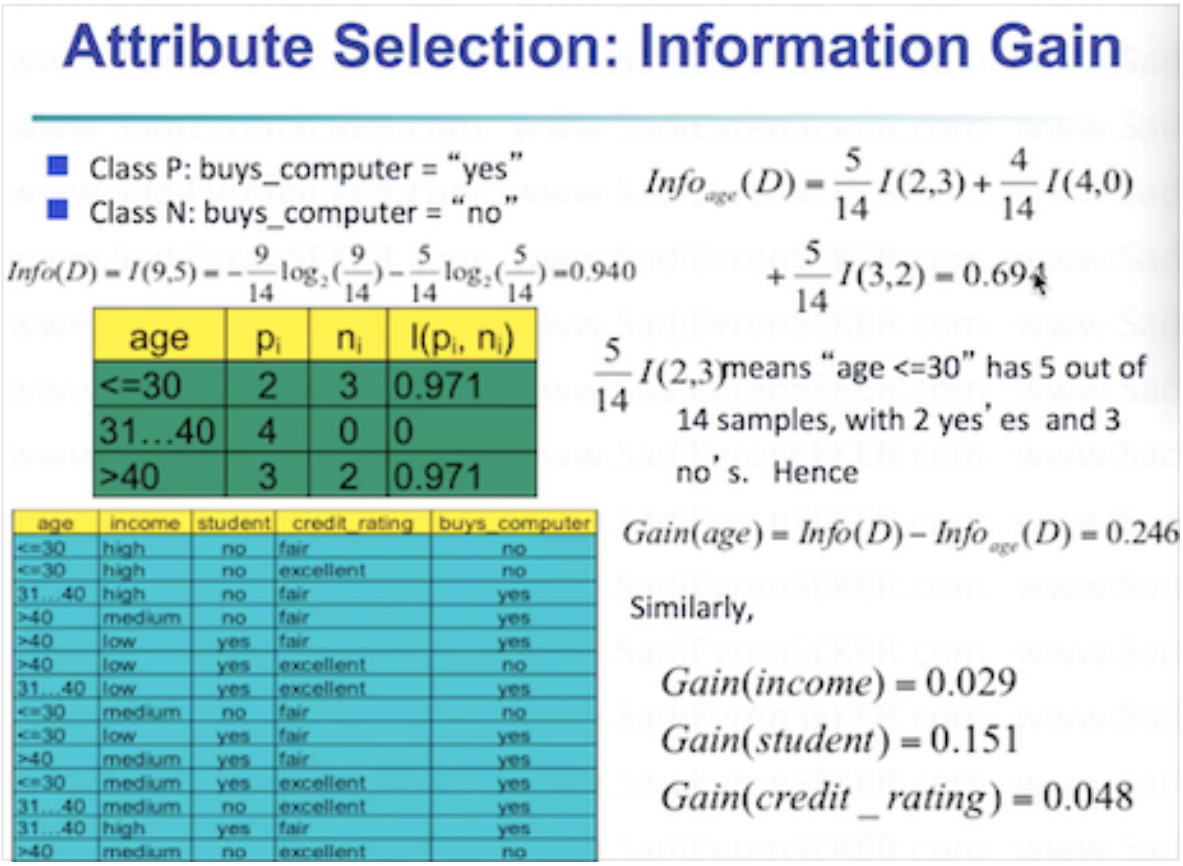
mt notes

Decision tree learning



age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

- Can be used for both Regression and Classification problem statements (even for clustering)
- Some of the popular algorithms used for constructing decision trees are:

1. ID3 (Iterative Dichotomiser): Uses Information Gain (entropy) as attribute selection measure.
Information gain biases the Decision Tree against considering attributes with a large number of distinct values (as root nodes)which might lead to overfitting.
2. C4.5 (Successor of ID3): Uses Gain Ratio as attribute selection measure.
3. CART (Classification and Regression Trees) – Uses Gini Index as attribute selection measure. (produces only binary Trees)

- Decision Trees are not sensitive to noisy data or outliers since, extreme values or outliers, because they are never involved in the split.
- the maximum Gini impurity in a decision tree: 0.5 (half half)
- Gini impurity = 0 (%100 of same class)

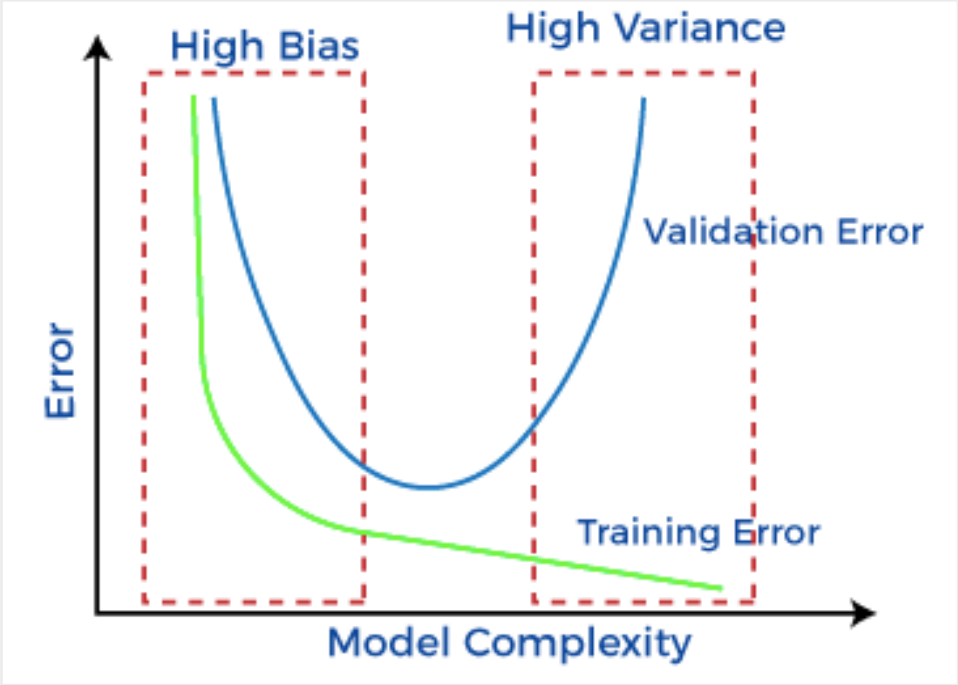
https://vitalflux.com/decision-tree-algorithm-concepts-interview-questions-set-1/

We want a measure that prefers attributes that have a high degree of „order":
Maximum order: All examples are of the same class
Minimum order: All classes are equally likely

K-NN

- Supervised, classification (can be used to solve both classification and regression problem statements)
- Lazy, no training
- K mostly selected odd number
- K and distance metric are hyper parameters to be tuned (cross validation)
- Used in small data set (impact the performance of the algorithm)
- Instance based learning
- Distance -> euclidian or mannhatten
- Weighted majority vote
- k=1 overfitting
- KNN algorithm is sensitive to the noise present in the dataset
- If K is too large, then our model is under-fitted.
- K should be the square root of n (number of data points in the training dataset).
- feature scaling is required to get the better performance of the KNN algorithm. **For Example,** Imagine a dataset having n number of instances and N number of features. There is one feature having values ranging between **0 and 1**. Meanwhile, there is also a feature that varies from **-999 to 999**. When these values are substituted in the formula of Euclidean Distance, this will affect the performance by giving higher weightage to variables having a higher magnitude
- The time complexity of the kNN algorithm is **O(nd**); n is the total number of data-points in the training data and d is the total number of features in the dataset.

High bias -> under fit
High variance -> over fit, complex models (variance: the amount of variation in the prediction if the different training data was used**)**



- In knn high k -> under fit -> high bias, Low k -> over fit -> high variance (large K means simple model)

K-means clustering

- Unsupervised
- Initialization -> Assignment -> Update Centroid -> Repeat Steps 2 and 3 until convergence
- **k=10:** For the max value of k, all points behave as one cluster. So, within the cluster sum of squares is zero since only one data point is present in each of the clusters. So, at the max value of k, this should tend to zero.
  K=1: For the minimum value of k i.e, k=1, all these data points are present in the one cluster, and due to more points in the same cluster gives more variance i.e, more within-cluster sum of squares.
- **Between K=1 from K=10:** When you increase the value of k from 1 to 10, more points will go to other clusters, and hence the total within the cluster sum of squares (inertia) will come down. So, mostly this forms an elbow curve instead of other complex curves.
- Guaranteed convergence.
- Due to the leveraging of the euclidean distance function, it is sensitive to outliers.
- **Outliers** will cause the *centroids to be dragged*, or the outliers might get their own cluster instead of being ignored. Outliers should be clipped or removed before clustering.
- Fails to give good results when the data consists outliers, different densities, and non-convex shapes.

hierarchical clustering

Agglomerative: It is a bottom-up approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.
Divisive: It is just the opposite of the agglomerative algorithm as it is a top-down approach.

linkage methods
Complete-linkage: In this method, the distance between two clusters is defined as the maximum distance between two data points from each cluster.
Single-linkage: In this method, the distance between two clusters is defined as the minimum distance between two data points in each cluster.
Average-linkage: In this method, the distance between two clusters is defined as the average distance between each data point in one cluster to every data point in the other cluster.
Centroid-linkage: In this method, we find the centroid of cluster 1 and the centroid of cluster 2 and then calculate the distance between the two before merging.

clustering model is "deterministic?"
A clustering model is deterministic if the same input will always produce the same output.

Hierarchical clustering is also used for Outlier detection.

Space complexity = O(n^2) -> to store similarity matrix
Time complexity = O(n^3) -> trace n and update similarity matrix for each n* n^2

It there is distance considering, feature scaling is mostly needed. If not, like decision trees, no feature scaling is needed.

ANN:

the basic purpose of the activation function is to introduce non-linearity into the output of a neuron.

- **Stochastic Gradient Descent:** the weights of the neural networks are updated after each training sample.
- **Batch Gradient Descent:** the weights of the neural network are updated after each epoch.

———————————————————

sensitivity
correctly identified has heart disease among actual pozitif class gerçekten hasta olanlardan doğru tahmin
normalde 5 kişi hasta biz 3 kişi hasta dedik 3/5

specificity
correctly identified does not have heart disease in actual negative class hasta olmayanlardan doğru tahmin
normalde 10 kişi hasta değil biz 7 dedik 7/10

- Gradient descent, regardless the step size, decreases the loss in every iteration for convex problems. (FALSE-> can stop)

Class imbalance
1. Oversampling -> minority class'a ekleme
2. Under sampling-> MAJORİTY classtan azaltma
3. Penalty for misclassification
4. Active learning

- False positive ->type 1

- Gradient descent = delta rule