Systematic sampling

Sistematik örnekleme, popülasyonu düzenli aralıklarla örnekler. Yukarıdaki şekilde, her satırda yukarıdan soldan sağa bakıldığında, her beşinci kahveden biri örneklenir.

Sistematik örnekleme - aralığın tanımlanması

- Pandas ile sistematik örnekleme, basit rastgele örneklemeden biraz daha zordur.
- Zor olan kısım, belirli bir örneklem büyüklüğü için her satır arasındaki aralığın ne kadar büyük olması gerektiğini belirlemektir. Beş kahvelik bir örneklem büyüklüğü istediğimizi varsayalım. Popülasyon büyüklüğü, tüm veri setindeki satır sayısıdır ve bu durumda 1338'dir.
- Aralık, popülasyon büyüklüğünün örneklem büyüklüğüne bölünmesidir, ancak cevabın tam sayı olması istendiğiz için // operatörü ile tam sayı bölmesi gerçekleştirilir. Bu, standart bölmeye benzer, ancak kesirli kısmı atar. 1338 / 5 267.6'dır ve kesirli kısmı atarsak iki 267. Bu nedenle, beş kahvelik sistematik bir örneklem elde etmek için veri setindeki her iki yüz altmış yedinci kahve seçilir.

The trouble with systematic sampling

- Yine de sistematik örnekleme ile ilgili bir sorun vardır.
- Kahvelerin ağızda bıraktığı tat özelliğine ilişkin istatistiklerle ilgilenilsin. Bunu incelemek için önce reset_index kullanarak DataFrame'de çizilebilecek bir indeks değerleri sütunu oluşturulur.
- Ağızda kalan tadı indekse karşı grafiğe dökmek bir model gösterir. Önceki satırlar genellikle sonraki satırlardan daha yüksek tat puanlarına sahiptir.
- Bu, hesaplanan istatistiklere önyargı getirir. Genel olarak, sistematik örnekleme kullanmak yalnızca böyle bir grafikte örüntü yoksa, yani sadece gürültü gibi görünüyorsa güvenlidir.

Making systematic sampling safe

- Sistematik örneklemenin güvenli olmasını sağlamak için, örneklemeden önce satır sırası rastgele hale getirilir.
- sample() fonksiyonu, n'nin belirttiği mutlak satır sayısı yerine, örneklemde döndürülecek veri kümesinin oranını belirtmeyi sağlayan frac adlı bir argümana sahiptir.
- frac değerini bir olarak ayarlamak, tüm veri kümesini rastgele örnekler. Aslında bu, satırları rastgele karıştırır. Daha sonra, indekslerin sıfırdan başlayarak sıralanacak şekilde sıfırlanması gerekir.
- drop eşittir True olarak belirtmek önceki satır indekslerini temizler ve başka bir reset_index çağrısına zincirleme yapmak bu yeni indeksleri içeren bir sütun oluşturur.
- Karıştırılmış veri kümesiyle grafiğin yeniden çizilmesi, tat ve dizin arasında hiçbir örüntü göstermez. Bu iyidir, ancak satırları karıştırdıktan sonra sistematik örneklemenin basit rastgele örneklemeyle aynı olduğunu unutmayın.

Stratified and weighted random sampling

Tabakalı örnekleme, alt grupları içeren bir popülasyonu örneklemeye olanak sağlayan bir tekniktir.

Örneğin, kahve derecelendirmeleri ülkelere göre gruplandırılabilir. value_counts yöntemini kullanarak ülkelere göre kahve sayısı elde edilirse, bu altı ülkenin en fazla veriye sahip olduğu görülür.

Örnekleme alt grupları hakkında düşünmeyi kolaylaştırmak için analizi bu altı ülkeyle sınırlayalım. Nüfusu filtrelemek ve yalnızca bu altı ülkeye karşılık gelen satırları döndürmek için isin() fonksiyonu kullanılabilir. Bu filtrelenmiş veri kümesi coffee_ratings_top olarak saklanır.

Proportional stratified sampling

Örneklemdeki her ülkenin oranlarının popülasyondakilerle yakından eşleşmesini önemseniyorsa, basit rastgele örneklem almadan önce veriler ülkelere göre gruplanabilir. Gruplamadan sonra sample() fonksiyonu çağrılarak, her ülke içinde basit bir rastgele örneklem alınır. Artık her ülkenin tabakalı örneklemdeki oranları popülasyondakilere çok daha yakındır.

Weighted random sampling

Tabakalı örneklemenin daha da fazla esneklik sağlayan yakın bir akrabası ağırlıklı rastgele örneklemedir.

- Her bir satırın örnekleme alınma olasılığını ayarlayan bir ağırlık sütunu oluşturulur.

Cluster Sampling

- Katmanlı örneklemeyle ilgili bir sorun, her alt gruptan veri toplanması gerekliliğidir.
- Örneğin, veri toplamanın pahalı olduğu durumlarda, örneği toplamak için fiziksel olarak bir yere seyahat edilmesi gerektiğinde, bu analizi aşırı pahalı hale getirebilir.
- Bunun için, cluster sampling adı verilen daha ucuz bir alternatif vardır.

Stratified Sampling vs Cluster Sampling

Stratified Sampling

- Popülasyonu alt gruplara bölünmelidir.
- Ardından her birinden basit rastgele örnekleme yapılır.

Cluster Sampling

- Bazı grupları seçmek için basit rastgele alt örnekleme kullanılır.
- Sadece bu alt gruplarda basit rastgele alt örnekleme kullanılır.

Multistage sampling (Çok aşamalı örnekleme)

Küme örneklemesi, çok aşamalı örneklemenin özel bir durumudur. İkiden fazla aşama kullanmak mümkündür. Yaygın bir örnek, eyaletler, ilçeler, şehirler ve mahalleler gibi çeşitli düzeylerde idari bölgeleri içerebilen ulusal anketlerdir.

Calculating mean cup points

Bir popülasyon parametresi, toplam fincan puanlarının ortalamasını hesaplayalım. Popülasyon parametresi bir alanın ortalaması olduğunda, buna genellikle popülasyon ortalaması denir. Gerçek yaşam senaryolarında, genellikle popülasyon ortalamasının ne olduğunu bilemeyeceğimizi unutmayın. Ancak burada olduğundan, bu seksen bir nokta dokuz değerini ölçmek için altın standart olarak kullanabiliriz. Şimdi, tartıştığımız örnekleme tekniklerinin her birini kullanarak aynı değeri hesaplayacağız. Bunlar, genellikle örnek ortalamaları olarak adlandırılan ortalama nokta tahminleridir. Basit ve tabakalı örnek ortalamaları popülasyon ortalamasına gerçekten yakındır. Kümeleme örneklemesi o kadar yakın değildir, ancak bu tipiktir. Kümeleme örneklemesi, daha az veri kullanırken neredeyse aynı derecede iyi bir yanıt vermek için tasarlanmıştır.

Relative error of point estimates (Nokta tahminlerinin göreli hatası)

Örneklem büyüklüğünün hesaplanan nokta tahminlerinin doğruluğunu nasıl etkilediğini göreceğiz.

Relative errors

Population Parameter

population_mean = df_coffee['total_cup_points'].mean()

Point Estimate

sample_mean = df_coffee.sample(n=sample_size)['total_cup_points'].mean()

Popülasyon ile örneklem ortalaması arasındaki farkı değerlendirmek için en yaygın ölçüm, **relative error** dur.

Relative Error, iki sayı arasındaki mutlak farktır; yani, eksi işaretler görmezden gelinir ve popülasyon ortalamasına bölünür. Burada, yüzdeye dönüştürmek için yüz ile çarpılır.

Relative Error as percentage

*relative_error_percentage = 100 * abs(population_mean - sample_mean) / population_mean*

Bağıl hatanın örneklem büyüklüğü arttıkça azaldığı görülmektedir. Öncelikle, mavi çizgi gerçekten gürültülü, özellikle küçük örneklem büyüklükleri için. Örneklem büyüklüğü küçükse, hesaplanan örneklem ortalaması, örneğe bir veya iki rastgele satır daha ekleyerek büyük ölçüde farklı olabilir. İkincisi, çizginin genliği, başlangıçta oldukça diktir. Küçük bir örneklem büyüklüğü olduğunda, sadece birkaç örneklem daha eklemek bize çok daha iyi doğruluk sağlayabilir. Grafiğin daha sağında, çizgi daha az diktir. Zaten büyük bir örneklem büyüklüğü varsa, örneğe birkaç satır daha eklemek o kadar fazla fayda sağlamaz. Son olarak, grafiğin en sağında, örneklem büyüklüğünün tüm popülasyon olduğu yerde, bağıl hata sıfıra düşer.

Bir örnekleme dağılımı oluşturma

Örnek ortalaması gibi nokta tahminlerinin, örnekte hangi satırların yer aldığına bağlı olarak nasıl değiştiği görüldü.

Örneğin, 30 kahveden oluşan basit bir rastgele örneklemden ortalama fincan puanlarını hesaplamak için kullanılan aynı kod, her seferinde biraz farklı bir cevap verir. Bu değişimi görselleştirmeye ve nicelemeye çalışalım.

Yaklaşık örnekleme dağılımları

Son uygulama, tekrar sayısının artırılmasının örneklem ortalamalarının bağıl hatasını etkilemediğini, aksine dağılımın daha tutarlı bir şekle sahip olmasını sağladığı görüldü.

Dört altı yüzlü zar atışı durumunu ele alalım. Pandas belgelerinde tanımlanan ve itertools paketini kullanan expand_grid işlevini kullanarak tüm olası atış kombinasyonlarını üretilebilir. Toplam 1296 zar atma kombinasyonu vardır.

Tam örnekleme dağılımı (Exact sampling distribution)
mean_roll dağılımını görmenin en iyi yolu bir çubuk grafiği çizmektir. Öncelikle, mean_roll kategoriye dönüştürülür ve bar grafiği çizilir. Bu, ortalama atışın tam örnekleme dağılımıdır çünkü her bir zar atış kombinasyonunu içerir.

Standart hatalar ve Merkezi Limit Teoremi (Standard errors and the Central Limit Theorem)

Gauss dağılımı (normal dağılım olarak da bilinir) istatistikte önemli bir rol oynar. Ayırt edici çan şeklindeki eğrisi bu ders boyunca ortaya çıkmıştır.

Merkezi limit teoreminin sonuçları

Yukarıda anlatılanşey, özünde, merkezi limit teoreminin bize söylediği şeydir. Bağımsız örneklerin ortalamaları normal dağılımlara sahiptir. Sonra, örneklem büyüklüğü arttıkça iki şey görülür. Bunlar; ortalamaların dağılımı normale yaklaşır ve bu örnekleme dağılımının genişliği daralır.

Popülasyon noktalarının standart sapmasını ele alalım. Yaklaşık 2.7'dir. Karşılaştırmak gerekirse, NumPy kullanarak her bir örnekleme dağılımından örnek ortalamalarının standart sapması alınırsa, çok daha küçük sayılar elde edilir ve bunlar örneklem boyutu arttıkça azalır. Pandas'ın std() ile bir popülasyon standart sapması hesaplandığında, std() varsayılan olarak bir örneklem standart sapmasını hesapladığından ddof'u sıfıra eşit olarak belirtmek gerekir. NumPy'nin std fonksiyonunu kullanarak popülasyonun bir örneği üzerinde bir standart sapma hesaplandığında, örnekleme dağılımındaki bu hesaplamalarda olduğu gibi, bir ddof belirtmelidir.

Karekök örneklem büyüklüğü üzerinden nüfus ortalaması

Merkezi limit teoreminin bir diğer sonucu da, popülasyon standart sapması örneklem büyüklüğünün karekköüne bölündüğünde, o örneklem büyüklüğü için örnekleme dağılımının standart sapmasının bir tahmininin elde edilmesidir. Örnekleme sürecinde yer alan rastgelelik nedeniyle tam olarak doğru değildir, ancak oldukça yakındır.

Standard error

Örneklem büyüklüğünün örnekleme dağılımının standart sapması üzerindeki etkisini görüldü. Örnekleme dağılımının bu standart sapmasının özel bir adı vardır: **standart hata**. Popülasyon standart sapmasını tahmin etmekten örnekleme sürecinden ne düzeyde değişkenlik bekleyeceğimize dair beklentiler belirlemeye kadar çeşitli bağlamlarda faydalıdır.

Standart hata (SE), bir tahmindeki değişkenliği veya belirsizliği, tipik olarak ortalama gibi bir popülasyon parametresinin tahminini niceleyen istatistiksel bir ölçüdür. Farklı örnekler çekilirse örnek ortalamasının (veya başka bir istatistiğin) gerçek popülasyon ortalamasından ne kadar farklılaşmasının beklendiğine dair bir gösterge sağlar.

Ortalamanın Standart Hatası (SEM): Standart hatanın en yaygın kullanımı ortalama bağlamındadır. Ortalamanın standart hatası (SEM), örnek ortalamasının popülasyon ortalamasının bir tahmini olarak kesinliğini niceliksel olarak ifade eder. Örnek ortalamasının popülasyon ortalamasından ne kadar farklılaşma olasılığının olduğunu söyler.

Bu formül, standart hatanın örneklem büyüklüğü arttıkça azaldığını, yani daha büyük örneklerin popülasyon ortalamasının daha kesin tahminlerini verdiğini göstermektedir.

Standart Sapmadan Farkı: Standart sapma, tek bir örneklemdeki veri noktalarının yayılımını ölçerken, standart hata, popülasyondan alınan farklı örneklerdeki örnek ortalamalarının değişkenliğini ölçer. Standart sapma, bireysel veri noktalarının ne kadar yayıldığını söylerken, standart hata, örnek istatistiğinizin gerçek popülasyon istatistiğinden ne kadar sapma olasılığının olduğunu söyler.

Yorumlama: Daha küçük bir standart hata, örnek ortalamasının gerçek popülasyon ortalamasına daha yakın olma olasılığını gösterir. Daha büyük bir standart hata, daha fazla değişkenlik olduğunu, yani örnek ortalamasının popülasyon ortalamasından daha uzak olabileceğini gösterir.

Standart Hatanın Kullanımları:

- Hipotez testinde, standart hata güven aralıklarını hesaplamak ve t-istatistik veya z-istatistik gibi istatistikleri test etmek için kullanılır.
- Güven aralıklarında, hata payı genellikle standart hatanın bir katıdır ve popülasyon parametresinin muhtemelen bulunacağı bir aralık sağlar.

Örnek

Diyelim ki ortalama boyu 170 cm ve standart sapması 10 cm olan 100 kişilik bir örneğimiz var. Ortalamanın standart hatası şu şekilde olacaktır:

Bu, 100 kişilik birçok örnek alırsak, örnek ortalamalarının ortalama olarak popülasyon ortalamasından yaklaşık 1 cm farklı olacağı anlamına gelir.

BootStrapping'e Giriş

Şimdiye kadar çoğunlukla yerine koymadan örnekleme üzerinde duruldu.

With or without

Yerine koymadan örnekleme, bir deste kart dağıtmaya benzer. Bir oyuncuya maça ası dağıtıldığında, maça ası başka bir oyuncuya dağıtılamaz. Yerine koymayla örnekleme, zar atmaya benzer. Altı atılırsa, bir sonraki atışta yine altı gelebilir. Yerine koymayla örneklemeye bazen yeniden örnekleme de denir.

Yerine koymadan basit rastgele örnekleme

Yerine koymadan basit bir rastgele örneklem aldındığında, veri kümesindeki her satır örneklemde yalnızca bir kez görünebilir.

Yerine koyma ile basit rastgele örnekleme

Yerine koyma ile örnekleme yapılırsa, veri kümesinin her satırının birden fazla kez örneklenebileceği anlamına gelir.

Neden yerine koyma ile örnekleme?

Şimdiye kadar coffee_ratings veri kümesi tüm kahvelerin popülasyonu olarak ele alındı. Elbette dünyadaki tüm kahveleri içermiyor, bu yüzden kahve veri kümesinin sadece büyük bir kahve örneği olarak ele alınması gerekir. Tüm popülasyonun nasıl olduğunu hayal etmek için, veri kümesinde olmayan diğer kahvelerin de yaklaşık olarak hesaplanması gerekir. Örnek veri kümesindeki kahvelerin her biri, sahip olunmayan kahveleri temsil eden özelliklere sahip olacaktır. Yeniden örnekleme, diğer teorik kahveleri yaklaşık olarak hesaplamak için mevcut kahveleri kullanmayı sağlar.

Basit tutmak için, kahve veri kümesinin üç sütununa odaklanılsın. Hangi satırların örneklemde yer aldığını görmeyi kolaylaştırmak için reset_index yöntemini kullanarak index adlı bir satır dizin sütunu eklenecek.

Resampling with .sample()

Yerine koyarak örneklemek için, her zamanki gibi sample() çağırılır. Ancak replace argümanı True olarak ayarlanır. frac'ı 1 olarak ayarlamak, orijinal veri kümesiyle aynı boyutta bir örnek üretir.

Repeated coffees

İndeks sütununun değerlerini saymak, her kahvenin yeniden örneklenen veri kümesinde kaç kez yer aldığını gösterir. Bazı kahveler dört veya beş kez örneklendi

Missing coffees

Bu, bazı kahvelerin yeniden örneklemede yer almadığı anlamına gelir. Yeniden örnekleme veri kümesindeki farklı endeks değerlerinin sayısını, drop_duplicates üzerinde len kullanarak alarak, 854 farklı kahvenin dahil edildiği görülür. Bu sayıyı toplam kahve sayısıyla karşılaştırarak, 484 kahvenin yeniden örneklemede yer almadığı görülebilir.

Bootstrapping

Bootstrapping adı verilen bir teknik için yeniden örneklemeyi kullanacağız. Bir bakıma, bootstrapping bir popülasyondan örneklemenin tam tersidir. Örneklemede, veri seti popülasyon olarak ele alınır ve daha küçük bir örneğe geçilir. Bootstrapping'de, veri seti bir örnek olarak ele alınır ve bu teorik bir popülasyon oluşturmak için kullanılır. Bootstrapping'in bir kullanım örneği, örneklemeden kaynaklanan değişkenliği anlamaya çalışmaktır. Bu, örnekleme dağılımı oluşturmak için popülasyonun birden çok kez örneklenemediği durumlarda önemlidir.

Bootstrapping Süreci

Bootstrapping süreci üç adımdan oluşur.

- Orijinal veri kümesiyle aynı boyutta bir yeniden örnekleme elde etmek için değiştirmeyle rastgele örnekleme yapılır.
- Sütunlardan birinin ortalaması gibi bir istatistik hesaplanır. Ortalamanın burada her zaman tercih olmadığını ve bootstrapping'in karmaşık istatistiklerin de hesaplanmasına izin verdiği unutulmamalıdır.
- Bu önyükleme istatistiklerinden çok sayıda elde etmek için ilk iki adım birçok kez tekrarlanır.

Daha önceki derslerde benzer bir şey yapıldı. Basit bir rastgele örnek alındı, ardından bir özet istatistiği hesaplandı ve ardından örnekleme dağılımı oluşturmak için bu iki adım tekrarlandı. Bu sefer örnekleme yerine yeniden örnekleme kullanıldığında, bir önyükleme dağılımı (bootstrap distribution) elde edilir.

Bootstrapping coffee mean flavor

Yeniden örnekleme adımı için az önce görülen kod kullanılır: frac'ı bire ve replace'ı True'ya ayarlayarak sample() çağırmak. Bir bootstrapp istatistiğinin hesaplanması NumPy'den ortalama ile yapılabilir. Bu durumda, ortalama lezzet puanı hesaplanır. Birinci ve ikinci adımları bin kez tekrarlamak için kodu bir for döngüsüne alınır ve istatistikler bir listeye eklenir.

Sample sd vs. bootstrap distribution sd

Popülasyon ortalamasını tahmin etmede bu sınırlamaya sahip olunsa da, dağılımlar hakkında harika bir şey de varyasyonun da nicelleştirilebilmesi-dir. Örnek tatlarının standart sapması 0.343 civarındadır. Eğer bootstrap dağılımının standart sapması hesaplanırsa ve ddof bir olarak belirlenirse, o zaman tamamen farklı bir sayı elde edilir. Peki burada neler oluyor?

ample, bootstrap dist'n, pop'n standard deviations

Bootstrapp'in bir amacının, bir örnekten diğerine geçerken örnek istatistiğinde ne tür değişkenlik beklenebileceğini ölçmektir. Bu nicelik, o istatistik örnekleme dağılımının standart sapmasıyla ölçülen standart hata olarak adlandırılır. Bootstrapp ortalamalarının standart sapması, bu belirsizlik ölçüsünü tahmin etmenin bir yolu olarak kullanılabilir. Bu standart hatayı örneklem büyüklüğünün kareköküyle çarpılırsa, orijinal popülasyondaki standart sapmanın bir tahmini elde edilir. Standart sapma tahmini 0,346. Gerçek standart sapma 0,343, bu nedenle tahmin oldukça yakındır. Aslında, yalnızca örneklem standart sapmasını kullanmaktan daha yakındır.

Interpreting the standard errors

Tahmini standart hata, ilgili istatistik için bootstrap dağılım değerlerinin standart sapmasıdır. Bu tahmini standart hata ile örneklem büyüklüğünün karekökünün çarpımı, popülasyonun standart sapmasının gerçekten iyi bir tahminini verir. Yani, bootstrapping popülasyon ortalamasını tahmin etmede zayıf olsa da, popülasyon standart sapmasını tahmin etmede genellikle harikadır.

Confidence intervals

Bu dağılımları ölçmenin bir yolu, bir dağılımdaki değerlerin çoğunun nerede bulunduğuna dair iyi bir fikir veren “ortalamanın bir standart sapması içindeki değerler” fikridir. Burada, “güven aralığı” terimi tanımlanarak bir istatistiğe yakın değerler hakkında bilgi verilecektir.

Predicting the weather

Dünyanın en öngörülemez bölgelerinden biri olan ABD ve Kanada'nın kuzeyindeki Büyük Ovalar'da hava tahminleri yapan meteorologları düşünün. Rapid City, Güney Dakota, Ulusal Hava Durumu Servisi tahmin ofisine sahip 120 ABD şehri arasında en az tahmin edilebilir şehir olarak sıralanmıştır. Rapid City'deki bir haber kanalında meteoroloji uzmanı olarak işe başladığımızı varsayalım. İşimiz yarınki yüksek sıcaklığı tahmin etmek.

Our weather prediction

Elimizdeki en iyi tahmin araçlarını kullanarak hava durumu verilerini analiz ediyoruz ve 47 Fahrenheit derecelik bir yüksek sıcaklık tahmin ediyoruz. Bu durumda 47 derece bizim nokta tahminimizdir. Hava durumu değişken olduğundan ve birçok Güney Dakotalı yarınki gününü bizim tahminimize göre planlayacağından, bunun yerine yüksek sıcaklık için bir dizi makul değer sunmak istiyoruz. Hava durumu programımızda, yarın yüksek sıcaklığın kırk ila elli dört fahrenheit arasında olacağını bildiriyoruz.

We just reported a confidence interval!

Kırk ila elli dört fahrenheit'lik bu tahmin, yarının yüksek sıcaklığının bilinmeyen miktarı için bir güven aralığı olarak düşünülebilir. Tam sıcaklıktan emin olamasak da, bu aralıkta olacağından eminiz. Bu sonuçlar genellikle nokta tahmini ve ardından parantez veya köşeli parantez içinde güven aralığının alt ve üst sınırları olarak yazılır. Güven aralığı nokta tahmini etrafında simetrik

olduğunda, bunu nokta tahmini artı veya eksi hata payı, bu durumda yedi derece olarak gösterebiliriz.

47°F(40°F, 54°F) veya 47°F[40°F, 54°F] veya 47±7°F

Ortalama artı veya eksi bir standart sapma

Ortalamaya bir standart sapma ekleyip çıkararak bir güven aralığı oluştulursa, bootstrap dağılımında bu bir standart sapma güven aralığının dışında çok sayıda değer olduğu görülür.

Quantile method for confidence intervals

Değerlerin yüzde doksan beşi güven aralığına dahil edilmek isteniyorsa, kantiller kullanılabilir. Kantillerin dağılımları toplam verinin belirli bir oranını içeren bölümlere ayırmaktadır. Değerlerin ortadaki yüzde doksan beşini elde etmek için, 0.025 kuantilinden 0,975 kuantiline gidilir, çünkü bu iki sayı arasındaki fark 0.95'tir. Bu güven aralığının alt ve üst sınırlarını hesaplamak için NumPy'den quantile çağrılır, dağılım değerlerini ve kullanılacak quantile değerleri elde edilir.

Inverse cumulative distribution function (Ters kümülatif dağılım fonksiyonu)

Güven aralıklarını hesaplamak için ikinci bir yöntem daha vardır. Bunu anlamak için normal dağılımın ters kümülatif dağılım fonksiyonuna aşina olmak gerekir. Daha önce görülen çan eğrisi olasılık yoğunluk fonksiyonu ya da PDF'dir. Kalkülüs kullanarak bu integre edilirse, kümülatif dağılım fonksiyonu veya CDF elde edilir. Eğer x ve y eksenleri ters çevirilirse, ters CDF elde edilir. Ters CDF'yi elde etmek için scipy.stats kullanabilir ve norm.ppf çağırılabilir. Sıfır ile bir arasında bir nicelik alır ve bu nicelik için normal dağılımın değerlerini döndürür. Loc ve scale parametreleri varsayılan olarak standart normal dağılıma karşılık gelecek şekilde 0 ve 1 olarak ayarlanmıştır.

tandard error method for confidence interval

Güven aralığını hesaplamak için kullanılan bu ikinci yönteme standart hata yöntemi denir. İlk olarak, bootstrap dağılımının ortalaması olan nokta tahminini ve bootstrap dağılımının standart sapması ile tahmin edilen standart hata hesaplanır. Ardından, bootstrap dağılımıyla aynı ortalama ve standart sapmaya sahip normal dağılımın ters CDF'sini elde etmek için norm.ppf çağırılır. Yine, güven aralığı 7.78 ile 7.52 arasındadır, ancak bootstrapp dağılımı tamamen normal olmadığı için sayılar geçen seferkinden biraz farklıdır.

Hypothesis tests and z-scores

A/B testing

- 2013 yılında Electronic Arts veya EA, SimCity 5 adlı bir video oyunu piyasaya sürdü.
- Oyunun piyasaya sürülmesine kadar, ön sipariş satışlarını artırmak istediler. Hipotez testinde kökleri olan A/B testi adı verilen deneysel bir tasarım tekniği kullanarak farklı reklam senaryolarını test ettiler ve hangisinin satışları en çok artırdığını gördüler.
- Web sitesi ziyaretçileri bir kontrol grubu ve bir tedavi grubu (treatmant group) olarak ayrıldı. Her grup, oyunun ön sipariş satış sayfasının farklı bir versiyonunu gördü.

A/B testing results

- A/B testinin sonuçları şaşırtıcıydı.
- Reklamın olmadığı tedavi sayfası, reklamın olduğu kontrol sayfasından %43 daha fazla satışla sonuçlandı.

- Deney, daha fazla indirim reklamının daha fazla satışla sonuçlanacağı yönündeki sezginin yanlış olduğunu kanıtladı.
- Peki: %43'lük fark, kontrol ve tedavi grupları arasında anlamlı bir fark mıydı yoksa sadece rastgele bir şans mıydı?
- Bu cevabı elde etmek için, kamuya açık olmayan EA'nın orijinal veri kümesine ihtiyaç vardır.

Stack Overflow Developer Survey 2020

Stack Overflow her yıl, çoğunlukla yazılım geliştiricileri olan kullanıcılarına, Stack Overflow'u nasıl kullandıkları, işleri ve kullandıkları geliştirme araçları hakkında anket yapar. Şimdi, Veri Bilimcileri olarak tanımlanan kullanıcıların anket yanıtlarının bir alt kümesine bakacağız.

z-scores

Değişkenler keyfi birimlere ve aralıklara sahip olduğundan, hipotez test edilmeden önce değerlerin standartlaştırılması gerekir. Değerleri standartlaştırmanın yaygın bir yolu ortalamayı çıkarmak ve standart sapmaya bölmektir. Hipotez testi için, örnek istatistiğin alındığı, varsayılan parametre değerinin çıkarıldığı ve standart hataya bölündüğü bir varyasyon kullanılır. Sonuç z-skoru olarak adlandırılır.

$$\text{Standardized Value} = \frac{\text{Value} - \text{Mean}}{\text{Standard Deviation}}$$

$$z = \frac{\text{sample stat} - \text{hypoth.param.value}}{\text{Standard error}}$$

Standard normal (z) distribution

Aşağıda, ortalaması sıfır ve standart sapması bir olan normal bir dağılım olan standart normal dağılım için olasılık yoğunluk fonksiyonunun bir grafiği bulunmaktadır. Genellikle z-dağılımı olarak adlandırılır ve z-skorları bu dağılımla ilişkilidir

p-values

Hipotez testleri ceza davalarına benzer.

Ceza davaları

- İki olası gerçek durum vardır: sanık ya suçu işlemiştir ya da işlememiştir.
- Ayrıca iki olası sonuç vardır: suçlu veya suçsuz kararı.
- İlk varsayım sanığın suçsuz olduğudur ve savcılık ekibinin sanığın suçu işlediğine dair makul şüphenin ötesinde kanıt sunması gerekir ki suçlu kararı verilsin.

İlk programlama deneyiminin yaşı

Stack Overflow anketine tekrar bakılsın. age_first_code_cut değişkeni kullanıcının programlamaya ne zaman başladığını sınıflandırır. 14 yaşında veya daha büyüklerse yetişkin (adult) olarak sınıflandırılırlar; aksi takdirde çocuk (child) olarak sınıflandırılırlar. Önceki araştırmalar yazılım geliştiricilerinin %35'inin çocukken programlama yaptığını öne sürmüş olsun. Bu, eldeki veri

setiyle cevaplanabilecek bir soru ortaya çıkarır. Eldeki örnek, veri bilimcilerinin daha büyük bir oranının çocukken programlamaya başladığına dair kanıt sağlıyor mu?

Tanımlar
Bazı tanımları belirleyelim. .

- Bir hipotez, bir popülasyon parametresi hakkındaki bir ifadedir.
- Bu popülasyon parametresinin gerçek değeri bilinmemektedir; yalnızca verilerden çıkarımlarda bulunulabilir.
- Hipotez testleri, iki rakip hipotezi karşılaştırır. Bu iki hipotez, mevcut fikri temsil eden sıfır hipotezi ve mevcut fikre meydan okuyan yeni bir fikri temsil eden alternatif hipotezdir.
- Bunlar sırasıyla H

ve H

- ile gösterilir.
- Burada, sıfır hipotezi, çocukken programlamaya başlayan veri bilimcilerinin oranının yazılım geliştiricileri hakkındaki araştırmayı %35 oranında takip etmesidir.
- Alternatif hipotez ise bu oranın %35'ten büyük olmasıdır.

H

: Çocukken programlamaya başlayan veri bilimcilerinin oranı %35'tir.
H

: Çocukken programlamaya başlayan veri bilimcilerinin oranı %35'in üzerindedir.

Ceza davaları ve hipotez testleri

- Ceza davası karşılaştırmasına geri dönülürse, sanık suçlu veya suçsuz olabilir ve aynı şekilde, hipotezlerden yalnızca biri doğru olabilir.
- Başlangıçta, sanığın suçsuz olduğu varsayılır ve benzer şekilde, başlangıçta sıfır hipotezinin doğru olduğu varsayılır. Bu, yalnızca örnek bunu reddetmek için yeterli kanıt sağlarsa değişir.
- Alternatif hipotezin kabul edildiğini söylemek yerine, sıfır hipotezini reddetmek veya sıfır hipotezinin reddedilmesi gelenekseldir.
- Kanıt, sanığın suçu işlediğine dair "makul şüphenin ötesinde" ise, o zaman "suçlu" kararı verilir. "Makul şüphenin ötesinde" hipotez testine eşdeğer olan şey, önem düzeyi olarak bilinir.

One-tailed and two-tailed tests

```
 Stack Overflow hipotez testi için, sağ kuyrukta uç değerler aradığımızdan sağ
kuyruklu bir teste ihtiyacımız var.
```

- Bir dağılımın kuyrukları, PDF'sinin sol ve sağ kenarlarıdır.
- Hipotez testleri, örneklem istatistiklerinin sıfır hipotezin doğru olması durumunda istatistiğin dağılımı olan sıfır dağılımının kuyruklarında yer alıp almadığını belirler.
- Üç tür test vardır ve alternatif hipotezin ifadesi hangi türün kullanılması gerektiğini belirler.
- Varsayılan bir değere kıyasla bir fark olup olmadığı kontrol ediliyorsa, her iki kuyrukta da uç değerler aranır ve iki kuyruklu bir test gerçekleştirilir.

- Alternatif hipotezde “ az” veya “daha az” gibi ifadeler kullanılıyorsa, sol kuyruklu bir test uygulanır.
- “Daha büyük” veya ‘aşıyor’ gibi kelimeler sağ kuyruklu bir teste karşılık gelir.
- Stack Overflow hipotez testi için, sağ kuyrukta uç değerler arandığından sağ kuyruklu bir teste ihtiyaç vardır.

p-değerleri, sıfır hipotezi için desteğin gücünü ölçer veya başka bir deyişle, sıfır hipotezi doğru olduğu varsayıldığında bir sonuç elde etme olasılığını ölçer. Büyük p-değerleri, istatistiğimizin sıfır dağılımımızın kuyruğunda olma olasılığı düşük bir sonuç ürettiği anlamına gelir ve şans, sonuç için iyi bir açıklama olabilir. Küçük p-değerleri, istatistiğimizin sıfır dağılımımızın kuyruğunda olma olasılığı yüksek bir sonuç ürettiği anlamına gelir. p-değerleri olasılıklar olduğundan, her zaman sıfır ile bir arasındadır.

Calculating the z-score
p-değerini hesaplamak için önce z-puanı hesaplanmalıdır. Örnek istatistiği, bu durumda çocukken programlamaya başlayan veri bilimcilerin oranı hesaplanır. Sıfır hipotezinden varsayılan değer yüzde 35'tir. Standart hata bootstrap dağılımının standart sapmasından alınır ve z-puanı oranlar arasındaki farkın standart hataya bölünmesidir.

Calculating the p-value

Z-puanını, varsayılan sıfır ortalama ve bir standart sapma değerleriyle standart normal CDF'ye, norm.cdf()'ye aktarılır. Sol kuyruk testi değil, sağ kuyruk testi gerçekleştirildiği için, p-değeri norm.cdf sonucu bir eksilterek hesaplanır.

Statistical significance

p-value özeti

- p-değerleri, sıfır hipotezi için ne kadar kanıt olduğunu ölçer.
- Büyük p-değerleri, alternatif hipotez için kanıt eksikliğini gösterir, bunun yerine varsayılan sıfır hipotezine bağlı kalınır.
- Küçük p-değerleri, alternatif hipotez lehine bu orijinal varsayımdan şüphe etmeye neden olur.
- Küçük bir p-değeri ile büyük bir p-değeri arasındaki kesme noktasını(cutoff) ne belirler?

Significance level(Anlamlılık Düzeyi)

- Kesme noktası(cutoff) anlamlılık düzeyi olarak bilinir ve alfa(α) ile gösterilir.
- Uygun anlamlılık seviyesi veri setine ve çalışılan disipline bağlıdır.
- Yüzde beş en yaygın seçimdir, ancak yüzde on ve yüzde bir de popülerdir.
- Anlamlılık düzeyi hangi hipotezin destekleneceğine dair bir karar süreci sunar.
- Eğer p-değeri alfa değerinden küçük ya da eşitse, sıfır hipotezi reddedilir. Aksi takdirde, reddedilir.
  Eğer $p \leq \alpha$ Ho reddedilir aksi takdirde Ho reddedilemez.
- Test yapılmadan önce uygun anlamlılık düzeyinin ne olması gerektiğine karar verilmesi önemlidir.
- Aksi takdirde, istenilen hipotezi seçmeye olanak tanıyan bir anlamlılık düzeyine karar verme eğilimi ortaya çıkar.

p-value hesaplama

- İş akışı, anlamlılık düzeyinin ayarlanmasıyla başlar, bu durumda 0.05.
- Ardından, örnek ortalaması hesaplanır ve varsayılan ortalama belirlenir
- Z-skoru için, bootstrap dağılımından elde edilen standart hataya da ihtiyaç vardır.
- Daha sonra örnek ortalaması, varsayılan ortalama ve standart hata kullanarak z-skoru hesaplanor ve p-değerini elde etmek için standart normal CDF kullanılır.

Confidence intervals (Güven Aralıkları)

Popülasyon parametresinin potansiyel değerleri hakkında bir fikir edinmek için, anlamlılık düzeyinin bir eksiği (1 - α) kadar bir güven aralığı düzeyi seçmek yaygındır.

0.05 puanlık bir anlamlılık düzeyi için yüzde 95'lik bir güven aralığı kullanılır.

Bu kantil yöntemi kullanılarak yapılan hesaplamaya bir örnektir. Bu aralık, çocukken programlama yapan veri bilimcilerin popülasyon oranı için bir dizi makul değer sağlar.

Hata Türleri

Ceza davası benzetmesine dönülecek olursa, iki olası doğruluk durumu ve iki olası test sonucu vardır, bu da dört kombinasyon anlamına gelir.

	Truly didn't commit crime	Truly committed crime
Verdict not guilty	correct	they got away with it
Verdict guilty	wrongful conviction	correct

Bunlardan ikisi kararın doğru olduğunu gösterir.

Sanık suçu işlemediği halde suçlu olduğuna karar verilmişse, haksız yere mahkum edilmiştir. Sanık suçu işlemiş ancak karar suçsuz ise, suçu yanına kar kalmıştır. Bunların ikisi de adalette yapılan hatalardır.

Benzer şekilde, hipotez testi için de doğru yapmanın iki yolu ve iki tür hata vardır.

Sıfır hipotezi doğruyken alternatif hipotez desteklenirse, yanlış pozitif hata yapılmış olur. Alternatif hipotez doğruyken boş hipotez desteklenirse, yanlış negatif hata yapılmış olur. Bu hatalar bazen sırasıyla birinci tip ve ikinci tip hatalar olarak bilinir.

	actual H_0	actual H_A
chosen H_0	correct	false negative
chosen H_A	false positive	correct

Yapılan örnekteki olası hatalar

Veri bilimcilerin çocuk olarak kodlama yapması durumunda, anlamlılık düzeyine eşit veya daha düşük bir p-değeri elde edildiyse ve sıfır hipotezi reddedildiyse, yanlış pozitif bir hata yapılmış olabilir. Veri bilimcilerin daha yüksek oranda çocuk olarak kodlamaya başlandığı düşünmesine rağmen, bu tüm popülasyon için doğru olmayabilir. Tersine, p-değeri anlamlılık düzeyinden büyükse ve sıfır hipotezi reddedilemediyse, yanlış negatif bir hata yapılmış olabilir.

t-testlerinin gerçekleştirilmesitek bir değişken için bir test istatistiği olan z-skorunu hesaplamıştık.

Bir önceki bölümde, tek bir değişken için bir test istatistiği olan z-skorunu hesaplama analtılmıştı.

İki örneklemli problemler

- Burada, bir değişkendeki gruplar arasında örnek istatistiklerinin karşılaştırılmasıyla ilgilenilecektir.
- Stack Overflow veri setinde, converted_comp sayısal bir yıllık ücret değişkenidir.
- age_first_code_cut ise iki seviyeli kategorik bir değişkendir: kullanıcının programlamaya ne zaman başladığını tanımlayan çocuk ve yetişkin
- yaş grubu arasındaki ücret farklılıkları hakkında sorular sorulabilir. Örneğin, ilk olarak çocukken programlama yapan kullanıcılar, yetişkin olarak başlayanlara göre daha mı iyi ücret alıyor?

Test İstatistikleri

Popülasyon ortalamasını bilinmese de, örneklem ortalaması kullanarak tahmin edilebilir.

bir örneklem ortalamasını belirtmek için kullanılır. Daha sonra örnek ortalamasının hangi gruba karşılık geldiğini belirtmek için alt simgeler kullanılır. Bu iki örneklem ortalaması arasındaki fark, hipotez testi için test istatistiğidir. Daha önce anlatılan z-skorları bir tür standartlaştırılmış test istatistiğidir.

t-istatistiklerinden p-değerlerinin hesaplanması

t-dağılımları

Test istatistiği, t, bir t-dağılımını takip eder. t-dağılımlarının serbestlik derecesi veya kısaca df olarak adlandırılan bir parametresi vardır. Burada, bir serbestlik dereceli t-dağılımının PDF'sini sarı renkle ve normal dağılımın PDF'sini mavi çizgilerle gösteren bir çizgi grafiği yer almaktadır. Küçük serbestlik dereceleri için t-dağılımının normal dağılıma göre daha kalın kuyruklara sahip olduğuna dikkat edilmelidir, ancak bunun dışında benzer görünüyorlar.

Degrees of freedom

Serbestlik derecesini artırdıkça, t-dağılımı normal dağılıma yaklaşır. Aslında normal dağılım, sonsuz serbestlik derecesine sahip bir t-dağılımıdır. Serbestlik derecesi, veri örneğindeki mantıksal olarak bağımsız değerlerin maksimum sayısı olarak tanımlanır. Bu oldukça zor bir kavramdır, bu yüzden bir örnek deneyelim.

erbestlik derecelerinin hesaplanması

Veri kümemizde 5 bağımsız gözlem olduğunu ve değerlerden dördünün 2, 6, 8 ve 5 olduğunu varsayalım. Bu bilgiyle, beşinci değer artık bağımsız değildir; 4 olmalıdır. Örneklemdeki beş

gözlemin hepsi bağımsız olsa da, örneklem hakkında ek bir gerçeği bildiğimiz için - ortalaması 5 olduğu için - sadece 4 serbestlik derecesine sahibiz. İki örneklemli durumumuzda, gözlem sayısı kadar serbestlik derecesi vardır, eksi iki çünkü iki örneklem istatistiğini, her grubun ortalamasını biliyoruz.

df =

- 2

Hipotezler

: Ortalama maaş(ABD Doları cinsinden), ilk olarak çocuk olarak kodlayanlar ve ilk olarak yetişkin olarak kodlayanlar için aynıdır.

: Ortalama maaş(ABD Doları cinsinden), önce çocuk olarak kodlayanlar için önce yetişkin olarak kodlayanlara kıyasla daha yüksektir.

Bu bir "daha büyük" alternatif hipotezi olduğundan, sağ kuyruklu bir teste ihtiyaç vardır.

Significance level

Şimdi bir p-değeri hesaplayacağız, ancak önce bir anlamlılık düzeyine karar vermemiz gerekiyor. Birkaç olasılık var; 0.1'i kullanalım. Bu, p-değerinin 0.1'den küçük veya ona eşit olması durumunda alternatif lehine boş hipotezi reddedeceğimiz anlamına gelir.

P-değerlerinin hesaplanması: bir orana karşı bir değer

Daha önce, p-değerini elde etmek için z-skoru normal CDF ile dönüştürüldü. Sağ kuyruklu bir test olduğu için sonucu birden çıkardık. Yğne daha önce, örneklem bilgisini kullanarak test istatistiği standart hatası için bir yaklaşım kullanıldı. Bu yaklaşımı kullanmak daha fazla belirsizlik ekler ve bu yüzden bu bir z problemi yerine bir t problemidir. t dağılımı, tek bir istatistik hesaplamasında birden fazla tahmin kullanıldığında daha fazla belirsizliğe izin verir. Burada, çoklu tahminler örnek ortalamasına ve örnek standart sapmasına karşılık gelmektedir.

```
from scipy.stats import norm
1 - norm.cdf(z_score)
```

z-statistic : Bir popülasyon parametresini tahmin etmek için bir örnek istatistiği kullanırken gereklidir.
t-statistic : Bir popülasyon parametresini tahmin etmek için çoklu örneklem istatistiği kullanırken gereklidir.

Paired t-tests

Daha önce, iki grup arasındaki ortalamaların farkına ilişkin standartlaştırılmış bir test istatistiğinden bir p-değeri hesaplamak için t-dağılımı kullanılmıştı.

US Republican presidents dataset

ABD başkanlık seçimlerinin bir veri kümesidir. Her satır ilçe düzeyinde bir başkanlık seçimini temsil etmektedir. Veri setindeki değişkenler ABD eyaleti, bu eyalet içindeki ilçe ve 2008 ve 2012'de Cumhuriyetçi adaya verilen oy yüzdesidir.

ttest() kullanarak iki ortalama arasındaki farkları test etme

Pingouin paketi, hipotez testi için çeşitli farklı yöntemler sağlar ve sonuçları pandas DataFrame olarak döndürür. Çıktısı, scipy-dot-stats'ın benzer yöntemlerine göre çalışmak için biraz daha dostça olabilir. Pingouin'den bir yöntem ttest'tir ve dizi benzeri nesnelerle çalışır, bu nedenle ilk argüman Farklar Serisidir. Bunun gibi dönüştürülmüş bir tek örnek testi için y, sıfır olan boş hipotezden varsayılan fark değerini belirtir. Alternatif hipotez türü, sırasıyla iki kuyruklu, sol kuyruklu ve sağ kuyruklu testlere karşılık gelen iki taraflı, daha az veya daha büyük olarak belirtilebilir.Çıktısı. Test istatistiğinin değerini, serbestlik derecesini, alternatif yönü ve p-değerini tanıyabiliriz. Ek çıktı, bu dersin kapsamı dışında kalan daha gelişmiş istatistiksel kavramları ifade eder.

paired=True ile ttest()

Eşleştirilmiş veriler için ttestinin daha da az çalışma gerektiren bir varyasyonu vardır. İki eşleştirilmiş değişken arasındaki farkı hesaplamak yerine, her ikisini de doğrudan ttestine x ve y olarak geçirebilir ve paired değerini True olarak ayarlayabiliriz. İlk dört sütundaki sonuçların öncekiyle aynı olduğuna dikkat edin.

Unpaired ttest()

Eşleştirilmiş seçeneğini Doğru olarak ayarlamazsak ve bunun yerine eşleştirilmemiş bir t-testi yaparsak, sayılar değişir. Test istatistiği sıfıra daha yakındır, daha fazla serbestlik derecesi vardır ve p-değeri çok daha büyüktür. Verilerimiz eşleştirilmişken eşleştirilmemiş bir t-testi yapmak yanlış negatif hata olasılığını artırır.

ANOVA tests

Eşleştirilmemiş ve eşleştirilmiş durumlarda iki grubu nasıl karşılaştıracağı anlatıldı. Peki ya ikiden fazla grup varsa?

Çoklu dağılımları görselleştirme

Ortalama yıllık ücretin her bir iş tatmini seviyesi için farklı olup olmadığının bilinmek istendiği varsayılsın. Yapılacak ilk şey dağılımları kutu grafikleri ile görselleştirmektir. Seaborn'un boxplot yöntemi burada stack_overflow verilerini kullanarak yatay eksende converted_comp ve dikey eksende job_sat ile güzel bir seçenek sunmaktadır. “Çok memnun” diğerlerinden biraz daha yüksek görünüyor, ancak önemli ölçüde farklı olup olmadıklarını görmek için hipotez testleri kullanmak gerekmektedir.

Analysis of variance (ANOVA)

ANOVA testleri gruplar arasında farklılık olup olmadığını belirler. Anlamlılık düzeyi 0.2 olarak belirlendi. Bu değer birçok durumda olduğundan daha büyüktür, ancak daha sonra farklı sayıdaki grupları karşılaştırmanın sonuçlarını anlamaya yardımcı olacaktır. Birden fazla gruptaki değerleri karşılaştırmak için pingouin anova yöntemi kullanılacak. Verileri stack_overflow, bağımlı değişken olan dv'yi converted_comp ve aralarında hesaplama yapılacak grup sütunu job_sat olarak belirlenir. p-değeri, p-unc sütununda saklanır; bu sütun, alfa değerinden yüzde 20 daha küçük olan 0.0013 noktasıdır. Bu, iş tatmini kategorilerinden en az ikisinin ücret düzeyleri arasında önemli farklılıklar olduğu anlamına gelir, ancak bu bize hangi iki kategori olduğunu söylemez.

pairwise_tests()

Tüm bu hipotez testlerini tek seferde çalıştırmak için pairwise_tests yöntemi kullanıbilir. İlk üç argüman olan data, dv ve between anova metodu ile aynıdır. p-adjust'ı kısa bir süre sonra tartışacağız. Sonuç, A ve B'nin her satırda karşılaştırılan iki seviye olduğu bir DataFrame gösterir. Daha sonra, p-değerlerinin p-unc sütununa bakılır. Bunlardan üçü anlamlılık düzeyimiz olan nokta-ikiden küçüktür.

Grup sayısı arttıkça...

Bu durumda beş grup var ve sonuçta on çift elde edilir. Grup sayısı arttıkça, çift sayısı - ve dolayısıyla gerçekleştirilmesi gereken hipotez testi sayısı - dört kat artar. Ne kadar çok test yapılırsa, bunlardan en az birinin yanlış pozitif anlamlı sonuç verme olasılığı o kadar yüksek olur. Anlamlılık düzeyi 0.2 olduğunda, bir test yapılırsa, yanlış pozitif sonuç olasılığı 0.2. Beş grup ve on testle, en az bir yanlış pozitif olasılığı yaklaşık 0.7'dir. Yirmi grupla, en az bir yanlış pozitif alacağımız neredeyse garantidir.

Bonferroni correction
Bunun çözümü, p-değerlerini artırmak için bir ayarlama uygulamak ve yanlış pozitif alma şansını azaltmaktır. Yaygın bir ayarlama Bonferroni düzeltmesidir. Düzeltilmemiş p-unc sütununun aksine, düzeltilmiş p-değerlerine karşılık gelen p-corr sütununa bakıldığında, çiftlerden yalnızca ikisinin önemli farklılıklara sahip olduğu görülmektedir.

One-sample proportion tests

Hipotez testleri, bilinmeyen bir popülasyon oranının belli bir değere eşit olup olmadığını ölçtüğü anlatılmıştı. Örneklem istatistiğinin standart hatasını tahmin etmek için örneklem üzerinde bootstrapping kullanılmıştı. Standart hata daha sonra standartlaştırılmış bir test istatistiği olan z-skorunu hesaplamak için kullanıldı, bu da bir p-değeri elde etmek için kullanıldı, böylece boş hipotezin reddedilip reddedilmeyeceğine karar verilebildi. Bir bootstrap dağılımını hesaplamak hesaplama açısından yoğun olabilir, bu nedenle bu sefer test istatistiğini hesaplamak yerine bootstrap olmadan hesaplama yapılacaktır.

Standardized test statistic for proportions

Bir oran veya kısaca popülasyon oranı olan bilinmeyen bir popülasyon parametresi p ile gösterilir. Örneklem oranı p-hat ile gösterilir ve popülasyon oranı için varsayılan değer p-sıfır ile gösterilir. Daha önce anlatıldığıu gibi, standartlaştırılmış test istatistiği bir z-skorudur. Bu, örneklem istatistiği ile başlayıp ortalaması çıkarılarak ve ardından standart hatasına bölünerek hesaplanabilir. p-hat eksi p-hat ortalaması, p-hat standart hatasına bölünür. Python'da Örnekleme'den, p-hat ile gösterilen örnek ortalamalarının örnekleme dağılımının ortalamasının p, yani popülasyon oranı olduğunu hatırlayın. Sıfır hipotezi altında, bilinmeyen p oranının varsayılan popülasyon oranı p-sıfır olduğu varsayılır. Z-skoru artık p-hat eksi p-sıfırdır ve p-hat'in standart hatasına bölünür.

p: population proportion (unknown population parameter)

$\hat{p}$: sample proportion (sample statistic)

p_0: hypothesized population proportion

$$z = \frac{\hat{p} - \text{mean}(\hat{p})}{\text{SE}(\hat{p})} = \frac{\hat{p} - p}{\text{SE}(\hat{p})}$$

Assuming H_0 is true, $p = p_0$, so

$$z = \frac{\hat{p} - p_0}{\text{SE}(\hat{p})}$$

Simplifying the standard error calculations

Oranlar için, H-naught altında, p-hat denkleminin standart hatası p-sıfır çarpı bir eksi p-sıfır olarak basitleştirilebilir, gözlem sayısına bölünür ve sonra kare köklü hale getirilir. Bunu z-skor denkleminin yerine konulabilir. Bunu hesaplamak daha kolaydır çünkü sadece örneklemden elde edilen p-hat ve n ile bizim seçtiğimiz p-sıfır değerlerini kullanır.

$SE_{\hat{p}} = \sqrt{\frac{p_0 * (1 - p_0)}{n}}$ → Under H_0, $SE_{\hat{p}}$ depends on hypothesized p_0 and sample size n

Assuming H_0 is true,

$$z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0 * (1 - p_0)}{n}}}$$

- Only uses sample information ($\hat{p}$ and n) and the hypothesized parameter (p_0)

Neden t yerine z?

Neden burada z-dağılımını kullanıldığı, ancak Bölüm 2'de t-dağılımının kullanıldığı merak edilebilir. Bu, iki örneklem ortalaması durumu için test istatistiği denklemidir. Örneklemin standart sapması olan s, örneklem ortalaması olan x-bar'dan hesaplanır. Bu da x-bar'ın payda popülasyon ortalamasını tahmin etmek için, payda ise popülasyon standart sapmasını tahmin etmek için kullanıldığı anlamına gelir. Bu ikili kullanım, popülasyon parametresi tahminimizdeki belirsizliği artırır. t-dağılımları etkin bir şekilde daha kalın kuyruklu bir normal dağılım olduğundan, bu ekstra belirsizliği hesaba katmak için bunlar kullanılabilir. Aslında, t-dağılımı boş hipotezin yanlışlıkla reddedilmesine karşı ekstra tedbir sağlar. Oranlar için, payda yalnızca p-hat kullanılır, böylece belirsizlikle ilgili sorundan kaçınılır ve bir z-dağılımı uygundur.

$$t = \frac{(\bar{x}_{\text{child}} - \bar{x}_{\text{adult}})}{\sqrt{\frac{s^2_{\text{child}}}{n_{\text{child}}} + \frac{s^2_{\text{adult}}}{n_{\text{adult}}}}}$$

- s is calculated from $\bar{x}$
  - $\bar{x}$ estimates the population mean
  - s estimates the population standard deviation
  - $\uparrow$ uncertainty in our estimate of the parameter
- t-distribution - fatter tails than a normal distribution
- $\hat{p}$ only appears in the numerator, so z-scores are fine

Proportion tests using proportions_ztest()

Statsmodels'in proportions_ztest fonksiyonu z-skorunu daha doğrudan hesaplayabilir. Bu fonksiyon NumPy dizileri olarak iki nesne gerektirir: her yaş grubundaki hobici sayısı ve her yaş grubundaki toplam satır sayısı. Bu sayıları age_cat'e göre gruplandırarak ve y hobi sahibi sütununda value_counts'u çağırarak elde edebiliriz. Sayılar daha sonra okunabilir ya da dizileri oluşturmak için alt kümelere ayrılabilir. Ardından, statsmodels.stats.proportions'dan proportions_ztest'i içe aktarıyoruz ve dizileri count ve nobs argümanlarına aktarıyoruz. Bir farkı test ettiğimiz için, alternatif argümanını kullanarak bunun iki taraflı bir test olduğunu belirtiriz. proportions_ztest bir z-skoru ve bir p-değeri döndürür. P-değeri, belirttiğimiz yüzde beş anlamlılık düzeyinden daha küçüktür, bu nedenle iki yaş grubu arasında hobici oranında bir fark olduğu sonucuna varabiliriz.

Chi-square test of independence (Ki-kare bağımsızlık testi)

ANOVA'nın t-testlerini ikiden fazla gruba genişletmesi gibi, ki-kare bağımsızlık testleri de orantı testlerini ikiden fazla gruba genişletir.

Ki-Kare Testi, kategorik değişkenler arasında anlamlı bir ilişki olup olmadığını veya gözlenen verilerin belirli bir hipotez altında beklenen verilerden anlamlı bir şekilde sapıp sapmadığını belirlemek için kullanılan istatistiksel bir yöntemdir. Genellikle kategorik veriler için hipotez testinde kullanılır.

Değişkenlerin bağımsızlığı

Bir önceki ders yapılan oran testi pozitif sonuç vermiştir. Küçük p-değeri, hobici ve yaş kategorisi değişkenlerinin bir ilişkisi olduğuna dair kanıt olduğunu göstermiştir. Eğer hobicilerin oranı her yaş kategorisi için aynı olsaydı, değişkenler istatistiksel olarak bağımsız kabul edilirdi. Daha resmi bir ifadeyle, yanıt değişkenindeki başarı oranı açıklayıcı değişkenin tüm kategorilerinde aynı olduğunda iki kategorik değişken istatistiksel olarak bağımsız kabul edilir.

Değişkenlerin bağımsızlığı için test

Pingouin paketi, önceki videodaki oranlardaki farkı test etmek için dolaylı bir yol sunar. chi2_independence yöntemine veri olarak stack_overflow, x olarak hobbyist ve y olarak age_cat değerlerini aktarıyoruz. Düzeltme argümanı, örneklem büyüklüğünün çok küçük ve serbestlik derecesinin bir olduğu durumlar için geçici bir faktör olan Yates'in süreklilik düzeltmesinin uygulanıp uygulanmayacağını belirtir. Her grubun yüzden fazla gözlemi olduğundan, burada buna ihtiyacımız yok. Yöntem üç farklı pandas DataFrame döndürür: beklenen sayılar, gözlenen sayılar ve testle ilgili istatistikler. İstatistiklere bakalım ve pearson test satırı ile chi2 ve pval sütunlarına odaklanalım. P-değeri, z-testinde elde ettiğimiz yüz binde iki değeriyle aynıdır. Chi2 değeri, önceki videoda gördüğümüz z-skorumuzun kareli sonucudur.

Exploratory visualization: proportional stacked bar plot (Keşifsel görselleştirme: orantılı yığılmış çubuk grafiği)

Orantılı yığılmış çubuk grafiği kullanarak verileri inceleyelim. Her yaş grubundaki oranlar hesaplanarak başlanır. Ardından, bu tabloyu geniş biçime dönüştürmek için unstack yöntemi kullanılır. Çizim yöntemini kullanıp kind bar ve stacked True olarak ayarlandığında orantılı yığınlanmış bir çubuk çizimi elde edilir.

Ki-kare testini değişkenler yer değiştirmiş olarak çalıştırırsak, sonuçlar aynı olacaktır. Bu nedenle, sorularımızı “X değişkeni Y değişkeninden bağımsız mı?” yerine “X ve Y değişkenleri bağımsız mı?” şeklinde ifade ediyoruz, çünkü sıralama önemli değil.

Yöne ve Kuyruklar

Bu testte kuyruklar hakkında endişelenmedik ve aslında chi2_independence yönteminin alternatif bir argümanı yoktur. Bunun nedeni ki-kare test istatistiğinin gözlenen ve beklenen sayıların karesine dayanması ve kare sayıların negatif olmamasıdır. Bu da ki-kare testlerinin sağ kuyruklu testler olma eğiliminde olduğu anlamına gelir.

Sol kuyruklu ki-kare testleri istatistiksel adli tıpta bir uyumun şüpheli derecede iyi olup olmadığını tespit etmek için kullanılır çünkü veriler uydurulmuştur. Ki-kare varyans testleri iki kuyruklu olabilir. Yine de bunlar niş kullanımlardır.

hi-square goodness of fit test

Tek örneklem ki-kare testine uyum iyiliği testi denir, çünkü varsayılan verilerin gözlemlenen verilere ne kadar iyi uyduğu test edilir. Testi çalıştırmak için scipy.stats'taki chisquare yöntemi kullanılır. Chisquare için iki gerekli argüman vardır: gözlenen sayılar için dizi benzeri bir nesne, f_obs ve beklenen sayılar için bir tane, f_exp. Fonksiyon tarafından döndürülen p-değeri çok küçüktür, 0.01 anlamlılık düzeyinden çok daha düşüktür, bu nedenle oranların örnek dağılımının varsayılan dağılımdan farklı olduğu sonucuna varırız.

Assumptions in hypothesis testing (Hipotez testinde varsayımlar)

Şimdiye kadar görülen her hipotez testi veriler hakkında varsayımlarda bulunur. Sadece bu varsayımlar karşılandığında o hipotez testini kullanmak uygun olur.

Randomness (Rastgelelik)

İster bir ister birden fazla örnek kullanılsın, her hipotez testi her bir örneğin popülasyondan rastgele seçildiğini varsayar. Eğer rastgele bir örneklem yoksa, o zaman popülasyon temsil edilmeyecektir.

Bu varsayımı kontrol etmek için verilerin nereden geldiğinin bilinmesi gerekir. Bunu kontrol etmek için yapılabilecek istatistiksel veya kodlama testleri yoktur. Şüpheye düşülürse, veri toplamaya dahil olan kişilere veya örneklenen popülasyonu anlayan bir alan uzmanına sorulması gerekir.

Independence of observations (Gözlemlerin Bağımsızlığı)

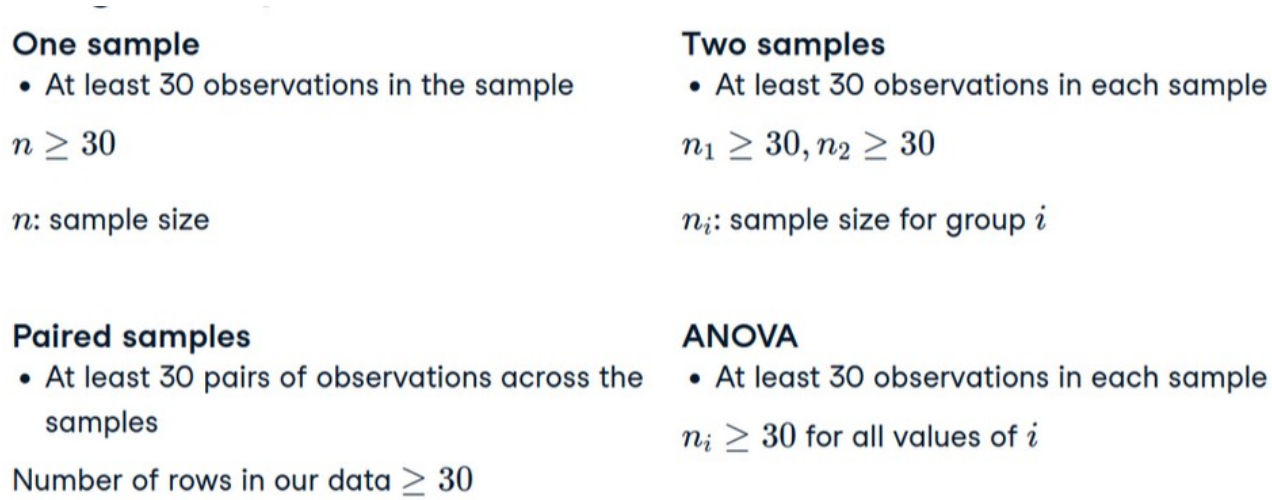
Testler ayrıca her bir gözlemin bağımsız olduğunu varsayar. İki örnek arasındaki bağımlılıklara izin verilen eşleştirilmiş t-testleri gibi bazı özel durumlar vardır, ancak bunlar hesaplamaları değiştirir, bu nedenle bu tür bağımlılıkların nerede meydana geldiğinin anlaşılması gerekir. Eşleştirilmiş t-testinde görüldüğü gibi, bağımlılıkların hesaba katılmaması yanlış negatif ve yanlış pozitif hata olasılığının artmasına neden olur. Bağımlılıkların hesaba katılmaması, analiz sırasında teşhis edilmesi zor bir sorundur. İdeal olarak, veriler analiz edilmeden önce tartışılması gerekir.

Large sample size (Büyük örneklem büyüklüğü)

Hipotez testleri ayrıca örneklemin Merkezi Limit Teoremi'nin geçerli olacağı kadar büyük olduğunu ve örneklem dağılımının normal dağıldığı varsayılabilir. Daha küçük örneklemler daha büyük belirsizliğe neden olur, bu da Merkezi Limit Teoreminin geçerli olmadığı ve örneklem dağılımının normal dağılmayabileceği anlamına gelebilir. Küçük bir örneklemin artan belirsizliği, tahmin etmeye çalışılan parametre üzerinde daha geniş güven aralıkları elde edileceği anlamına gelir. Merkezi Limit Teoremi geçerli değilse, örneklem üzerinde yapılan hesaplamalar ve bunlardan çıkarılan sonuçlar saçma olabilir, bu da yanlış negatif ve yanlış pozitif hata olasılığını artırır. Örneklemin “yeterince büyük” olması için ne kadar büyük olması gerektiği teste bağlıdır.

Large sample size: t-test (Büyük örneklem büyüklüğü: t-testi)

Tek örneklemli t-testleri için popüler bir sezgisel yöntem, örneklemde en az 30 gözleme ihtiyaç olduğudur. İki örneklemli durum veya ANOVA için, her gruptan otuz gözleme ihtiyaç vardır. Bu, çoğunluk grubunun büyütülerek bir azınlık grubun örnekleminin telafi edilemeyeceği anlamına gelir. Eşleştirilmiş durumda, otuz çift gözleme ihtiyaç vardır. Bazen bu testlerin her birinde 30'dan daha azıyla da kurtulunabilir; önemli olan boş dağılımın normal görünmesidir. Bu durum genellikle 30 civarında gerçekleşir ve bu biraz keyfi eşiğin nedeni de budur.



Large sample size: proportion tests

Tek örnek orantı testleri için, örneklem en az on başarı ve on başarısızlık içeriyorsa yeterince büyük kabul edilir. Başarı olasılığı sıfıra veya bire yakınsa daha büyük bir örneğe ihtiyaç olduğuna dikkat edilmelidir. İki örneklem durumunda, her bir örneklemden on başarı ve on başarısızlığa ihtiyaç vardır.

One sample

- Number of successes in sample is greater than or equal to 10

$n \times \hat{p} \geq 10$

- Number of failures in sample is greater than or equal to 10

$n \times (1 - \hat{p}) \geq 10$

n: sample size
$\hat{p}$: proportion of successes in sample

Two samples

- Number of successes in each sample is greater than or equal to 10

$n_1 \times \hat{p}_1 \geq 10$

$n_2 \times \hat{p}_2 \geq 10$

- Number of failures in each sample is greater than or equal to 10

$n_1 \times (1 - \hat{p}_1) \geq 10$

$n_2 \times (1 - \hat{p}_2) \geq 10$

Large sample size: chi-square tests

ki-kare testi biraz daha bağışlayıcıdır ve her grupta on yerine yalnızca beş başarı ve beş başarısızlık gerektirir.

- The number of successes in each group in greater than or equal to 5

$n_i \times \hat{p}_i \geq 5$ for all values of i

- The number of failures in each group in greater than or equal to 5

$n_i \times (1 - \hat{p}_i) \geq 5$ for all values of i

n_i: sample size for group i
$\hat{p}_i$: proportion of successes in sample group i

Sanity check

Yapabileceğimiz bir diğer kontrol de bir bootstrap dağılımı hesaplamak ve bunu bir histogram ile görselleştirmektir. Çan şeklinde bir normal eğri görülmüyorsa, varsayımlardan biri karşılanmamış demektir. Bu durumda, veri toplama süreci tekrar gözden geçirilmeli ve rastgelelik, bağımsızlık ve örneklem büyüklüğü varsayımlarından herhangi birinin sağlanıp sağlanmadığına bakılmalıdır.

Non-parametric tests

Peki şimdiye kadar görülen hipotez testlerinin varsayımları karşılanmazsa ne yapılır?

Parametric tests
Şimdiye kadar görülen testler parametrik testler olarak bilinir. Z-testi, t-testi ve ANOVA gibi testlerin hepsi popülasyonun normal dağıldığı varsayımına dayanır. Parametrik testler ayrıca Merkezi Limit Teoremi'nin geçerli olduğu “yeterince büyük” örneklem büyüklükleri gerektirir.

Non-parametric tests

Bu varsayımlardan emin olunmayan ya da varsayımların karşılanmadığından emin olunan durumlarda parametrik olmayan testler kullanılır. Bunlar normal dağılım varsayımlarını veya önceki derste görülen örneklem büyüklüğü koşullarını sağlamaz. Bu parametrik varsayımlar olmadan test yapmanın birçok farklı yolu vardır. Burada, sıralamalarla ilgili olanlara odaklanılacak. x listesini düşünün. x'in ilk değeri olan bir en küçük değerdir ve ikinci değer olan on beş en küçük beşinci değerdir. En küçükten en büyüğe doğru olan bu sıralamalar x'in elemanlarının rankları olarak bilinir. scipy.stats'in rankdata metodu ile bunlara erişilebilir.

Şimdi ne tür sonuçlar verdiğini görmek için parametrik olmayan bir test kullanalım. Parametrik olmayan testlerin, örneklem büyüklüğünün küçük olduğu veya verilerin normal dağıldığı varsayılamayan durumlarda parametrik alternatiften daha iyi çalıştığı unutulmamalıdır.

Wilcoxon-signed rank test
Frank Wilcoxon tarafından 1945 yılında geliştirilen ve geliştirilen ilk parametrik olmayan prosedürlerden biri olan Wilcoxon-işaretli sıra testini kullanacağız. Başka bir pingouin yöntemi kullanarak uygulamadan önce testin iç işleyişini gözden geçireceğiz.

Wilcoxon-signed rank test (Step 1)

Wilcoxon işaretli sıra testi, veri çiftlerindeki mutlak farkları hesaplamayı ve ardından bunları sıralamayı gerektirir. İlk olarak, eşleştirilmiş değerlerdeki farklar alınır.

Wilcoxon-signed rank test (Step 2)
Ardından, abs yöntemi kullanılarak farkların mutlak değerini alınırr ve bunlar abs_diff sütununa yerleştirilir.

Wilcoxon-signed rank test (Step 3)
Ardından, bu mutlak farklar scipy.stats'teki rankdata yöntemi kullanarak sıralanır.

Wilcoxon-signed rank test (Step 4)
Hesaplamanınson kısmı, W adı verilen bir test istatistiğinin hesaplanmasını içerir. W, sıraları iki gruba ayırmak için fark sütununun işaretlerini kullanır: biri negatif farklara sahip satırlar ve diğeri pozitif farklar için. T-eksi, negatif farklara sahip sıraların toplamı olarak tanımlanır ve T-artı, pozitif farklara sahip sıraların toplamıdır. Bu örnekte, tüm farklar negatiftir, bu nedenle T-eksi değeri beş sıranın toplamıdır ve T-artı sıfırdır. Test istatistiği W, T-eksi ve T-artı değerlerinden küçük olanıdır ve bu durumda sıfırdır. W'yi ve buna karşılık gelen p-değeri manuel hesaplama yerine pingouin yöntemini kullanarak hesaplanabilir.

Implementation with pingouin.wilcoxon()

pingouin'in wilcoxon yöntemi, eşleştirilmiş bir argümana sahip olmaması dışında ttest yöntemine çok benzer argümanlar alır. Fonksiyon sıfır W değeri döndürür - bizim manuel hesaplamamızla aynı! Bu, t-testinden elde edilen p-değerinden on kat daha büyük olan yaklaşık yüzde üçlük bir p-değerine karşılık gelir, bu nedenle küçük örnek boyutu göz önüne alındığında bu sonuçtan daha

emin olmalıyız. Wilcoxon testi, beş satırlık bu küçük örneklemi kullanarak 2008 Cumhuriyetçi yüzdelerinin 2012 yüzdelerinden daha küçük olduğuna dair kanıtımız olmadığını göstermektedir.

Non-parametric ANOVA and unpaired t-tests

Wilcoxon-Mann-Whitney test

Sayısal bir girdinin sıralamaları üzerinde hipotez testleri gerçekleştirerek normal dağılımlı verilerle ilgili varsayımlardan kaçınılabilir. Wilcoxon-Mann-Whitney testi, çok kabaca, sıralanmış veriler üzerinde bir t-testidir. Bu test, bir önceki derste görülen Wilcoxon testine benzer, ancak bunun yerine eşleştirilmemiş veriler üzerinde çalışır.

StackOverflow anketine ve dönüştürülen ücret ile katılımcıların kodlamaya başlama yaşı arasındaki ilişkiye geri dönelim. age_vs_comp adlı yeni bir DataFrame'de sadece bu iki sütuna odaklanarak başlıyoruz. Pingouin ile Wilcoxon-Mann-Whitney testi yapmak için öncelikle verilerimizi uzun formattan geniş formata dönüştürmemiz gerekir. Bu, pandas'ın pivot_table'dan farklı olarak toplama yapmayan pivot yöntemi ile gerçekleştirilir; bunun yerine, satırlar boyunca her grup için ham değerleri döndürür. Artık verilerimiz, her satır için converted_comp girdilerine karşılık gelen değerlerle birlikte adult ve child adlı iki sütunda bulunmaktadır. NaN yetişkin değeri bir çocuk girişine ve NaN çocuk değeri de bir yetişkin girişine karşılık gelir.

Wilcoxon-Mann-Whitney test

Anlamlılık düzeyi 0.01 olarak belirlensin. pingouin'den mwu kullanarak bir Wilcoxon-Mann-Whitney testi çalıştırılabilir. Karşılaştırmak istenilen iki sayı sütununa karşılık gelen x ve y argümanlarını kabul eder, bu durumda çocuk ve yetişkin. alternative, alternatif hipotezin türünü belirler, bu durumda, önce çocuk olarak kodlayanların önce yetişkin olarak kodlayanlardan daha yüksek bir gelire sahip olduğu, ki bu sağ kuyruklu bir testtir. Burada, p-değeri yaklaşık on üzeri negatif on dokuzuncu kuvvet olarak gösterilmektedir, bu da anlamlılık düzeyinden önemli ölçüde daha küçüktür.

Kruskal-Wallis test

ANOVA'nın t-testlerini ikiden fazla gruba genişletmesi gibi, Kruskal-Wallis testi de Wilcoxon-Mann-Whitney testini ikiden fazla gruba genişletir. Yani, Kruskal-Wallis testi ANOVA'nın parametrik olmayan bir versiyonudur. İş tatmini grupları arasında converted_comp açısından bir fark olup olmadığını araştırmak üzere Kruskal-Wallis testi yapmak için pingouin'in kruskal yöntemini kullanıyoruz. Wilcoxon-Mann-Whitney testinin aksine, kruskal yöntemi uzun veriler üzerinde çalıştığı için burada verilerimizi pivotlamamıza gerek yoktur. Veri olarak stack_overflow, bağımlı değişken olan dv'yi converted_comp olarak giriyoruz ve job_sat grupları arasında karşılaştırma yapıyoruz. Yine, buradaki p-değeri çok küçüktür ve anlamlılık düzeyimizden daha küçüktür. Bu, ortalama tazminat toplamlarından en az birinin bu beş iş memnuniyeti grubunda diğerlerinden farklı olduğuna dair kanıt sağlamaktadır.

Machine learning with scikit-learn

What is machine learning?

Makine öğrenimi, bilgisayarların açıkça programlanmadan verilerden karar vermeyi öğrenmesi sürecidir.

Examples of machine learning

Örneğin, içeriği ve göndereni göz önüne alındığında bir e-postanın spam olup olmadığını tahmin etmeyi öğrenmek. Ya da kitapları içerdikleri kelimelere göre farklı kategorilerde kümelemeyi öğrenmek, ardından herhangi bir yeni kitabı mevcut kümelerden birine atamak.

Unsupervised learning

Denetimsiz öğrenme, etiketlenmemiş verilerden gizli kalıpları ve yapıları ortaya çıkarma sürecidir. Örneğin, bir işletme müşterilerini, bu kategorilerin ne olduğunu önceden bilmeden satın alma davranışlarına göre farklı kategorilerde gruplamak isteyebilir. Bu, denetimsiz öğrenmenin bir dalı olan kümeleme olarak bilinir.

Supervised learning

Denetimli öğrenme, tahmin edilecek değerlerin zaten bilindiği ve daha önce görülmemiş verilerin değerlerini doğru bir şekilde tahmin etmek amacıyla bir modelin oluşturulduğu bir makine öğrenimi türüdür. Denetimli öğrenme, hedef değişkenin değerini tahmin etmek için özellikleri kullanır, örneğin bir basketbol oyuncusunun maç başına attığı sayılara göre pozisyonunu tahmin etmek gibi. Bu ders yalnızca denetimli öğrenmeye odaklanacaktır.

Types of supervised learning
İki tür denetimli öğrenme vardır. Sınıflandırma, bir gözlemin etiketini veya kategorisini tahmin etmek için kullanılır. Örneğin, bir banka işleminin hileli olup olmadığını tahmin edebiliriz. Burada iki sonuç olduğu için - hileli işlem veya hileli olmayan işlem, bu ikili sınıflandırma olarak bilinir. Regresyon sürekli değerleri tahmin etmek için kullanılır. Örneğin, bir model, hedef değişken olan mülkün fiyatını tahmin etmek için yatak odası sayısı ve mülkün büyüklüğü gibi özellikleri kullanabilir.

Naming conventions (İsimlendirme Kuralları)

Note that what we call a feature throughout the course, others may call a predictor variable or independent variable. Also, what we call the target variable, others may call dependent variable or response variable.

Before you use supervised learning

Denetimli öğrenme gerçekleştirmeden önce yerine getirilmesi gereken bazı gereksinimler vardır. Verilerimizde eksik değerler olmamalı, sayısal formatta olmalı ve pandas DataFrames veya Series ya da NumPy dizileri olarak saklanmalıdır. Bu, verilerin doğru formatta olduğundan emin olmak için önce bazı keşifsel veri analizi gerektirir. Uygun veri görselleştirmeleriyle birlikte tanımlayıcı istatistikler için çeşitli pandas yöntemleri bu adımda yararlıdır.

scikit-learn syntax

scikit-learn tüm denetimli öğrenme modelleri için aynı sözdizimini izler, bu da iş akışını tekrarlanabilir hale getirir. Denetimli öğrenme problemi için bir algoritma türü olan bir Modeli bir sklearn modülünden içe aktarılır. Örneğin, k-En Yakın Komşular modeli etiketleri veya değerleri tahmin etmek için gözlemler arasındaki mesafeyi kullanır. Model adında bir değişken oluşturur ve Model örneklendirilir. Bir model, özellikler ve hedef değişken hakkındaki kalıpları öğrendiği verilere uydurulur. Modeli, özelliklerin bir dizisi olan X'e ve hedef değişken değerlerin bir dizisi olan y'ye uyarlanır. Daha sonra modelin predict yöntemi kullanılarak altı yeni gözlem, X_new, geçiriyoruz. Örneğin, altı e-postadan alınan özellikler bir spam sınıflandırma modeline beslenirse, altı değerden oluşan bir dizi döndürülür. Bir, modelin e-postanın spam olduğunu tahmin ettiğini, sıfır ise spam olmadığını tahmin ettiğini gösterir.

Sınıflandırma zorluğu

Denetimli öğrenme etiketleri kullanır. Görünmeyen verilerin etiketlerini tahmin etmek için nasıl bir sınıflandırma modeli veya sınıflandırıcı oluşturulur?

Classifying labels of unseen data

Dört adım vardır. İlk olarak, kendisine iletilen etiketli verilerden öğrenen bir sınıflandırıcı oluşturulur. Daha sonra etiketsiz veriler girdi olarak iletilir ve bu görünmeyen veriler için etiket tahmininde bulunmayı sağlar. Sınıflandırıcı etiketli verilerden öğrendiği için buna eğitim verileri adı verilir.

1. Build a model
2. Model learns from the labeled data we pass to it
3. Pass unlabeled data to the model as input
4. Model predicts the labels of the unseen data

- Labeled data = training data

k-Nearest Neighbors

Sınıflandırma problemleri için popüler olan k-En Yakın Komşular adlı bir algoritma ele alınacak. k-En Yakın Komşular ya da KNN'nin amacı, herhangi bir veri noktasının etiketini tahmin etmek için k, örneğin üç, en yakın etiketli veri noktasına bakmak ve etiketsiz gözlemin hangi etikete sahip olması gerektiği konusunda oy kullanmalarını sağlamaktır. KNN, en yakın komşuların çoğunluğunun hangi etikete sahip olduğuna dayalı tahminler yapan çoğunluk oylamasını kullanır.

Model performansının ölçülmesi

Artık bir sınıflandırıcı kullanarak tahminler yapılabilir, ancak modelin doğru tahminler yapıp yapmadığı nasıl bilinebilir? Performansı nasıl değerlendirilebilir?

Model performansının ölçülmesi

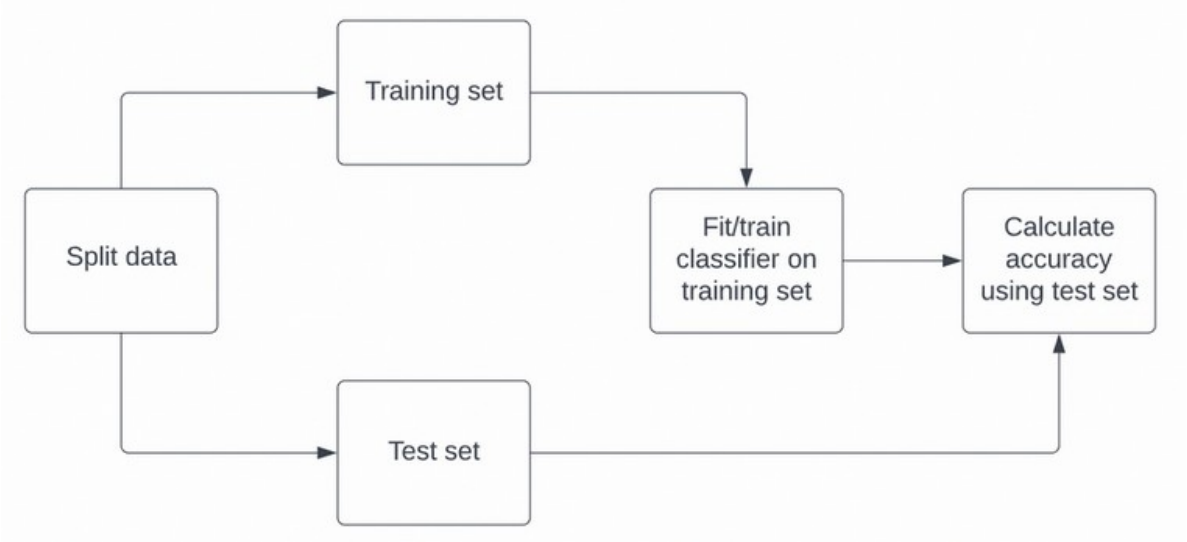
Sınıflandırmada, accuracy yaygın olarak kullanılan bir metriktir.Accuracy, doğru tahminlerin sayısının toplam gözlem sayısına bölünmesiyle elde edilir.

$$\frac{\textit{correct predictions}}{\textit{total observations}}$$

oğruluğu nasıl ölçülür? Sınıflandırıcıyı fit etmek için kullanılan veriler üzerinde accuracy hesaplanabilir. Ancak, bu veriler modeli eğitmek için kullanıldığından, performans, modelin görünmeyen verilere ne kadar iyi genelleme yapabileceğinin göstergesi olmayacaktır, ki asıl ilgilenilen de budur!

Hesaplama doğruluğu

Verileri bir eğitim kümesi ve bir test kümesi olarak ayırmak yaygındır.



Model complexity

Bir modelin bir gözleme hangi etiketi atayacağını belirleyen eşikler olan karar sınırları anlatılmıştı. Aşağıdaki şekillerde, k arttıkça, karar sınırı bireysel gözlemlerden daha az etkilenir ve bu da daha basit bir modeli yansıtır. Daha basit modeller veri kümesindeki ilişkileri daha az tespit edebilir, bu da underfitting olarak bilinir. Buna karşılık, karmaşık modeller genel eğilimleri yansıtmak yerine eğitim verilerindeki gürültüye karşı hassas olabilir. Bu durum overfitting olarak bilinir.

Model complexity and over/underfitting

Ayrıca k değeri bir model karmaşıklığı eğrisi kullanarak da yorumlanabilir. Bir KNN modeliyle, artan k değerleri kullanılarak eğitim ve test kümelerindeki doğruluk hesaplanabilir ve sonuçlar çizilebilir.

The basics of linear regression

Regresyon mekaniği

Verilere bir çizgi uydurmak istenir ve iki boyutta bu, y = ax + b biçimini alır. Tek bir özellik kullanmak basit doğrusal regresyon olarak bilinir, burada y hedef, x özellik ve a ve b öğrenmek istediğimiz model parametreleridir. a ve b ayrıca model katsayıları veya sırasıyla slope ve intercept olarak da adlandırılır. Peki a ve b için değerler nasıl doğru bir şekilde seçilir? Herhangi bir verilen çizgi için bir hata fonksiyonu tanımlanabilir ve ardından bu fonksiyonu en aza indiren çizgi seçilebilir. Error fonksiyonlarına ayrıca loss veya cost fonksiyonları da denir.

Loss Function

Bu dağılım grafiğini kullanarak bir kayıp fonksiyonunu görselleştirelim. Doğrunun gözlemlere mümkün olduğunca yakın olmasını istiyoruz. Bu nedenle, uyum ve veri arasındaki dikey mesafeyi en aza indirmek istiyoruz. Bu yüzden her gözlem için, gözlem ile doğru arasındaki dikey mesafeyi hesaplıyoruz. Bu mesafeye residual denir.

Ordinary Least Squares

Artıkların toplamını en aza indirmeyi deneyebiliriz, ancak o zaman her pozitif artık her negatif artığı iptal edecektir. Bundan kaçınmak için kalıntıların karesini alırız. Tüm kareli kalıntıları toplayarak kalıntı kareler toplamını veya RSS'yi hesaplarız. RSS'yi en aza indirmeyi amaçladığımız bu doğrusal regresyon türüne Sıradan En Küçük Kareler veya OLS denir.

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Ordinary Least Squares (OLS): minimize RSS

Linear regression in higher dimensions

x1 ve x2 olmak üzere iki özellik ve y olmak üzere bir hedef olduğunda, bir doğru y = a1x1 + a2x2 + b biçimini alır. Bu nedenle, doğrusal bir regresyon modeli uydurmak için a1, a2 ve intercept b olmak üzere üç değişken belirtilir. Çoklu doğrusal regresyon modeli kurmak, n sayıda özellik için bir katsayı, a n ve b belirtmek anlamına gelir. Çoklu doğrusal regresyon modelleri için scikit-learn, özellik ve hedef değerleri için birer değişken bekler.

$$y = a_1x_1 + a_2x_2 + b$$

- To fit a linear regression model here:
  - Need to specify 3 variables: $a_1,\ a_2,\ b$
- In higher dimensions:
  - Known as multiple regression
  - Must specify coefficients for each feature and the variable b

$$y = a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_nx_n + b$$

- scikit-learn works exactly the same way:
  - Pass two arrays: features and target

R-squared

Doğrusal regresyon için varsayılan metrik, özellikler tarafından açıklanan hedef değişkendeki varyans miktarını ölçen R-kare değeridir. Değerler sıfır ile bir arasında değişebilir; bir, özelliklerin hedefin varyansını tamamen açıkladığı anlamına gelir. Burada sırasıyla yüksek ve düşük R-kare değerlerini görselleştiren iki grafik yer almaktadır.

Mean squared error and root mean squared error

Bir regresyon modelinin performansını değerlendirmenin bir başka yolu da artık kareler toplamının ortalamasını almaktır. Bu, ortalama karesel hata veya MSE olarak bilinir. MSE, hedef değişkenimizin karesi cinsinden ölçülür. Örneğin, bir model bir dolar değerini tahmin ediyorsa, MSE dolar kare cinsinden olacaktır. Dolara dönüştürmek için, kök ortalama karesel hata veya RMSE olarak bilinen karekökü alabiliriz.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- MSE is measured in target units, squared

$$RMSE = \sqrt{MSE}$$

- Measure $RMSE$ in the same units at the target variable

Cross-validation

Test seti üzerinde R-kare hesaplanıyorsa, döndürülen R-kare, verileri bölme şekline bağlıdır! Test setindeki veri noktaları, üzerinde hesaplanan R-kare değerinin modelin görülmeyen verilere genelleme yeteneğini temsil etmediği anlamına gelen bazı özelliklere sahip olabilir. Esasen rastgele bir bölünmeye bağlı olan bu bağımlılıkla mücadele etmek için çapraz doğrulama adı verilen bir teknik kullanılır.

Cross-validation basics

Veri kümesi beş gruba veya kümeye bölünerek başlanır. Daha sonra ilk küme test kümesi olarak bir kenara ayırılır, model kalan dört kümeyle oluşturulur, test kümesi ile de tahmin yapılır ve R-kare gibi ilgili metrik hesaplanır. Ardından, ikinci küme test kümesi olarak ayırılır, kalan verilerle model kurulut, test kümesinde tahmin yapılır ve ilgili metrik hesaplanır. Ardından benzer şekilde üçüncü küme, dördüncü küme ve beşinci küme ile devam edilir. Sonuç olarak, ortalama, medyan ve %95 güven aralıkları gibi ilgili istatistiklerin hesaplayabileceği beş R-kare değeri elde edilir.

Cross-validation and model performance

Veri kümesi beş kümeye bölündüğü için bu işleme 5 katlı çapraz doğrulama denir. Eğer 10 kat kullanılırsa, buna 10-kat çapraz doğrulama denir. Daha genel olarak, k kat kullanılırsa, buna k-kat çapraz doğrulama veya k-kat CV denir. Bununla birlikte, bir değiş tokuş söz konusudur. Daha fazla küme kullanmak hesaplama açısından daha pahalıdır. Bunun nedeni, daha fazla kez fitting ve tahmin yapılıyor olmasıdır

Regularized regression

Şimdi regresyonda overfitting'den kaçınmak için kullanılan bir teknik olan regularization'u inceleyelim.

Why regularize?

Doğrusal bir regresyon modeli oluşturmak, her özellik için bir katsayı, a, ve intercept, b, seçmek için bir kayıp fonksiyonunu en aza indirmektir. Bu nedenle, loss fonksiyonunu büyük katsayıları cezalandıracak şekilde değiştirmek yaygın bir uygulamadır. Buna regularization denir.

Ridge regression

İncelenecek ilk regularized regresyon türü ridge olarak adlandırılır. Ridge ile, Sıradan En Küçük Kareler kayıp fonksiyonu artı her bir katsayının kare değerini bir sabit olan alpha ile çarparak kullanırız. Dolayısıyla, kayıp fonksiyonunu en aza indirirken, modeller büyük pozitif veya negatif değerlere sahip katsayılar için cezalandırılır. Ridge kullanırken, uyum sağlamak ve tahmin etmek için alfa değerinin seçilmesi gerekir. Esasen, modelin en iyi performans gösterdiği alfa seçilebilir. Ridge için alfa seçmek KNN'de k seçmeye benzer. Ridge'deki alfa, bir modelin parametrelerini seçmek için kullanılan bir değişken olan hiperparametre olarak bilinir. Alfa model karmaşıklığını kontrol eder. Alfa sıfıra eşit olduğunda, büyük katsayıların cezalandırılmadığı ve aşırı uyumun meydana gelebileceği OLS gerçekleştilir. Yüksek bir alfa, büyük katsayıların önemli ölçüde cezalandırıldığı anlamına gelir ve bu da yetersiz uyuma yol açabilir.

- Loss function = OLS loss function + $\alpha * \sum_{i=1}^{n} {a_i}^2$
- Ridge penalizes large positive or negative coefficients
- α: parameter we need to choose
- Picking α is similar to picking `k` in KNN
- Hyperparameter: variable used to optimize model parameters
- α controls model complexity
  - α = 0 = OLS (Can lead to overfitting)
  - Very high α: Can lead to underfitting

Lasso Regression

Kayıp fonksiyonunun OLS kayıp fonksiyonu artı her bir katsayının mutlak değerinin alfa gibi bir sabitle çarpımı olduğu lasso adı verilen başka bir düzenli regresyon türü daha vardır.

- Loss function = OLS loss function + $\alpha * \sum_{i=1}^{n} |a_i|$

Lasso regression for feature selection

Lasso regresyonu aslında özelliğin önemini değerlendirmek için kullanılabilir. Bunun nedeni, daha az önemli özelliklerin katsayılarını sıfıra küçültme eğiliminde olmasıdır. Katsayıları sıfıra küçülmeyen özellikler lasso algoritması tarafından seçilir.

Modeliniz ne kadar iyi?

Classification metrics

Model performansını ölçmek için doğru sınıflandırılmış etiketlerin oranı olan accuracy kullanılabilmektedir. Ancak doğruluk her zaman kullanışlı bir ölçüt değildir.

Class imbalance

Bir banka işleminin hileli olup olmadığını tahmin etmek için bir model düşünün; işlemlerin yalnızca %1'i gerçekten hilelidir. Her işlemi doğru olarak sınıflandıran bir model oluşturulabilir; bu model %99 doğruluk oranına sahip olacaktır! Ancak, sahtekarlığı gerçekten tahmin etme konusunda berbat bir iş çıkarır, bu nedenle asıl amacında başarısız olur. Bir sınıfın daha sık görüldüğü duruma sınıf dengesizliği denir. Burada, doğru işlemler sınıfı, hileli işlemler sınıfından çok daha fazla örnek içerir. Bu, uygulamada sık karşılaşılan bir durumdur ve modelin performansını değerlendirmek için farklı bir yaklaşım gerektirir.

Confusion matrix for assessing classification performance

Dolandırıcılık işlemleri örneği gibi ikili bir sınıflandırıcı verildiğinde, karışıklık matrisi adı verilen performansı özetleyen 2'ye 2'lik bir matris oluşturulabilir.

- Confusion matrix

	Predicted: Legitimate	Predicted: Fraudulent
Actual: Legitimate	True Negative	False Positive
Actual: Fraudulent	False Negative	True Positive

Sınıflandırma performansının değerlendirilmesi

Genellikle, ilgilenilen sınıf pozitif sınıf olarak adlandırılır. Dolandırıcılığı tespit etmek amaçlandığı için, pozitif sınıf gayrimeşru bir işlemdir. Peki karışıklık matrisi neden önemlidir? İlk olarak, doğruluk elde edilebilir: doğru tahminlerin toplamının matrisin toplamına bölünmesiyle elde edilir.

- Accuracy:

$$\frac{tp + tn}{tp + tn + fp + fn}$$

Precision

İkinci olarak, karışıklık matrisinden hesaplanabilecek başka önemli ölçütler de vardır. Kesinlik, tüm pozitif tahminlerin toplamına bölünen doğru pozitiflerin sayısıdır. Buna pozitif tahmin değeri de denir. Buradaki durumda bu, doğru etiketlenmiş hileli işlemlerin sayısının hileli olarak sınıflandırılan toplam işlem sayısına bölünmesiyle elde edilir. Yüksek hassasiyet, daha düşük bir yanlış pozitif oranına sahip olmak anlamına gelir. Buradaki sınıflandırıcı için bu, daha az sayıda yasal işlemin hileli olarak sınıflandırılması anlamına gelir.

- Precision

$$\frac{true\ positives}{true\ positives + false\ positives}$$

Recall

Recall, doğru pozitiflerin sayısının doğru pozitifler ve yanlış negatiflerin toplamına bölünmesiyle elde edilir. Buna duyarlılık da denir. Yüksek geri çağırma, daha düşük bir yanlış negatif oranını yansıtır. Buradaki sınıflandırıcı için bu, çoğu hileli işlemi doğru tahmin etmek anlamına gelir.

- Recall

$$\frac{true\ positives}{true\ positives + false\ negatives}$$

- High recall = lower false negative rate

F1 Score

F1 score

- F1 Score: $2 * \frac{precision * recall}{precision + recall}$

F1-skoru, hassasiyet ve geri çağırmanın harmonik ortalamasıdır. Bu metrik hassasiyet ve geri çağırmaya eşit ağırlık verir, bu nedenle hem model tarafından yapılan hata sayısını hem de hata türünü hesaba katar. F1 puanı, benzer hassasiyet ve geri çağırma değerlerine sahip modelleri tercih eder ve her iki metrikte de makul ölçüde iyi performans gösteren bir model aranıyorsa faydalı bir metriktir.

Logistic regression and the ROC curve

Logistic regression for binary classification

Adına rağmen, lojistik regresyon sınıflandırma için kullanılır. Bu model, bir gözlemin ikili bir sınıfa ait olma olasılığını (p) hesaplar. Örnek olarak diyabet veri seti kullanılırsa, p >= 0.5, veri bir olarak etiketlenir, bu da bir bireyin diyabet olma olasılığının daha yüksek olduğu tahminini temsil eder; p < 0.5, diyabet olmama olasılığının daha yüksek olduğunu temsil etmek için sıfır olarak etikenir.

Linear decision boundary

Lojistik regresyonun, aşağıdaki şekilde de görüldüğü gibi doğrusal bir karar sınırı ürettiği unutulmamalıdır.

Predicting probabilities

Lojistik regresyonun predict_proba yöntemi çağırılarak ve test özellikleri aktarılarak her bir örneğin bir sınıfa ait olma olasılıklarını tahmin edilebilir. Bu, her iki sınıf için de olasılıkları içeren 2 boyutlu bir dizi döndürür; bu durumda, birey sırasıyla hileli veya hileli değildir. Pozitif sınıf olasılıklarını temsil eden ikinci sütun kesilir ve sonuçlar y_pred_probs olarak saklanır.

Probability thresholds

Scikit-learn'de lojistik regresyon için varsayılan olasılık eşiği 0.5 beştir. Bu eşik KNN gibi diğer modeller için de geçerli olabilir. Peki bu eşik değiştirildiğinde ne olur?

The ROC curve

Farklı eşik değerlerinin doğru pozitif ve yanlış pozitif oranlarını nasıl etkilediğini görselleştirmek için bir receiver operating characteristic, veya ROC eğrisi kullanılabilir. Burada noktalı çizgi, etiketleri rastgele tahmin eden bir şans modelini temsil etmektedir.

Eşik sıfıra eşit olduğunda, model tüm gözlemler için bir tahmininde bulunur, yani tüm pozitif değerleri doğru tahmin eder ve tüm negatif değerleri yanlış tahmin eder.

- Eşik bire eşitse, model tüm veriler için sıfır tahmin eder, bu da hem doğru hem de yanlış pozitif oranların sıfır olduğu anlamına gelir.
- Eşiği değiştirirsek, bir dizi farklı yanlış pozitif ve doğru pozitif oranları elde ederiz.
- Eşiklerin çizgi grafiği eğilimi görselleştirmeye yardımcı olur.

Plotting the ROC curve

ROC eğrisini çizmek için sklearn-dot-metrics'ten roc_curve import edilir. Daha sonra roc_curve fonksiyonu çağırılır; test etiketlerini ilk argüman olarak ve tahmin edilen olasılıkları ikinci argüman olarak iletilir. Sonuçlar üç değişkene ayırılır: yanlış pozitif oranı, FPR; doğru pozitif oranı, TPR; ve eşikler. Daha sonra FPR ve TPR ile birlikte sıfırdan bire noktalı bir çizgi çizilebilir.

ROC AUC

D oğru pozitif oranı bir ve yanlış pozitif oranı sıfır olan bir modelimiz varsa, bu mükemmel model olacaktır. Bu nedenle, AUC olarak bilinen bir metrik olan ROC eğrisi altındaki alan hesaplanır. Puanlar sıfır ile bir arasında değişir ve bir idealdir.

Hyperparameter tuning

Ridge ve lasso regresyonunda alfa için bir değer seçmemiz gerekmişti. Aynı şekilde, KNN'yi kurmadan ve tahmin etmeden önce n_neighbors değeri seçilir. Bir modeli kurmadan önce belirlenen alfa ve n_neighbors gibi parametrelere hiperparametreler denir. Yani, başarılı bir model oluşturmak için temel bir adım doğru hiperparametreleri seçmektir.

Choosing the correct hyperparameters

Çok sayıda farklı değer denenebilir, hepsini ayrı ayrı fit edilebilir, ne kadar iyi performans gösterdikleri görülebilir ve en iyi değerler seçilebilir! Buna hiperparametre ayarlama denir. Farklı hiperparametre değerleri fit edilirken, hiperparametrelerin test setine overfit edilmesini önlemek

için çapraz doğrulama kullanılır. Veriler yine de bölünebilir, ancak eğitim setinde çapraz doğrulama gerçekleştirilebili. Test setini alıkonulur ve ayarlanmış modeli değerlendirmek için kullanılır.

Grid search cross-validation

Hiperparametre ayarı için bir yaklaşım, denemek için olası hiperparametre değerlerinden oluşan bir grid seçilen grid search olarak adlandırılır. Örneğin, bir KNN modeli için iki hiperparametre arasında arama yapılabilir - metrik türü ve farklı sayıda komşu. Burada üçerlik artışlarla 2 ile 11 arasında n komşu ve iki metrik vardır: öklid ve manhattan. Bu nedenle, aşağıdaki gibi bir değerler tablosu oluşturulabilir.

Limitations and an alternative approach

Grid Search harikadır. Ancak, fit() sayısı hiperparametre sayısının değer sayısı ile çarpımının kat sayısı ile çarpımına eşittir. Bu nedenle, iyi ölçeklenmez! Yani, her biri 10 değer içeren bir hiperparametre için 3 kat çapraz doğrulama yapmak 30 fit() anlamına gelirken, her biri 10 değer içeren 3 hiperparametre üzerinde 10 kat çapraz doğrulama yapmak 900 fit() eşittir! Ancak, başka bir yol daha vardır.

RandomizedSearchCV

Tüm seçenekleri kapsamlı bir şekilde aramak yerine rastgele hiperparametre değerleri seçen rastgele bir arama gerçekleştirilebilir.

Preprocessing data

Scikit-learn eksik değer içermeyen sayısal verilere ihtiyaç duyar. Şimdiye kadar kullanılan tüm veriler bu formatta idi. Ancak, gerçek dünya verilerinde durum nadiren böyle olacaktır ve bunun yerine model oluşturmadan önce verilerin önceden işlenmesi gerekir.

Dealing with categorical features

Örneğin renk gibi kategorik özellikler içeren bir veri kümesine sahip olalım. Bunlar sayısal olmadığı için scikit-learn bunları kabul etmeyecektir ve bunların sayısal özelliklere dönüştürülmesi gerekir. Bu durum, özelliğin her kategori için bir tane olmak üzere kukla (dummy) değişkenler adı verilen çoklu ikili özelliklere bölünerek başarılır. Sıfır, gözlemin o kategoride olmadığı, bir ise olduğu anlamına gelir.

Dummy variables

Elektronik, Hip-Hop ve Rock gibi on değere sahip bir tür özelliği olan bir müzik veri kümesiyle çalıştığımızı varsayalım. Her tür için ikili özellikler oluştulur. Her şarkının bir türü olduğundan, her satırda on sütundan birinde 1 ve geri kalanında sıfır olacaktır. Eğer bir şarkı ilk dokuz türden herhangi biri değilse, o zaman dolaylı olarak bir rock şarkısıdır. Bu da sadece dokuz özelliğe ihtiyaç olduğu anlamına gelir, dolayısıyla Rock sütununu silinebilir. Bu yapılmazsa, bilgileri tekrar edilmiş olur ki bu da bazı modeller için sorun teşkil edebilir.

Dealing with categorical features in Python

Kukla değişkenler oluşturmak için scikit-learn'in OneHotEncoder'ı veya pandas'ın get_dummies'i kullanılabilir. Burada get_dummies kullılacktır.

Handling Missing Data

Belirli bir satırda bir özellik için hiçbir değer olmadığında, buna eksik veri denir. Bunun nedeni gözlem yapılmamış olması veya verilerin bozuk olması olabilir. Sebep ne olursa olsun, bu durumun çözülmesi gerekir.

Dropping missing data

Yaygın yaklaşımlardan biri, tüm verilerin %5'inden daha azını oluşturan eksik gözlemleri kaldırmaktır. Bunu yapmak için pandas'ın dropna yöntemi kullanılır ve subset parametresine %5'ten az eksik değer içeren sütunların bir listesi iletilir. Bu durumda ilgili sütunlarda eksik değerler varsa, satırın tamamı kaldırılır. DataFrame tekrar kontrol edildiğinde daha az eksik değer oldupu görülür.

Imputing values

Diğer bir seçenek de eksik verileri düzeltmektir. Bu, eksik değerlerin ne olabileceği konusunda eğitimli bir tahmin yapmak anlamına gelir. Belirli bir özellik için eksik olmayan tüm girdilerin ortalaması çıkarılabilir. Medyan gibi diğer değerler de kullanılabilir. Kategorik değerler için genellikle en sık rastlanan değer impute edilir. Veri sızıntısı olarak bilinen bir kavram olan test seti bilgilerinin modele sızmasını önlemek için veri yüklemeden önce verilerin bölünmesi gerekmektedir.

Imputing within a pipeline

Ayrıca, bir dizi dönüşümü çalıştırmak ve tek bir iş akışında bir model oluşturmak için kullanılan bir nesne olanpipeline kullanılarak da impute edilebilir.

Centering and scaling

Veriler Neden Ölçeklendirilir?

Müzik veri setindeki bazı özellik değişkenlerinin aralıklarını kontrol etmek için describe fonksiyonu kullanalım. Aralıkların büyük ölçüde değiştiği görülür: duration_ms -1 ile ile 1617333 arasında değişmekte, speechiness yalnızca 0.023400 ile 0.710 arasında değişmekte ve loudness yalnızca negatif değerlere sahip!

Birçok makine öğrenimi modeli, kendilerini bilgilendirmek için bir tür mesafe kullanır, bu nedenle çok daha büyük ölçeklerde özellikler varsa, modeli orantısız bir şekilde etkileyebilirler. Örneğin KNN, tahminlerde bulunurken mesafeyi açıkça kullanır. Bu nedenle, aslında özelliklerin benzer ölçekte olması istenir. Bunu başarmak için, genellikle ölçekleme ve merkezleme olarak adlandırılan, verilerimizi normalleştirebilir veya standartlaştırabiliriz.

Veriler nasıl ölçeklendirilir?

Verileri ölçeklendirmenin birkaç yolu vardır: herhangi bir sütun verildiğinde, ortalama çıkarılabilir ve varyansa bölünebilir, böylece tüm özellikler 0 etrafında merkezlenir ve 1 varyansa sahip olur. Buna standardizasyon denir. Ayrıca minimum değer çıkarılıp veri aralığına bölünebilir, böylece normalleştirilmiş veri kümesi minimum 0 ve maksimum 1 değerine sahip olur. Ya da bunun yerine veriler -1 ile 1 arasında olacak şekilde ortalanabilir. Burada standartlaştırma yaılacaktır, ancak scikit-learn'in diğer ölçeklendirme türleri için kullanılabilir işlevleri mevcuttur.

CV and scaling in a pipeline

Cross validation'ı bir pipeline ile nasıl kullanabileceğimize de bakalım. Önce ardışık düzen oluşturulur. Daha sonra bir sözlük oluşturarak hiperparametre alanı belirlenir: anahtarlar ardışık düzen adım adı, ardından çift alt çizgi ve ardından hiperparametre adıdır. Karşılık gelen değer, söz konusu hiperparametre için denenecek değerlerin bir listesi veya dizisidir. Bu durumda, KNN modelinde n_neighbors değerini ayarlıyoruz. Daha sonra verilerimizi eğitim ve test kümelerine ayırıyoruz. Daha sonra GridSearchCV nesnesini örnekleyerek, boru hattımızı geçirerek ve param_grid bağımsız değişkenini parametrelere eşit olarak ayarlayarak parametrelerimiz üzerinde bir ızgara araması gerçekleştiriyoruz. Daha sonra bunu eğitim verilerimize uyarlarız. Son olarak, test setimizi kullanarak tahminler yaparız.

Evaluating multiple models

Different models for different problems

Bu karmaşık bir sorudur ve cevabı içinde bulunulan duruma göre değişir. Ancak, bu kararı verirken yol gösterebilecek bazı ilkeler vardır. Veri setinin boyutu bir rol oynar. Daha az özellik daha basit bir model anlamına gelir ve eğitim süresini kısaltabilir. Ayrıca, Yapay Sinir Ağları gibi bazı modellerin iyi performans göstermesi için çok fazla veri gerekir. Paydaşlara tahminlerin nasıl yapıldığını açıklanabilmesi için yorumlanabilir bir modele ihtiyaç olabilir. Model katsayılarının hesaplanıp yorumlayanabildiği doğrusal regresyon buna bir örnektir. Alternatif olarak, en doğru tahminleri elde etmek için esneklik önemli olabilir. Genel olarak, esnek modeller veriler hakkında daha az varsayımda bulunur; örneğin, bir KNN modeli özellikler ve hedef arasında doğrusal bir ilişki olduğunu varsaymaz.

Her şey metriklerde

Scikit-learn çoğu model için aynı yöntemlerin kullanılmasına izin verir. Bu, onları karşılaştırmayı kolaylaştırır! Regresyon modelleri kök ortalama karesel hata veya R-kare değeri kullanılarak değerlendirilebilir. Benzer şekilde, sınıflandırma modellerinin tümü doğruluk, bir karışıklık matrisi ve ilgili metrikleri veya ROC AUC kullanılarak analiz edilebilir. Bu nedenle, bir yaklaşım birkaç model ve bir metrik seçmek ve ardından herhangi bir hiperparametre ayarı yapmadan performanslarını değerlendirmektir.

Ölçeklendirme

KNN, doğrusal regresyon ve lojistik regresyon gibi bazı modellerin performansının verilerimizi ölçeklendirmekten etkilendiğini hatırlayın. Bu nedenle, modelleri kutudan çıkar çıkmaz değerlendirmeden önce verilerimizi ölçeklendirmek genellikle en iyisidir.