
upGrad
Raho Ambitious

upGrad

#LifeKoKaroLift

SGC Coaching:

Articulate your Journey | Activate Students' Vigour| Accelerate Mutual Growth

JumpStart to an engaging session with your group!

using these slides for reference

Agenda: Session 7

upGrad

Classification model –(5 mins)

Focused Teaching –(55 mins)

Decision trees, Random Forest and Bagging

Confusion matrix

Doubt resolution –(30 mins)

Discuss - Notebooks

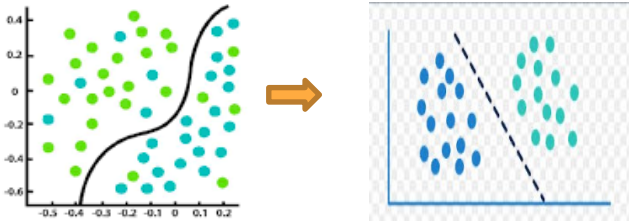
Focussed Teaching: Classification model
Decision trees

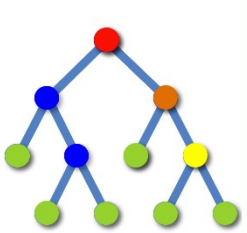
upGrad

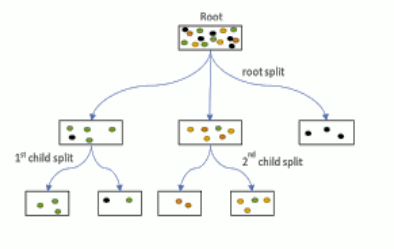
Supervised learning

Goal is to separate the classes

Decision trees







CART

CHAID TREES

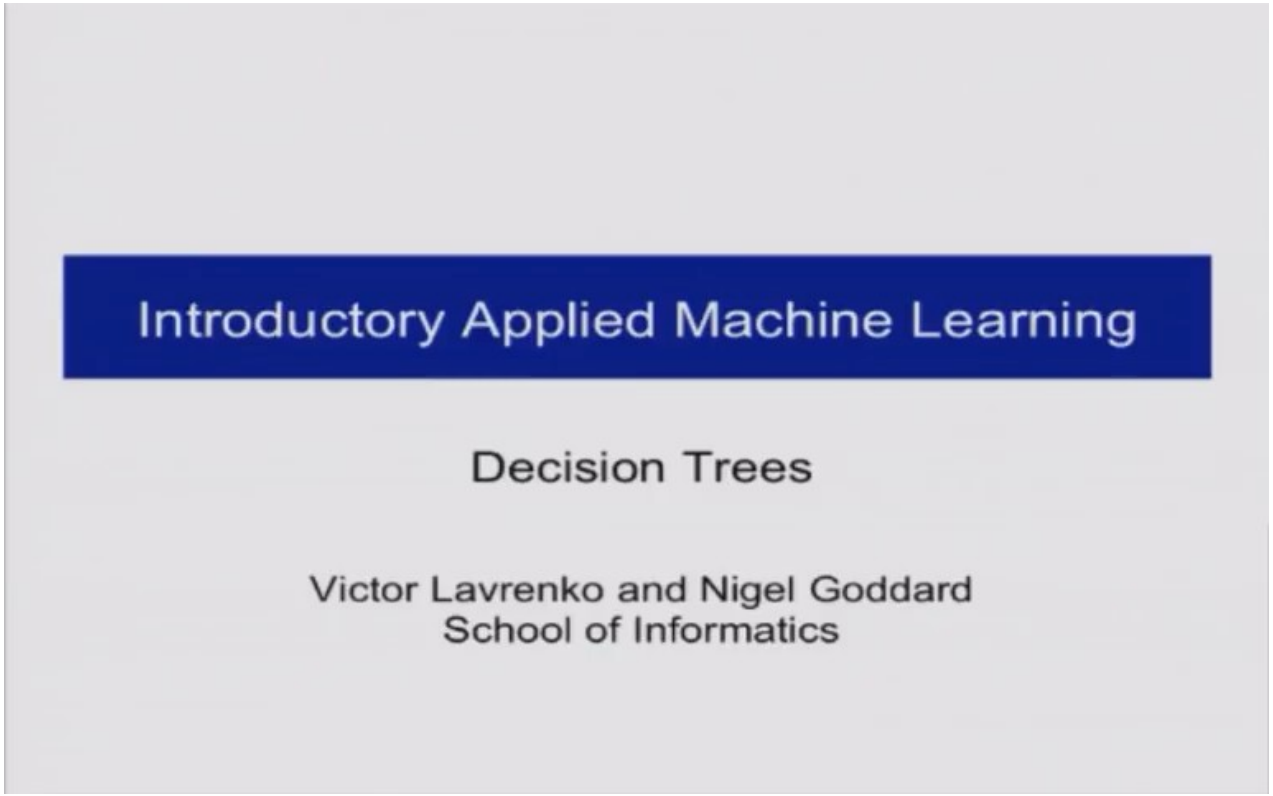
- Target variable
- These are Classes
- Convert continuous into target variable

- Goal separate the classes
- We can use any form of separator
- Many methods predict Y value – at a threshold of 0.5 the classes are separated

- Trees split the target variable
- Basis the predictor value splits
- CART gives only binary trees/ regression solution

Focussed Teaching : Decision tree logic

upGrad



Focussed Teaching : Decision tree logic

upGrad

A general algorithm for a decision tree can be described as follows:

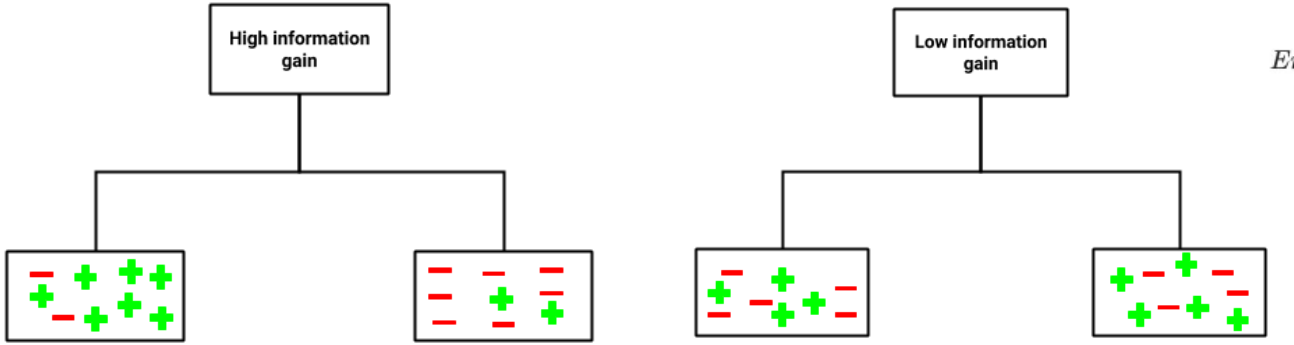
1. Pick the best attribute/feature. The best attribute is one which best splits or separates the data.
2. Ask the relevant question.
3. Follow the answer path.
4. Go to step 1 until you arrive to the answer.

The best split is one which separates two different labels into two sets.
Decision trees divide the feature space into axis-parallel rectangles or hyperplanes

Focussed Teaching : Decision tree logic

upGrad

Information gain is a statistical property that measures how well a given attribute separates the training examples according to their target classification



$$Entropy : \sum_{i=1} -p * log_2(p_i)$$
$$p_i = Probabilityofclassi$$

$$InformationGain = Entropy(parentnode) - [AverageEntropy(children)]$$

Check for more features at https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Focussed Teaching : Decision tree - code along

upGrad

```
#Importing required libraries import pandas as pd import numpy as np from sklearn.datasets import load_iris from
sklearn.tree import DecisionTreeClassifier from sklearn.model_selection import train_test_split
#Loading the iris data
data = load_iris()
print('Classes to predict: ', data.target_names)
#Extracting data attributes
X = data.data
 ### Extracting target/ class labels
y = data.target
print('Number of examples in the data:', X.shape[0])
#Using the train_test_split to create train and test sets. X_train, X_test, y_train, y_test = train_test_split(X, y,
random_state = 47, test_size = 0.25)
#Importing the Decision tree classifier from the sklearn library.
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(criterion = 'entropy')
#Training the decision tree classifier.
clf.fit(X_train, y_train)
#Predicting labels on the test set.
y_pred = clf.predict(X_test)
#Importing the accuracy metric from sklearn.metrics library
from sklearn.metrics import accuracy_score
print('Accuracy Score on train data: ', accuracy_score(y_true=y_train, y_pred=clf.predict(X_train)))
print('Accuracy Score on test data: ', accuracy_score(y_true=y_test, y_pred=y_pred))
clf = DecisionTreeClassifier(criterion='entropy', min_samples_split=50)
clf.fit(X_train, y_train)
print('Accuracy Score on train data: ', accuracy_score(y_true=y_train, y_pred=clf.predict(X_train)))
print('Accuracy Score on the test data: ', accuracy_score(y_true=y_test, y_pred=clf.predict(X_test)))
```

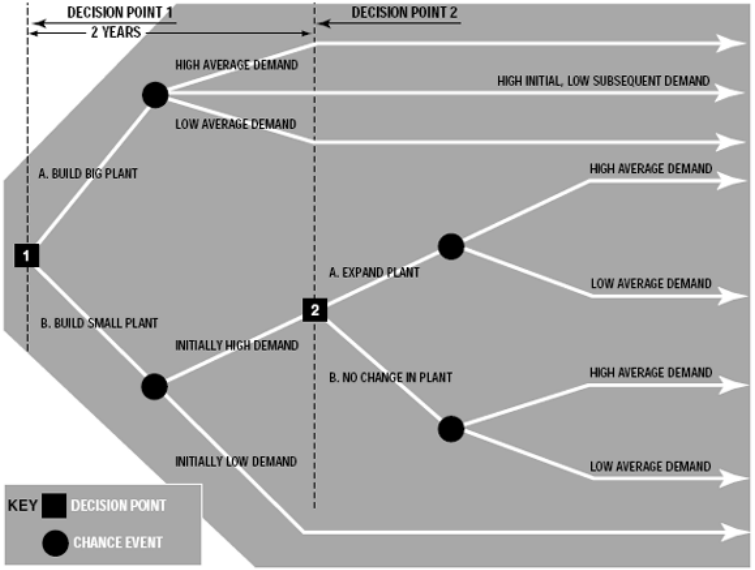
#Output: Out: Accuracy Score on train data:
0.9553571428571429 Accuracy Score on test data:
0.9736842105263158

Focussed Teaching: Business case study
Check the HBR article

upGrad

The management of a company that I shall call Stygian Chemical Industries, Ltd., must decide whether to build a small plant or a large one to manufacture a new product with an expected market life of ten years.

EXHIBIT III. Decisions and Events for Stygian Chemical Industries, Ltd.



Pls refer to article : https://hbr.org/1964/07/decision-trees-for-decision-making

Focussed teaching: Decision trees - Pros and Cons

upGrad

Pros	Cons
Easy to use	Prone to overfitting
Can handle both categorical and numerical data	Need a measurement as to how well they are doing
Resistant to outliers	Need to be careful with parameter tuning
New features can be easily added	Can create biased learned trees

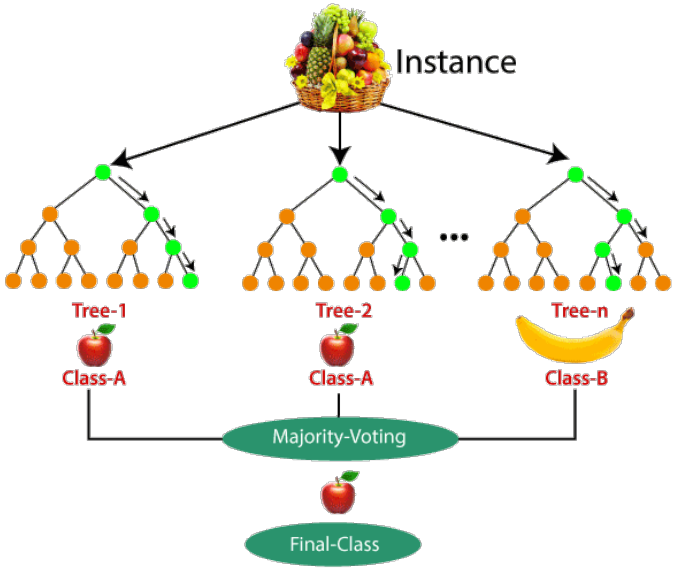
We can limit the tree growth by specifying the number of branches, min samples for split and min samples for a leaf node – called pruning of decision trees
This will to a level provide us a better solution

Focussed Teaching : Ensemble model
Random Forest

upGrad

Rather than depending one decision tree lets create a lot of trees and take the best result
This will be a good solution that is not an overfit solution
This is called an ensemble methodology – this ML is Random Forest

Random Forest logic

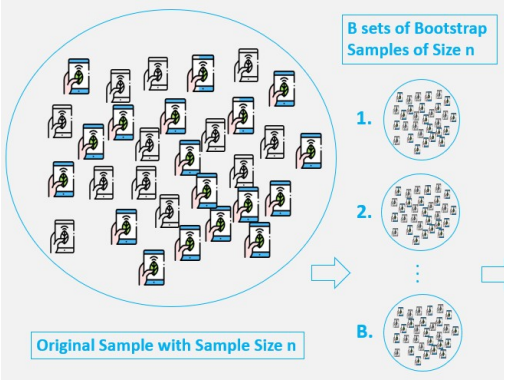


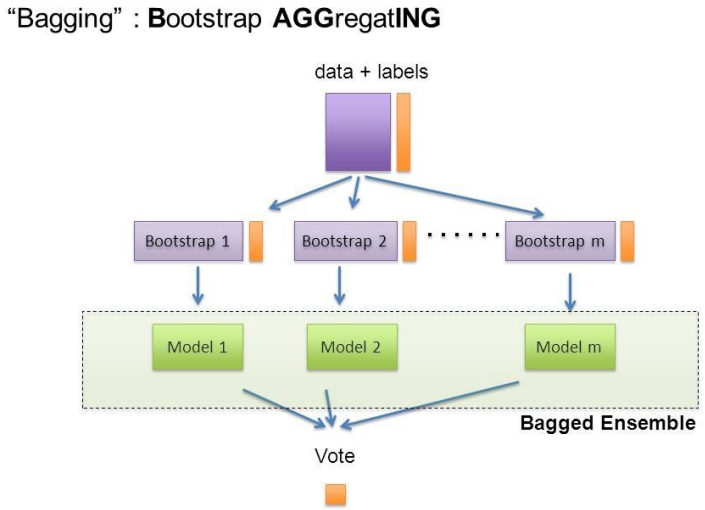
Focussed Teaching : Ensemble model
Bagging

upGrad

Rather than depending one decision tree lets create a lot of trees we will aggregate the solution
This is called an ensemble methodology

We can create a variety of data using Bootstrapping

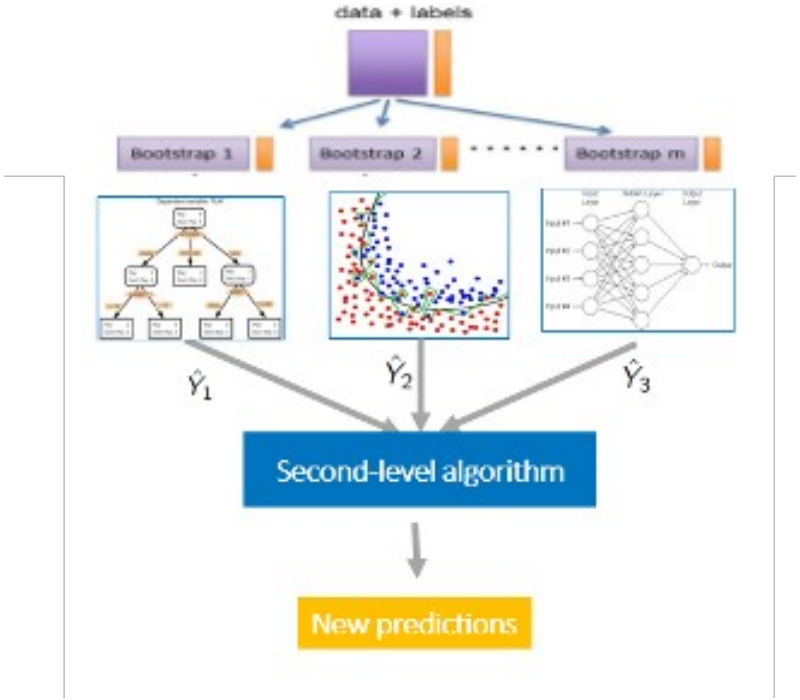




Focussed Teaching : Ensemble model
Blending

upGrad

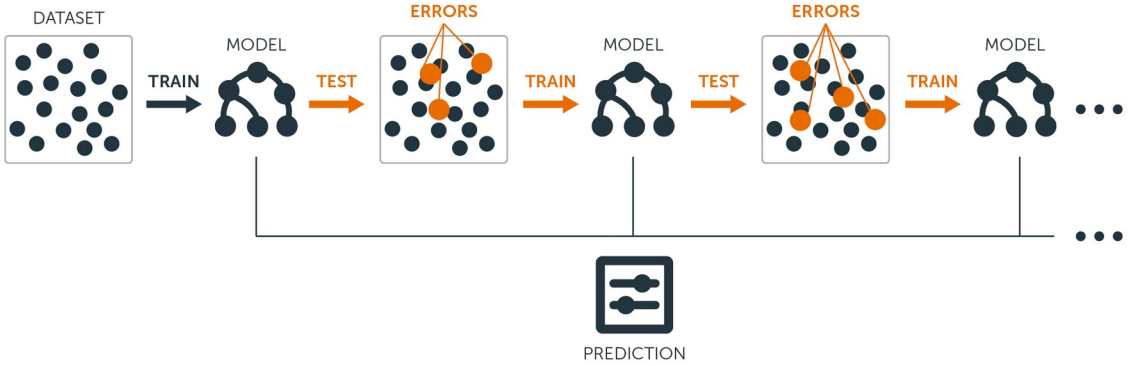
Rather than depending on decision trees and taking a vote, we can use two sets of algorithm
1/ first level – for ML
2/ second level – in place of voting



Focussed Teaching : Ensemble model
Boosting

upGrad

We can use decision trees to improve performance
The initial model – weak learners can be combined to give a high performing prediction
This works by reducing errors (y predicted vis-à-vis y actual)
These are called Boosting algorithm
We have 3 boosting methods – ADA Boost , Gradient Boost & XG Boost



Personalized Feedback and Doubt resolution

upGrad

Address individual concerns and provide specific feedback on their notebooks

- What are they missing?
- What can they do better?

Doubt Resolution

Questions??