1. **Veri Ön İşleme**

**S1:** Ön-işleme nedir?

**C1:** Üzerinde çalışılacak veri setinde,

* Eksik veri: Verilerin girilmemiş olması, veri içerisinde null değerlerin olması
* Yanlış veri türü: Maaş değerinin -10 olması, yaşın 200 girilmesi
* Farklı girilen veri: Tarih için 1995 veya 02.12.1995 gibi değişik yazımlar (standart olmalı! VT mantığında da bunu constrain/kısıt’lerle sağlarız)

bulunabilir. Bunları ele almamız eğitilen algoritmanın başarımını etkileyecektir. Bu durumlarda eldeki veriler kullanılmadan önce veri ön işleme çalışması yapılmalıdır. Bu çalışmalardan “eksik veri” adımını inceleyelim:

**Eksik Verilerin Temizlenmesi:** Bu durumda kullanılabilecek yöntemler şu şekildedir:

* Sütundaki diğer verilerin ortalama değerini alarak eksik verilerin yerine yazmak
* Sütunda en çok tekrar eden değerleri eksik verilerin yerine yazmak

Bunun dışında ön-işleme adımlarında karşımıza çıkan önemli bir iki adım aşağıdaki gibidir

**Kategorik Verilerin Kodlanması:**  Bazı algoritmalar kategorik verilerle çalışamaz. Bunların sayısal hale çevrilmesi/kodlanması gerekir.

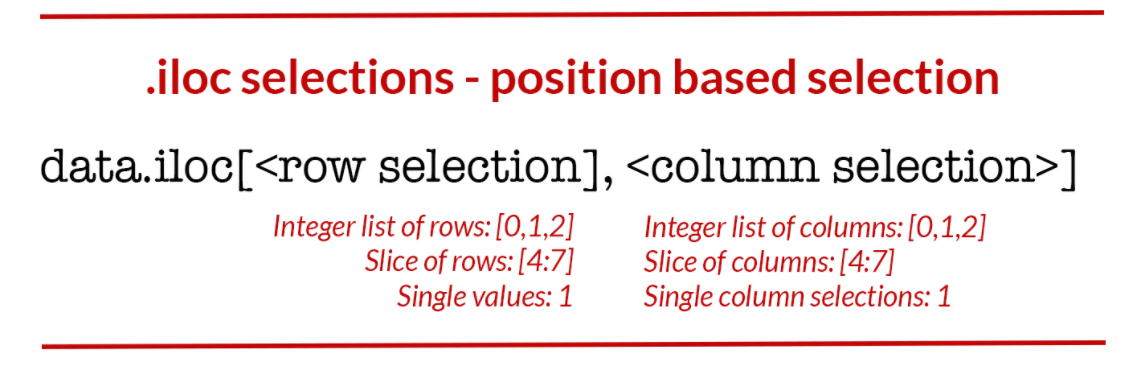
**Öznitelik Ölçekleme:** Verilerin algoritma eğitimde kullanılırken normalize/standardize edilmeli yani aynı değer aralıklarına getirilmelidir. Normalizasyon bir kolonda bütün değerlerin en büyük değere bölünmesi gibi düşünülebilir (tüm değerler 0-1 aralığına girer).

1. **Veri Ön İşleme Uygulaması**
2. Anaconda Kurulumu
3. <https://www.anaconda.com/download/> adresinden Python 3.5 üstü versiyonu kurunuz.
4. Veri Seti İndirme İşlemleri: Bu derste kullanacağımız veri setleri <https://www.superdatascience.com/machine-learning/> adresinde yer almaktadır.
5. Kullanılacak Temel Kütüphaneler:

* scikit-learn: Veri analitiği ve makine öğrenmesi uygulamalarında temel yöntemleri içeren python kütüphanesi.
* Numpy: Pythonda matematiksel işlemlerin temel paketidir.
* Matplotlib : Veri görselleştirme işlemleri için kullanılır.
* Pandas: Veri analizi ve veri ön işleme işlemlerinde kullanılan python kütüphanesidir.

1. Veri Seti Seçme ve Görüntüleme

**Iloc:** Veri seçme işlemi için kullanılır. İlk parametre seçilecek satırı, 2. Parametre seçilecek sütunu ifade eder. “ : ” şeklindeki gösterim tüm satırın seçileceği anlamına gelir. Son sütunu seçmek için -1 kullanılabilir.



**Not1:** iloc, loc kullanımları için aşağıdaki kaynağı okuyunuz

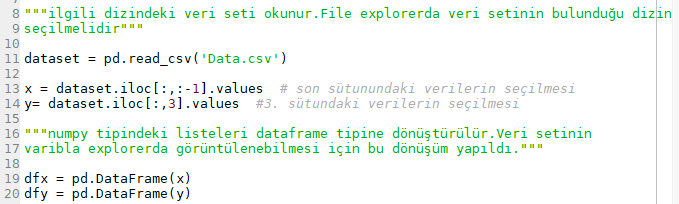
<https://medium.com/@cemayan/pandas-ile-veri-analizi-2-data-selection-missing-values-4ecec921be87>

**Not2:** Python’da liste indisleri 0’dan başlar. Örneğin [0:1,1:3] 0’ıncı satırı (ilk satırı) ve birinci ve ikinci sütunları (gerçekte 0 dan başladığı için 2 ve 3 nolu sütunlar da çalışıyoruz)

Normalde seçtiklerine bir ekle..

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 2 | 3 |
| Sıfır |  |  |  |
| birinci |  |  |  |

**Dataframe** : Listeleri satır ve sütunlardan oluşan veri yapılarına dönüştürür



pd 🡪 pandas’tan pd olarak import edilmiş

[ : , : -1] 🡪 son sütun hariç hepsinin seçilmesi (x ve y eğitim/test için ayrılıyor aşağıda gelecek!)

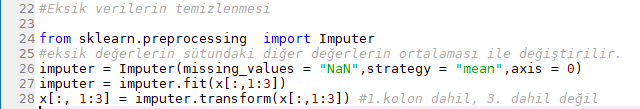
1. Eksik Verilerin Temizlenmesi/Yerine Değer Atanması

**Imputer**: Eksik verileri gidermek için kullanılır**.**

**Imputer.fit() :**Eğitim verileri üzerinde istatistikleri (ortalama, median..) hesaplar.

**Imputer.transform():** Hesaplanan ortalama vs. gibi “değeri” boş alana uygular.

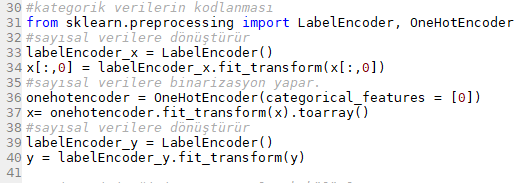
Fit ve transformun ayrı ayrı yapılması veri kayıplarını engellemek içindir. Eğitim ve test verilerinin aynı olduğu biliniyorsa fit\_transform() metodu da kullanılabilir.



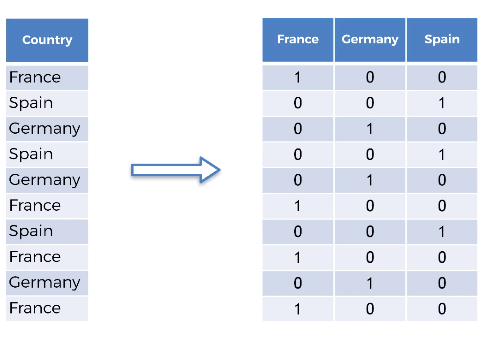
Notlar:

* axis= 0 sütun boyunca değiştirme/güncelleme, 1 satır boyunca değiştirme/güncelleme
* imputer = Imputer(missing\_values='NaN', strategy='mean',axis=0)
* imputer isimli örnek oluştur, boş alanlara NaN yaz, boş alanlara yazılacak değeri ‘mean’ ile hesapla ve sütun-kolon boyunca çalış.
* imputer = imputer.fit(x[:, 1:3]) 🡪 Şimdi bu işlemin X değerine fit edilmesi (tüm bu anlatılan işlemin x üzerinde yapılması, NaN ve mean işlemini yap)
* X[:, 1:3]=imputer.transform(x[:, 1:3]) 🡪 transform ile hesaplanan ortalama değerleri ilgili kolona yansıt ve x’i yeni değerleri ile elde et. NOT: fit hesap kitabı yaparken gerçek değiştirmeyi transform yapıyor.
* fit\_transform metodu : Bu işlemleri tek seferde yapar 🡪 fit\_transform(X[, y])

1. Kategorik Verilerin Kodlanması (algoritmalar kategorik veri istemiyor!)



**labelEncoder**: sklearn.preprocessing kütüphanesinde bulunur. Veriyi birebir sayısallaştırmaya yarar. Kategorik her veriye sayısal bir değer atar (kaç grup varsa o kadar sayı kullanır!). Örneğin aşağıdaki ülkeler örneği için 1, 2, 3 gibi değerler kullanır ama pratik olarak bunun anlamını çözmek daha zor olduğu için genellikle OnHotEncoder kullanılır.

**OneHotEncoder**: sklearn.preprocessing kütüphanesinden bulunan encoder bu kategorik verilerin binarizasyonunu gerçekleştirmemizi sağlar. Şekil 1deki gibi x matrisindeki 1. Kolon olan country kolonunda var olan değere “1” verip olmayanlara “0” vermektedir.

Burada örneğin fransanın kodu 100, almanyanın ki 010 şeklinde düşünülebilir.

1. Veri Setinin Test ve Eğitim Olarak Bölümlenmesi

**Sklearn.modelselection.train\_test\_split:** Liste veya matrisleri rastgele eğitim ve test verilerine ayırır. Her seferinde aynı örnekleri aynı grupta tutmak istiyorsak (deneyin aynı şekilde tekrarlanabilmesi için!) random\_state =0 veya 1 veya 42 veya başka ‘integer’ kullanılabilir. Tamamen random olması isteniyorsa o zaman random\_state parametresi kullanılmaz.

test\_size = Test için ayrılacak verilerin oranını gösterir. 0-1 arasında değer alır.

train\_size = Eğitim için a ayrılacak verilerin oranını gösterir. 0-1 arasında değer alır.

test\_size + train\_size = 1 olacağı için birinin yazılması yeterlidir.



Not: python da tahmin yapılacak kolon (y) ayrılıyor..

|  |  |  |  |
| --- | --- | --- | --- |
| Country | Age | Salary | Purchased |
| France | 44 | 72000 | No |
| Spain | 27 | 48000 | Yes |
| Germany | 30 | 54000 | No |
| Spain | 38 | 61000 | No |
| Germany | 40 |  | Yes |
| France | 35 | 58000 | Yes |
| Spain |  | 52000 | No |
| France | 48 | 79000 | Yes |
| Germany | 50 | 83000 | No |
| France | 37 | 67000 | Yes |
| Turkey | 50 | 60000 | ????  MODEL CANLIYA GEÇECEK |

|  |
| --- |
|  |
|  |
|  |