DERS 1

Makine Öğrenmesi Nedir = İnsanların öğrenme şeklini taklit etmek için, veri ve algoritmaların kullanımını yapan bilgisayar ve istatistik bilimi dalıdır.

DERS 2

Makine Öğrenmesi Nerelerde Kullanılır = Spesifik olarak bir alanı yok, her alanda kullanılmaya başlandı ve her alanda kullanılmaya devam edilecek gibi görünüyor.

DERS 3

Supervised Learning (Denetimli Öğrenme) = Hem bağımlı değişken x, hem de bağımsız değişken y varsa buna Supervised Learning deniliyor | (Konut fiyatları) y = B0 + B1 \* x(Oda Sayısı) + B0 + B1 \* x(Cephe yönü) + … + diye gider | (Etiketli (Label) Gözlemler) Ana veriden gelen etiketler 2’ye ayrılırlar:

1. Eğitim (Training set)

-Makinanın öğrenme yapısı gibi

1. Test set

-Tahmin modeli üzerinden yapıyor

ETİKET = Bir çocuk, yolda annesiyle yürürken bir kedi gördü. Gördüğü kedi de 4 ayak, kuyruk, göz, kulak, tüyleri vardı ve annesi çocuğa dedi ki, 'O bir kedi.' Biraz daha yürüdüler ve bir köpek gördü. Gördüğü köpeğin 4 ayak, kuyruk, göz, kulak, tüyleri vardı ve buna çocuk 'kedi' dedi. Ancak anne, 'Hayır, çocuğum, bu kedi değil, çünkü kediler miyavlar, bu ise miyavlamıyor,' dedi. Bu sayede kediye eşsiz olan bir şey yüklenmiş oldu: miyavlaması. Makine öğrenmesinde kedilere 0 etiketi, köpeklere ise 1 etiketi verdiğimizde, ne kadar çok veri verirsek, o kadar iyi öğretmiş oluyoruz. Her bir kedi resmine 1 sayısı değilse 0 sayısı geliyor. Bu duruma da y(Kedi) = f(x)(DATA RESİMLER) deniyor. Her bir f(x), y değerinin etiketi olmuş oluyor. Olay bu.

Eğitim seti = Makinanın öğrenmesi için verilen veriler

Test seti = Sonuç olarak, biz bir model çıktısı vereceğiz. Model çıktısı üzerinden doğruluğunu test etme işlemine tabi tutacağız.

\*\* Hangi Modeller Kullanılır: 1- Regresyon | 2- Sınıflama (Y değişkeninin kategorik olmasından dolayı). Yukarıda bahsettiğim etiket konusu sınıflamaya girerken, Regresyon ise ev fiyatı ile oda sayısı arasında bir bağlantı arandığında kullanılır.\*\*

DERS 4

Unsupervised Learning (Denetimsiz Öğrenme) = Burada bağımlı değişken yok, kardeşim. Bunların hepsi bağımsız değişkenler. Sen spesifik olarak bunları ayır, sınıflandır ve bir öğrenme sürecine gir, diyoruz. Kümeleme yöntemi, boyut indirgeme ve kümeleme dediği şey, benzerleri ayrı şekilde kümeliyor.

DERS 5

KAGGLE SİTESİ

DERS 6

PYTHON + NORMAL DERS

Eğitim ve Test Verisi Ayırma

DERS 7

Overfitting and Underfitting (Aşırı ve Eksik Öğrenme) = Aşırı öğrenme sorunu; Üniversite sınavına hazırlandığını düşün ve tüm kitabı yalayıp yuttun ve sınava girdin, aşırı zor geldi. Çünkü sınav yoruma dayalı sorular sordu, sen ise ezberlemeye dayalı bir yöntem kullandın. Burada tüm kitap senin eğitim veri setin, makina sensin. Ama bizim yaptığımız bir şeyleri öğrenmek değil, onları sadece ezberlemek oldu.

Eksik öğrenme sorunu; Anne ile çocuğun kedi hakkında konuştuğunu düşün. Bu sefer çocuk sadece beyaz bir kedi görüyor. Tüm gördüğü kediler de beyaz renkli. Günlerden bir gün siyah bir kedi görüyor, bu siyah kedinin bir kedi mi yoksa başka bir şey mi olduğuna karar veremiyor. İşte buna eksik öğrenme deniyor, yani eğer veri setin az sayıda ise bu tür sorunlar ortaya çıkabilir.

Eğitim veri setimiz, Test veri setimiz, Gerçek hayattaki veri setleri.

Test V.S veya Gerçek V.S iyi ise Optimal (Gerçeğe en yakın olan model yapısıdır)

Test V.S veya Gerçek V.S iyi değil ama Eğitim V.S çok iyi sonuçlar veriyorsa = Aşırı öğrenme sorunu vardır

Test V.S veya Eğitim V.S iyi değilse = Eksik öğrenme problemi ortaya çıkar

DERS 8

Önyargı(Bias(Underfitting)) ve Varyans(Overfitting) = Hata = İndirgenmiş Hata + İndirgenemez Hata | İndirgenmiş hata, model üzerinde iyileştirmeler yaparak hatayı minimize etmeyi amaçlar ve iki şeyden oluşur: 1. Bias (Yanlılık) 2. Varyans. İndirgenemez hata ise, ne yaparsak yapalım bu hatayı azaltamayacağımız hata türünü ifade eder

Bias = Elimizdeki verilere göre gerçek olması gereken modelden uzaklaşma durumudur. Örneğin, çok küfür eden bir toplumdan veri seti aldın ve makineye bu durumu öğretirken, makinenin düşündüğü şey, herkesin küfür ettiği, bu yüzden bu durumun normal olduğunu düşünür, ona göre hareket eder. İşte bu duruma Bias denir

Varyans = Üniversite hayatında 4 yıl boyunca sana öğretilen her şeyi öğrendin ama iş hayatına geldiğinde üniversitede öğrendiğin bilgilerin çoğunun sana işte yaramadığını fark ettin. İşte bu durum, sana işe yaramayacak verileri zihninde tutup üzerlerine kafa yormanın vakit kaybına yol açtığını gösteriyor

DERS 9

Model Validation (Model Doğrulama) = Doğrulama, eğitim veri seti içerisinde kullanılır. Model rastgele olarak veri setinin içinden örneklem çeker, ancak bu çekilen örnekleri doğrulamak için eğitim veri setini bölme işlemi yapar

Hold-out, Verisetini eğitim ve test olarak 2’ye ayırıyor. “Dataseti küçükse bununla çalışılır.”

k-fold cross validation (k katlamalı Çapraz doğrulama) = Hold-out geliştirilmiş hali | Eğitim data setinin içinde bir daha eğitim ve test olarak 2’ye ayırıyor.

DERS 10

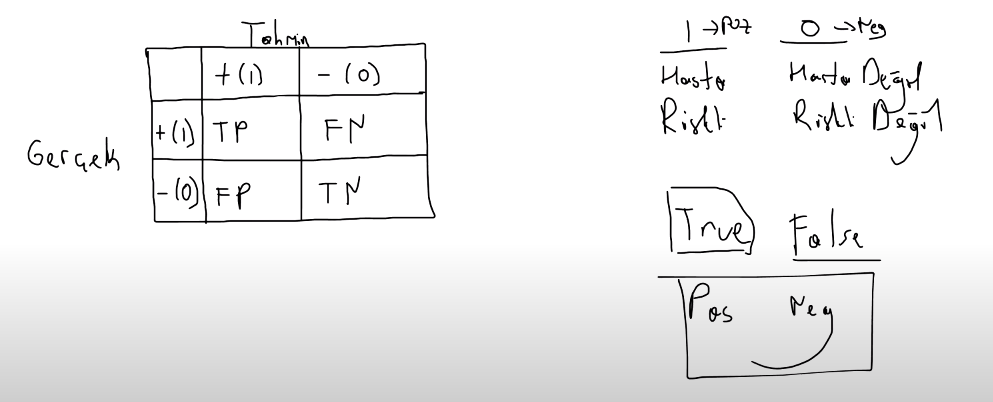
PYTHON + NORMAL DERS

Regression Model Performance (Regresyon Model Başarısı) = Regresyon model başarısı, bir regresyon modelinin, belirli bir problemin çözümü için ne kadar iyi performans gösterdiğini değerlendiren ölçütler.

Mean Squared Error (MSE)(Ortalama Hata Karesi) = Karesini alma sebebimiz, negatif değerlerden kurtulmak ve bir sayının karesini aldıkça hata ortalamasının daha yukarı çıkacak olmasıdır. Bu da bize hataların dağılımının ne kadar uzak olduğu hakkında basit bir bilgi verecektir

Root-Mean-Square (RMSE)(Ortalama Karekök Sapması) = Yukarıda bulduğumuz hatanın karekökünü alır. Bu, hataların standart sapmasını ortaya çıkarır

Mean Absolute Error (MAE)(Ortalama mutlak hata) = Hataların karesini almaktan ziyade mutlak değerini alıyor, yani sadece negatif işaretleri ortadan kaldırıyor

DERS 11-12

Confusion Matrix (Hata Matrisi) 1. Bölüm

Confusion Matrix (Hata matrisi) =

T= True | F = False | P = Pozitif | N = Negatif | P ve N değerlerini Tahminin üzerinden alıyor.

1. Doğruluk (Accuracy): (TP + TN) / (TP + FP + TN + FN) | Modeli yüzde kaç tahmin ettiğini verir
2. Hata Oranı = (FP + FN) / (TP + FP + TN + FN) | Daha az kullanılır
3. Hassasiyet(Kesinlik)(Precision): TP / (TP + FP) | Hassasiyet, modelinizin spam olarak tahmin ettiği değerlerin ne kadarının gerçekten spam olduğunu ölçer. Yani, modelinizin spam olarak etiketlediği örneklerin kaçı gerçekten spamdır. Bu, modelinizin yanlış pozitif (spam olmayan bir e-postayı spam olarak işaretleme) hatalarını ne kadar az yaptığınızı gösterir
4. Duyarlılık (Recall): TP / (TP + FN) | Duyarlılık, gerçek spam olan örneklerin kaçını doğru bir şekilde tespit ettiğinizi ölçer

Kesinlik ve Duyarlılık arasında hangisine bakılacak = Eğer hangisi daha maliyetli ise ona bakılır, yani kanseri olan birine sen kanser değilsin dersen, bu durum çok fazla maliyetli olacağı için FN (Yanlış Negatif) duyarlılığına bakılacak

Confusion Matrix (Hata Matrisi) 2. Bölüm

F1-Score (F1-Skoru): 2 \* (Hassasiyet \* Duyarlılık) / (Hassasiyet + Duyarlılık) | Hem kesinlik hem de duyarlılık üzerinden bir işlem gerçekleştiriyor |

ROC Curve (ROC Eğrisi) = TP ile FP oranları birbiriyle kıyaslamasını gösteren bir eğri yapısı.

DERS 13

PYTHON

Simple Linear Regression (Basit Doğrusal Regresyon)

DERS 14

PYTHON

Multiple Linear Regression (Çoklu Doğrusal Regresyon) = Birden fazla bağımsız değişken var, bunun içinde

DERS 15

PYTHON

Multiple Linear Regression (Çoklu Doğrusal Regresyon-2) = DERS ÖZETİ: Eksik verileri sklearn kütüphanesi ile doldurma işlemi |

R2 skoru: R2 skoru, bir regresyon modelinin veriye ne kadar iyi uyduğunu ve bağımsız değişkenlerin bağımlı değişkenin varyansını ne kadar iyi açıkladığını ölçen bir istatistiksel metriktir.

mean\_squared\_error (Ortalama Kare Hata): Regresyon modellerinin performansını değerlendirmek için kullanılan bir metriktir.

mean\_absolute\_error (Ortalama Mutlak Hata): Bir Regresyon modelinin performansını değerlendirmek için kullanılan bir metriktir.

DERS 16

PYTHON

Multiple Linear Regression (Çoklu Doğrusal Regresyon-3) = DERS ÖZETİ: Kategorik değişkenleri regresyon modeline nasıl entegre ediyoruz, onun işlemlerini yapacağız. Label ve One-Hot olarak ayrılıyorlardı. Eğer hiyerarşi varsa, Label yoksa One-Hot kullanıyoruz

DERS 17

PYTHON + NORMAL DERS

Model Tuning = Model üzerinde ince ayar yapma işlemi. Radyo dinlerken frekansını bulursun, ama ses gene biraz cızırtılı gelir. Biz o cızırtıyı gidermek için çok az frekansla oynarız, ki ses net gelsin. İşte bu örneğe dayanarak, model üzerinde o ince ayarı yapma gibi düşünebilirsiniz. O ince ayarı yapmaya yarayan aracın adı Hyperparameter (Hiper parametre).

DERS 18

Polinomal Regresyon = Doğrusal değil, polinomsal hali sadece Excel'de örnek yapıldı

DERS 19

PYTHON

Polinomal Regresyon-2

DERS 20

PYTHON

Polinomal Regresyon-3

DERS 21

Ridge Regression (Ridge Regresyon) = EKK'nın çoklu doğrusal problem ve aykırı gözlem problemleri bazen çözüm bulmakta çok zorlanabiliriz. Bu nedenle, bu EKK'nın sorunlarını daha fazla minimize etmek amacıyla yeni bir model geliştirilmiştir. İşte bu model, Ridge Regresyon modeli olarak adlandırılır. Formülünde EKK'dan farklı olarak bir Hyperparameter ve Ceza Terimi (Penalty Term) eklenir. Değişken sayısı gözlem sayısından fazla ise ve bazı değişkenleri çıkarmadan önemini azaltmak istiyorsak, Ridge Regresyon kullanılabilir.

DERS 22

PYTHON

Ridge Regression (Ridge Regresyon-2) = Ridge Regresyon'un Python'da nasıl uygulandığını öğreniyoruz.

DERS 23

PYTHON

Ridge Regression (Ridge Regresyon-3)

DERS 24

PYTHON

Lasso regression (Lasso Regresyon) = Ridge Regression ile benzer bir yapıya sahip, ancak Ridge'de gereksiz olanları minimize ederken bu yöntem doğrudan ortadan kaldırıyor

DERS 25

PYTHON

Lasso regression (Lasso Regresyon-2) = Alfa değerini, yani hiperparametre ayarlama işlemlerini.

DERS 26

PYTHON

Elastic Net Regression (ElasticNet Regresyon) = Hem Lasso'yu hem de Ridge'yi beraber kullanan bir yapıdır.

DERS 27 (A)

PYTHON

Doğrusal Regresyon Örnek = Bağımsız değişkenler arasında bir ilişkinin olup olmadığını kontrol ettik. Grafik, Korelasyon ve VIF ile

DERS 28

Principal Component Analysis (PCA) (Temel Bileşenler Analizi) = WinRAR gibi programların amacı büyük verileri sıkıştırarak küçültmektir, böylece işimiz hızlanır. Temel bileşenler analizi ise WinRAR gibi çok fazla bağımsız değişken içeren verileri basitleştirir, gereksiz bilgileri atarak daha düzgün ve doğru bir şekilde, yani hataları minimize ederek sunar. Bu Temel Bileşenler Analizi içinde 3 önemli kavram bulunur.

1. Öznitelik = Bağımsız değişken yapıları | Modele bir vektör yardımıyla girer
2. Özdeğer
3. Özvektör

DERS 29

Özdeğer ve Özvektör = Örnekleri ve açıklaması notta var

DERS 30

PYTHON

PCA Örnek-1 = Standartlaştırma işlemi ve verileri küçültme ve sıkıştırma işlemi yapıyoruz, ancak bu küçültme ve sıkıştırma işlemini yaparken en az veri kaybetmeyi amaçlıyoruz. İşte bu, tekrar bize bir hiperparametre olarak dönüyor. PCA Örnek-2'de bu hiperparametreyi ayarlayacağız

DERS 31

PYTHON

PCA Örnek-2 = En az bağımsız değişken kullanarak en iyi modeli açıklayanı bulmak, radyonun frekansını ayarlama gibi.

DERS 32

PCA Geometrisi = 1. Amacımız doğru açıyı bulabilmek. 2. Max varyansı ortaya çıkaran açı, bizim doğru açımızdır. En yakın doğru noktalara, bu noktaların yansımalarını alır, mesafeleri ölçer, ardından varyansı maksimize etmeye çalışan açıyı da bulur. “Genel\_mantık” resminde değerlerin karesini alıp topluyorduk, işte ona özdeğer denir. En uygun açıya da 1. Birleşen Özvektör denir.

DERS 33

PYTHON

LDA Nedir = Veri sınıfları arasındaki varyansı maksimize etmeye çalışılan bir yöntem.

DERS 34

Sohbet muhabbet

DERS 35

Support Vector Regression = Hata yapma toleransımızı düzenleyen bir yapı ortaya çıkarıyor. Bir güven aralığı gibi bir çizgi olduğunu düşün. O çizgiler arasında bir doğru oluşturur

DERS 36

PYTHON

SVR Örnek-1 = Basit bir şekilde Support Vector Regression Python'da nasıl kullanılır.

DERS 37

PYTHON

SVR Örnek-2 = Support Vector Regression kurulumu ve hangi modelin daha iyi çalıştığının tespit işlemleri.

DERS 38

PYTHON

SVR Model Tuning = SVR'de hyperparameter işlemlerini halletme konusu.

DERS 39

Decision Tree Regressor (Karar Ağacı Regresyonu) = Yalnızca koşullu kontrol ifadelerini içeren bir algoritmanın bir görüntüleme yöntemi. Koşulların bir ağaç yapısıyla dallarıyla görüntülenmesini sağlayan algoritma yapısı.

DERS 40

PYTHON

Karar Ağacı Regresyonu-2 = Karar ağacı regresyon yapısının Python içinde nasıl kurulacağı.

DERS 41

PYTHON

Karar Ağacı Regresyonu-3 = Karar ağacı regresyon yapısının Python içinde hyperparameter işlemlerini halletme.

DERS 42

Bootstrap aggregating (Bagging Karar Ağacı Regresyonu) = Overfitting hatasına düşmemek için kullanılan yöntem, veri içinde birden fazla karar ağacı üretmeye çalışıyor

DERS 43

PYTHON

Bagging Karar Ağacı Regresyonu-2 = Bagging yapısının Python içinde nasıl kurulacağı.

DERS 44

PYTHON

Bagging Karar Ağacı Regresyonu-3 = Bagging yapısının Python içinde hyperparameter işlemlerini halletme.

DERS 45

Random Forest Regresyon = "Bagging," verilerden rastgele örnekler çekerek yeni bir model oluşturur ve bu örneklerin tahminlerinin ortalamasını alırken, bu oluşturma sırasında hatalar ortaya çıkabilir. Bu hataların bazıları çok büyük olduğu için ortalama tahminleri çok fazla etkileyebilir. Bu nedenle, bu durumda "Rastgele Ağaç Regresyonu" olarak adlandırılan yeni bir yaklaşım geliştirilmiştir.| Bagging'den farklı olarak, Random Forest, değişkenleri göz önünde bulundurarak örneklem alır. Yani, Bagging tüm bağımsız değişkenlerle çalışırken, Random Forest istediği bağımsız değişkenlerle çalışır.

DERS 46

PYTHON

Random Forest Regresyon-2 = Random Forest Regresyon yapısının Python içinde nasıl kurulacağı.

DERS 47

PYTHON

Random Forest Regresyon-3 = Random Forest regresyon yapısının Python içindeki hyperparameter işlemleri. Karar Ağacı, Bagging ve Random Forest'ı aynı anda uygulayarak farklarını gözlemlemek.

DERS 48

PYTHON

Toplu Model Çözümü = linear regression | polynomial regression | ridge regression | lasso regression | elastic net regression | support vector regression | decision tree regression | bagging regression trees | random forest regression |

DERS 49

Classification (Sınıflandırma Algoritmaları) = Önemli olan kelime etiketleme yapısı yapısının olması gereklidir | Kategori olması gerekiyor bunu yapmamız için.

DERS 50

Logistic Regression (Lojistik Regresyon Nedir) = Sigmoid fonksiyonu kullanılır. S dalgası şeklinde bir görüntüye sahiptir. Maksimum olabilirlik yöntemini kullanırız. Çok değişkenli, bağımsız değişkenin birden fazla olduğu durumu ifade eder. Sıralı log regresyon ise birbirleri arasında hiyerarşik bir düzenin olup olmadığıyla ilgilidir.

DERS 51

PYTHON

Logistic Regression (Lojistik Regresyon-2) = Lojistik Regresyon yapısının Python içinde nasıl kurulacağı.

DERS 52

PYTHON

Lojistik Regresyon-3 = Python'da normal yapısını kurduk ve hata değerlerine baktık. Burada r2 ve mse gibi değerlerden ziyade hata matrisi üzerinden hata tahminine bakılıyor.

DERS 53

PYTHON

Lojistik Regresyon-4 = Eğer bağımlı değişken 2'den fazla kategorik değişkenden oluşuyorsa, ne yapmamız gerektiği sorusunun cevabı

DERS 54

PYTHON

Lojistik Regresyon-5 = Bağımlı değişkenler arasında bir hiyerarşi varsa, ne olur onun örneğini yapıyoruz.

DERS 55

PYTHON

K-Nearest Neighbors (KNN) = En yakın komşu ölçümü mesafesine bakar. Eğer en yakın olanlardan hangisi çoğunluksa, ona göre işlem yapılır. Eğer bir eşitlik durumu söz konusu ise, gruplar arasında minimum mesafeye sahip olanı seçer.

DERS 56

PYTHON

K-Nearest Neighbors (KNN-2) = KNN model kurulumu. KNN model kurulumunda hata verirse “pip install --upgrade threadpoolctl” komutunu kullanarak hatayı çözüyor.

DERS 57

PYTHON

K-Nearest Neighbors (KNN-3) = K-Nearest Neighbors (KNN-3) yapısının Python içindeki hyperparameter işlemleri.

DERS 58

Support Vector Machine (SVM Nedir) = Regresyonda, 2 doğru arasında kalanlar bizim en optimal doğrularımız olurken, sınıflamada ise iki sınıfı ayıracak en iyi doğruyu arıyoruz.

DERS 59

PYTHON

Support Vector Machine (SVM-2) = Support Vector Machine modelinin kurulumu.

DERS 60

PYTHON

Support Vector Machine (SVM-3) = Support Vector Machine yapısının Python içindeki hyperparameter işlemleri.

DERS 61

Bayes Theorem (Bayes Teoremi) = Bağımlı değişkenlerde katergori sayısı daha fazla olduğunda başarılı sonuçlar ortaya çıkarır.

DERS 62

Naive Bayes Classifier (Naive Bayes Nedir) = Notlar 48. Sayfa da örnek var. Bunlar 3 dala ayrılırlar:

1. Bernoulli dağılımı = Sadece 2 sonucu varsa kullanılan bir yapıdır. Evet Hayır gibi
2. Multinomial Distribution = Bir frekans sayımı yapıyo 1,2,3,4 diye ilerlerim diyor
3. Gauss Normal Dağılım = Normal dağılımda bildiğimiz normal dağılım şartlarına uyması gerekiyor.

DERS 63

PYTHON

DERS 64

PYTHON

Sms Spam Detection-1 = Bayes yapısıyla ilgili bir örnek. Bir veri seti üzerinde ne anlamlar çıkarabiliriz, nelere dikkat etmemiz lazım. Verimizi uygun olan çalışma modeline uygun bir şekilde nasıl hazırlarız onun pratiğini yapıyoruz.

DERS 65

PYTHON

Sms Spam Detection-2 = Örneğe devam.