**2.1.1 – eksik veriler**

Nan değerlere ortalama yazdırma

imputer = SimpleImputer(missing\_values=np.nan, strategy='mean')

**3.2.2 \_şablon\_veriler**

Veriler.csv Ülke yaş boy kilodan ciniyet tahmin eden lineer reg hale getirme

encoder: Kategorik to numeric

Sayısal olmayan verileri sayısal hale getirme

2 değişken varsa le yeterli-true/false 1/0

Ancak fazla değişken varsa ohe kullanmak gerekir. Çünkü aralarındaki ilişki doğrusal değil

le = preprocessing.LabelEncoder()

ohe = preprocessing.OneHotEncoder()

numpy dizileri dataframe donusumu

dataframe birlestirme islemi

#verilerin egitim ve test icin bolunmesi

ölçekleme-standar scaler- sc-veri benzeştirme

lineer reg

**6\_ECDF\_histogram**

ECDF histogram cizdirme datacamp

**7\_lineer reg**

Satışlar csv deki aylara göre satışlardan regresyon elde ettik

Sort\_index()

**8\_multireg**

Veriler.csv deki verilerden tahminler elde ettik model oluşturduk

Ve bunları karşılaştırarak sonuçlar elde ettik

import statsmodels.api as sm

#dummy variable

#model cinsiyet tahmin

# backward elemination,

# p value

model = sm.OLS(boy,X\_l).fit()

OLS tablosu

R2 değerleri

**9\_tenis v1\_v2**

Tenis verilerinden humudity i bulmaya çalıştık

-tüm tabloya le uygulama

veriler2=veriler.apply(preprocessing.LabelEncoder().fit\_transform)

genel uygulamalar

ikisi arasında ufak tefek farklar var incelenebilir.

**3.3.3 polreg(silindi) 🡪 3.6.1 lin,pol, svr,dt, rf,silindi🡪3.7.1**

linner reg plot çizdirme

plynomal reg

lineer regresyonu poly ile girdilerin ^2 ^3 alarak daha doğru hale getirdik

tahminler yaptık single değer için

r2\_score

svr

decision tree

Z=X+0,5 – farklı değerlere ne sonuç veriyor model

Random forest

Model grafikleri

rbf-gausian

**4\_Titanic model evo sololearn**

Accuracy,TP TN gibi kapsamlı açıklamalar var

k-fold – train test spliti farklı farklı yerlerden ayırma

ikili girdili for döngüsü

accuracy

model.score

**3.5.1**\_svr support vektör regressiom 🡪 3.6.1 de devam ettiği için silindi

rbf-gausian

**10.3\_ödev8**

OLS grafik-p value

veriler.corr()-korelasyon

**4.9 şablon sınıflandırma- sololearn model evo benzeri,**

KNeighborsClassifier

K değeri önemli, k 1 seçilirse tüm örneklerin yaşam sansıolur . k=n olursa %51 olan tüm uzayı domine eder.

Genel yaklaşım K değeri= karekök(eğitim boyutu(X\_train))/2

sklearn.svm import SVC

from sklearn.naive\_bayes import GaussianNB

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

# 7. ROC , TPR, FPR değerleri

Hangi veriler yanlış

**13\_KMeans AgglomerativeClustering**

Gözetimsiz verilerde kullanılır. (belli bir sonuç yok. Survived-not survived gibi)

Rastgele noktalar belirleyerek cluster yapıyor.

Cluster sayısı belirleme önemli. Ona göre noktalar beliliyor. N\_cluster

Buna göre veriyi bölüyor.

For döngüsüyle en yüksek puanlı k nokta sayısı bulunabilir.

Agglomerative tüm clusterları ayrı ayrı bölüyor. Sonra en yakın 2 clusterı birleştiriyor. Sonra yakın clusterları birleştiriyor. Böyle böyle gruplar . çıkıyor.

Scatterla çizme

AgglomerativeClustering

dendogram

APRİORY

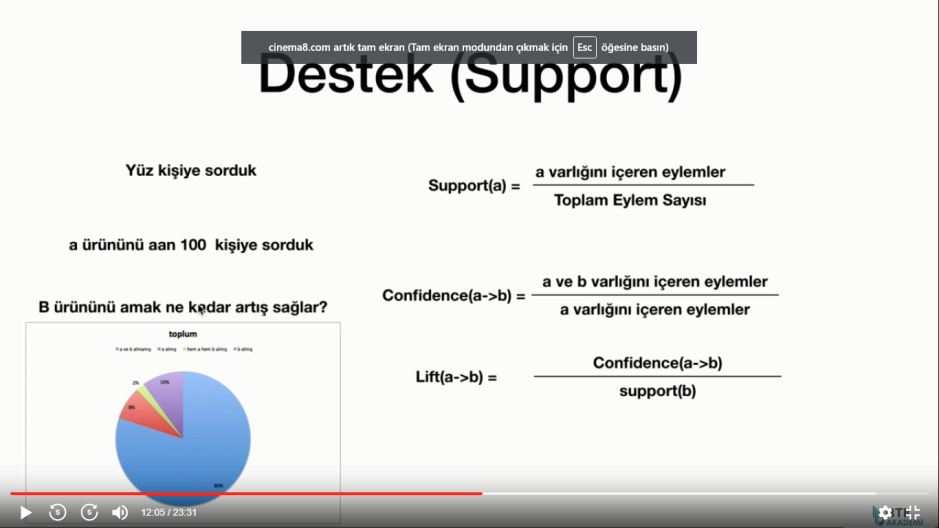
associacion rule model- bunu alan bunu alır(örn: bebek bezi alan, bira da alır) netflix de bu algoritma

corolation or causality

köpek balık saldırısı artınca dondurma satışı da artıyor (corolation)

havalar ısınınca dondurma satışı artıyor (causality)

**Destek (support)**

****

Lift 1 den büyükse- a ürünün üzerine odaklanmak b ürünü satışınıda arttırır. Bu örnekte lift=1,66

Lift 1 den küçükse- ?

Sklearnda yok

ECLAT

Equivalance class Transformation

Reinforced learning

**15\_Reinforced learning**

THY reklam tıklaması, hangi reklam daha başarılı

Random selection

UCB( upper confidance selection) –ödül kazanma, kazanamama

5 farklı reklam var 1 tanesi daha başarılı başarılı olanı göstermeye çalışıypruz

18\_NLP DDİ

Restoran yorumlarının iyi kötü olarak sınıflandırılması

Re.sub – cümle içinden istemediğimiz bölümü çıkarma

yorum = re.sub("[^a-zA-Z]"," ", yorumlar["Review"] [0])

karakter olmayan her şeyi çıkaraır

alfanumerik karakterleiri silme

büyük küçük harf değiştirme

stopwords – anlamsız kelimeler. Bunları bulma

NLTK- tool kit

STEMMER-kelimeleri eklerinden ayırma

CountVectorizer – kelimeleri suffix matrix haline getirdi

**Sololearn 7\_neural Networks**

Creating Artificial Dataset- sistemden hazır database çekme

s= X[y==0][:,0] – y 0 olan tüm satırları seç

güzel bir ayırma var- (100,2) lik bir veri var. Bu veri içinde Y=0 olanlar ayrı x=0 olanlar ayrı olacak şekilde scatter yapıldı

MLP solver- yapay sinir ağlarındaki bir regresyon gibi bir yöntem

MNIST dataset- handwrite predicting

Drawing the Digits-matshow

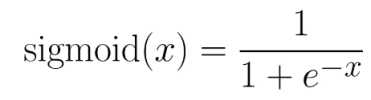
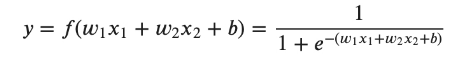
1x64 Bir data satırını çekip 8x8 hale getirip onu bastırdık. matshow

Yanlış tahmin ettiğimiz satırlar( incorrect = X\_test[y\_pred != y\_test])

Ve onların tek tek plotları

MLP weights

--------------------------------

Yukarıdaki formül bir sinir nöronun girdiye vereceği çıktı formülü

**19\_Yapay sinir ağları (YSA) artificial neural network(ann)**

Perceptron- geri yansıtma

Önce ağırlıkları değiştiriypruz

Gradyan alçalış (gradiend descendent)- belli adımlarlar la optimun-m noktaya iniyor. Ama çok daha iyi bir noktayı bulmamızı engeller.

Stochastic gradient descendent- yukardaki kötüleşen durumu biraz daha iyi hala getirir.

Satır satır alıp değerleri güncellersek stochastic yaklaşım, 100 satırı okursak batch(yağın) yaklaşım

Back propagation: ( en önemli yaklıım)- kaç tekrarın olacağını belirlemek önemli. Learning rate, ne kadar öğrendiği. Epox? Kaç defa de öğrenceği.

Derin öğrenme kütüphaneleri:

* Pytorch
* Caffe
* Keras- tensorflow üzerinde hızlı işlemler
* tensorFlow- google detaylı
* Theno – işlemleri gpu ya dağıtma

ohe = ColumnTransformer – ohe yi direk data üzerinde uygulayarak işleri hızlandırıyor(?not: yalnız ayrı ayrı columlara dönüştümek yerine 2 lik sisteme dönüştürüyor.

Rectifier.

Optimizer – adam SGD skotastik gradient descending

Losses

classifier = Sequential()

dense

**20\_PCA Boyut indirgeme / LDA**

Pca- Wine veri tabanını kullandık.

13 tane features olan bir kümeyi düzlem değiştirerek 2ye indiriyoruz

from sklearn.decomposition import PCA

başarı çok az düştü ama kullandığımız veri azaldı.

Pca gözetimsiz- tüm veriler aynı

LDA-gözetimli – data pointin hangi sınıfta olduğu önemli, sınıfları ayırır.

https://sebastianraschka.com/Articles/2014\_python\_lda.html

**21\_K-FOLD model değerlendirme grid search**

4\_titanc model evo da benzerini yaptık

k-fold

grid search

from sklearn.model\_selection import GridSearchCV

farklı farklı parametraleri denemek ve en başarılısını bulmak SVM parametrelerini tek tek bakmak

**22\_XGboost**

<https://xgboost.readthedocs.io/en/latest/>

yüksek verilerle iyi sonuç veriyor

hızlı çalışıyor

problem ve yorumun mümkün olması(çok önemli)

**23\_Model kaydetme ve çalıştırma.**

Urlden data indirme

Fit ettiğin veriyi kaydetme böyle tekrar tekrar fit etmeye gerek yok.

**DÖKÜMANTASYON**

Linearly seperable – tek bir doğru ile çözüm kümesi ayrılabiliyormu?

1. **Hatalı satırları bulma**

incorrect = X\_test[y\_pred != y\_test]

incorrect\_true = y\_test[y\_pred != y\_test]

incorrect\_pred = y\_pred[y\_pred != y\_test]

bunu data frame ekleme

--------------------

df = pd.DataFrame(x\_test, columns=["0","1","2"])

df["actual"] = y\_test

df["predicted"] = y\_pred

incorrect = df[df["actual"] != df["predicted"]]

df["incorrect"]=df["actual"] != df["predicted"]

s= X[y==0][:,0] – y 0 olan tüm satırları seç

güzel bir ayırma var- (100,2) lik bir veri var. Bu veri içinde Y=0 olanlar ayrı x=0 olanlar ayrı olacak

1. **Belli s**ayıdan küçük df i ayırma y nin 3 ten küçük olduğu satırları ayırdık (7\_neural net)

X5 = X[y <= '3']

Neural network coeeficents

print(len(mlp.coefs\_))  
print(mlp.coefs\_[0].shape)  
print(mlp.coefs\_[1].shape)  
  
Output:  
2  
(10, 50)  
(50, 4)

10 inputs / 50 nodes in the hidden layer / 4 outputs

---

y\_pred=(y\_pred>0.5)

0.5 ten büyük olanlar 1 olmayanlar 0

1. **1 den 10 a kadar yazdırma**

range(1,10) -plot if içinde kullanılabilir

np.arrange

1. **Dizi işlemleri:**

**Dizi oluşturma**

oduller=[0]\*d 10 elemanlı her elemanı sıfır olan bir dizi

np.arange

**iloc işlemleri**

X= veriler.iloc[:,3:-1] 3. Den sondan 1 öncekine kadar satırlar

**İNDEXİNG**

geo =veriler.iloc[:,4:5].values --🡪size: (1000,)

geo =veriler.iloc[:,4].values-🡪 size(1000,1)

neden böyle oluyor bilmiyorum ama size (1000,) olunca şşlemlerde hata veriyor

yorumlar.shape[0]- kaç row var

x\_train**.sort\_index() –** index numarasına göre sıralama

1. **GÖRSELLEŞTİRME**
   1. X kümesi 2 sutundan oluşsun. Y\_tahminde bu satırların ait olduğu sınıflar. Her sınıfı farklı renkte gösterme için kod(c ile color aynı değil, color=”b” blue yapar)

plt.scatter(X[:,0],X[:,1],c=Y\_tahmin)

yada

plt.scatter(X[Y\_tahmin==0,0],X[Y\_tahmin==0,1],s=100 ,color="red")

plt.scatter(X[Y\_tahmin==1,0],X[Y\_tahmin==1,1],s=100 ,color="blue")

plt.scatter(X[Y\_tahmin==2,0],X[Y\_tahmin==2,1],s=100 ,color="green")

ikiside aynı ikincide y nin true olduğu x satırlarını yazdırıyor. Burada c ile color aynı

--------------------------

X = cv.fit\_transform(derlem).toarray()

1. **Fit Train**

gnb.fit(X\_train,y\_train)- x le yi karşılaştırıp öğrenir

gnb.predict(X\_test) – x teste ilgili kuralları uygulayıp tahmin eder

gnb.fit\_transform—

1. **Katogorik Nümerik**

Aşağıdaki kod kullanılabilir.

le = preprocessing.LabelEncoder()

ohe = preprocessing.OneHotEncoder()

geo = ohe.fit\_transform(veriler.iloc[:,4:5].values).toarray()

geo = pd.DataFrame(data=geo , index=range(geo.shape[0]), columns=["fr","ge","sp"])

gender=le.fit\_transform(veriler.iloc[:,5:6])

gender=pd.DataFrame(data=gender , index=range(gender.shape[0]), columns=["male"])

X = pd.concat([veriler.iloc[:,3:4],geo,gender,veriler.iloc[:,6:-1]], axis=1 )

1. **Pandas numpy**

.values dataframe ->array of object

Pd.DataFrame array of object->dataframe



Add a scalar with operator version which return the same results. Tüm tadaframe bir sayı ekleme

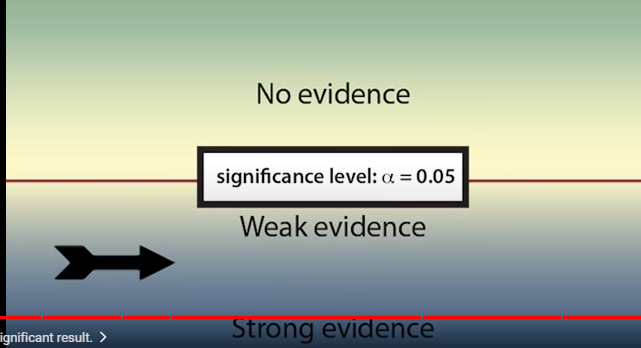
df.add(1)

X=Z+ 1 buda çalışıyor

1. **Regresyon bilgileri(Tahmin Metodları)**
   1. **P-value**

P değeri 0,05 altındaysa (küçüldükçe) H0 ın hatalı olma ihtimali artar(null hipotesis)

P değeri 0,05 üzerindeyse (büyüdükçe) H1 ın hatalı olma ihtimali artar(alternative hipotesis)

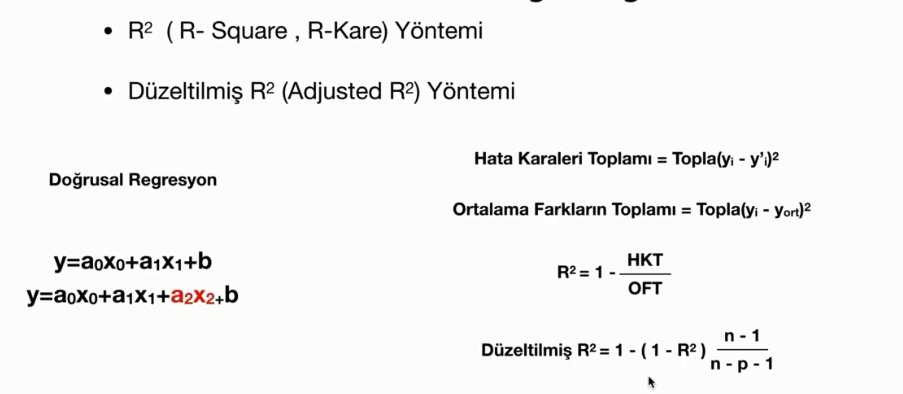
****

0.5 no evidance that H0 is wrong

0,05 weak evidance

0,001 strong evidance H0 is wrong

* 1. **R2 değeri (R square)**

****

Herkese ortalama değer yazsak R2=0 olur. En basit algoritma , R2 –sonsuza kadar gidebilir.

Mükemmel algoritma yazsak R2=1 olur. En iyi algoritma

Düzeltilmiş R2 yöntemi

Doğrusal regresyonda yeni bir feature eklendiğinde;

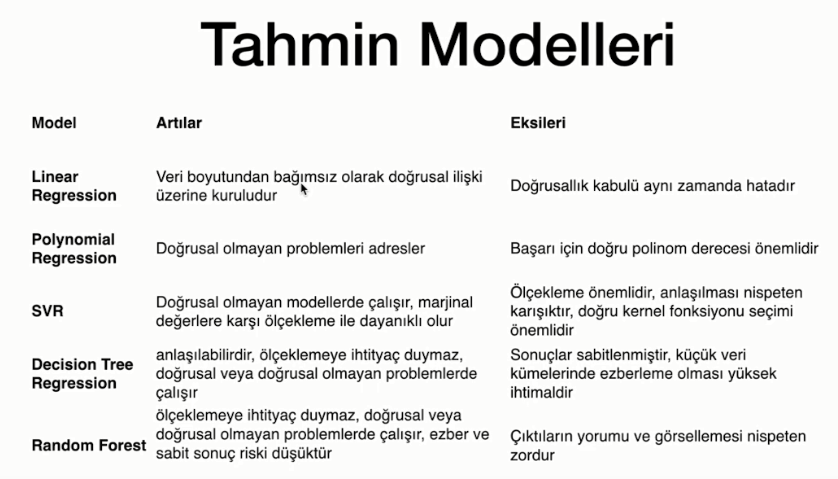
Eğer bu özellik olumlu bir etki yapıyorsa , R2 artar

Eğer bu özellik olumsuz bir etki yapacaksa , denklemdeki çarpanı 0 olur(neden?). bu nedenle R2 değeri değişmez.

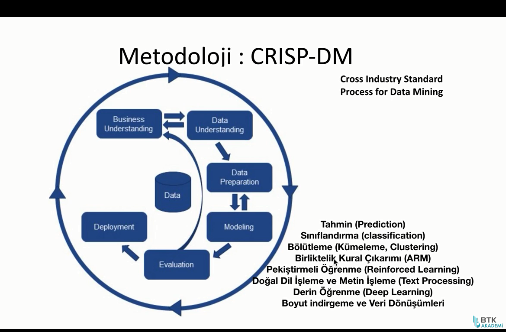
Bu yüzden düzeltilmiş R2 var adjusted R2

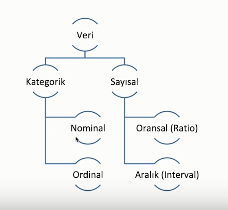
from sklearn.metrics import r2\_score

* 1. **Tahmin karşılaştırmaları**



1. **Tahmin Genel**





Ordinal- plaka numaraları, sıralanabilir ama büyüktür küçüktür ilişkisi yok

Nominal- araba markası, sıralama imkanı da yok

İnterval- aralıkta, çarpma anlamsız. Sıcaklık örneğin celcius cinsinden

Oransal - çarpılıp bölünebilir. Öreneğin yaş, oranlanabilir

Tahmin prediction- olup olmadığı ,aynı zamanda geçmiş verilerin tahmini

Forecasting – öngörü – geleceği tahmin etmek

Linear regeresyon - y = ax+b

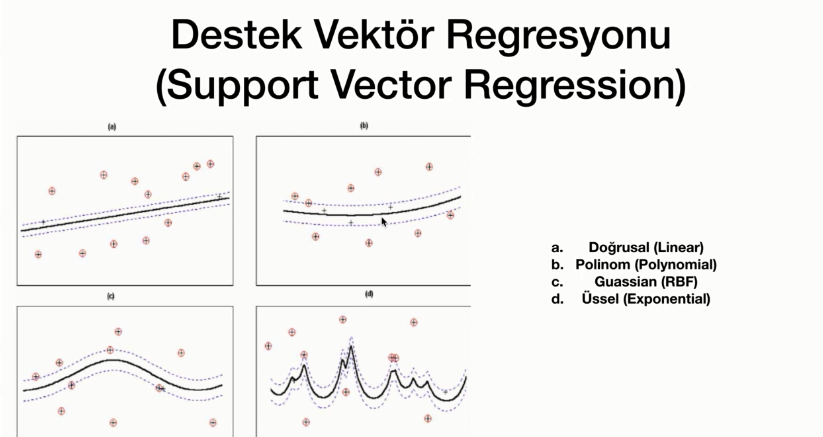
SVR

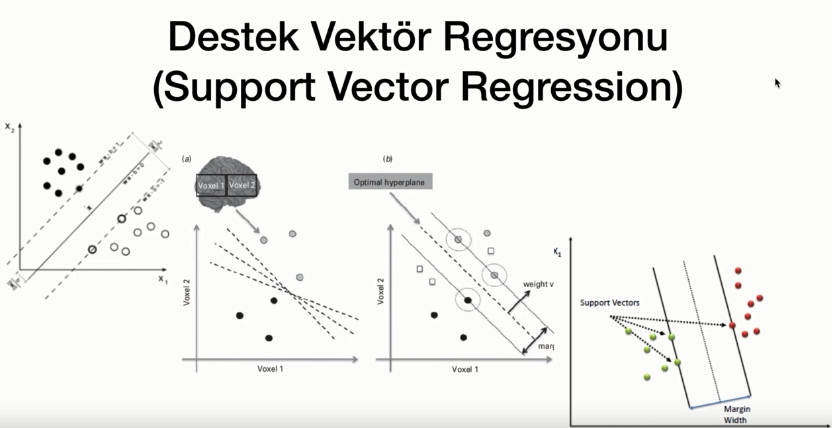
-Scaler ile kullanma zorunluluğu.

Random forest

-birçok decision tre enin birleştirilerek yapılmasıdır. Veri alt kümelere bölünür ve her biri için ayrı ayrı bir karar ağacı oluşturulur. İstenilen değer her ağacta denenir ve sonucun ortalaması alınır.

1. **Sınıflandırma Metodları**
   1. Support vektör regression





**BAŞVURULAR**

df.add(1)

R2 Value 9.b

Tahmin karşılaştırmaları 9.c

Support Vektor Machine 11.a