.kaggle.com/nirajvermafcb/comparing-various-ml-models-roc-curve-comparison>

1. Veri Seti Açıklaması

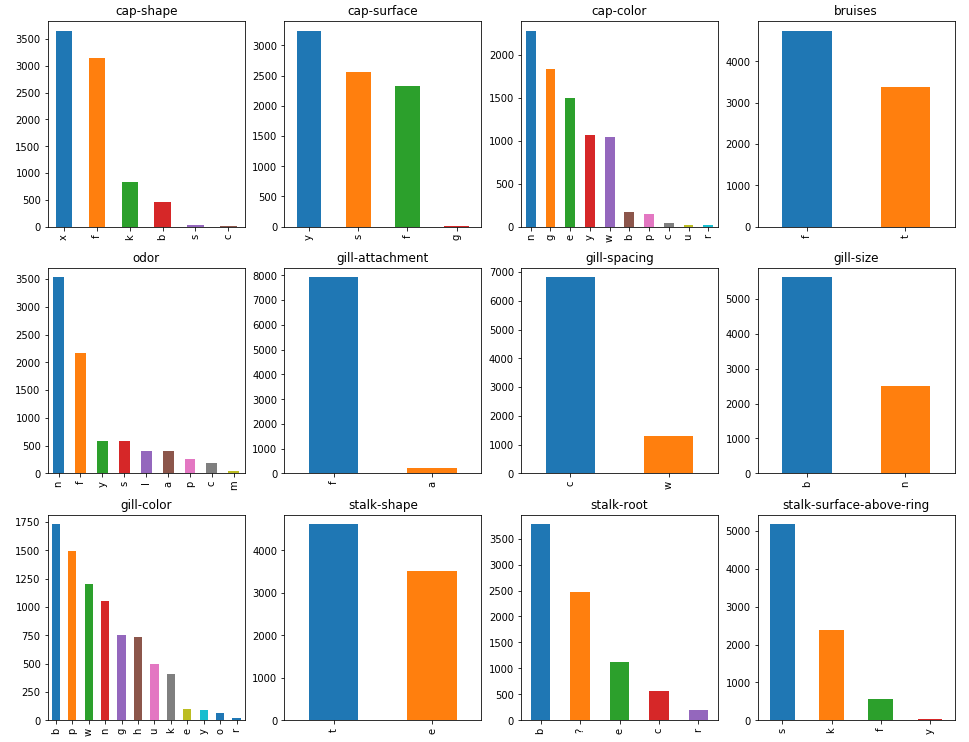
Veri seti içerisinde 8124 satırda çeşit çeşit bulunan mantarlar ve 22 sütunda bu mantarları niteleyen özellikler bulunmaktadır.

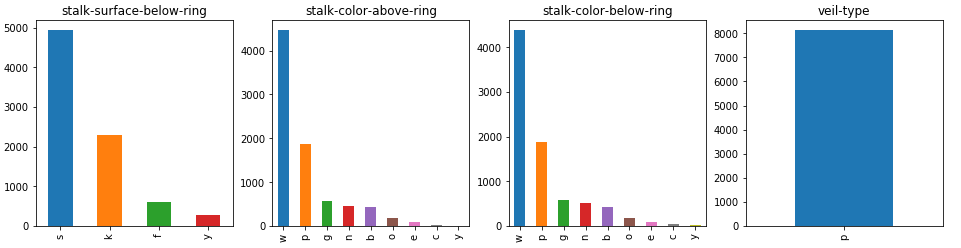
Veri seti özellikleri

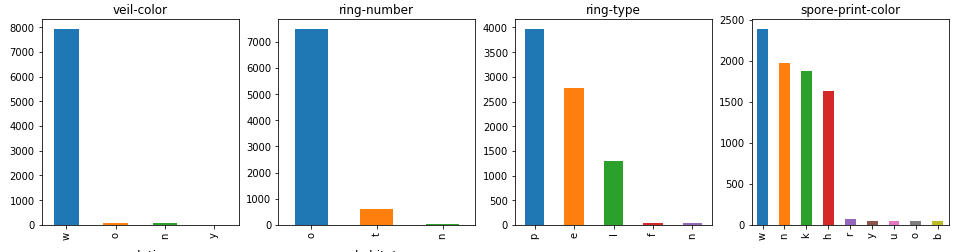
* Cap-shape : Başlık şekili – Kategorik veri tipi
* Cap-surface : Başlık yüzeyi – Kategorik veri tipi
* Cap-color : Başlık rengi - Kategorik veri tipi
* Bruises : Çürükler - Kategorik veri tipi
* Odor : Koku – Kategorik veri tipi
* Gill : Mantarın alt kısmı - Kategorik veri tipi
* Gill-spacing : Alt kısım boşluğu - Kategorik veri tipi
* Gill-size : Alt kısım boyutu - Kategorik veri tipi
* Gill-color : Alt kısım rengi - Kategorik veri tipi
* Stalk-shape : Sap şekli - Kategorik veri tipi
* Stalk-root : Sap kökü - Kategorik veri tipi
* Stalk-surface : Sap yüzeyi - Kategorik veri tipi
* Stalk-surface-below-ring : Sap yüzey alt halka - Kategorik veri tipi
* Stalk-surface-above-ring : Sap yüzey üst halka - Kategorik veri tipi
* Stalk-color-below-ring : Alt halka sap rengi - Kategorik veri tipi
* Veil-type : peçe tipi : Kategorik veri tipi
* Veil-color : peçe rengi - Kategorik veri tipi
* Ring-number : Halka sayısı - Kategorik veri tipi
* Ring-type : Halka tipi - Kategorik veri tipi
* Spore-print-color : Kök baskı rengi : Kategorik veri tipi
* Population : Nüfus - Kategorik veri tipi
* Habitat : Yetişme Ortamı - Kategorik veri tipi

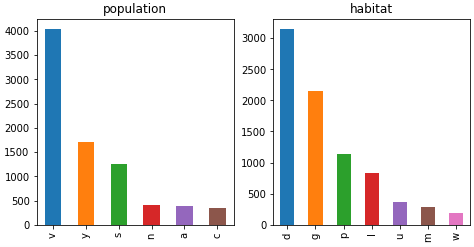
1. Temel Tanımlayıcı İstatistikler

* cap-shape -> Count : 8124 – unique : 6 – top : x – freq : 3656
* cap-surface -> Count : 8124 – unique : 4 – top : y – freq : 3244
* cap-color -> Count : 8124 – unique : 10 – top : n – freq : 2284
* bruises -> Count : 8124 – unique : 2 – top : f – freq : 4748
* odor -> Count : 8124 – unique : 9 – top : n – freq : 3528
* gill-attachment -> Count : 8124 – unique : 2 – top : f – freq : 7914
* gill-spacing -> Count : 8124 – unique : 2 – top : c – freq : 6812
* gill-size -> Count : 8124 – unique : 2 – top : b – freq : 5612
* gill-color -> Count : 8124 – unique : 12 – top : b – freq : 1728
* stalk-shape -> Count : 8124 – unique : 2 – top : t – freq : 4608
* stalk-root -> Count : 8124 – unique : 5 – top : b – freq : 3776
* stalk- surface-above-ring-> Count : 8124 – unique : 4 – top : s – freq : 5176
* stalk- surface-below-ring-> Count : 8124 – unique : 4 – top : s – freq : 4936
* stalk- color-above-ring-> Count : 8124 – unique : 9 – top : w – freq : 4464
* stalk- color-below-ring-> Count : 8124 – unique : 9 – top : w – freq : 4384
* veil-type -> Count : 8124 – unique : 1 – top : p – freq : 8124
* veil-color -> Count : 8124 – unique : 4 – top : w – freq : 7924
* ring-number -> Count : 8124 – unique : 3 – top : o – freq : 7488
* ring-type -> Count : 8124 – unique : 5 – top : p – freq : 3968
* spore-print-color -> Count : 8124 – unique : 9 – top : w – freq : 2388
* population -> Count : 8124 – unique : 6 – top : v – freq : 4040
* habitat -> Count : 8124 – unique : 7 – top : d – freq : 3148



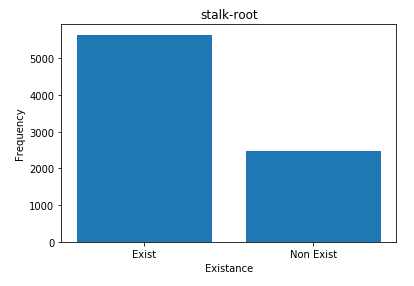






* 1. Kayıp Değerler

Veri seti null değerler içermemektedir. Fakat tanımsız değerler veri seti sağlayıcısı tarafından “?” olarak işaretlenmiştir.



Burada bulunan grafikte “stalk-root” sütununda bulunan “?” ile tanımlanmış boş değerler gözükmektedir. Buradaki değerlerin içerisinde 2480 değerde “?” bulunmaktadır.

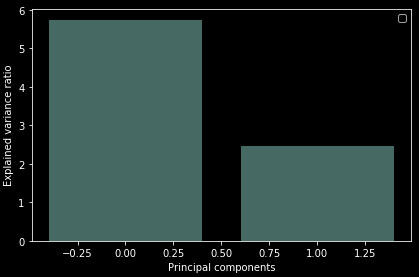
Bu “?” ile işaretlenmiş olan değerleri lojistik regresyon kullanarak tahmin etmeye çalıştık. İlk çalışmamızda modelde overfit problemi mevcuttur. Bu problemi lojistik regresyonun regulasyon parametresini değiştirerek çözdük. Bu tahmin edilen değerleri diğer tanımlanmış verilerle birleştirdik. Bu tahmin etme işleminde lojistik regresyondan gelen eğitim verisi doğruluk oranı 0.95, test verisi doğruluk oranı 0.93 olarak gözlemlenmiştir.

Veri seti içerisinde bulunan “veil-type” sütununda sadece tek bir değer bulunduğu için yapacağımız gözleme herhangi bir katkısı bulunmayacağından bu sütunu veri setimi içerisinden çıkarıyoruz.

* 1. Aykırı Değerler

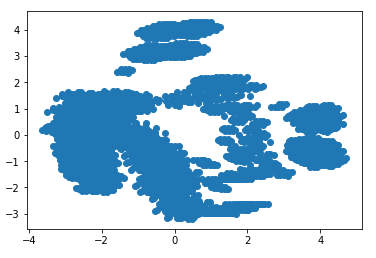
Aykırı olan değerleri grafikleştirerek gösterebiliriz. Fakat veri setimizde 22 boyut bulunmaktadır. Verileri görselleştirebilmek için temel bileşenler analizi kullanıyoruz.

Verilerimizi 2 boyutta görselleştirmek istediğimiz için temel bileşenler analizinde en yüksek varyansı açıklayan ilk 2 sütunu alıyoruz.



Yukarıdaki çubuk grafiğinde en yüksek varyansı açıklayan ilk 2 temel bileşen ve bunlara karşılık gelen varyans değerleri görülmektedir.

Bu temel bileşen analizi grafiklendirildiğinde herhangi bir aykırı değere rastlanmamaktadır.



1. Veri Yönetimi Süreçleri

Veri seti tek bir dosyadan gelmektedir. Bu nedenle herhangi bir dosya birleştirme işlemi bulunmamaktadır.

Kayıp değerlerin nasıl kontrol altına alındığı bilgisi bölüm 3.1 de anlatılmıştır.

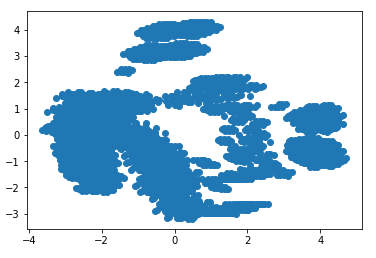
Aykırı değerler grafikler ile gösterilebildiği için ve veri seti içerisinde 22 boyut bulunduğu için temel bileşenler analizi uygulanarak 2 boyuta düşürülmüştür. Bu boyut indirgeme sonucu herhangi bir aykırı değer bulunmadığı bölüm 3.2 de grafiklendirilmiştir.

Veri setine bölüm 3.2 de temel bileşenler analizi uygulayarak veriyi dönüştürdük. Temel bileşen analizine uygun hale getirebilmek için veriye standart ölçeklendirme işlemi uyguladık.

1. Veri Görselleştirme

Projemizin konusu kümeleme algoritmaları olduğu için sonuç sütunumuz bulunmamaktadır. Bu nedenle her bir özelliğin sonuç sütunu ile ilişkisi gösterilemez.

Projemiz içerisinde temel bileşenler analizi ile boyut indirgeme metodunu kullandığımız için 22 boyutluk veri setimizi 2 boyuta düşürdük ve elimizde en yüksek varyansı açıklayan 2 boyut oluşmuş oldu. Bu iki boyutun birbirleri arasındaki ilişki gösteren scatter plot aşağıdaki grafikte bulunmaktadır.



1. Makine Öğrenmesi Uygulamaları

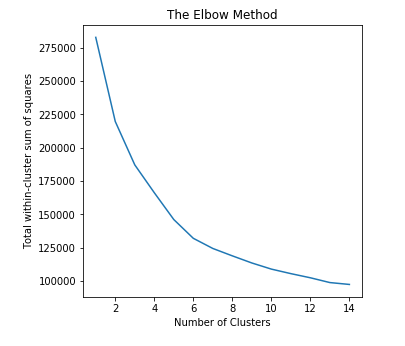
Bu proje kapsamında mantar veri seti üzerinde kümeleme(clustering) metodları kullanılmıştır.

* 1. K - Means Kümeleme Algoritması

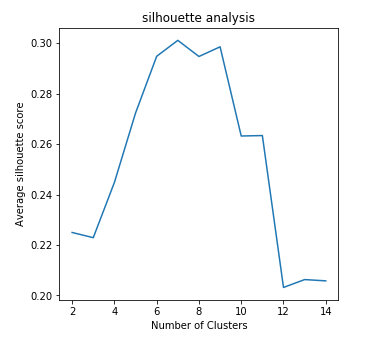
Burada ilk olarak önemli olan durum küme boyutlarının ne kadar olması gerektiğidir.

En iyi modeli seçmek için k-means algoritmasının inertia ölçüm metriğinin minimize edilmesi gerekmektedir. Bu metrik her bir örneğin kendi küme merkezine olan uzaklığının karesinin toplamıdır.

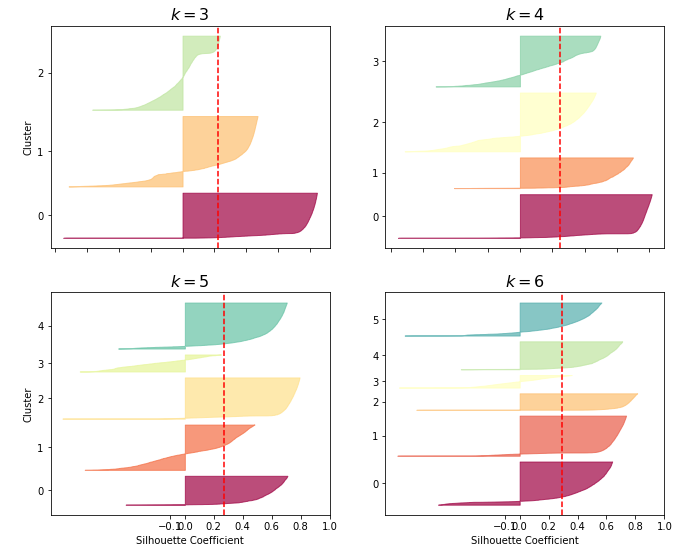
Bu nedenle inertia değerini en aza indiren bir küme boyutu bulmamız gerekiyor. Ancak, küme boyutunu artırdıkça, intertia değeri daha düşük değerlere gider. Bu yüzden optimum bir değer bulmamız gerekiyor. Inertia çizimi, küme büyüklüğünün seçilmesine yardımcı olabilir. Grafik yakınlaşmaya başladığında bir küme büyüklüğü değeri seçebiliriz. Bu yönteme dirsek yöntemi denir.



Bu değer ile kümelerin birbirinden ne kadar farklı olduğunu bulmuş oluruz. +1 ile -1 arasında oluşan değer +1 e ne kadar yakın olursa küme sayısının o kadar iyi olduğunu teyit etmiş oluruz. Kümelerin kendi içlerinde ve kendi aralarındaki ortalama uzaklık arasında oluşturulan denklem ile bu değer bulunur.

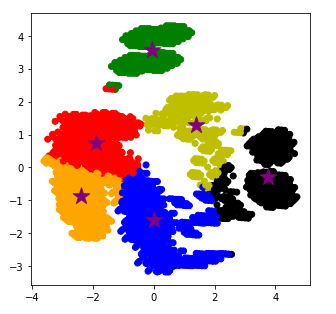


Her iki grafiğide analiz ettiğimizde küme büyüklüğü olarak 6 değerini seçmek en uygun çözüm olacaktır.



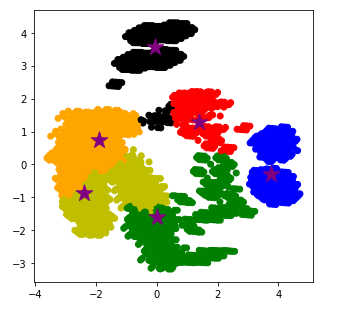
Silhouette analizine göre kümelerin sihouette katsayılarının 1 e yakın olması istenmektedir.

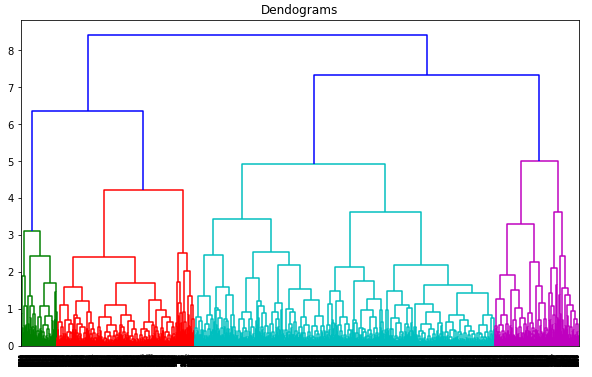
K = 6 da bu durum görülmektedir.



* 1. Agglomerative Kümeleme Algoritması

Agglomerative Clustering, başlangıçta her bir veri noktasına bir küme diyerek işleme başlar. Daha sonra birbirine en yakın olan kümeleri birleştirerek farklı ve yeni bir küme oluşturur ve en sonunda bu şekilde kümeleri birleştire birleştire devam ederek tek bir kümede bütün veriler birleşinceye kadar bu işleme devam eder.





Yukarıdaki dendogramda görüldüğü gibi dendrogramın uygun yerinden çekilen çizgiden geçen dikey çizgilerin altında kalan kümelere bakıldığında en uygun 6 kümenin verimiz için en uygun kümeleme sayısı olduğu tespit edilmiştir.