Kaggle Learn'deki "Veri Temizleme" kursuna hoş geldiniz!

Veri temizleme, veri biliminin önemli bir parçasıdır ancak oldukça sinir bozucu olabilir. Metin alanlarınızdan bazıları neden bozuk? Eksik değerlere ne yapmalısınız? Tarihleriniz neden doğru biçimlendirilmemiş? Tutarsız veri girişlerini nasıl hızlıca temizleyebilirsiniz? Bu kursta, bu sorunlarla neden karşılaştığınızı ve daha da önemlisi, onları nasıl düzelteceğinizi öğreneceksiniz!

Bu kursta, en yaygın veri temizleme sorunlarından bazılarıyla nasıl başa çıkacağınızı öğrenecek ve böylece verilerinizi daha hızlı analiz etmeye başlayabileceksiniz. Gerçek, dağınık verilerle beş uygulamalı alıştırma yapacak ve veri temizlemeyle ilgili en sık sorulan sorularınızdan bazılarını yanıtlayacaksınız.

Bu not defterinde, eksik değerlerle nasıl başa çıkacağımıza bakacağız.

Take a first look at the data[¶](https://www.kaggle.com/code/alexisbcook/handling-missing-values" \l "Take-a-first-look-at-the-data" \t "_self)

Yapmamız gereken ilk şey, kullanacağımız kütüphaneleri ve veri setini yüklemek.

Gösterim amacıyla, Amerikan Futbolu maçlarında meydana gelen olaylardan oluşan bir veri seti kullanacağız. Aşağıdaki alıştırmada, yeni becerilerinizi San Francisco'da verilen yapı ruhsatlarından oluşan bir veri setine uygulayacaksınız.

*# modules we'll use*

import pandas as pd

import numpy as np

*# read in all our data*

nfl\_data = pd.read\_csv("../input/nflplaybyplay2009to2016/NFL Play by Play 2009-2017 (v4).csv")

*# set seed for reproducibility*

np.random.seed(0)

/opt/conda/lib/python3.7/site-packages/IPython/core/interactiveshell.py:3553: DtypeWarning: Columns (25,51) have mixed types.Specify dtype option on import or set low\_memory=False.

exec(code\_obj, self.user\_global\_ns, self.user\_ns)

Yeni bir veri seti edindiğinizde yapmanız gereken ilk şey, ona bir göz atmaktır. Bu, verilerin doğru bir şekilde okunduğunu görmenizi sağlar ve verilerde neler olup bittiğine dair bir fikir verir. Bu durumda, NaN veya None ile temsil edilecek eksik değerler olup olmadığına bakalım.

*# look at the first five rows of the nfl\_data file.*

*# I can see a handful of missing data already!*

nfl\_data.head()

| Date | GameID | Drive | qtr | down | time | TimeUnder | TimeSecs | PlayTimeDiff | SideofField | ... | yacEPA | Home\_WP\_pre | Away\_WP\_pre | Home\_WP\_post | Away\_WP\_post | Win\_Prob | WPA | airWPA | yacWPA | Season |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 2009-09-10 | 2009091000 | 1 | 1 | NaN | 15:00 | 15 | 3600.0 | 0.0 | TEN | ... | NaN | 0.485675 | 0.514325 | 0.546433 | 0.453567 | 0.485675 | 0.060758 | NaN | NaN | 2009 |
| 1 | 2009-09-10 | 2009091000 | 1 | 1 | 1.0 | 14:53 | 15 | 3593.0 | 7.0 | PIT | ... | 1.146076 | 0.546433 | 0.453567 | 0.551088 | 0.448912 | 0.546433 | 0.004655 | -0.032244 | 0.036899 | 2009 |
| 2 | 2009-09-10 | 2009091000 | 1 | 1 | 2.0 | 14:16 | 15 | 3556.0 | 37.0 | PIT | ... | NaN | 0.551088 | 0.448912 | 0.510793 | 0.489207 | 0.551088 | -0.040295 | NaN | NaN | 2009 |
| 3 | 2009-09-10 | 2009091000 | 1 | 1 | 3.0 | 13:35 | 14 | 3515.0 | 41.0 | PIT | ... | -5.031425 | 0.510793 | 0.489207 | 0.461217 | 0.538783 | 0.510793 | -0.049576 | 0.106663 | -0.156239 | 2009 |
| 4 | 2009-09-10 | 2009091000 | 1 | 1 | 4.0 | 13:27 | 14 | 3507.0 | 8.0 | PIT | ... | NaN | 0.461217 | 0.538783 | 0.558929 | 0.441071 | 0.461217 | 0.097712 | NaN | NaN | 2009 |

Evet, bazı değerler eksik gibi görünüyor.

# How many missing data points do we have?[¶](https://www.kaggle.com/code/alexisbcook/handling-missing-values#How-many-missing-data-points-do-we-have?)

Tamam, artık bazı eksik değerlerimiz olduğunu biliyoruz. Bakalım her sütunda kaç tane eksik değer var.

*# get the number of missing data points per column*

missing\_values\_count = nfl\_data.isnull().sum()

*# look at the # of missing points in the first ten columns*

missing\_values\_count[0:10]

Date 0

GameID 0

Drive 0

qtr 0

down 61154

time 224

TimeUnder 0

TimeSecs 224

PlayTimeDiff 444

SideofField 528

dtype: int64

Bu çok fazla gibi görünüyor! Veri setimizdeki değerlerin yüzde kaçının eksik olduğunu görmek, bu sorunun ölçeğini daha iyi anlamamıza yardımcı olabilir:

*# how many total missing values do we have?*

total\_cells = np.product(nfl\_data.shape)

total\_missing = missing\_values\_count.sum()

*# percent of data that is missing*

percent\_missing = (total\_missing/total\_cells) \* 100

print(percent\_missing)

24.87214126835169

Vay canına, bu veri setindeki hücrelerin neredeyse dörtte biri boş! Bir sonraki adımda, eksik değerler içeren bazı sütunlara daha yakından bakacağız ve onlara ne olduğuna dair bir şeyler anlamaya çalışacağız.

# Figure out why the data is missing[¶](https://www.kaggle.com/code/alexisbcook/handling-missing-values" \l "Figure-out-why-the-data-is-missing" \t "_self)

Burası, veri biliminin "veri sezgisi" adını verdiğim kısmına girdiğimiz noktadır. Bununla demek istediğim, "verilerinize gerçekten bakmak, neden bu şekilde olduğunu ve bunun analizinizi nasıl etkileyeceğini anlamaya çalışmaktır." Bu, özellikle bu alanda yeniyseniz ve çok fazla deneyiminiz yoksa, veri biliminin sinir bozucu bir parçası olabilir. Eksik değerlerle uğraşmak için, değerin neden eksik olduğunu anlamak için sezgilerinizi kullanmanız gerekecektir. Bunu anlamanıza yardımcı olmak için kendinize sorabileceğiniz en önemli sorulardan biri şudur:

**Is this value missing because it wasn't recorded or because it doesn't exist?**

Eğer bir değer, **var olmadığı için** eksikse (örneğin, çocuğu olmayan birinin en büyük çocuğunun boyu gibi), o değerin ne olabileceğini tahmin etmeye çalışmanın bir anlamı yoktur. Bu değerleri muhtemelen **NaN** olarak tutmak istersiniz.

Diğer yandan, eğer bir değer **kaydedilmediği için** eksikse, o sütundaki ve satırdaki diğer değerlere dayanarak ne olabileceğini tahmin etmeye çalışabilirsiniz. Bu işleme **imputation (değer atama)** denir ve bir sonraki aşamada nasıl yapılacağını öğreneceğiz! :)

Şimdi bir örnek üzerinde çalışalım. nfl\_data veri çerçevesindeki eksik değerlerin sayısına baktığımda, "TimesSec" sütununda çok fazla eksik değer olduğunu fark ettim:

*# look at the # of missing points in the first ten columns*

missing\_values\_count[0:10]

Date 0

GameID 0

Drive 0

qtr 0

down 61154

time 224

TimeUnder 0

TimeSecs 224

PlayTimeDiff 444

SideofField 528

dtype: int64

Dokümantasyona bakarak, bu sütunun, oyunun yapıldığı sırada oyunda kalan saniye sayısını içerdiğini görebiliyorum. Bu, bu değerlerin var olmadığı için değil, büyük olasılıkla kaydedilmediği için eksik olduğu anlamına gelir. Bu nedenle, onları sadece NA olarak bırakmak yerine ne olmaları gerektiğini tahmin etmeye çalışmamız mantıklı olacaktır.

Öte yandan, "PenalizedTeam" gibi, çok fazla eksik alan içeren başka sütunlar da var. Ancak bu durumda, eğer bir ceza yoksa hangi takımın cezalandırıldığını söylemenin bir anlamı olmadığı için alan eksiktir. Bu sütun için, ya boş bırakmak ya da "hiçbiri" gibi üçüncü bir değer ekleyip bu değeri NA'ların yerine kullanmak daha mantıklı olacaktır.

**İpucu:** Henüz yapmadıysanız, veri seti dokümantasyonunu okumak için harika bir yer! Başka bir kişiden aldığınız bir veri setiyle çalışıyorsanız, daha fazla bilgi almak için o kişiye ulaşmayı da deneyebilirsiniz.

Çok dikkatli bir veri analizi yapıyorsanız, eksik değerleri doldurmak için en iyi stratejiyi bulmak amacıyla her sütuna tek tek bakacağınız nokta burasıdır. Bu not defterinin geri kalanında, eksik değerlerle size yardımcı olabilecek, ancak muhtemelen bazı faydalı bilgileri de kaldıracak veya verilerinize biraz gürültü (yanlış bilgi) katacak bazı "hızlı ve kirli" teknikleri ele alacağız.

# Drop missing values[¶](https://www.kaggle.com/code/alexisbcook/handling-missing-values" \l "Drop-missing-values" \t "_self)

Eğer aceleniz varsa veya değerlerinizin neden eksik olduğunu anlamak için bir nedeniniz yoksa, elinizdeki seçeneklerden biri, eksik değerler içeren tüm satırları veya sütunları kaldırmaktır. (Not: Önemli projeler için bu yaklaşımı genellikle önermiyorum! Verilerinizi gözden geçirmek ve veri setinizi gerçekten tanımak için eksik değerlere sahip tüm sütunlara tek tek bakmak genellikle bu çabaya değer.)

Eksik değerlere sahip satırları düşürmek istediğinizden eminseniz, **pandas**'ın bunu yapmanıza yardımcı olacak kullanışlı bir fonksiyonu (dropna()) vardır. Haydi bu fonksiyonu NFL veri setimizde deneyelim!

*# remove all the rows that contain a missing value*

nfl\_data.dropna()

| Date | GameID | Drive | qtr | down | time | TimeUnder | TimeSecs | PlayTimeDiff | SideofField | ... | yacEPA | Home\_WP\_pre | Away\_WP\_pre | Home\_WP\_post | Away\_WP\_post | Win\_Prob | WPA | airWPA | yacWPA | Season |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

Aman Tanrım, sanırım tüm verilerimiz silinmiş! 😱 Bunun nedeni, veri setimizdeki her satırda en az bir eksik değer olması. Bunun yerine, en az bir eksik değere sahip tüm sütunları kaldırsak daha başarılı olabiliriz.

*# remove all columns with at least one missing value*

columns\_with\_na\_dropped = nfl\_data.dropna(axis=1)

columns\_with\_na\_dropped.head()

| Date | GameID | Drive | qtr | TimeUnder | ydstogo | ydsnet | PlayAttempted | Yards.Gained | sp | ... | Timeout\_Indicator | Timeout\_Team | posteam\_timeouts\_pre | HomeTimeouts\_Remaining\_Pre | AwayTimeouts\_Remaining\_Pre | HomeTimeouts\_Remaining\_Post | AwayTimeouts\_Remaining\_Post | ExPoint\_Prob | TwoPoint\_Prob | Season |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 2009-09-10 | 2009091000 | 1 | 1 | 15 | 0 | 0 | 1 | 39 | 0 | ... | 0 | None | 3 | 3 | 3 | 3 | 3 | 0.0 | 0.0 | 2009 |
| 1 | 2009-09-10 | 2009091000 | 1 | 1 | 15 | 10 | 5 | 1 | 5 | 0 | ... | 0 | None | 3 | 3 | 3 | 3 | 3 | 0.0 | 0.0 | 2009 |
| 2 | 2009-09-10 | 2009091000 | 1 | 1 | 15 | 5 | 2 | 1 | -3 | 0 | ... | 0 | None | 3 | 3 | 3 | 3 | 3 | 0.0 | 0.0 | 2009 |
| 3 | 2009-09-10 | 2009091000 | 1 | 1 | 14 | 8 | 2 | 1 | 0 | 0 | ... | 0 | None | 3 | 3 | 3 | 3 | 3 | 0.0 | 0.0 | 2009 |
| 4 | 2009-09-10 | 2009091000 | 1 | 1 | 14 | 8 | 2 | 1 | 0 | 0 | ... | 0 | None | 3 | 3 | 3 | 3 | 3 | 0.0 | 0.0 | 2009 |

*# just how much data did we lose?*

print("Columns in original dataset: **%d** **\n**" % nfl\_data.shape[1])

print("Columns with na's dropped: **%d**" % columns\_with\_na\_dropped.shape[1])

Columns in original dataset: 102

Columns with na's dropped: 41

Oldukça fazla veri kaybettik, ancak bu noktada verilerimizden tüm NaN'ları başarıyla kaldırdık.

# Filling in missing values automatically[¶](https://www.kaggle.com/code/alexisbcook/handling-missing-values" \l "Filling-in-missing-values-automatically" \t "_self)

Bir diğer seçenek de eksik değerleri doldurmaya çalışmaktır. Bir sonraki bölümde, düzgün yazdırılabilmesi için NFL verilerinin küçük bir alt bölümünü alıyorum.

*# get a small subset of the NFL dataset*

subset\_nfl\_data = nfl\_data.loc[:, 'EPA':'Season'].head()

subset\_nfl\_data

| EPA | airEPA | yacEPA | Home\_WP\_pre | Away\_WP\_pre | Home\_WP\_post | Away\_WP\_post | Win\_Prob | WPA | airWPA | yacWPA | Season |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 2.014474 | NaN | NaN | 0.485675 | 0.514325 | 0.546433 | 0.453567 | 0.485675 | 0.060758 | NaN | NaN | 2009 |
| 1 | 0.077907 | -1.068169 | 1.146076 | 0.546433 | 0.453567 | 0.551088 | 0.448912 | 0.546433 | 0.004655 | -0.032244 | 0.036899 | 2009 |
| 2 | -1.402760 | NaN | NaN | 0.551088 | 0.448912 | 0.510793 | 0.489207 | 0.551088 | -0.040295 | NaN | NaN | 2009 |
| 3 | -1.712583 | 3.318841 | -5.031425 | 0.510793 | 0.489207 | 0.461217 | 0.538783 | 0.510793 | -0.049576 | 0.106663 | -0.156239 | 2009 |
| 4 | 2.097796 | NaN | NaN | 0.461217 | 0.538783 | 0.558929 | 0.441071 | 0.461217 | 0.097712 | NaN | NaN | 2009 |

Pandas'ın **fillna()** fonksiyonunu kullanarak bir veri çerçevesindeki eksik değerleri doldurabiliriz. Elimizdeki seçeneklerden biri, **NaN** değerlerinin neyle değiştirileceğini belirtmektir. Burada, tüm **NaN** değerlerini 0 ile değiştirmek istediğimi söylüyorum.

*# replace all NA's with 0*

subset\_nfl\_data.fillna(0)

| EPA | airEPA | yacEPA | Home\_WP\_pre | Away\_WP\_pre | Home\_WP\_post | Away\_WP\_post | Win\_Prob | WPA | airWPA | yacWPA | Season |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 2.014474 | 0.000000 | 0.000000 | 0.485675 | 0.514325 | 0.546433 | 0.453567 | 0.485675 | 0.060758 | 0.000000 | 0.000000 | 2009 |
| 1 | 0.077907 | -1.068169 | 1.146076 | 0.546433 | 0.453567 | 0.551088 | 0.448912 | 0.546433 | 0.004655 | -0.032244 | 0.036899 | 2009 |
| 2 | -1.402760 | 0.000000 | 0.000000 | 0.551088 | 0.448912 | 0.510793 | 0.489207 | 0.551088 | -0.040295 | 0.000000 | 0.000000 | 2009 |
| 3 | -1.712583 | 3.318841 | -5.031425 | 0.510793 | 0.489207 | 0.461217 | 0.538783 | 0.510793 | -0.049576 | 0.106663 | -0.156239 | 2009 |
| 4 | 2.097796 | 0.000000 | 0.000000 | 0.461217 | 0.538783 | 0.558929 | 0.441071 | 0.461217 | 0.097712 | 0.000000 | 0.000000 | 2009 |

Ayrıca biraz daha akıllı davranıp eksik değerleri, aynı sütunda hemen ardından gelen değerle değiştirebilirim. (Bu, gözlemlerin mantıksal bir sıraya sahip olduğu veri kümeleri için çok mantıklıdır.)

*# replace all NA's the value that comes directly after it in the same column,*

*# then replace all the remaining na's with 0*

subset\_nfl\_data.fillna(method='bfill', axis=0).fillna(0)

| EPA | airEPA | yacEPA | Home\_WP\_pre | Away\_WP\_pre | Home\_WP\_post | Away\_WP\_post | Win\_Prob | WPA | airWPA | yacWPA | Season |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 2.014474 | -1.068169 | 1.146076 | 0.485675 | 0.514325 | 0.546433 | 0.453567 | 0.485675 | 0.060758 | -0.032244 | 0.036899 | 2009 |
| 1 | 0.077907 | -1.068169 | 1.146076 | 0.546433 | 0.453567 | 0.551088 | 0.448912 | 0.546433 | 0.004655 | -0.032244 | 0.036899 | 2009 |
| 2 | -1.402760 | 3.318841 | -5.031425 | 0.551088 | 0.448912 | 0.510793 | 0.489207 | 0.551088 | -0.040295 | 0.106663 | -0.156239 | 2009 |
| 3 | -1.712583 | 3.318841 | -5.031425 | 0.510793 | 0.489207 | 0.461217 | 0.538783 | 0.510793 | -0.049576 | 0.106663 | -0.156239 | 2009 |
| 4 | 2.097796 | 0.000000 | 0.000000 | 0.461217 | 0.538783 | 0.558929 | 0.441071 | 0.461217 | 0.097712 | 0.000000 | 0.000000 | 2009 |