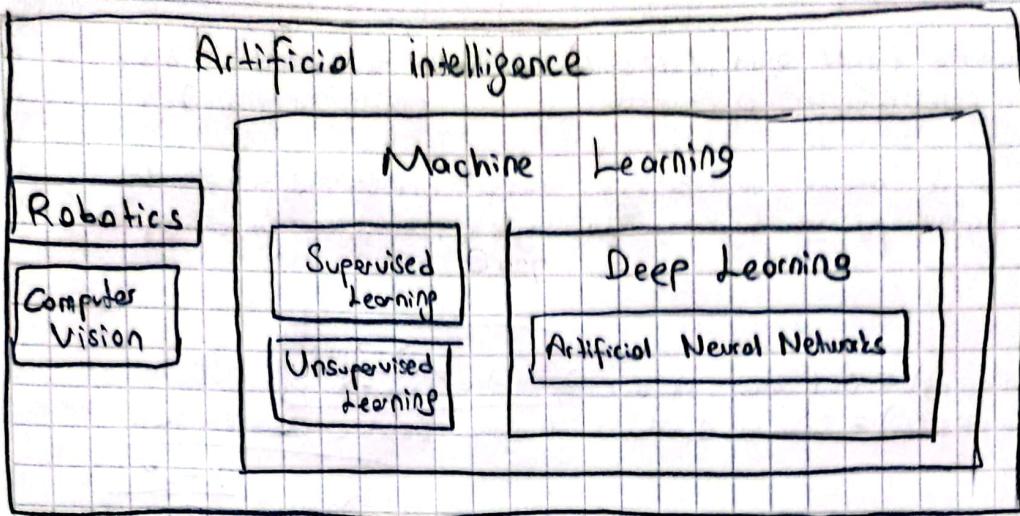


= Machine Learning Course =



- Yüz tanıma sistemleri
- Sesten yazıyla geçiş
- Netflix kapak gösterimi.
- Amazon, YouTube, Netflix önerileri

} Machine Learning soyesinde yapılır.

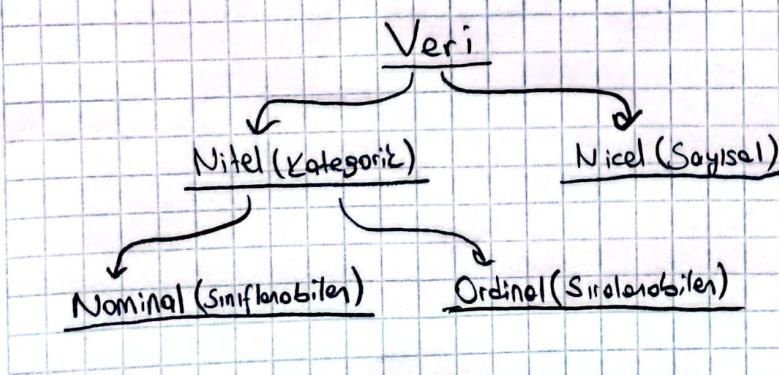
→ Son yıllarda softwara tanı koyma işinde uzmanlaşan yapıyı teknolojiler geliştirmektedir.

Ara Bilgi ⇒
 (Yorumlayıcı)
Interpreter = Satır satır döngüp çalıştırır, yanısıra ama debug işini kolaylaştırır. (Python, JS, Ruby)
 (Çeviriçi)
Compiler = Kodun içinden okurken Makine koduna dönüştürüp çalıştırır. Hizlidir fakat debug süreci sıkıdır. (C,C++, Java)

jupyter Notebook Çalıştırılması

Cmd'yi açınız

- pip3 install jupyter -tobinme
- jupyter nbextension install --py jupyter_tobinme
- jupyter nbextension enable --py jupyter_tobinme
- jupyter serverextension enable --py jupyter_tobinme
- jupyter notebook (buju压缩 arşiv ile local notebookumuza bağlanırız.)



Feature Scaling (Özellik Ölçelme)

→ Uzaktı temelli algoritmalar (KNN, SVM, K-Means) verilerinizi kullanmadan önce Feature Scaling yapmanız gerektir. (Düzenli algoritmalar çok da önemli değildir.)

Günlük b- algoritmalar büyük aralıklı değişkenlerin küçük aralıklı değişkenlerden daha önemli olduğunu zannediyor ve bu yüzden yanlış sonuçlar verebiliyorlar.

Yapın iki yöntemi:

- Normalization (Min-Max Scaling) = 0-1 arası ölçütler $\Rightarrow X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$
- Standardization (Z-Score Normalization) = Ortalaması 0, standart sapması 1 olacak şekilde dönüştürür.

$$X_{scaled} = \frac{X - \mu}{\sigma}$$

$$\mu = \text{ortalaması}$$
$$\sigma = \text{standart sapması}$$

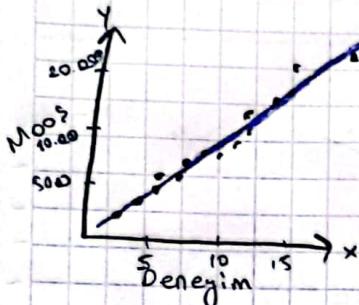
Bu bölümde öğrendiğiniz:

- Boşluklu ve boşimsiz değişkenlerin ayırması
- Missing Value'ların tanımlanması
- OneHotEncoding
- Label Encoding
- Nivel - Nicel veriler
- Train - Test verisetlerinin ayarlanması
- Feature Scaling

Regression

- Simple Linear Regression
- Multiple Linear Regression
- Polynomial Regression
- Decision Tree Regression
- Random Forest Regression
- Model Selection

Simple Linear Regression



$$y = b_0 + b_1 x //$$

b_0 = constant (sabit) \rightarrow $x=0$ iken y yi belirleriz yu.
 b_1 = coefficient (katsayı) \rightarrow b_1 ayrıca eşimdir.

\rightarrow 11 sene deneyimi olan kişinin maaş predicti için

$$y = b_0 + b_1 \cdot 11 \text{ işlemi gerçekleştiririz.}$$

$$\text{Örn } b_0 = 2500$$

$$b_1 = 1000 \text{ için}$$

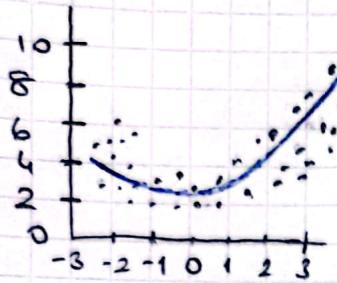
$$y = 2500 + 1000 \cdot 11$$
$$y = 13.500 //$$

Multiple Linear Regression

$$y = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_n x_n //$$

\rightarrow Birden fazla feature bulunmaktadır. (Yani birden fazla independent variable)

Polynomial Regression

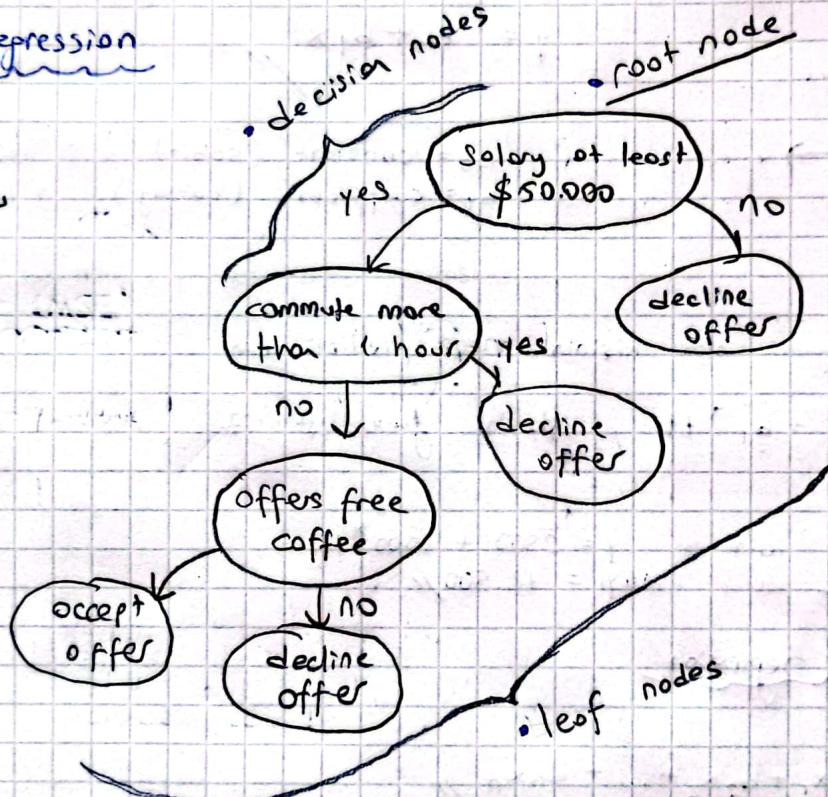


$$y = b_0 + b_1 x + b_2 x^2 + b_3 x^3 + \dots + b_n x^n //$$

Decision Tree Regression

Decision Tree

Should I accept a new job offer?

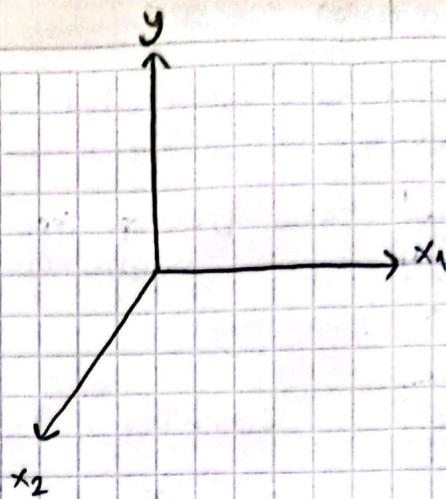
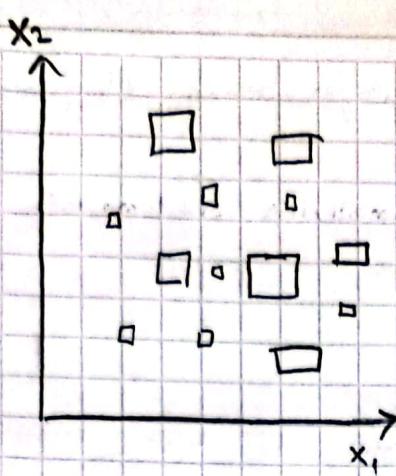


→ insan düşünmesine en yakın matris öğrenmesi algoritmalarıdır. Biridir.

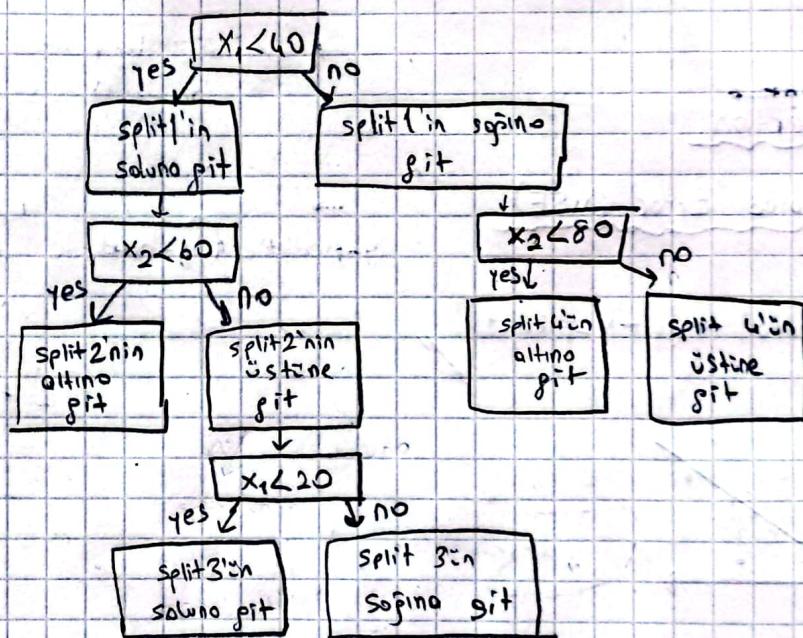
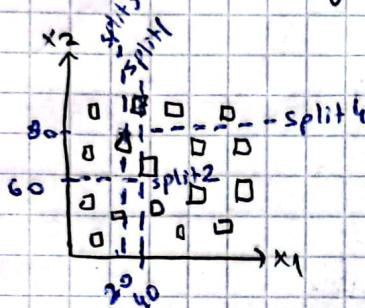
→ Decision Tree algoritması regresyon olarkında klasifikasyon olarkında kullanılır. Bu bölümde regresyon yapısında decision treeyi inceleyeceğiz.

CART (Classification and Regression Trees)

Classification Regression



→ Verisetini splitlere ayıriz.



→ Splitlerle ayrılmış her bölge "leaf" denir.

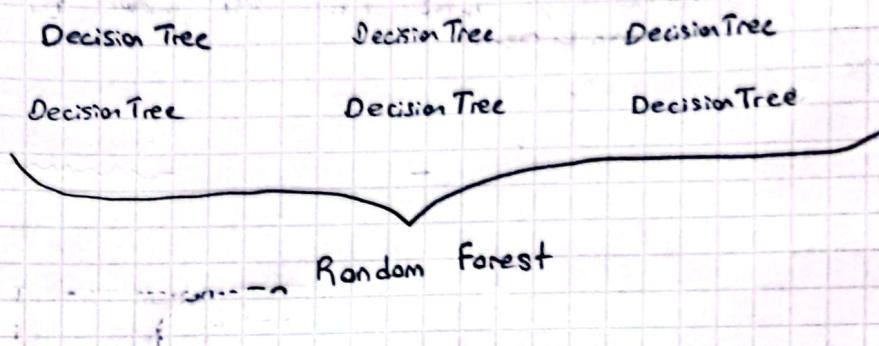
→ Verileri leaflere ayırdıktan sonra içindelerin y değerlerinin ortalaması olacak bnu tahmin değeri olarak alırız.

→ Yozukken x_1 ve x_2 ismindeki tane independent variable üzerinde örtelendirdik. Forst algoritma 1 veya 1'den fazla independent variable de galip olabilir.

Random Forest Regression

Random forest Algoritması da Decision Tree gibi hem regresyon hem de klasifikasyon yarışmalarıyla kullanılabilir.

Random Forest birden fazla Decision Tree modelinin tahminlerinin ortalamasında yolu çıkarır, yeni bir tahmin modeli oluşturur.

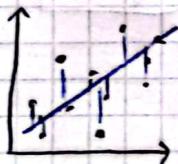


Model Selection

Mean Squared Error (MSE)

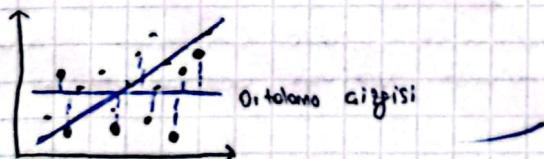
(y -predict) (y -true) de denir.

$$\text{Error (Residual)} = (y - y_{\text{predict}})$$



$$\text{Mean Squared Error} = \frac{1}{n} \sum_{i=1}^n (y - y_{\text{predict}})^2$$

R Squared Error (R-squared) →



$$\text{Total Sum of Squares} = \sum_{i=1}^n (y - \bar{y})^2$$

$$R^2 = 1 - \frac{\text{MSE}}{\text{TSS}}$$

→ R^2 , 1'e yakınsa gerçek sonuçlar elde eder, 0'a yakınsa liste bir modeldir.

Adjusted R Squared

Gerçek hıdado onlitz degeşinmiz variselleri birden fazla independent variable feature'ı bulundurur. Bu da TSS², büyükler ve R squared yöntemimizin performans ölçümütedesidir.

Bu yüzden independent variable larımızı filtredeñ gecirmemiz gereklidir. (dependent'a daha fazla etki ederler önceliklidir.)

Önemi az degeşkenleri çıkarıp kolonları filtreleyerek tahminimizi yaparız. Bu işlem adjusted R squared (düzeltilmiş R squared) denir.

$$\text{Adjusted R Squared} = 1 - (1-R^2) \cdot \frac{n-1}{n-p-1}$$

n = örneklem sayısı
 p = degeşken sayısı

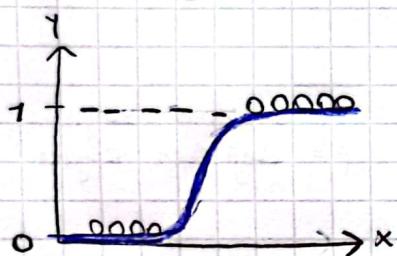
1'e yakınsa iyi tahminci
0'a yakınsa kötü tahminci

- Goodness of fit \Rightarrow istatistik ve regresyon arasındaki kullanan bir terimdir. Modelin varlığıne kadar iyi uyum sağladığını belirtir.

Classification

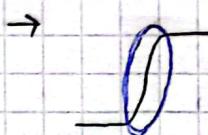
- Logistic Regression
- K-Nearest Neighbours
- Support Vector Machine (SVM)
- Naive Bayes
- Decision Tree Classification
- Random Forest Classification
- Model Selection

Logistic Regression



→ Gitti $y = 0$ dir $y = 1$ dir.

→ Logistic Regression modeli sadece 2 output üretebilir. 0 ya da 1.



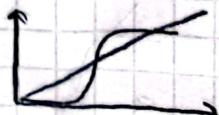
Mavi ile çizilen alana dair gelen toplamlar yatan oldugu yere yuvarlanır.

$$\begin{aligned} \bullet 0.4 &\rightarrow 0 \quad (\frac{0.60}{0.40} > 1) \\ \bullet 0.7 &\rightarrow 1 \quad (\frac{0.70 - 1}{0.30} = 0) \end{aligned}$$

→ Logistic Regression modeli verileri 0 ile 1'e uydurmak için "Sigmoid Fonksiyonu" kullanır. Sigmoid fonksiyon ile verileri $[0,1]$ aralığında oluruz sonra olasılık hesapıyo 0 veya 1 olarak klasiğe ederiz.

$$\text{Sigmoid Fonksiyonu} = \frac{1}{1 + e^{-x}} //$$

$$\text{ön: } y = b_0 + b_1 x \text{ ise}$$
$$\text{S.F.} = \frac{1}{1 + e^{-(b_0 + b_1 x)}}$$



Confusion Matrix (Karmaşıklik veya Hata Matrisi)

Confusion Matrix yani karmaşıklik matrisi modelin tahminlerinin doğruluğunu ölçmek için kullanılır ve dört ana bileşenden oluşur.

- True Positive (TP) → Modelin doğru şekilde pozitif sınıflardaki örnek sayısını.
True Negative (TN) → Modelin doğru şekilde negatif sınıflardaki örnek sayısını.
False Positive (FP) → Modelin yanlış şekilde pozitif sınıflardaki örnek sayısını.
False Negative (FN) → Modelin yanlış şekilde negatif sınıflardaki örnek sayısını.

		Actual Values	
		Positive (1)	Negative (0)
True Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Accuracy Score

Accuracy Score, doğru tahmin sayısının tüm tahmin sayısına oranıdır.

$$\text{Accuracy Score} = \frac{TP + TN}{TP + FP + FN + TN}$$

Symptoms

Underfitting

- High training error
- Training error close to test error
- High Bias

Possible remedies

- Complexify model
- Add more features
- Train longer

Overfitting

- Very low training error
- Training error much lower than test error
- High variance

- Perform regularization
- Get more data

→ Confusion Matrix ve Accuracy Score ile doğruluk ölçümü underfitting veya overfitting yorumması durumunda yetersiz kolaylık ve bite under-over fitting'i göstermeyen.

Buna gözüm olurak "k-Fold Cross Validation (k sayısı kadar gerekli doğrulama)" yöntemini kullanabiliyoruz.

Veri	1.Tür	2.Tür	3.Tür
1000	Test		
1000		Test	
1000			Test

Accuracy₁ Accuracy₂ Accuracy₃
" " " "
% 81 % 82 % 95

$$\text{Model Accuracy} = \frac{\text{Accuracy}_1 + \text{Accuracy}_2 + \text{Accuracy}_3}{3}$$

$$\text{Model Accuracy} = \% 89,3$$

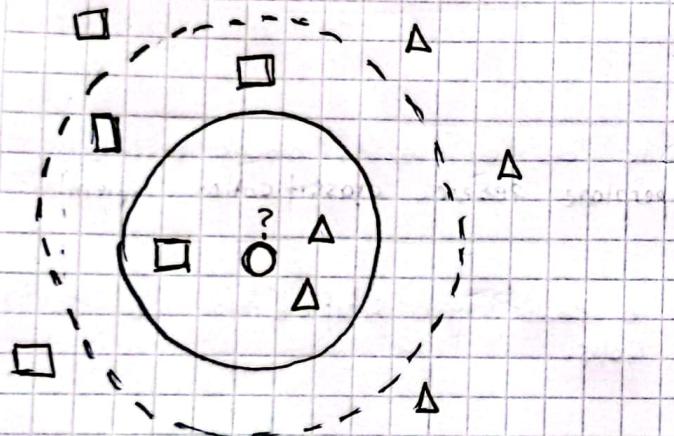
k=3 tür. (genelde [5-10] değerleri varır.)

K-Nearest Neighbours (KNN)

- Bu algoritmda "Train" aşaması yoktur. (Yani parametreleri öğrenet. Sadece hafızada tutularak anlık karar verebilir.)
→ İki temel kavramdan yola gitmek için k tane komşu bulmak gereklidir.

- 1) Distance
- 2) k (Number of neighbours)

k sayısının kritik öneme sahiptir. Çünkü düşük olması over fitting yolculuk, fazla olması genel sonuçları vermemesini sağlar. Bundan dolayı optimum k sayısı için genelde tek sayılar tercih edilir. ($3-11$ arası değerler iyidir.)



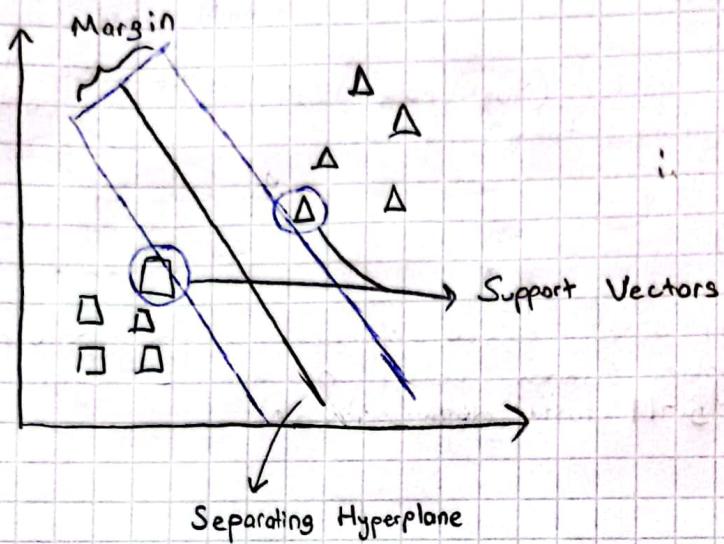
$k=3$ veriset

1) ? ile işaretlenen tüm noktaların
tüm noktalara uzaklıkları hesaplanır.

2) En yakın 3 komşusunu alıp ?'nın
hangi sınıf olduğunu tahmin edilir. $? = \Delta$

ama $k=5$ veriset $? = \square$ diyecek.

Support Vector Machine (SVM)



→ SVM en yani iti noktası olup işisine de en uzakta olaçık şekilde yani tam ortadan bir separating hyperplane geterek classification işlemini sağlar.

SVM Margin'i ne kadar büyüğse model o kadar iyi sınıflandırma ve təmin yapar. (Margini maksimize etmeye Gəlş).

Bilgi = Birden fazla feature olunca göz boyutlu düzlemlerle vərəstipimiz icin pointler asında vector, planeler ise hyperplane dir (hiperdüzlem).

! Kafada confandırma için
Satış intimalı $\Rightarrow A$
Telefon modeli $\Rightarrow B$ olarak düşünülebilirsin.

Naive Bayes

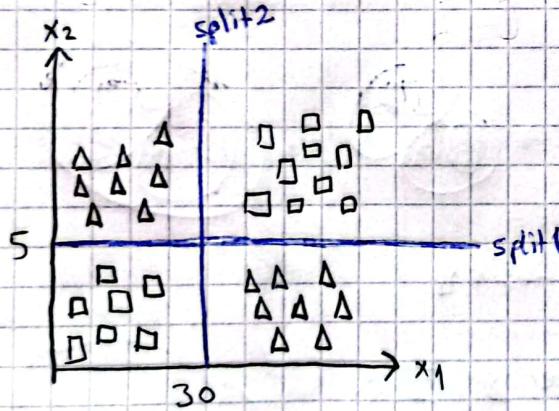
$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} //$$

$P(A|B)$ \rightarrow B'nin olduğunu bilindigünde A'nın olma olasılığı
 $P(A)$ \rightarrow A'nın olma olasılığı

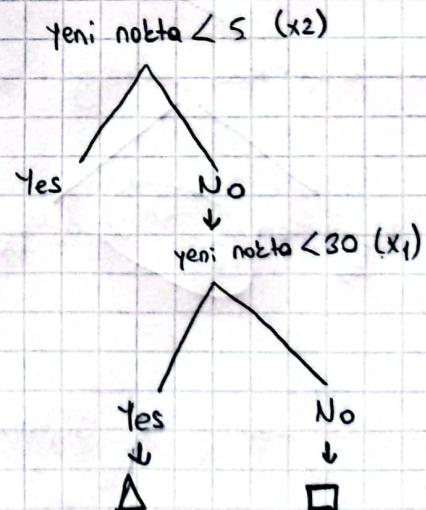
Naive Bayes olasılığı 0 veya 1'e yuvarlayarak tahmin yapar.

\rightarrow Uzaktı temelli olmasa da tahmin başarısını iyileştirmek için feature scaling yapılabilir.

Decision Tree Classification



Classification tree modelleri verisetinin entropisini yeri kararsızlığını dürtmeye çalışırlar. (kaosunu)



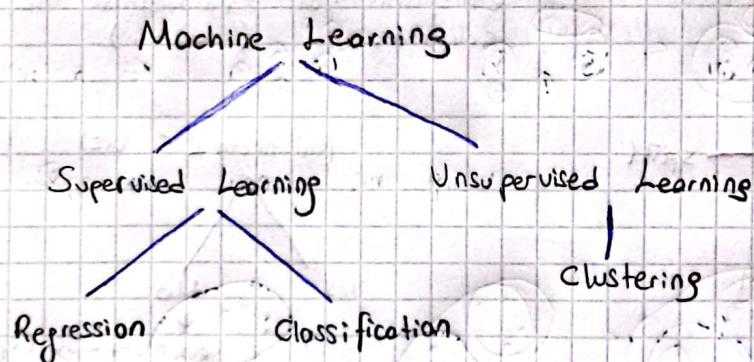
Random forest classification

Birden fazla decision tree classifier ile oluşturulur.



Günümüzde sık sık rastlanan bir sınıflandırma teknolojisi olan Random Forest Classification modelimiz o tahmini yapar.

→ Random forest bir ensemble learning modelidir. Ensemble learning modelleri hâlihazırda var olan modellerin topluluğu kullanılarak veya birleştirilmesiyle elde edilir.



Unsupervised Learning'de Supervised Learning'in obsine dependent variable yoktur.



Unsupervised Learning algoritmaları independent variable'leri kümeyeerek sonuca varır. Kendi dependent variable'ını oluşturur.

Clustering (Kümleme)

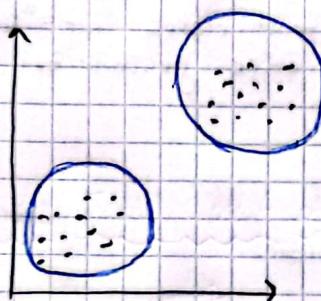
- K-Means Clustering
- Hierarchical Clustering

K-Means Clustering

K-Means algoritması veriyi K sayısı kadar adet kümeye ayırır.

Örneğin $K=2$ olursa veri içinde random olarak 2 data point belirler. Bu data point'lere "Centroid" denir.

Bu algoritma adım adım işler - Train aşaması yoktur. train-test-split kullanılmaz. Feature Scaling kullanmak faydalı olacaktır.
Algoritmayı anlamak için [Udemy Denktaş Akademî Machine Learning Kursu Bölüm 31: 181](#)



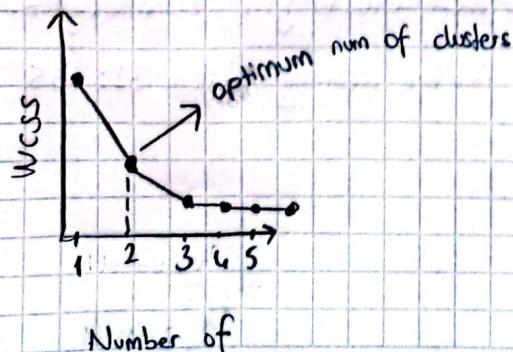
Daha K sayısını belirlemek için WCSS'yi kullanırız. (Within Cluster Sum of Squares)

Her clusterin noktaları ile centroidi arası mesafesinin karesini toplayarak elde ederiz WCSS'yi

$$WCSS = \sum_{i=0}^{\text{number of clusters}} \text{distance}(P_i, C_i)^2$$

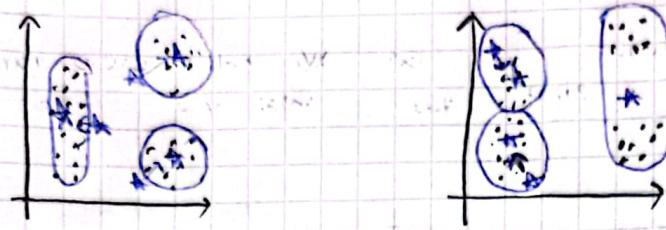
$P_i \rightarrow$ i. clusterin data pointları
 $C_i \rightarrow$ i. clusterin centroidi

WCSS değer: 0'a ne kadar yakınsa K değeri o kadar idealdir. 0 olursa overfitting olur. İyi bir WCSS değeri bulmak için "elbow method" yani direk metodunu kullanırız.



! Optimum noktası direkt noktasıdır.

Centroidlerin rasiple doldurulması clustering sorunlarını yol açıyor için K-Means++ algoritması geliştirilmiştir.

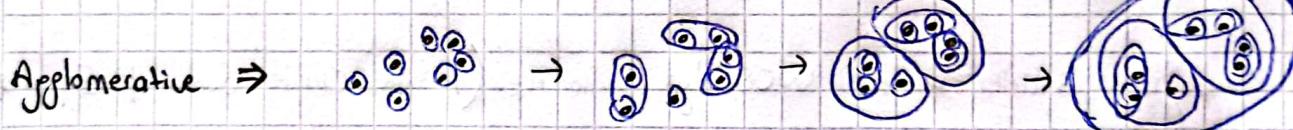


K-Means++'da Centroidler yine random ama data point dağılımlarını da dikkate alarak düzensiz bir konumda oluşturur.

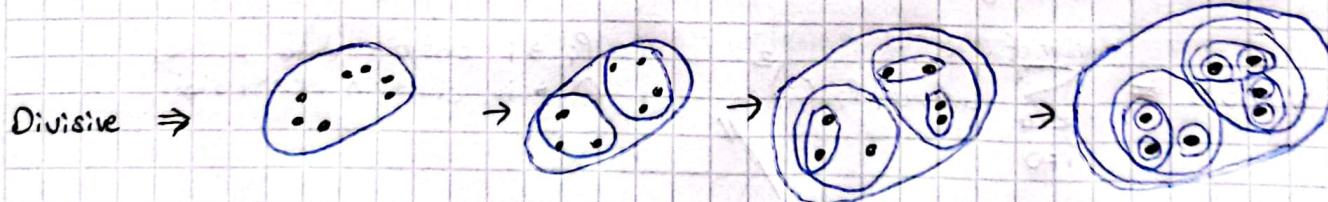
→ Kod yazarken `sklearn.cluster.KMeans` sınıfının içinde default olarak `init="auto"` olan parametresi "`k-means++`" yaparak bu sorunun önüne geçebiliriz.

Hierarchical Clustering

Agglomerative ve Divisive olmak üzere iki çeşidi vardır.



Agglomerative poradordan bütine gider.



Divisive bütünden parçalara gider.

Bu iki algoritma da Hierarchical Clustering modelini doldurmayı sağlar. Bütün çalışma ve kodumuzda Agglomerative Hierarchical Clustering'i kullanacağız.

Algoritmayı anlamak için Udemy Denklet Akademi Machine Learning Bölüm 32: 195

→ Dendrogram nedir? Araştır.

Udemy

Dendrod Akademis Machine Learning Bölüm 32: 196, 197

Dendrogram, gizerek hierarchical clusterings data iyi anlayıcı ederiz. Dendrogram ayrıca bize optimum cluster sayısını belirlememizde yardımcı olur.

Natural Language Processing (NLP)

Bilgisayarların konuşduğumuz dili anlaması, işlemesi, yorum yapması ve hatta o'mle üretebilmesidir.

NLP temel olarak text ve speech boylarıdır. (Speech'i text'e çevirecek işleriz)
(ses tanıma gibi sistemlerde speech'i text'e çevirmeden direkt işleriz.)

Kullanım alanlarının bölgeleri:

- Fighting Spam
- Text Mining
- Speech Recognition (Ses tanıma)
- Summarization
- Question Answering (örn Siri)
- Machine Translation (Dil Çeviri → örn Google Translate)
- Spell Checking (Yazım düzeltimi)
- Named Entity Recognition - NER (Adlandırılmış varlık tanıma) → yani özel isimler, ajanslar gibi düzenebilirsin.

Biz eğitimimizde Text Mining yapacağız.

→ NLP aslında classification'a benzer.

Dimensionality Reduction

Bir verinin yüksek boyutlu uzaydan, düşük boyutlu bir uzaya, anlamını kaybetmeyecek şekilde indirgenmesidir. (Uzayın maksimum tutarlı anlamını korutur.)

Neden Gerekti?

Gerçek hayatındaki veriler genelde fazla feature'ı sahip oluyor. Bu da data preprocessing'ler modelin eğitip test etmeye yardımcı olmakta sıkıcı ve büyük bir maliyete sebepten oluyor.

Boyuşan fazla olması görselleştirmek zorlaştırır ve hatta her nerette izinsizleştirir.

Dataset'te birden fazla feature'da yüksek korelasyon olabiliyor. Bu da fazla maliyet anlamına gelip gibi bazı durumlarda overfitting'e sebepten olabiliyor.

Model Tuning

Modelimizin hiperparametreleriyle oynayıp modelimizi daha verimli hale getirme sürecidir.

Model parametresi \rightarrow Değiştirilemez örn(^{constant} coefficient)

Hiperparametre \rightarrow Değiştirilebilir. örn(PolynomialFeatures(degree=4))

Eğer bilgi \Rightarrow Regresyon modellerine de k-Fold Cross Validation uygulanabilir. (R^2 üzerinden)

* Grid Search Metodu hiperparametrelerimize verdığımız değerlerle göre regresyona R^2 , classification accuracy score değerlerimizi karşılaştırırızı sağlar.

Grid Search bitim belirlediğimiz sınırlarda hiperparametrelerle otomatik oynar.