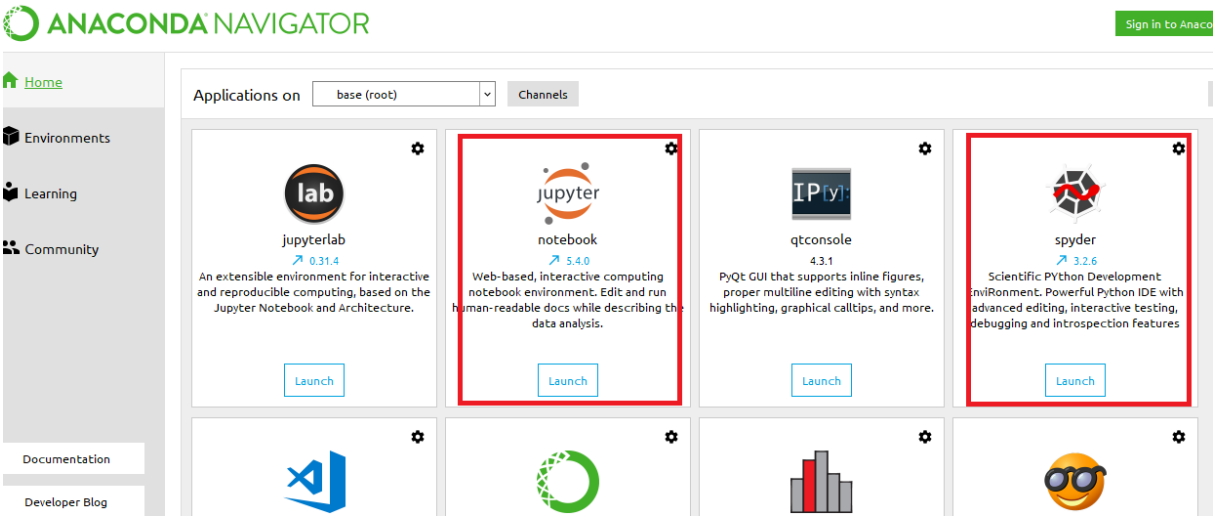
0.0 Makine Öğrenmesi Giriş

Ders İşleyiş Biçimi, Yazılım-Araç Gereç, Lab vs:

- Ders kısa teori ve python code'u üzerinden deneysel uygulama şeklinde gerçekleşecektir.
- Deneyleri Anaconda ortamında Spider (Python kodlama ortamı) ve/veya Jupyter Notebook (Python kodlama ortamı) ile kendi laptoplarımızda sınıf ortamında çalışacağız.



0. 1 Sorularla Makine Öğrenmesi Felsefesi

S1: Makine Öğrenmesi, İstatistiksel Öğrenme, Veri Analitiği Nedir, Yapay Zekâ? Neden bu bilime ihtiyaç var?

Problem Tanımı: Veri çok hızlı büyüyor ve içinde altın (para!) değerinde bilgi saklı. Bunun analizini insanlarla yapmak artık imkânsız. Bunu makinelere -otomatik- yaptırmak mümkün mü?

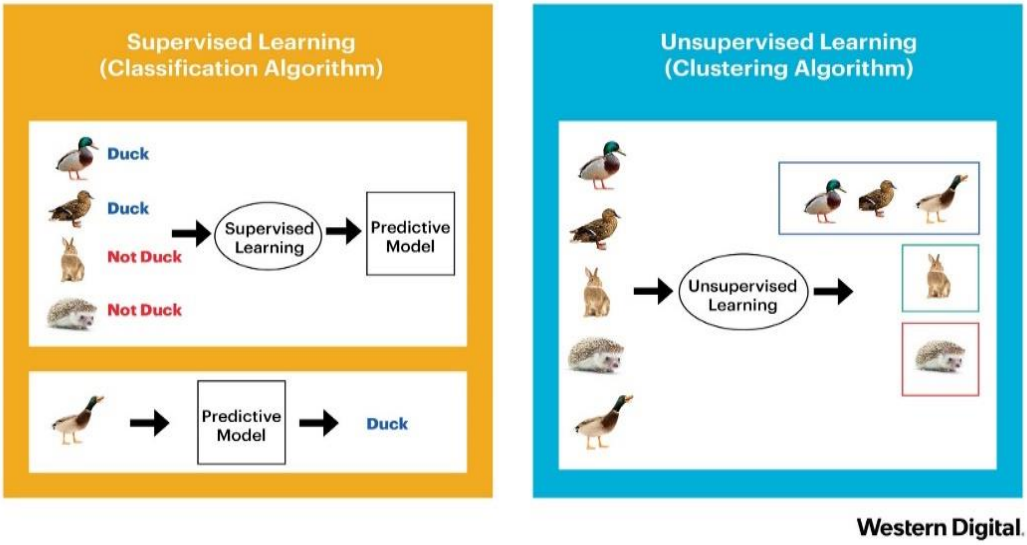
Bu sorulara veri bilimi EVET diyor ve yukarıda adı geçen birbirine benzer alanlar ortaya çıkıyor.

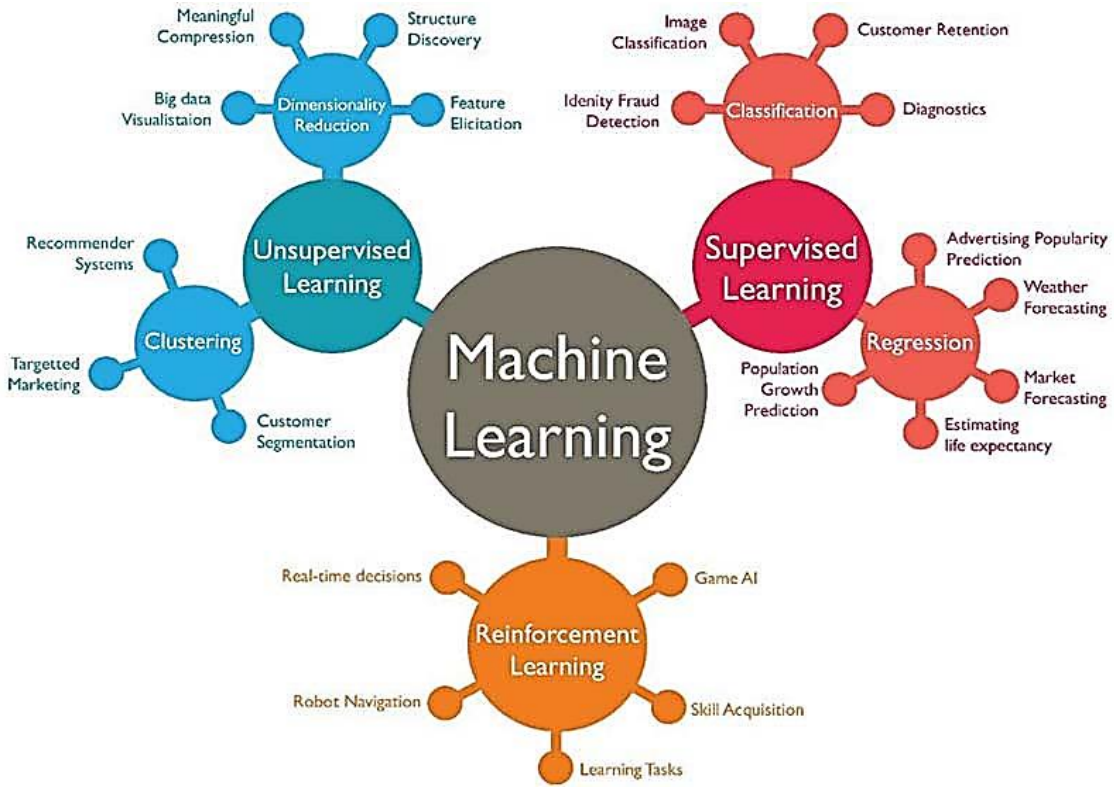
S2: Makine Nasıl Öğrenir? Kullanılan Yöntemler?

Makineler Supervised ve Unsupervised yöntemlerle öğrenirler. Bu iki grubun dışında kalan üçüncü öğretim yöntemi Reinforcement öğrenmedir. Bunlara ek olarak supervised kabul edilebilecek bir analiz yöntemi "Association Rule Mining- Market Sepeti Analizi" de yöntemlere eklenmelidir.

- Supervised (Öğreticili/Gözetimli) Öğrenme: Past/Eldeki verilerle eğitilen algoritmanın (train aşaması) daha önce eğitimi sırasında hiç görmediği verilerle performansının ölçülmesi (test aşaması) süreci. Yani algoritmayı eski veriyle eğit, yeni veriyle (daha önce karşılaşmadığı veriyle) test et : 80/20 kuralı 60-40 70-30 (validation!) train-valid-test
- Unsupervised (Öğreticisiz) Öğrenme :Veri içinde birbirine benzer kümeler bulma. Benzerlik ise verinin türüne bağlı olmakla birlikte renkler gibi düşünürsek verinin birbirine yakın renkler şeklinde kümelenmesi yöntemidir. KÜMELEME

Her iki yöntemin felsefesini aşağıdaki şekille karşılaştıralım...





S3: Veri Derken Matematiksel Model Olarak Neye Benziyor? Neye Benzetip Çalışıyoruz?

Sayısal Tahmin Örneği: Ev fiyatı tahmin etme verisinin çok küçük bir bölümü

HavuzBuyukluk	Havuz	Yas	SatisYili	Fiyat
0	0	5	2010	215000
120	0	6	2010	105000
0	0	6	2010	172000
0	0	4	2010	244000
0	0	3	2010	189900
0	0	6	2010	195500
0	0	4	2010	213500
144	0	1	2010	191500
0	0	3	2010	236500

Bu verileri kullanarak aşağıdakini tahmin edeceğiz! Nasıl olabilir? Yorum??

HavuzBuyukluk	Havuz	Yas	SatisYili	Fiyat
118	1	7	2011	???

Veri Seti Kaynağı: https://www.kaggle.com/c/house-prices-advanced-regression-techniques

Bu örnekte problem ev fiyatının değişik emlak özeliklerine göre (büyüklük, yaş, havuz var yok, konum vs) tahmin edilmesi.

AMAÇ: Eldeki bu veriyle bir modeli eğitmek ve sonra "HavuzBuyukluk", "Havuz", "Yas", "SatisYili" değerleri bilinen bir binanın "Fiyat" tahminini yapmaktır.

118	1	7	2011	???

SORU: Bunu emlakçının kullanacağı hale nasıl getirebiliriz? Ne tür sorunlar var? Nelere dikkat edeceğiz? (Veri değişiyor örneğin! Modelin durumu?)

- Modelin güncellenmesi (modelin başarımı aynı kalmaz çünkü piyasa isteklerine göre emlak fiyatları değişir..)
- Modelin servis haline gelmesi, YANİ???
- Modeli kullanacak kişi sayısı

Bu python dilinde "fit(train)-80: modeli eğit, oluştur" ve "predict (test)-20: oluşan modelde yeni değerleri girip sonucu tahmin et" şeklinde özetlenebilir.

Sınıfsal/Kategorik Tahmin Örneği: Churn analizi denilen firma/servis sağlayıcıların müşterilerini kayıp etmemek adına "bırakma" riski olan müşteriyi önceden tahmin edip (prediction) ona göre müşteriye özel kampanya/reklam ile elde tutma mantığı. Çoğunlukla Telekom/GSM firmaları için uygulanan bir yöntem.

Cinsiyet	Yas	OdemeTipi	KonusmaSuresi	Churn
male	64	credit card	98	loyal
male	35	cheque	118	churn
female	25	credit card	107	loyal
female	39	credit card	177	churn
male	39	credit card	90	loyal
female	28	cheque	189	churn
female	21	credit card	102	loyal
male	48	credit card	141	loyal
female	70	credit card	153	churn
male	36	credit card	46	loyal
male	22	credit card	51	loyal

male	34	Kredi kartı	134	????

Burada problem son kolonda verilen bilgilere göre (Cinsiyet, Yas, OdemeTipi, KonusmaSuresi), müşterinin sadık (loyal), terk etmiş (churn) olup olmayacağını belirlemek üzerini bir sınıflandırma (kategori belirleme) problemidir.

Veri Seti Kaynağı: http://docs.rapidminer.com/studio/getting-started/customer-churn-data.xlsx

Daha gerçekçi veri seti için Kaggle'da "customer churn prediction" araması yapınız.

S4: Neden Çok Sayıda Algoritma Var? Yani örneğin sayısal tahmin yapma demek olan "regresyon" problemleri için onlarca, kategorik tahmin yapmak demek olan "sınıflandırma" problemleri için onlarca algoritma var?

- Veri Madenciliği DENEYSEL bir bilimdir. (AutoML var artık!!!)
- Her problemi çözen, her veri tipine uygun evrensel bir algoritma yoktur.
- Algoritmalar hatta farklı algoritmaların birlikte çalıştığı ensemble/topluluk öğrenme modelleri var.

  A : 90; B: 80 buradan ensemble A ve B beraber çalışıyor → 95

S5:

i) ML/Data Mining Geliştirme Ortamları?

- Görsel Ortamlar: WEKA, Knime, Rapid Miner gibi masaüstü ortamları ile çalışma biçimi buna benzeyen Microsoft Azure /AWS gibi ortamlar.
- Kodlama Ortamları : Python Anaconda (Spider, Vscode ortamları kullanılabilir), Google Colab/Jupyter, AWS Jupyter, Kaggle/Jupyter

ii) Hangi Dil?: Python ve R çok kullanılıyor. Python hem bu alanda hem de Big Data (Büyük Veri) alanında yoğun kullanılıyor. Bunun için Python tercih edilebilir.

Not: Deep Learning (bir ML yöntemi!) çalışmalarında çok kullanılan Keras kütüphanesi de Python tarafından destekleniyor.

iii) Hangi kaynaklar? : Çalışmalarımızda veri seti, kısa konu başlığı aramaları için kaggle.com, kod tabanlı ve kısa teknik notlar için github.com ve hemen hemen her konuda kod/anlatım için medium.com (ve benzeri siteler) kullanılıyor.

S6: Eğitilen Algoritmalar, Geliştirilen Yöntemler Nasıl Kullanılabilir?

Yaptığımız analizler ya şirket/firma için bir problemi çözmek veya elde ettiğimiz modeli servis olarak kullanıcılara sunmaktır. Servis bağlamında, yüz tanıma, doğal dil işleme, bir tahmin probleminin çözümüne dışardan erişme isteniliyorsa bu hedefe:

Ya AWS/Azure gibi bir firmadan (ya da bu işe adanmış servis sağlayıcıları var) destek alıp modeli servis haline getiririz (aslında kullanırız).

Ya da kendimiz örneğin Python kodunu Flask yardımıyla servis şeklinde oluşturabiliriz. Bu tarz bir hackaton da planlanıyor

"Neden servise ihtiyaç var" dersek: Bir chatbot, otomatik mail cevaplama hizmeti ve benzeri bir yazılım projesinin bir parçası, bir modülü geliştirilmiş olsun. Bu model YA yazılım projesinde çalışan kişilerin birikimine (JavaScript gibi bir başka dil) çevrilecek VEYA onlara bir API üzerinden hizmeti kullanma bilgisi şeklinde paylaşılacaktır.

Örnek : Aşağıda bu tarz bir çalışma için model verilmiştir. "python machine learning service flask" gibi aramalarda karşımıza benzer sonuçlar çıkabilir. Araştırınız.

https://www.analyticsvidhya.com/blog/2017/09/machine-learning-models-as-apis-using-flask/

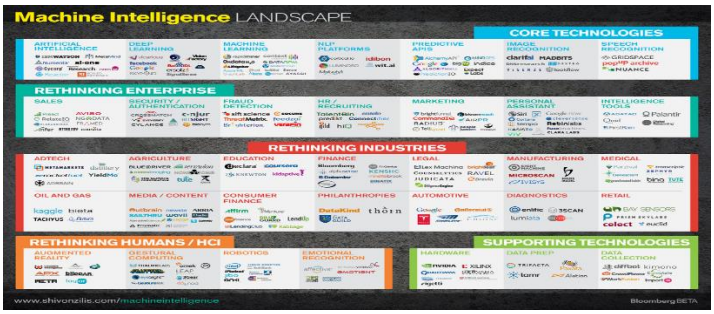
S7: Deep Learning ile Çoğu Problem Çözülebiliyor mu? Diğer algoritmalara ihtiyaç var mı hala?

Deep Learning çok katmanlı (deep'liği buradan geliyor) bir Neural Network algoritmalar grubudur. Çok alanda çok ciddi performans göstermiş özellikle görüntü/ses tanıma da, dil işlemede yoğun kullanılmaya başlanmıştır. Burada Deep Learning vs Diğer geleneksel algoritmalar kıyasında ortaya çıkan kritik sonuç şudur:

DL çok yoğun GPU gücü ve eğitim için çok veri istiyor. Bu nedenle aynı veya yakın performansı gösteren bir geleneksel algoritma standart bir CPU ile gerçeklenebildiği için DL'ye tercih edilebilir. Eğitim verisi az olan problemler için DL kullanımı gerçekçi olamaz.

Daha detaylı bir kıyaslama için aşağıdaki linki BİRLİKTE inceleyelim:

https://www.analyticsvidhya.com/blog/2017/04/comparison-between-deep-learning-machine-learning/



Bu resmi linkten gözden geçirelim..

SON NOT: ML için bir dil öğrenmek? (Örneğin ne kadar python bilmeliyiz?)

Temel olarak veri bilimi için öğrenilmesi gereken standart işlemler vardır.

i) Verinin Ön İşlemesi,

ii) Öznitelik Seçimi/Transformasyonu,

iii) Değişik Metodların denenmesi (Test/Train veya Cross Validation bölmesi) ve başarım sonuçlarının elde edilmesi (Nümerik tahminlerde Mean Squared Error/RMSE, Kategorik tahminlerde Accuracy, Fmeasure gibi) ve bu sonuçlara bakarak uygun algoritmanın tespiti,

iv) Parameter Tuning (seçilen modelin ideal parametrelerinin bulunması)→ başarım değişiyor

v) Modelin deploye edilmesi, canlıya alınması, servis haline gelmesi

Bu sıralama bağlamında bir dili öğrenmek bu adımların nasıl yapıldığını öğrenmek gibi düşünülmelidir.

scikit_cheat_sheet üzerinden python kodlarına bakalım