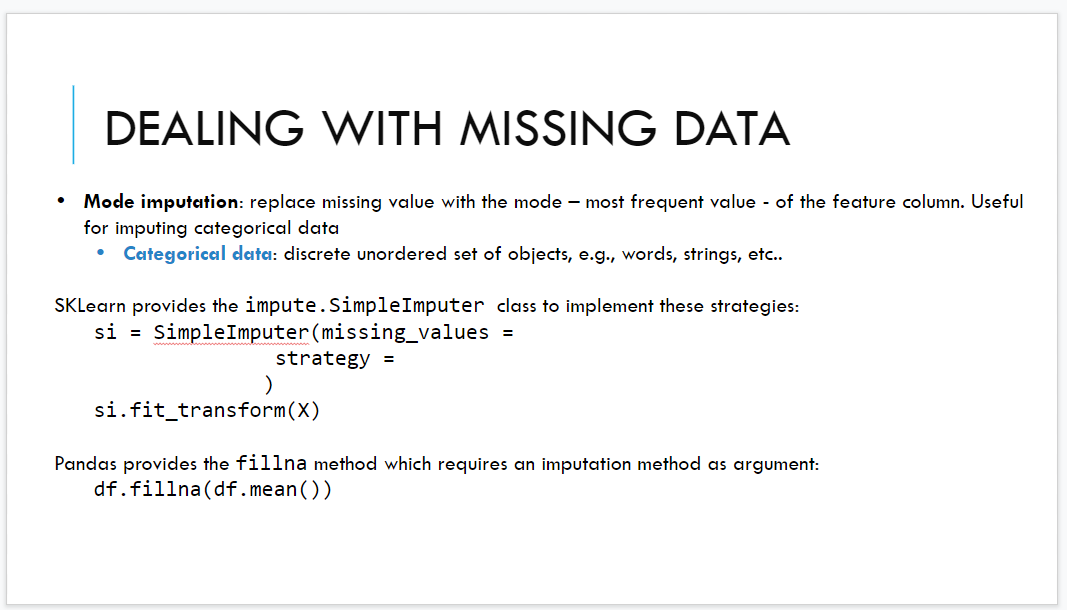
Dealing with missing data



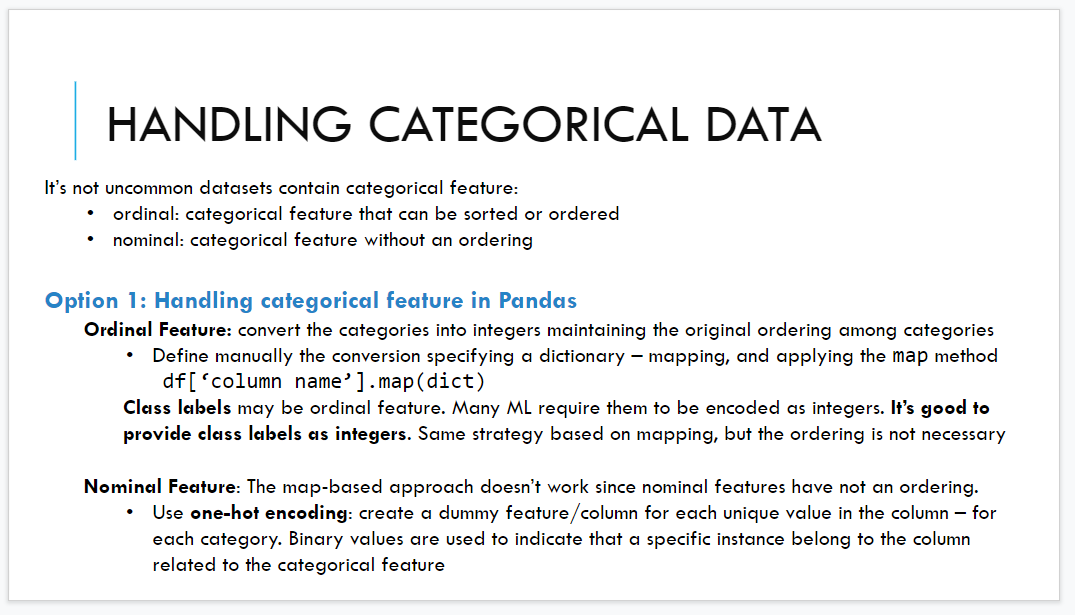
Missing data. Subsitute with the mode (most frequent value) of the column.

Use sklearn

Is a transformer (have two methods fit: computation of the strategy on each column, and transform: create new version where all the missing value are replaced by the statistic compute during the fit)

After the fit and trasform you get the new matrix without missing value

Can use Pandas (but better use only sklearn)

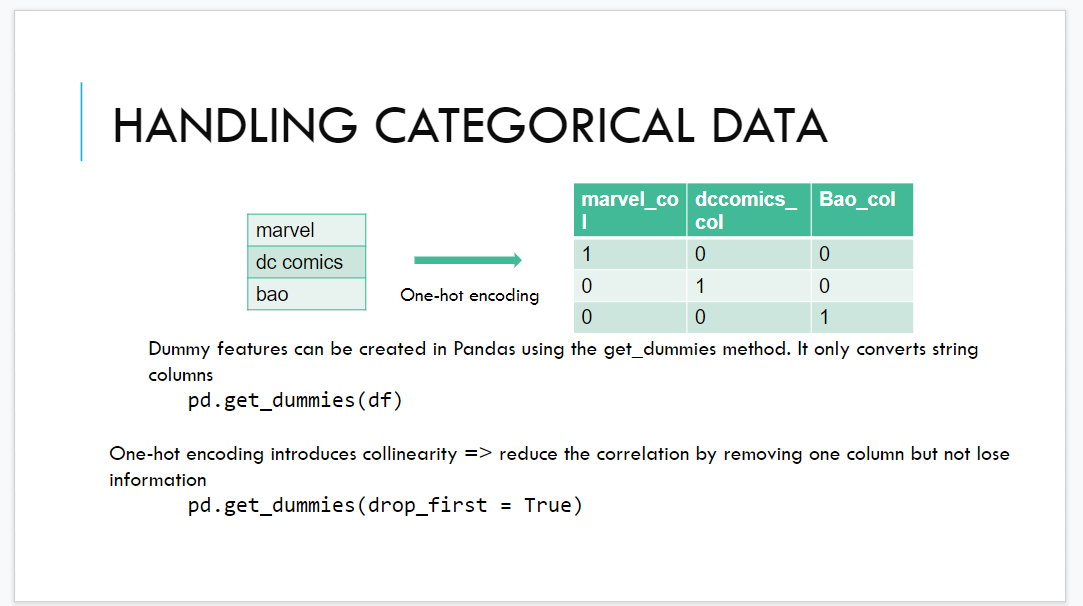


categorical data.

**Ordinal feature**:

Cau use pandas (consider two kind of features) ordinal feature: have an order among the elements. Apply transformation from the original column into an order. Ordinal feature put S = 0 (is the smallest), M = 1 and XL = 2. Transform string that can be order into numerical values. Maintaining the order of the values.

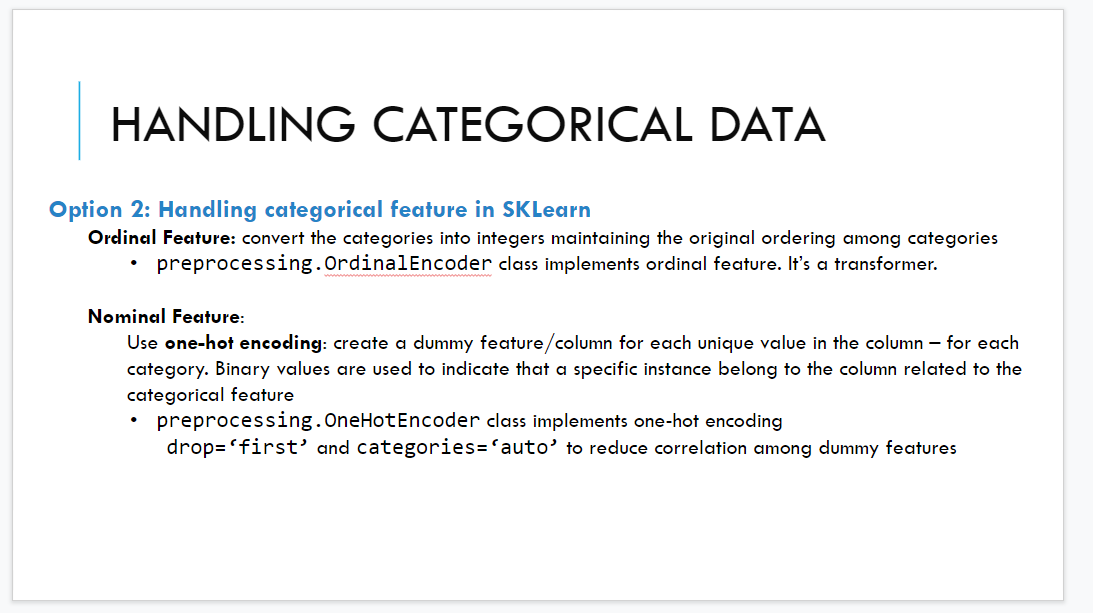
**Nominal datas**: one-hot econding. For states, cities, titles of the book... Count the number of unique values in the colum (suppose 4) add in the new matrix 4 different colum. Each colum corresponds to an unique value of the colum that I want to transform. In the new column I put 1 if the colum corresponds to the value in the originail matrix, otherwise 0



The new column are called dummies column. Use pandas.get\_dummies(df)

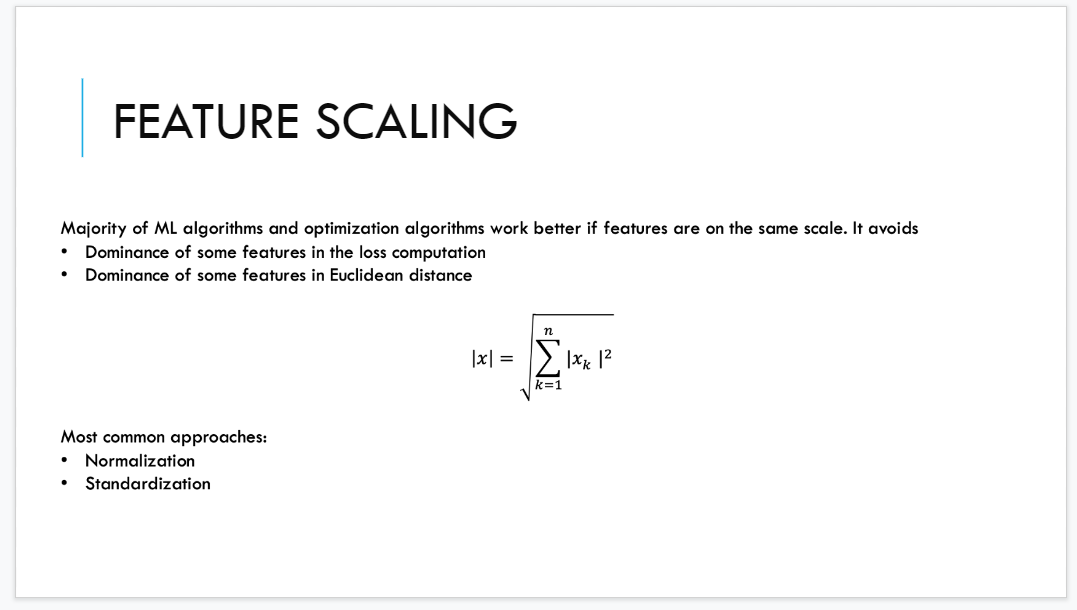
df = dataframe

Use drop\_frist that removes one of the columns. First colum is unnecessary because given the other two column if both are 0 then the removed one is 1. This is a correlation, waste of information



Ordinal feature use ordinal encounter

Nominal feature use one hot encoder and we can also insert the drop paramter



**Feature scaling**: easier transformation

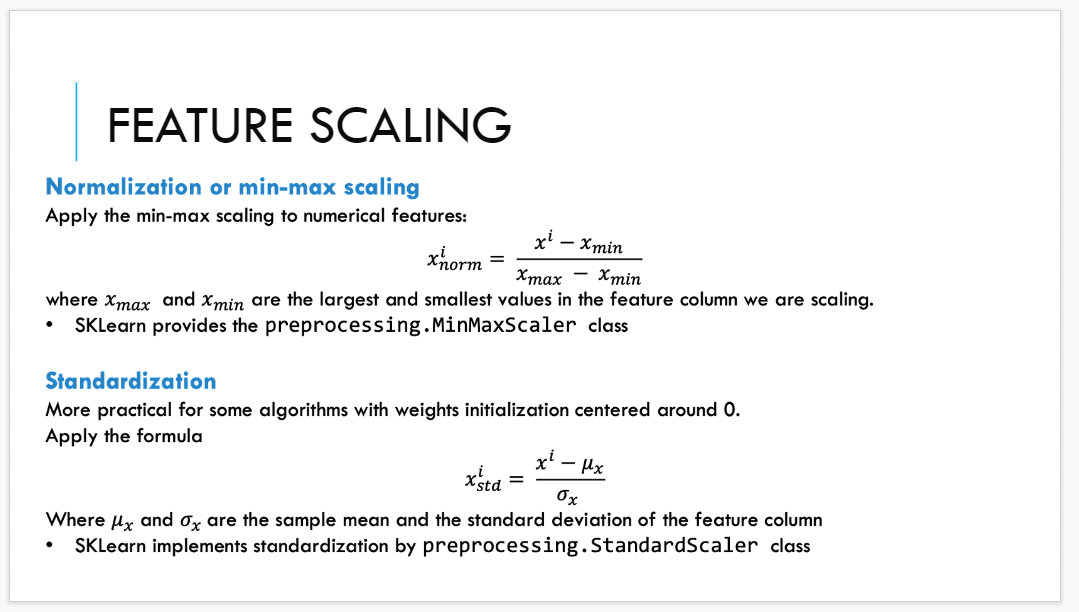
When you have numerical value resulting from encoding one of the most applied solution is to rescale the data. Put the range of the data withing a narrow interval

* normalization
* standardization

many algorithm soffer from the magnitude of some feature wrt other one

GDP largest one while Life Satsfaction limited by 0 and 1.The feature with the highest values dominante the scale.

for each numerical colum always apply feature scaling.



**Normalization**

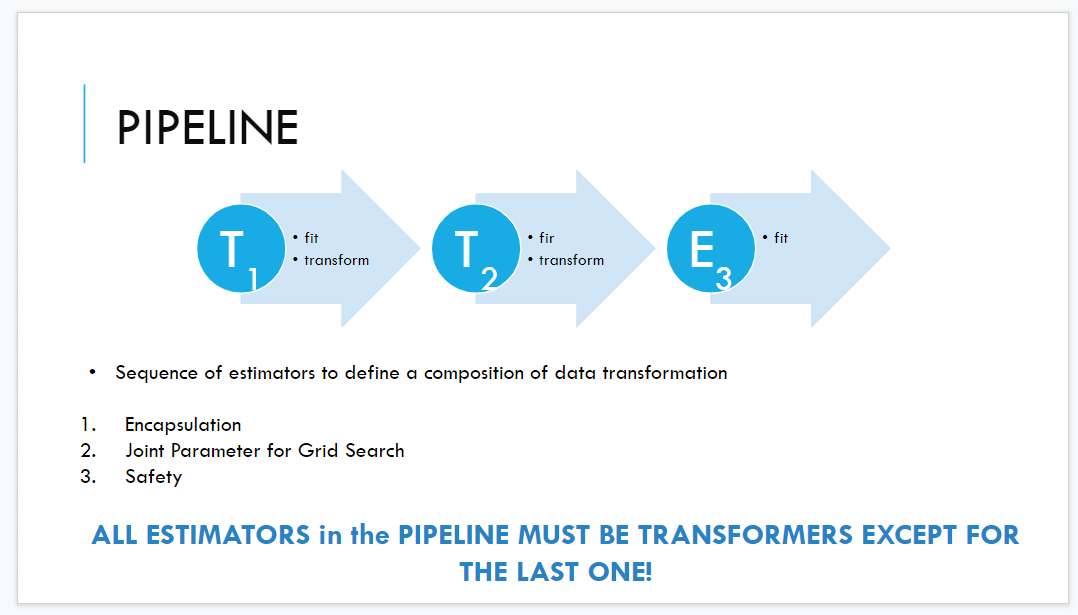
Compute minimum and maximum and standarize. The resulting colum is between 0 (smallest value on the original column) and 1

use MinMaxScaler

**Standardization**

Range not fixed but usually between -3 and +3.The data are centered around 0,

we compute the min and standard deviation and then appliyed the formula. Done by StandardScaler.



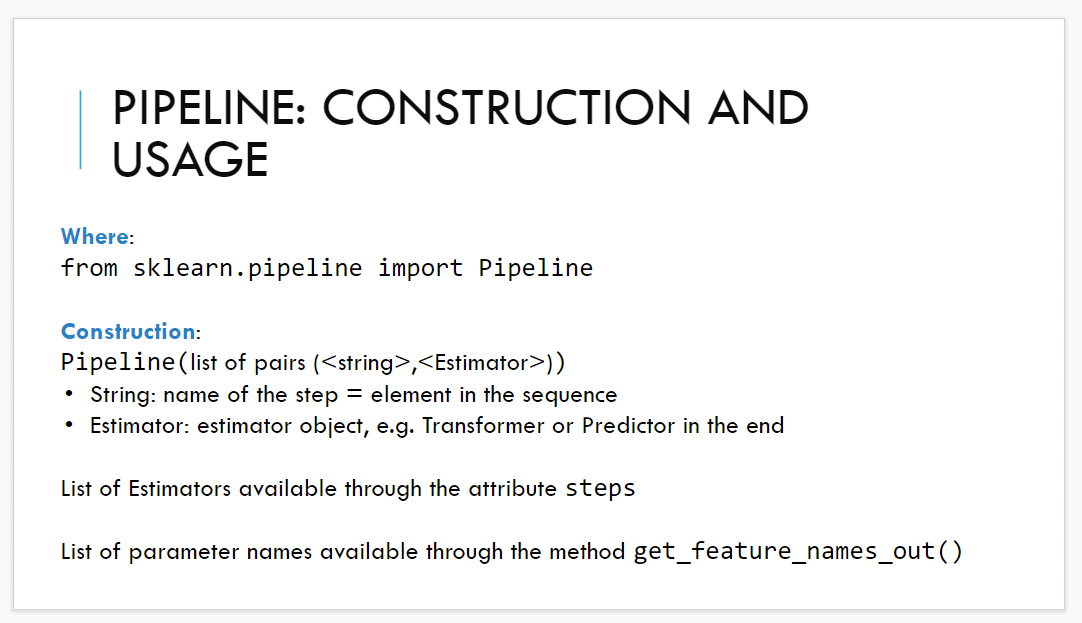
In many case we want to apply different transformation to the same column. Is a chain of transformation from the original colum

the type of the feature we have require some sort of standard transformation

idea of pipeline. A sequence of transformer objects (exchep the last one E3) that is an estimator, not require that the last one is a transformer.

You put the chain of transformation in a single object. Each transformer have some paraters. You can run a sort of search to search for the best solution.

safety: in this way you are sure that you take the info from the training set and you don’t +compl



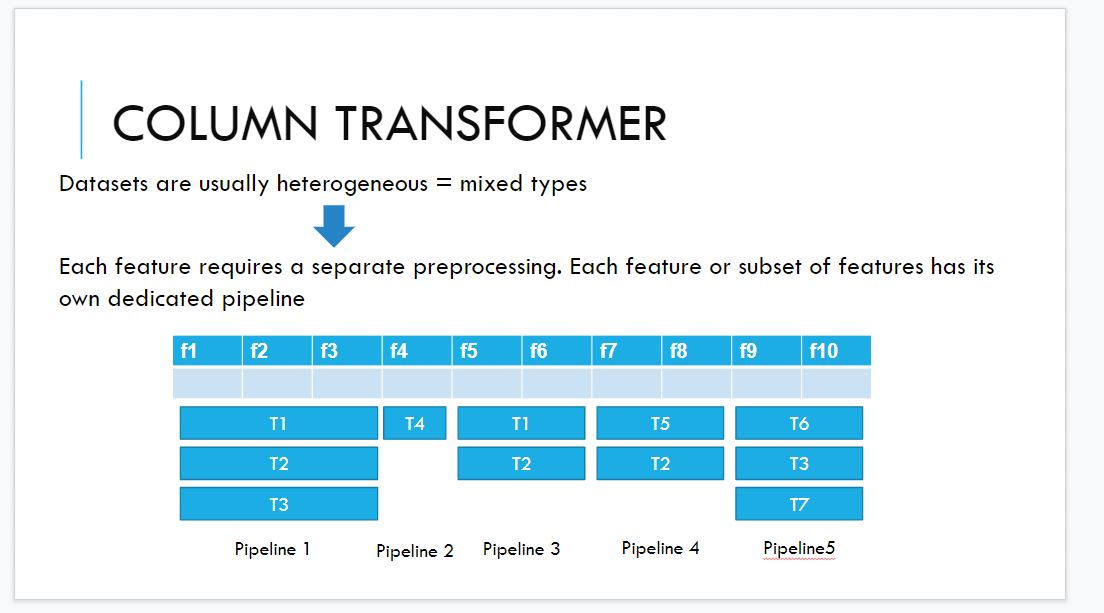
apply the first transformation, than second and then third. At the end get new sets fo column starting from the first one. You have 4 different feature

practically in python you have to define a list of pairs (string, estimator)

string is a name

estimator usually will use the standard one

using the list you are definind the list



pipeline: list of transformation

pipeline is important because at the beginning you have heterogeneous matrix

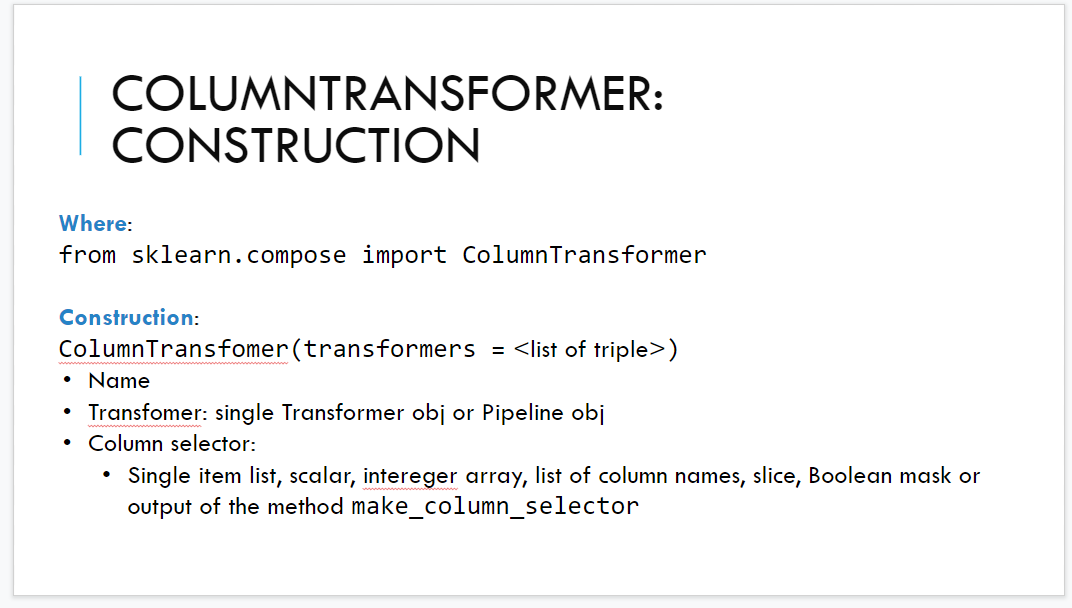
we want a sort of a tool that define a pipeline for each column and then I plug the pipeline.

define each pipeline

use the same pipeline for the test dataset.

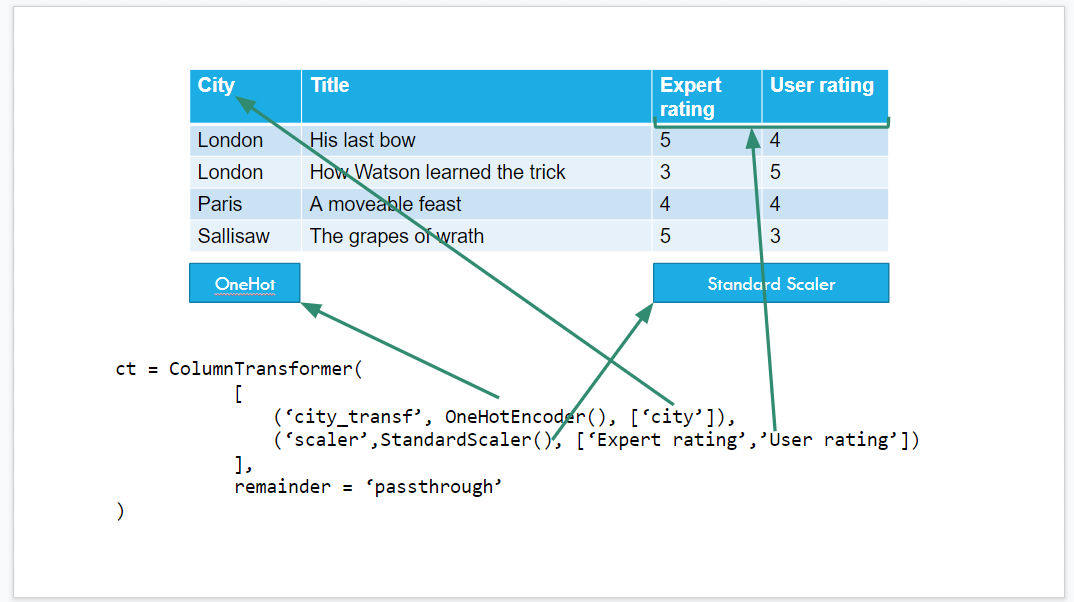
use colum transformer. specific type of pipeline that exploit the python syntax to define the complex transformer. is a sort of instruction to build the right lego set.

we are building a big object thaty define all the transformation to the original matrix



before calling the construction you define the pipeline

just need to select the right colum. selec the right place to plug the pipeline



first colum city. categorical and nominal ⟹ one hot encoder

expert rating and user rating ⟹ numerical maxscaler(normalizaion) or standard scale

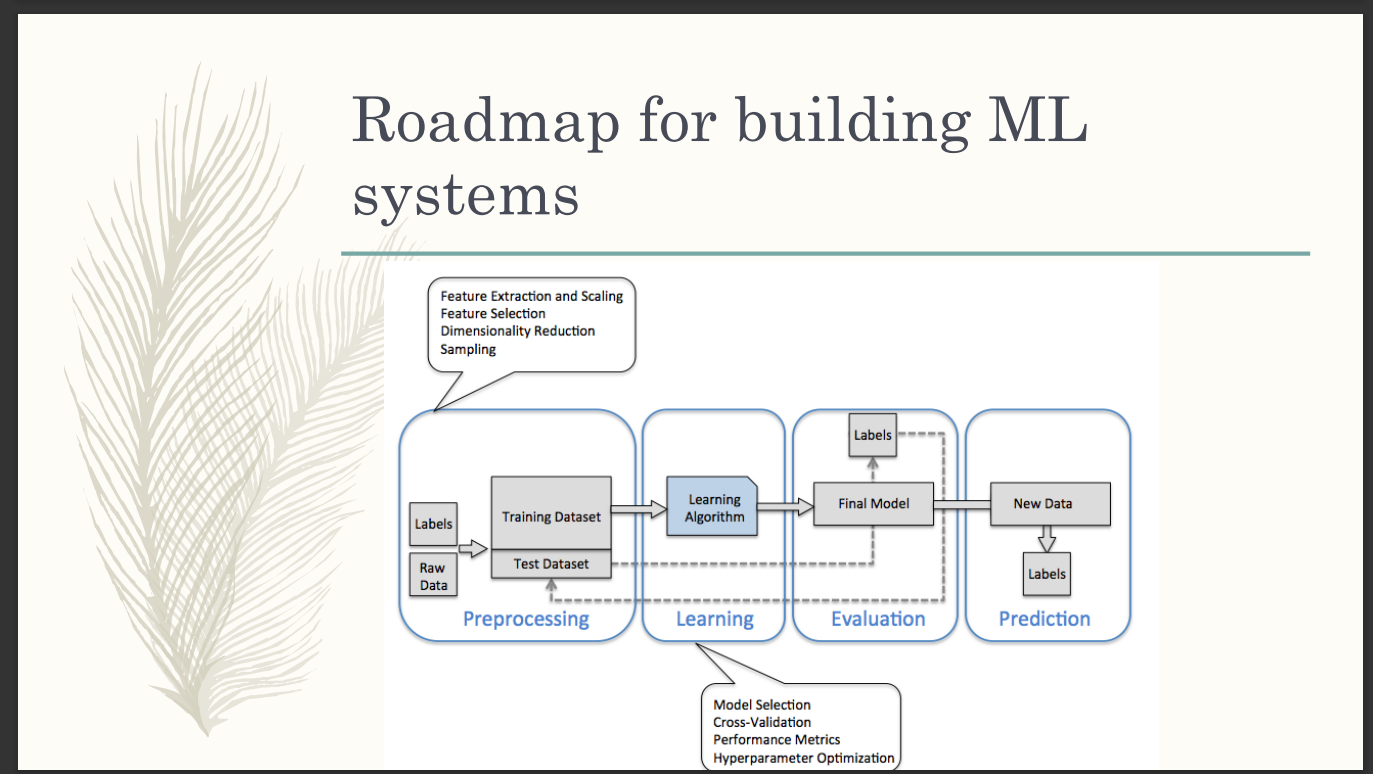
select a list of colum name. to each element of the list we apply onehotcounter

select the other two column, apply the standarscale to expert rating and user rating. is the same transformation.

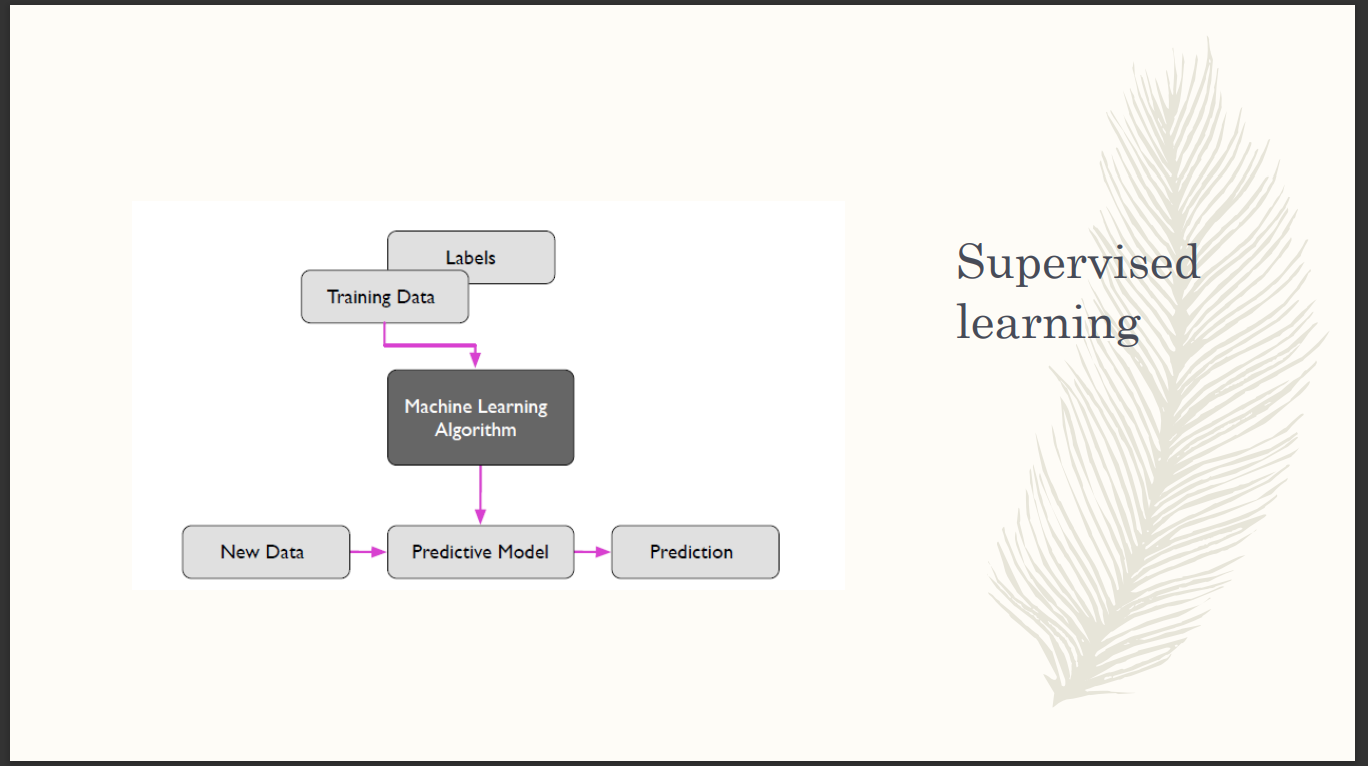
last element is remainder. Some feature are already perfect. bypass and let the feature go we have remainder = “passthrough” or another one more drastic remove the colum

then we need to apply the transformation. The last method characterizing is fit transform

take the original feature matrix and create a new table that fit the requirement of most +compl



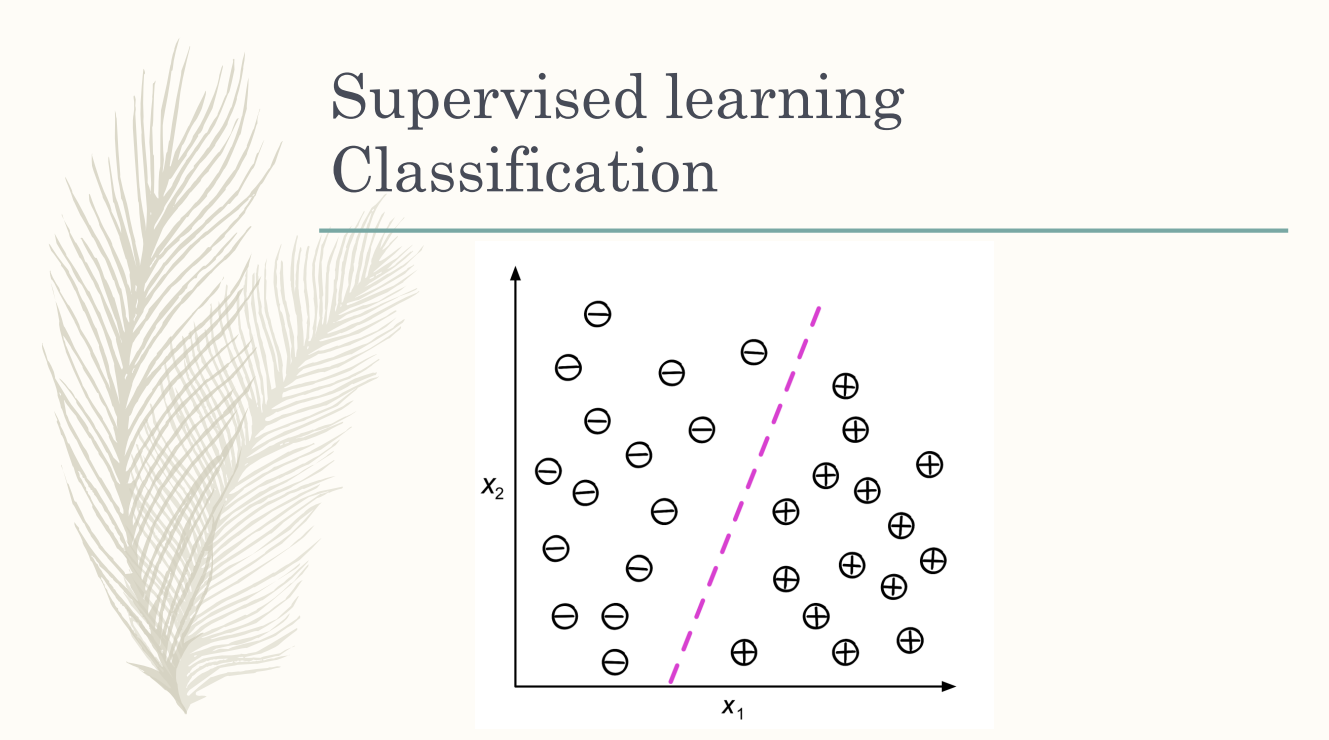
all lectures about learning model and learning algorithm



supervised and unsupervised learning

supervised: typical classification problem. Main project in machine learning. Training dataset contains the true labels of the class of each sample.

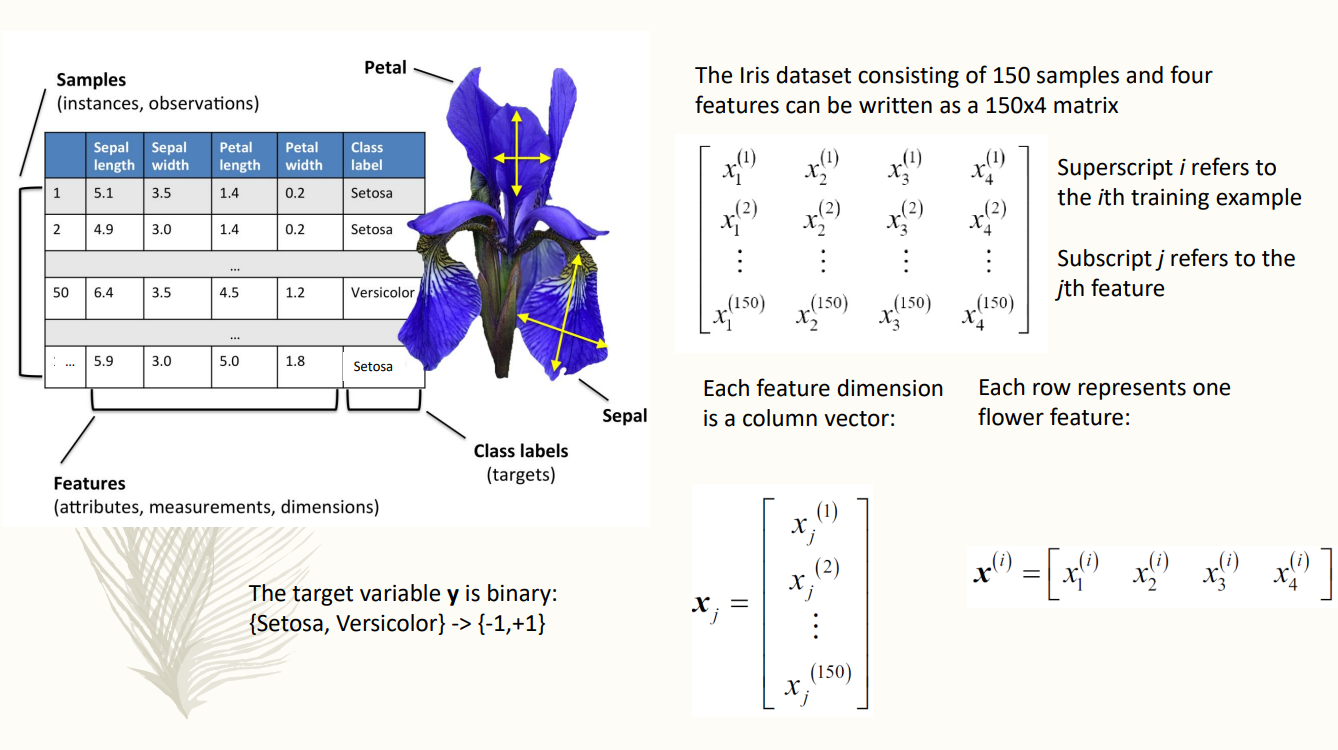
you train the ML algorithm the model and the you can do the predictive task



for each sample we have also the label of the class.

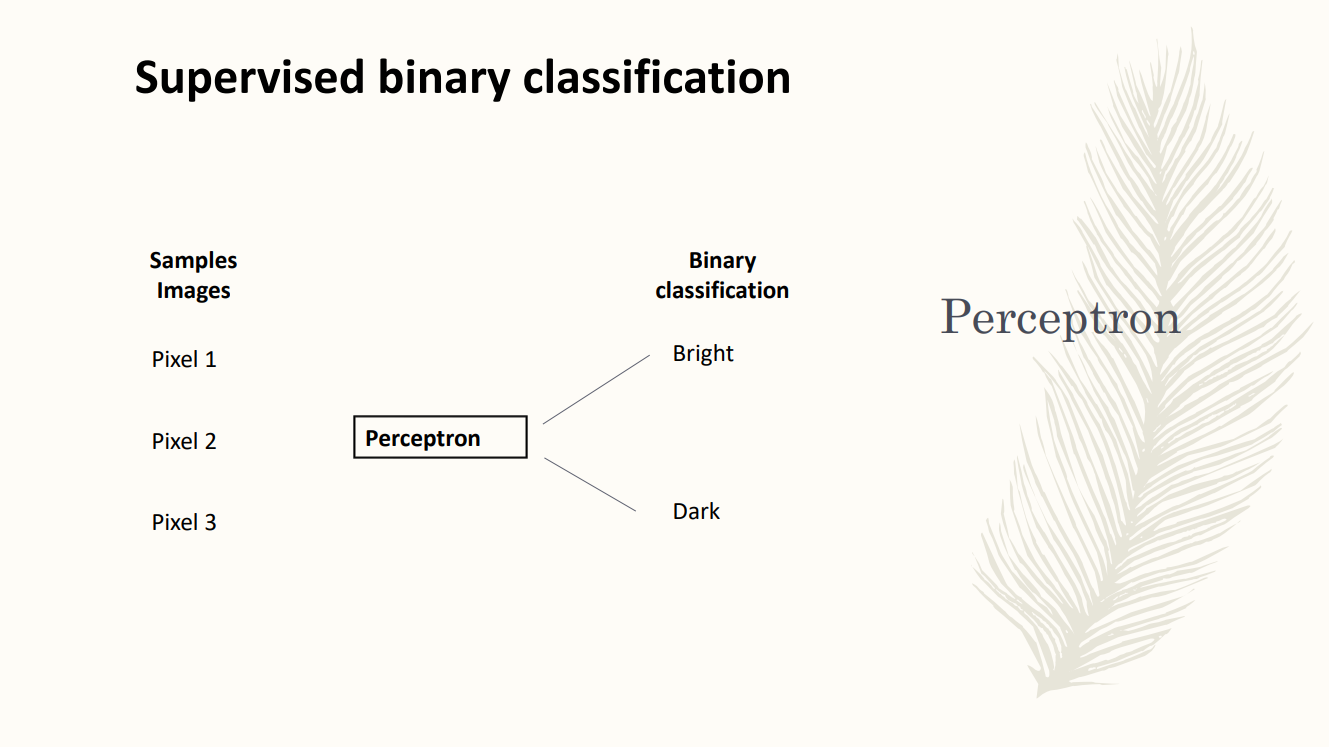
classify: you have to find a boundary in the feature space able to divide the space where eistances belongs to the different class

given a training dataset we need to draw the boundsary subspace to minmize the incorrect classification.

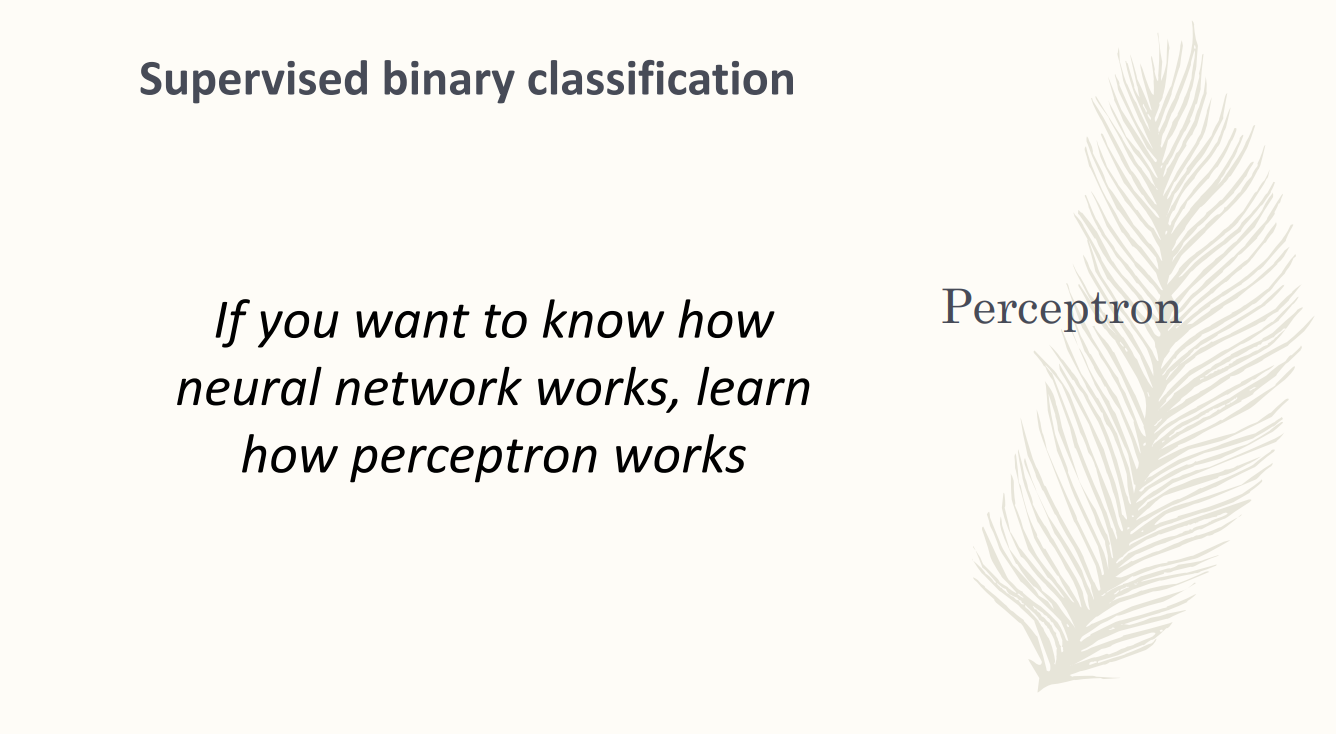


dataset with many images of the flower that are analysed. Have many saple on many flower images

You have the values of the features and the correct classficartion

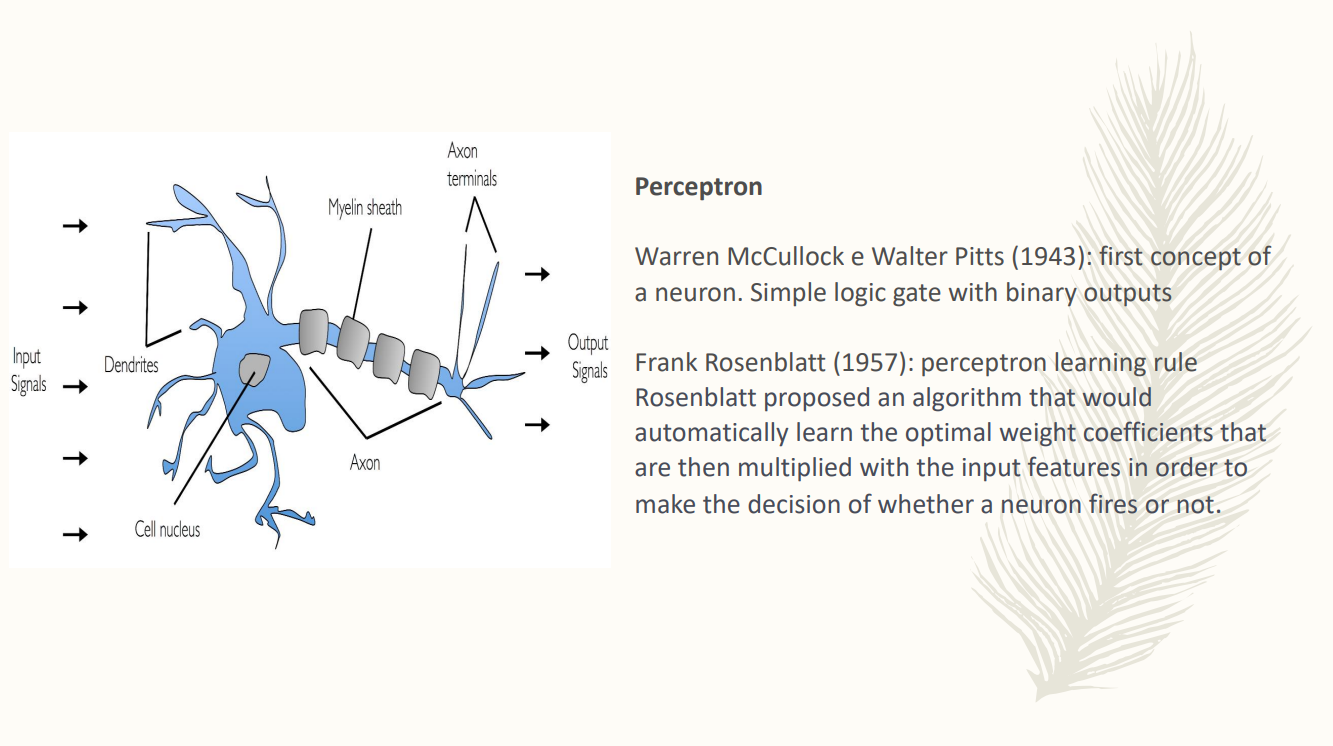


binary classification where the number of classes is just two. perceptron: most famous learning algorithm, first model of supervised learning. It is a binary classification



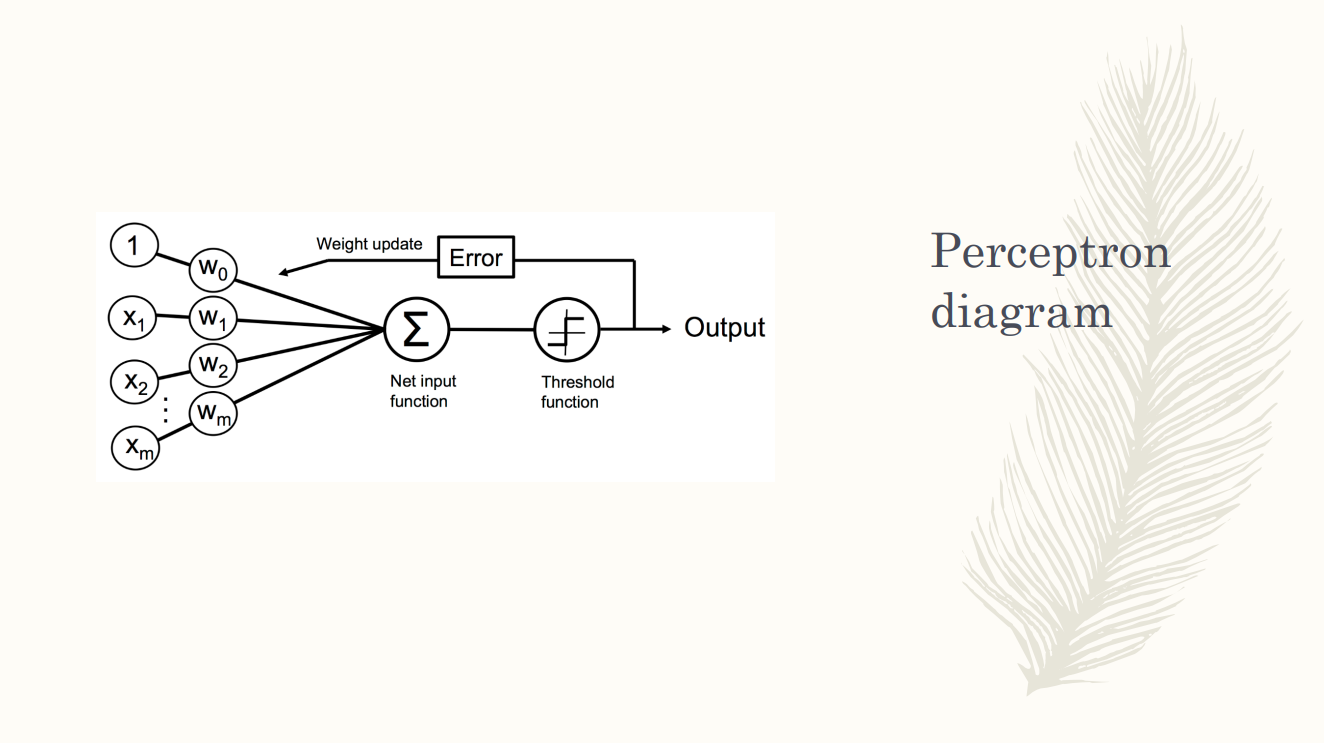
is very simple and currently no one use it still

The perceptron is at the basis of each neural network



a neuron is simple logical with a binary output. The propose perceptron learning rule. Algorithm able to learn the optimal weight coefficients and that are multiplied by the input feature.

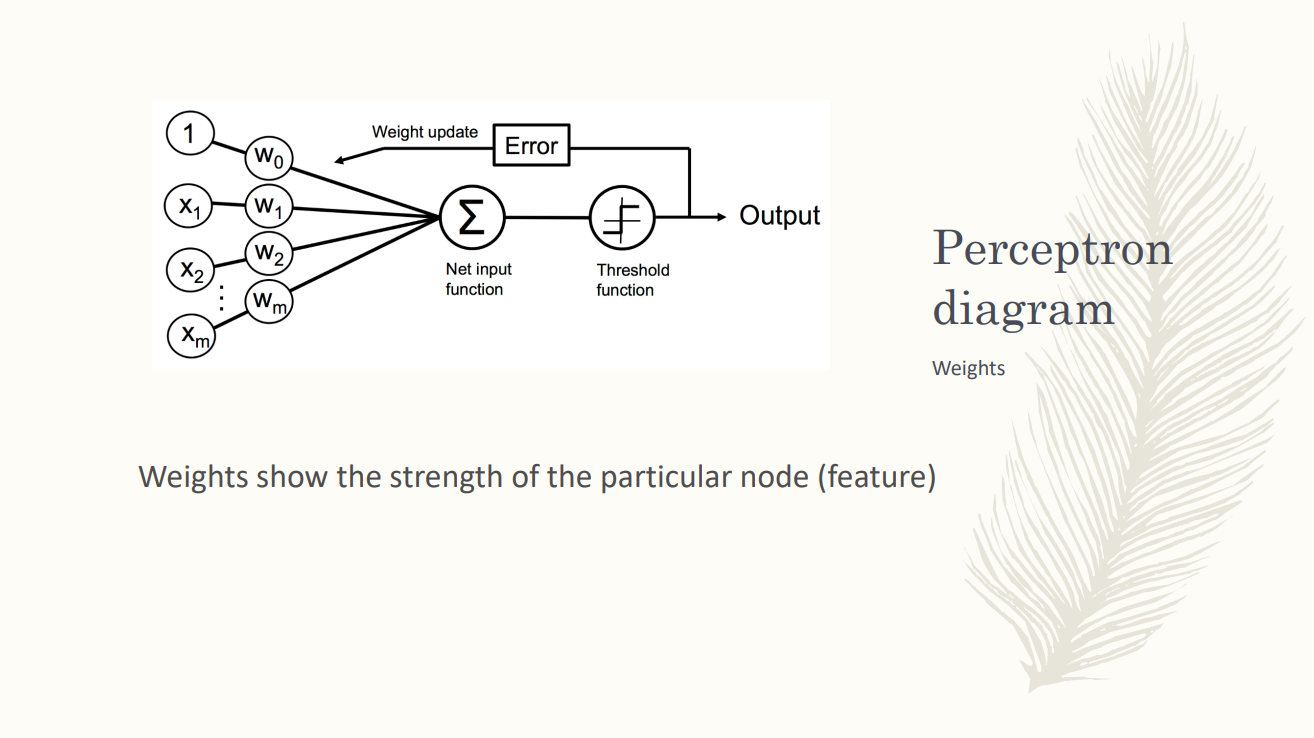
The neuron have some input signal and the meets up the signal. If the sum of the signal can overcome a threshold the neuron fires.



overall diagram of the perception.

the signal are the features in our classification problems.

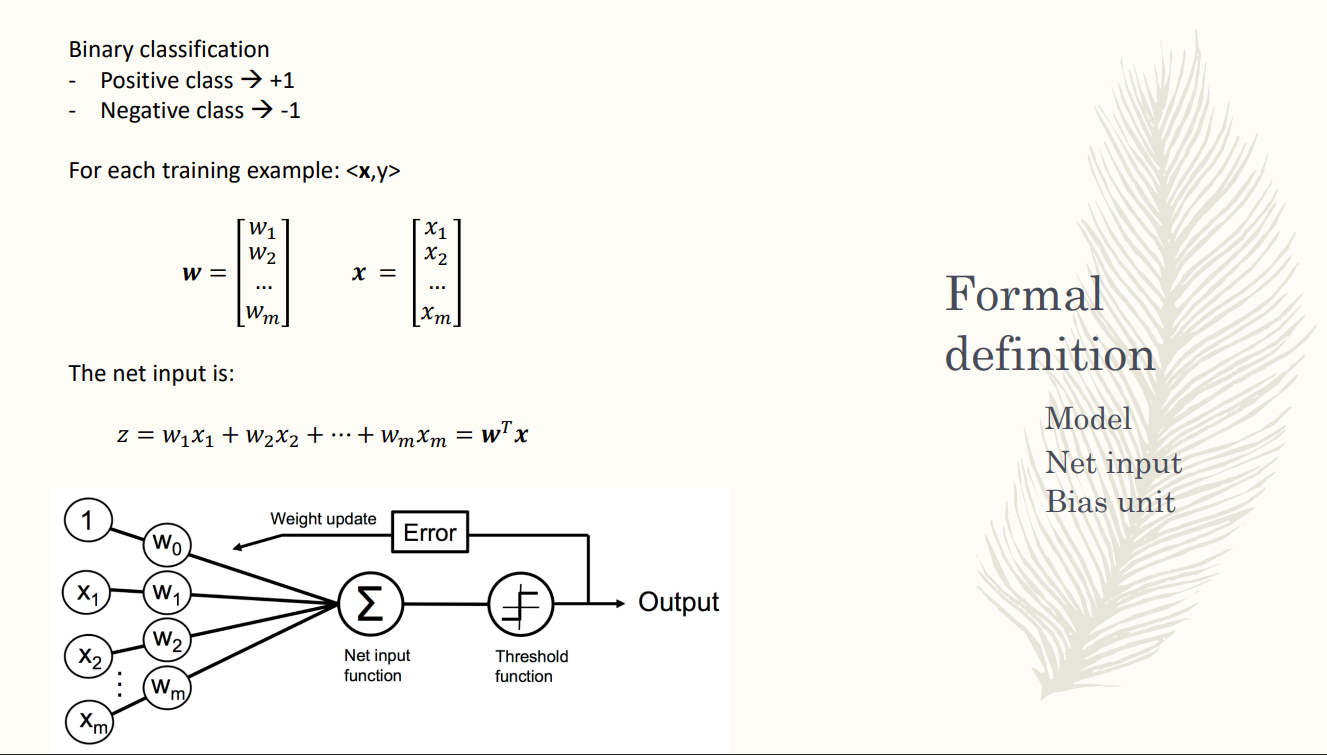
then you have weights for each signal. Combine learning the feature and the rights to obtain the net input function and then you decide the class introducing ther threshold function. fi the net input is abow you have an output, below you have the other one



each weight is associated to one feature. indicates the strength of one feature. Instead of summing up all the feature we give a weight and tell how much important is that feature.

the weights are parameters of the model.

we don’t fix a priori the weight. the learning algorithm will try to find the best set of wights to have the best set of predictions.

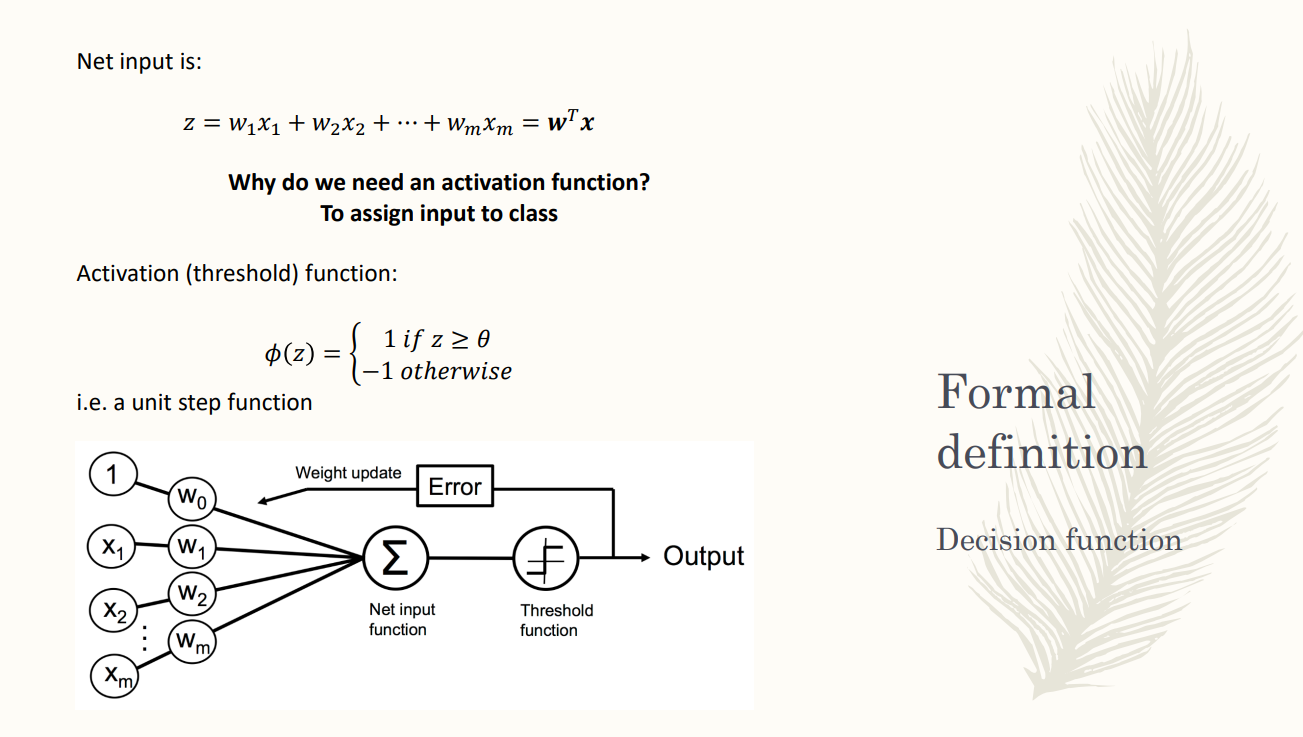


use leaner combination of the feature using as coefficients the weights. combine all the values of the features in a linear combination where each feature is weighted by the corresponding weight.

the learn combination of the feature is net input function.

and we csan write it in a vector form. weight are in the interval 0,1

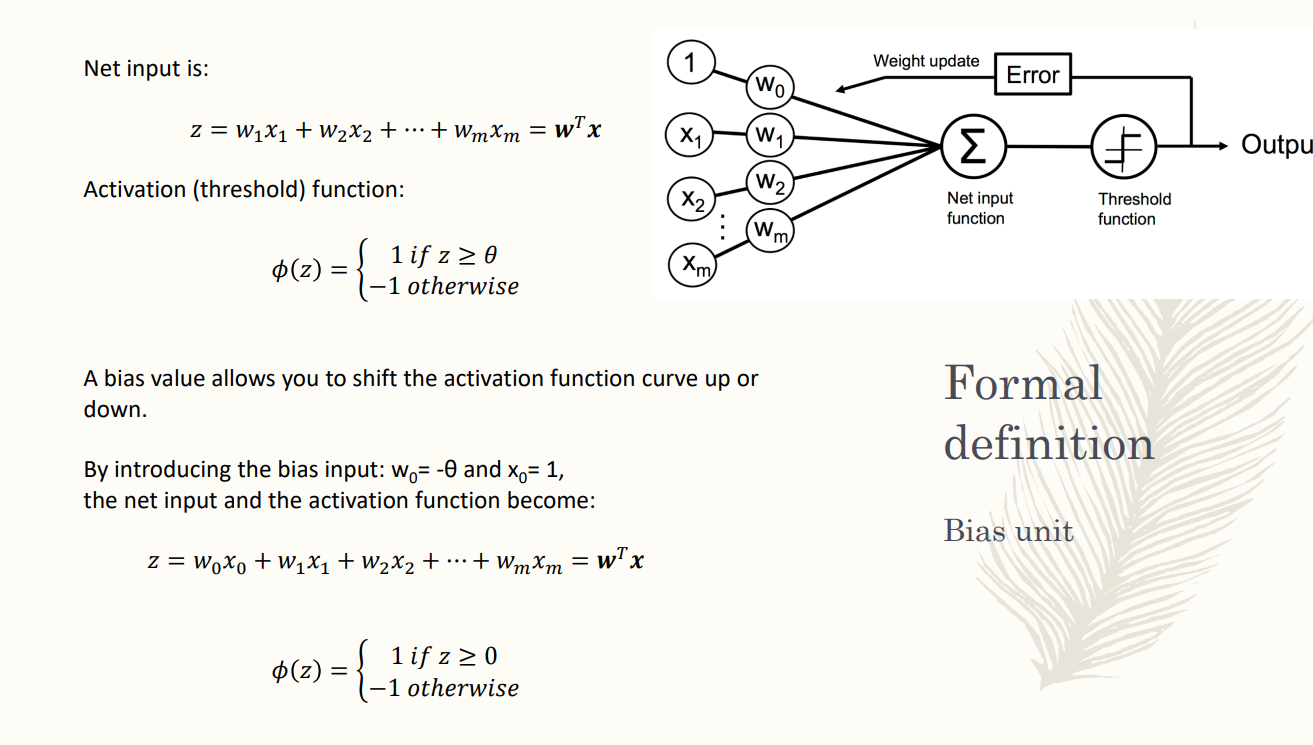
z = linear combination of the feature with their rispoective weights



we need an activation function. if the sum is about a threshold the neuron fire. We need a function that help us performing the classification

Using an activation function (decision function, treshdolg function) +compl

perception model use the most simple activation function unit activation function



φ = 1 if is ≥ a threshold. We are fixing a threshold θ if the net input is above or equal to the threshold then the sample belongs to the positive class, if it is less it belongs to the negative class

after a linear combination we have the treshold function

the threshold is fixed. It is an hyperparateters.

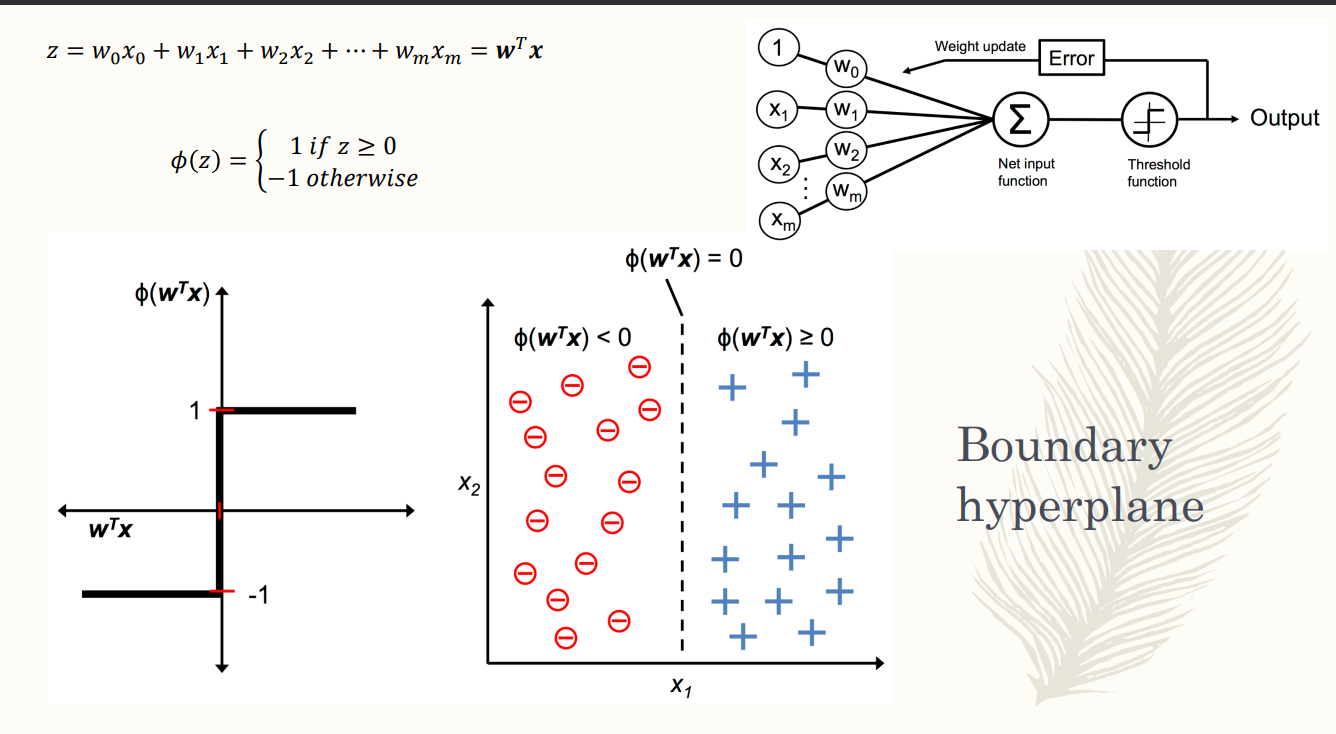
parameters: part of the model

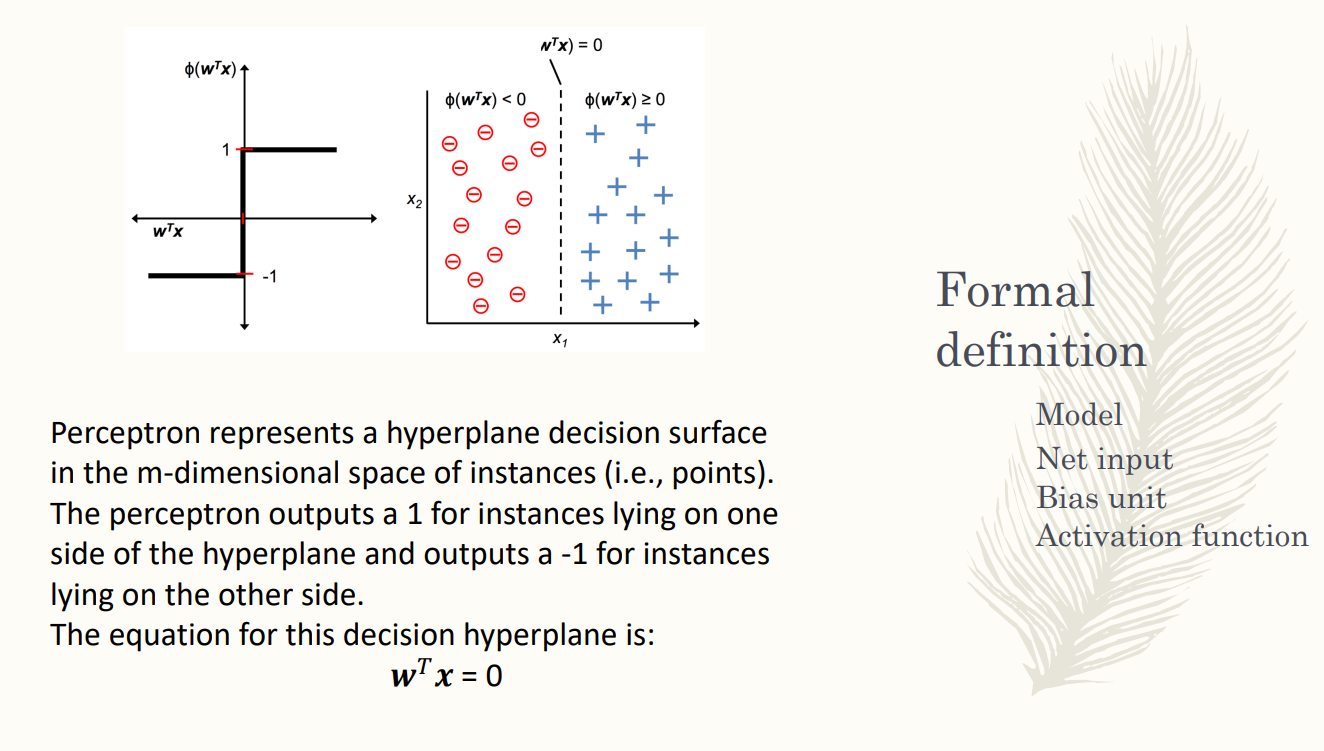
hyperparaters: fixed a priori in the model, can’t learn that. you decide it and can not be changed

usually we prefere to have model without hyperparamters or have a limited numbr of it.

the threshold is an hyperparaterm. we can have a feature that is not a feature of the sample but x0 is always equal to one. Is a variable that is always equal to 1 for all sample. **bias unit**. If we say that w0 = -θ we can rewrite the leaner combination to add it to the real combination of features.

it allows to shift the sativation function up and down





how to assign a sample to a class

we can rewrite how to assign a sample to a class by the activation function is equal to the sign w^t x

when the sign of the learn cobmintion is positive. the sign decides with class is assign to a sample

How to find the best weigths. Introduction the cost function in order to obtain the best +compl

the idea of the cost function is to mimized the number of cost classification. find the set of werghts that the leads to the minimum number of missclassfied sample.

if the true label is different form the predict one we have a missification. find the best best set of wehgts that minimized the number of sample that are not correctly classified

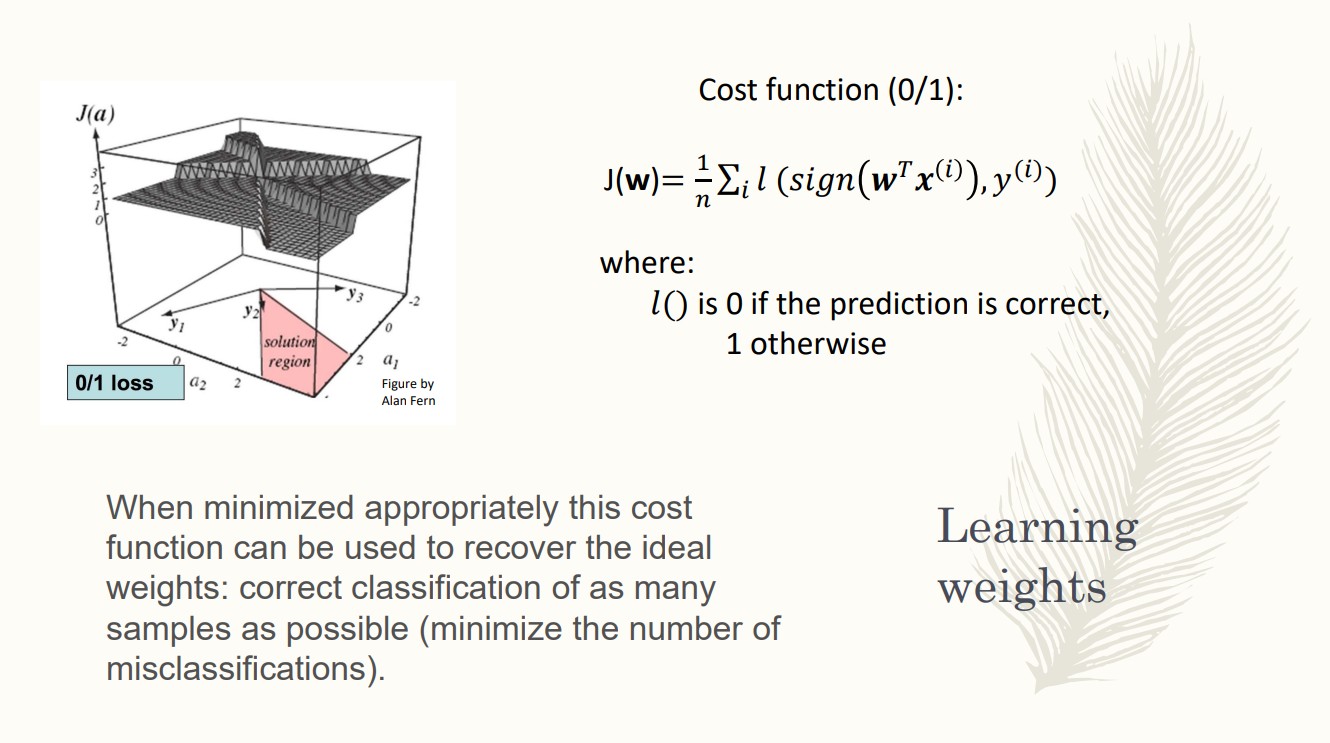
we have a correct classification when the sign of the net input is equal to the true label given by the training dataset

for each sample correctly classified we have a sample equal to 0. to all sample not correctly classified wee have 1

the cost function countries how mant sample have a predicted label different front ehtrue one. it is a cost function because if you minimized the cost function +compl

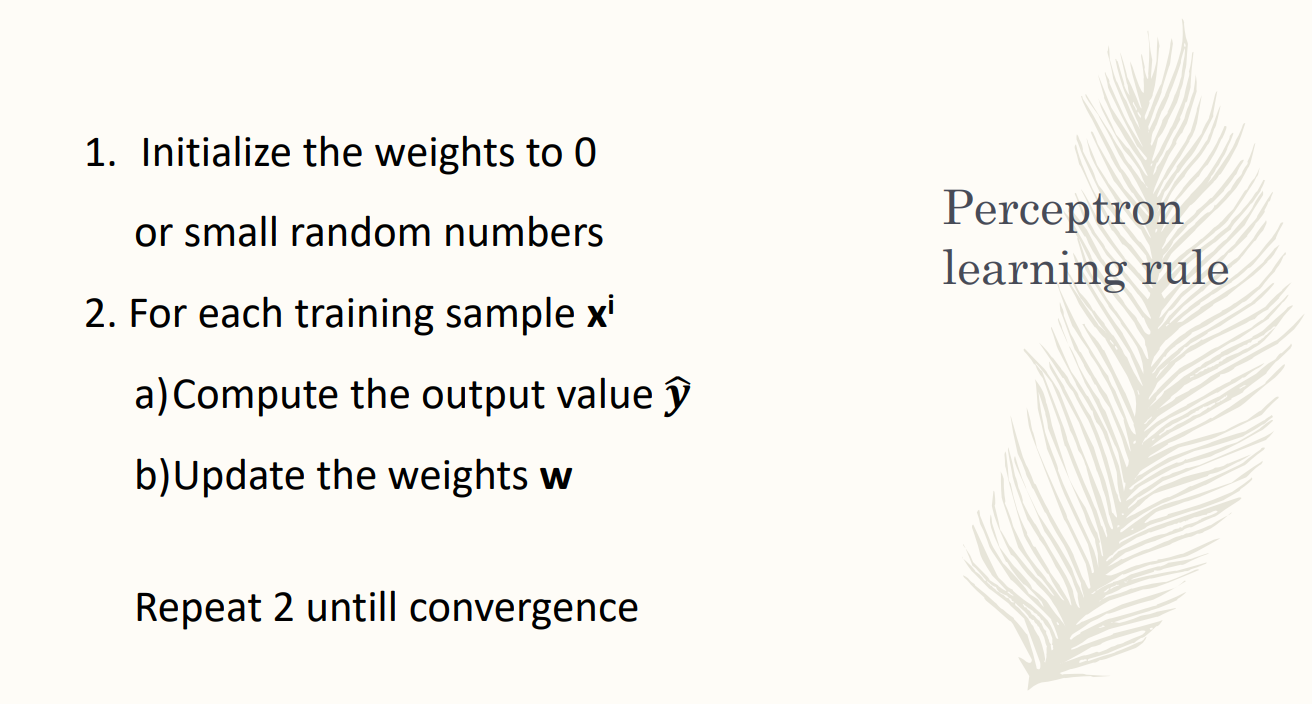
find a set of weights that minimized the cost function.

METTERE ALTRE SLIDES PRIMA



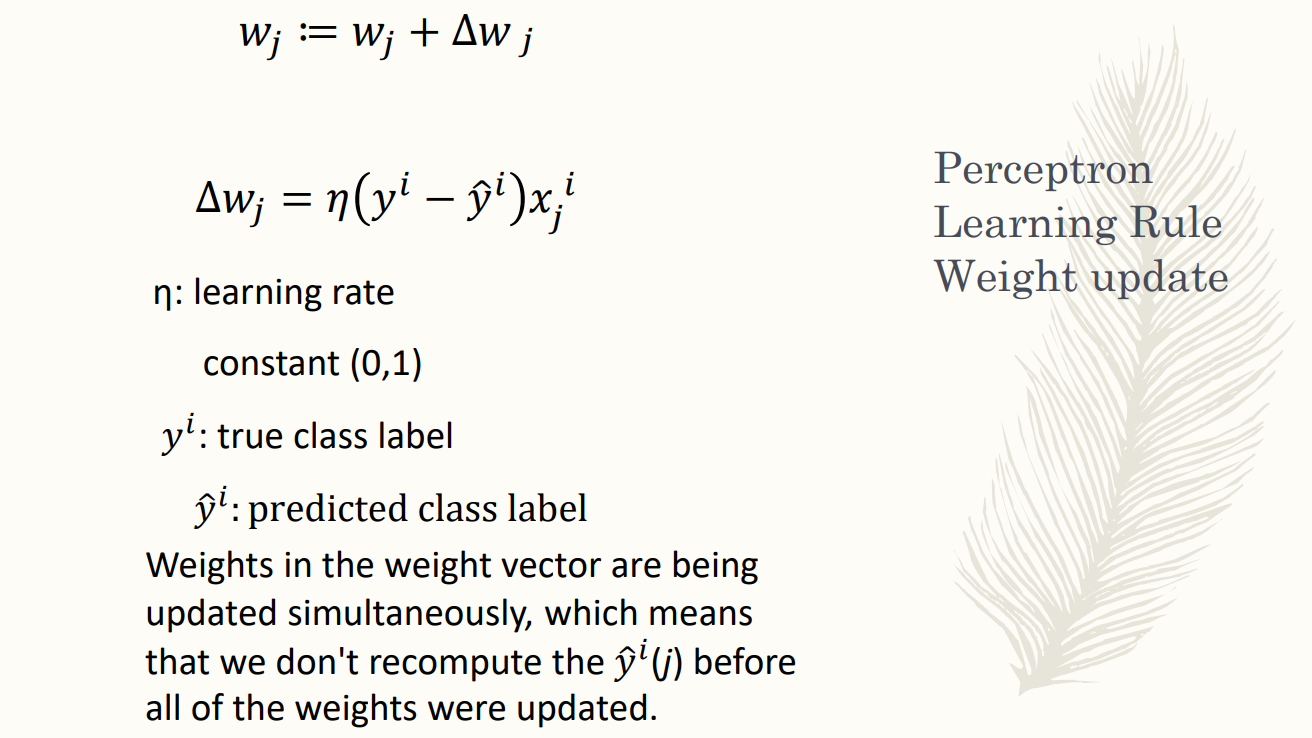
we want to find a set of weights mimizing the set function.

the cost function is a set iof function, can not fnd the mimium.



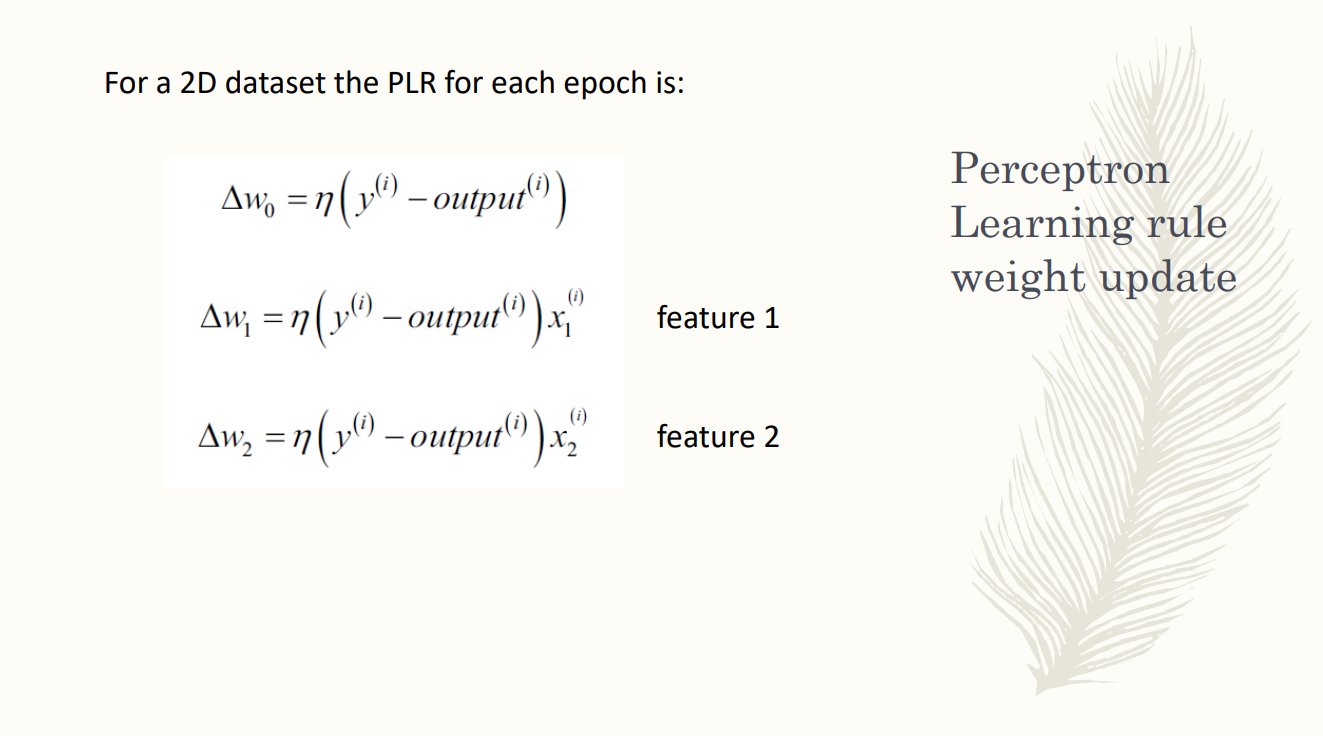
first learning aghortim introduced. the eprcetion learning rule initizled all the weights to 0 or a small number.

commute the predicted value of the class. we can compute for each sample the predicted class. then we update the weights according the the eprceipn rule and repat the step until we reacht ehconvergence



you have to fix it a priori

the difference between the true class and the rpedicted one t



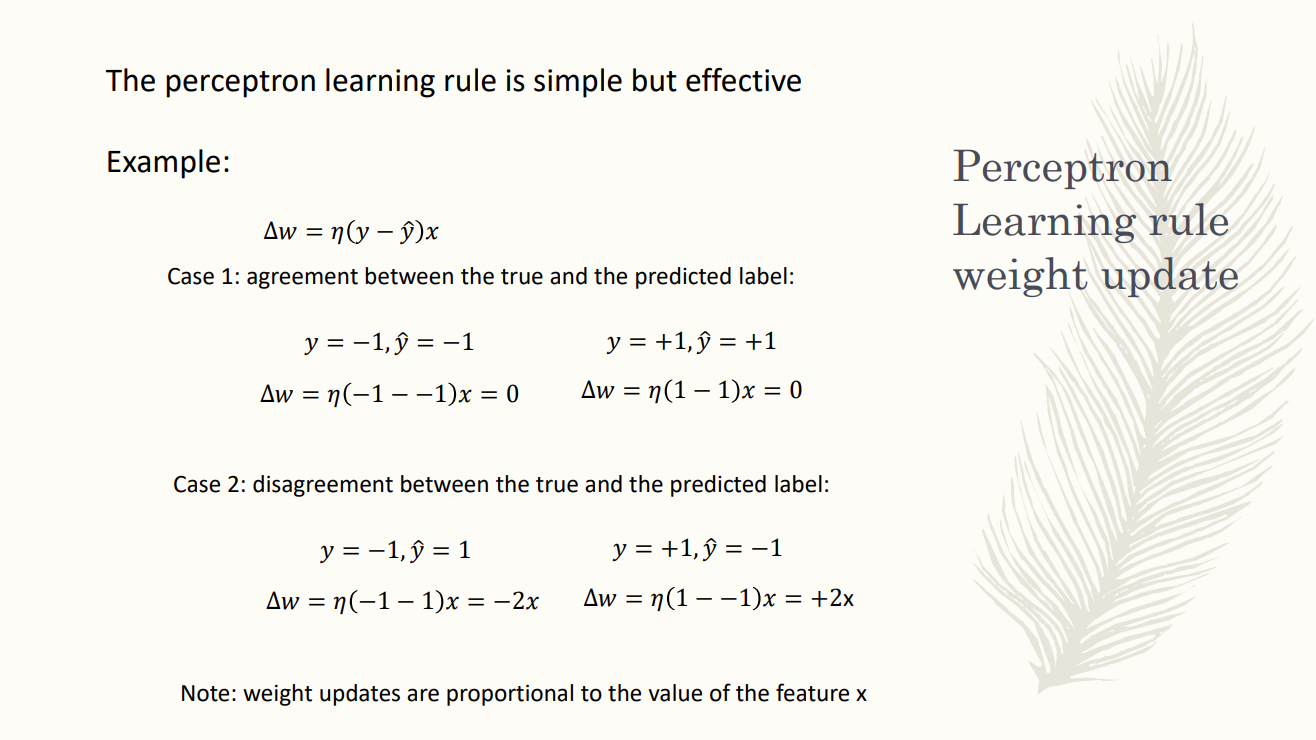
for each sample you update the weight according to the rule. call Epoch.

then start again and have new epoch. you go on with epoch until you reach the convergence.

for each sample starting from the first you update iteratively the weights. for each sample you have to update all the weights.

if you have two features you have to update the bias unit.

for each epoch and for each sample you update the weight

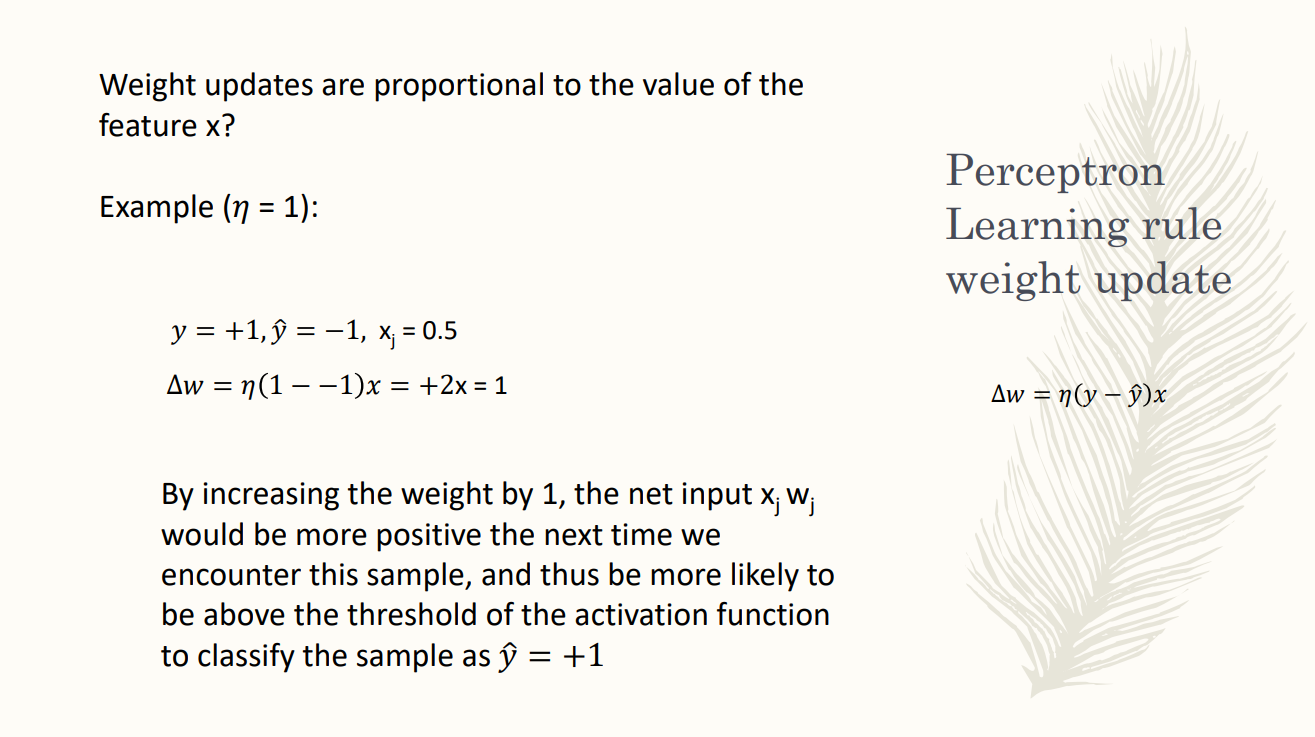


you have an agrement between the true and predicted label. You compute the predicted label and the true label of the sample. With the weights fixed the predicted weight of the sample is ewqual to the true label

if you compute how the perception rule would change you uptain that having the true and the predicted label equal you don’t update the eight and remains the same. the perception rule makes the wright do not change for the sample

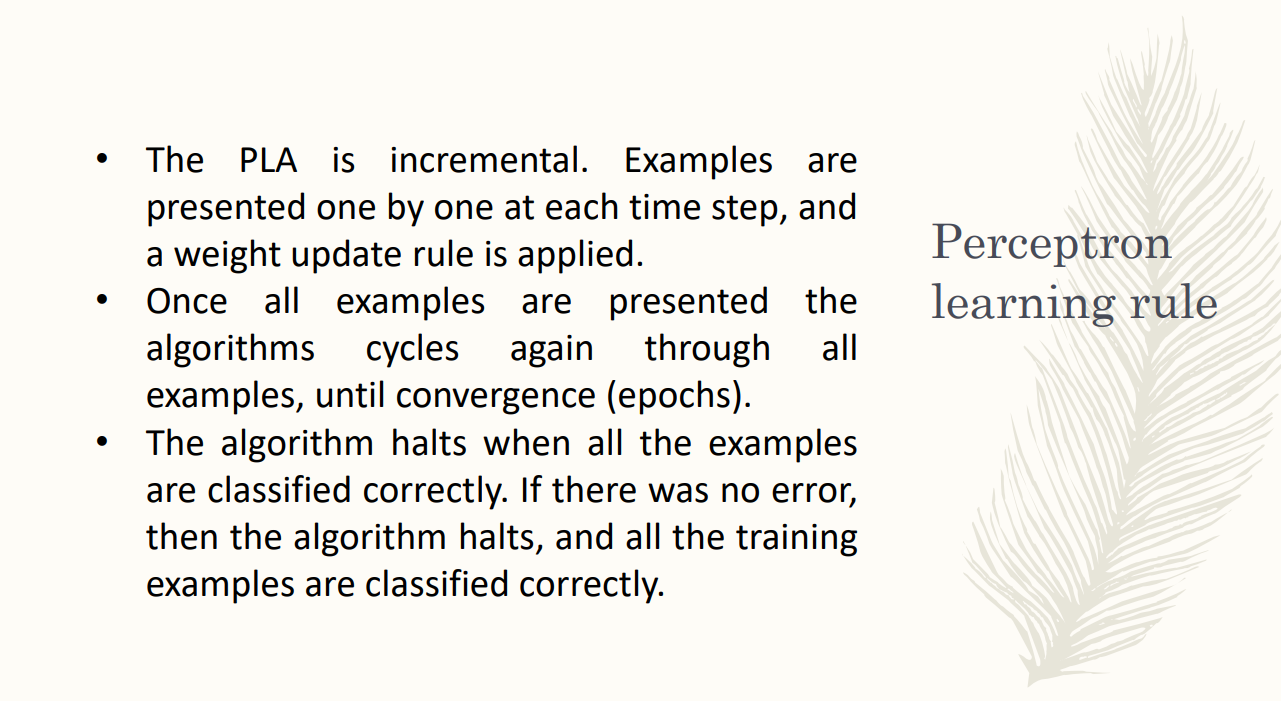
try to update the weights in order to shift the outcomes according to the disagreement.

you update the wrights in orderto maker the outcme+compl



you change the weights but you don’t change the learning.

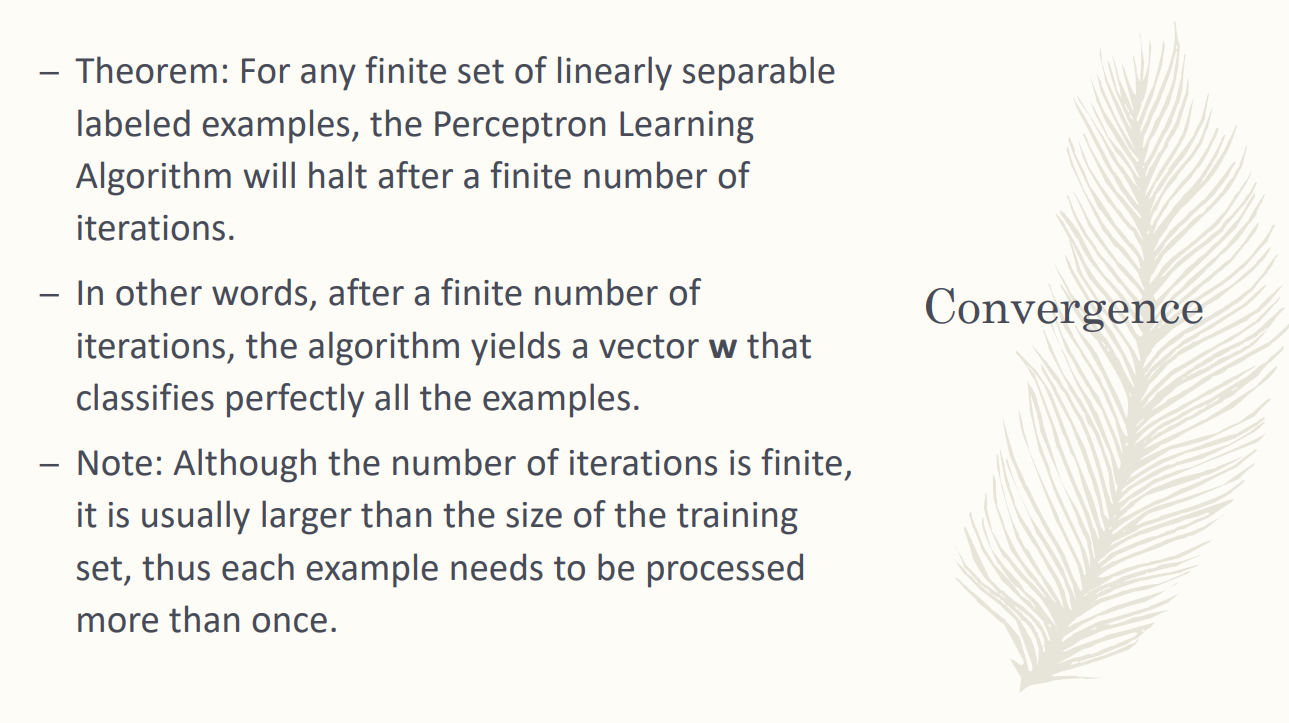
FOR TOMORROW REMEMBER TRHE FUNCTION



sample are presented one by one. anaylze one sample upstate the wefith

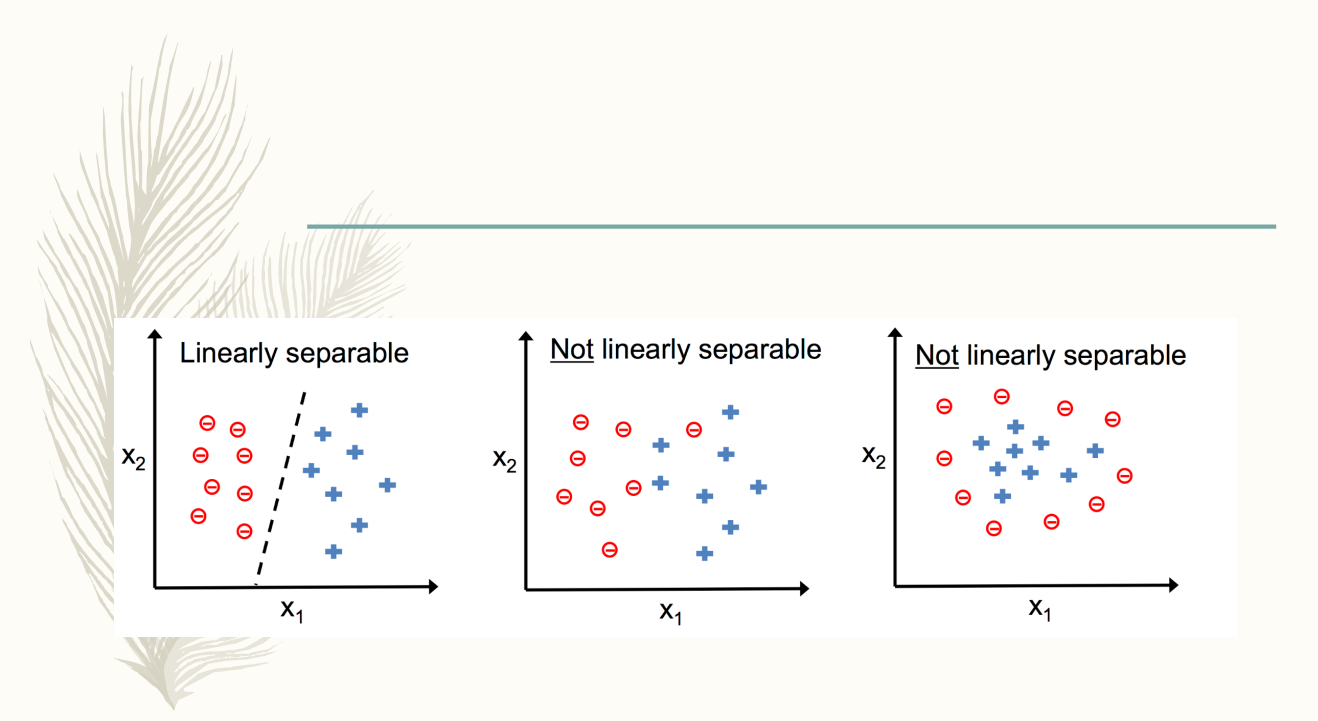
if you don’0t have update you reach the convergence

the algorithm stop when for all the sample in the epoch you don’t have udptte of the weights. that is all the predicted labels are equal to the true labels



the perception learning laghortim will always reacht ehcovnergence i linealry separated class.

it ∃ an htyperplane dividing positive and negative points. you have the convergence in a finite number of iteration.



of your problem is not linearly separstre the perception does not add. can’t have the correct solution and the perception does not give a solution