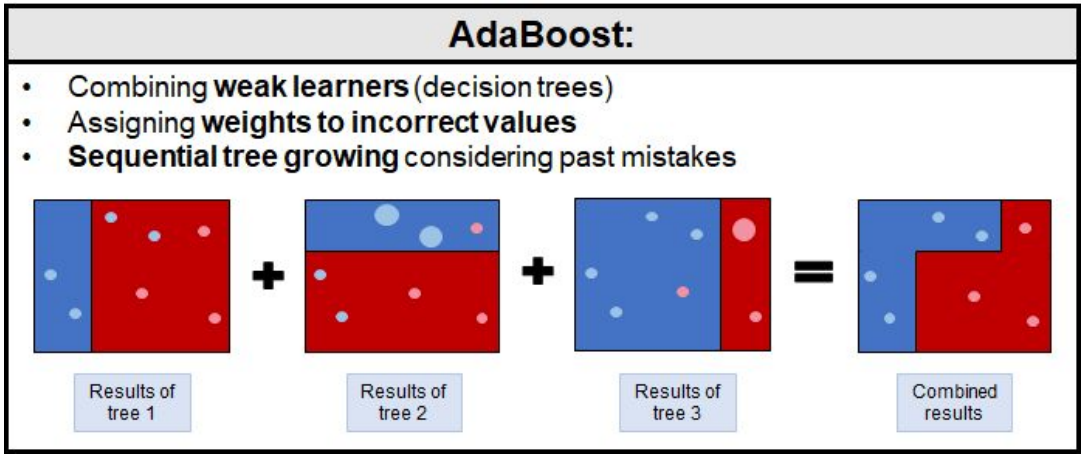
Boosting

AdaBoost

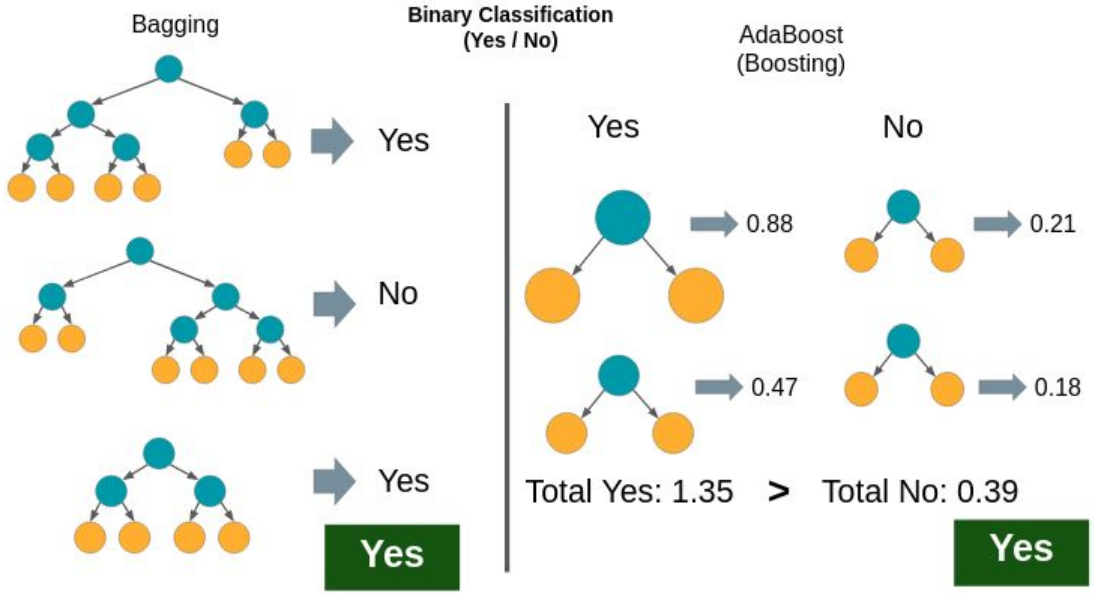
- AdaBoost Adaptive boosting is a boosting algorithm, which also works on the principle of the stagewise addition method where multiple **weak** learners are used for getting **strong** learners.



How AdaBoost Works

- **Step 1:** A weak classifier (e.g. a decision stump) is made on top of the training data based on the weighted samples. Here, the weights of each sample indicate how important it is to be correctly classified. Initially, for the first stump, we give all the samples equal weights.
- **Step 2:** We create a decision stump for each variable and see how well each stump classifies samples to their target classes.
- **Step 3:** More weight is assigned to the incorrectly classified samples so that they're classified correctly in the next decision stump. Weight is also assigned to each classifier based on the accuracy of the classifier, which means high accuracy = high weight!
- **Step 4:** Reiterate from Step 2 until all the data points have been correctly classified, **the error function does not change**, or the **maximum iteration level** has been reached.

How AdaBoost Works



An Example of How AdaBoost Works

- Suppose we have the sample data below, with three features (x1, x2, x3) and an output (Y). *Note that T = True and F = False.*

x1	x2	x3	Y
T	T	F	T
T	T	F	F
T	F	F	T
T	T	T	F
F	T	F	F
T	F	F	T

An Example of How AdaBoost Works

- **Step 1: Assign a sample weight for each sample**

$$sample\ weight = \frac{1}{\#\ of\ samples}$$

x1	x2	x3	Y	Sample Weight
T	T	F	T	1/6
T	T	F	F	1/6
T	F	F	T	1/6
T	T	T	F	1/6
F	T	F	F	1/6
T	F	F	T	1/6

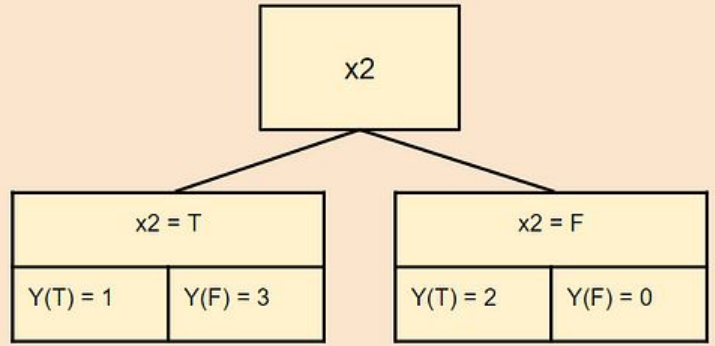
An Example of How AdaBoost Works

- **Step 2: Calculate the Gini Impurity for each variable**

This is done to determine which variable to use to create the first stump.

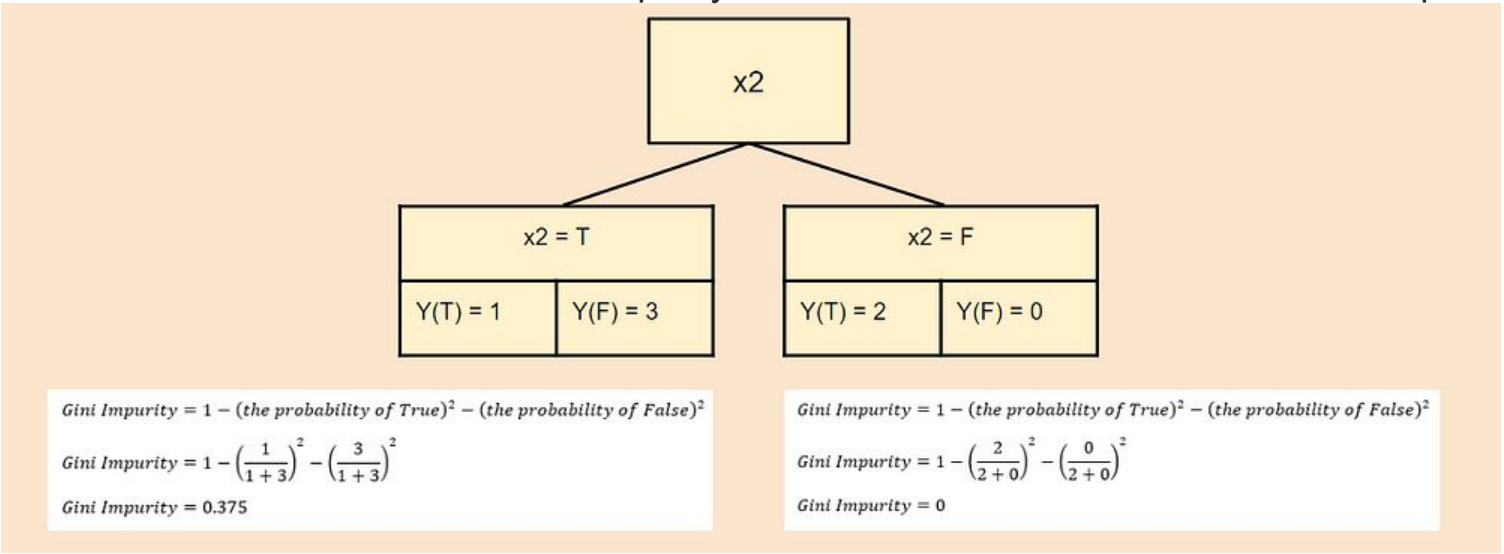
$$Gini\ Impurity = 1 - (the\ probability\ of\ True)^2 - (the\ probability\ of\ False)^2$$

x2	Y = T	Y = F
X2 = T	1	3
X2 = F	2	0



An Example of How AdaBoost Works

x2 has the lowest Gini Impurity, so x2 will be used to create the first stump.



$$Total\ Impurity = 0.375\left(\frac{4}{4+2}\right) + 0\left(\frac{2}{4+2}\right)$$

$$Total\ Impurity = 0.25$$

the Gini Impurity for x2 = 0.25.

An Example of How AdaBoost Works

- **Step 3: Calculate the Amount of Say for the stump that was created**

Total Error is equal to the sum of the weights of the incorrectly classified samples.

Since one of the samples was incorrectly classified for x2, the total error is equal to 1/6.

$$Amount\ of\ say = \frac{1}{2}\log\left(\frac{1 - total\ error}{total\ error}\right)$$

$$Amount\ of\ say = \frac{1}{2}\log\left(\frac{1 - \frac{1}{6}}{\frac{1}{6}}\right) = 0.35$$

An Example of How AdaBoost Works

- **Step 4: Calculate the new sample weights for the next stump**

we're going to increase the sample weights of samples that were incorrectly classified and decrease the sample weights of samples that were correctly classified using the following equations:

$$\textit{New Sample Weight For } \textbf{\textit{Incorrect}} \textit{ Samples} = \textit{Sample weight} * e^{\textit{amount of say}}$$

$$\textit{New Sample Weight For } \textbf{\textit{Correct}} \textit{ Samples} = \textit{Sample weight} * e^{-\textit{amount of say}}$$

An Example of How AdaBoost Works

Need to divide each weight by 0.84

x1	x2	x3	Y	New Sample Weight	Normalized Sample Weights
T	T	F	T	0.24	0.29
T	T	F	F	0.12	0.14
T	F	F	T	0.12	0.14
T	T	T	F	0.12	0.14
F	T	F	F	0.12	0.14
T	F	F	T	0.12	0.14

Total: 0.84

An Example of How AdaBoost Works

- Step 5: Create a **bootstrapped dataset** with the odds of each sample being chosen based on their new sample weights.

x1	x2	x3	Y	Normalized Sample Weights
T	T	F	T	0.29
T	T	F	F	0.14
T	T	F	T	0.29
T	T	T	F	0.14
T	T	F	T	0.29
T	F	F	T	0.14

An Example of How AdaBoost Works

- **Step 6: Repeat the process n number of times**
- Lastly, this process is repeated until n number of stumps are created, each with its own amount of say. Once this is done, the model is complete and new points can be classified.
- New points are classified by running them through all of the stumps and seeing how they're classified. Then, the amount of say is summed for each class, and the class with the higher amount of say is the classification of the new point.

Difference Between Boosting Algorithms

Algorithms	Gradient Boosting	AdaBoost	XGBoost	CatBoost	LightGBM
Year	–	1995	2014	2017	2017
Handling Categorical Variables	May require preprocessing like one-hot encoding	No	NO	Automatically handles categorical variables	No
Speed/Scalability	Moderate	Fast	Fast	Moderate	Fast
Memory Usage	Moderate	Low	Moderate	High	Low
Regularization	NO	No	Yes	Yes	Yes
Parallel Processing	No	No	Yes	Yes	Yes
GPU Support	No	No	Yes	Yes	Yes
Feature Importance	Available	Available	Available	Available	Available

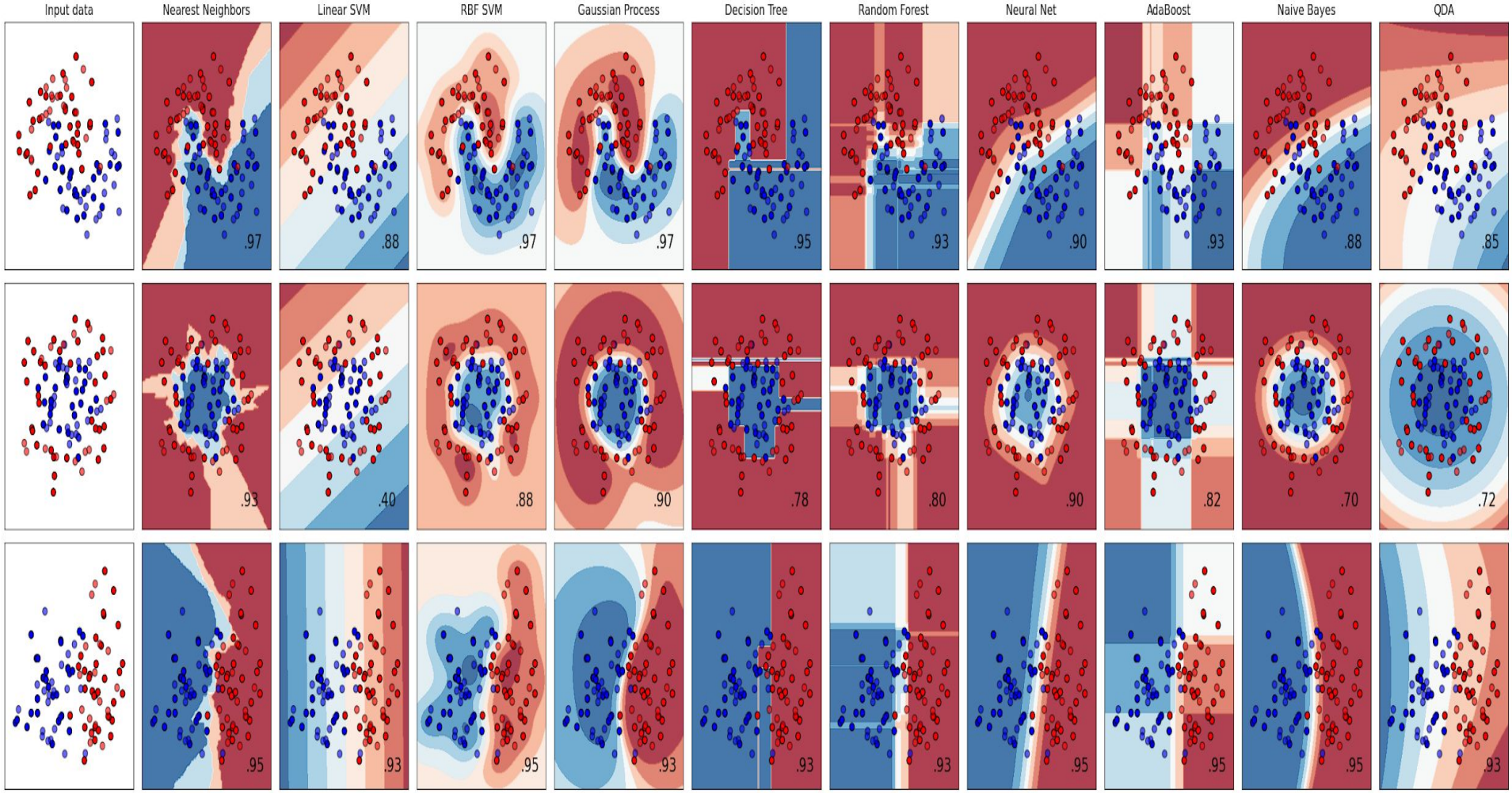
https://www.geeksforgeeks.org/gradientboosting-vs-adaboost-vs-xgboost-vs-catboost-vs-lightgbm/

AdaBoost Code

https://www.analyticsvidhya.com/blog/2015/11/quick-introduction-boosting-algorithms-machine-learning/

Copy Code

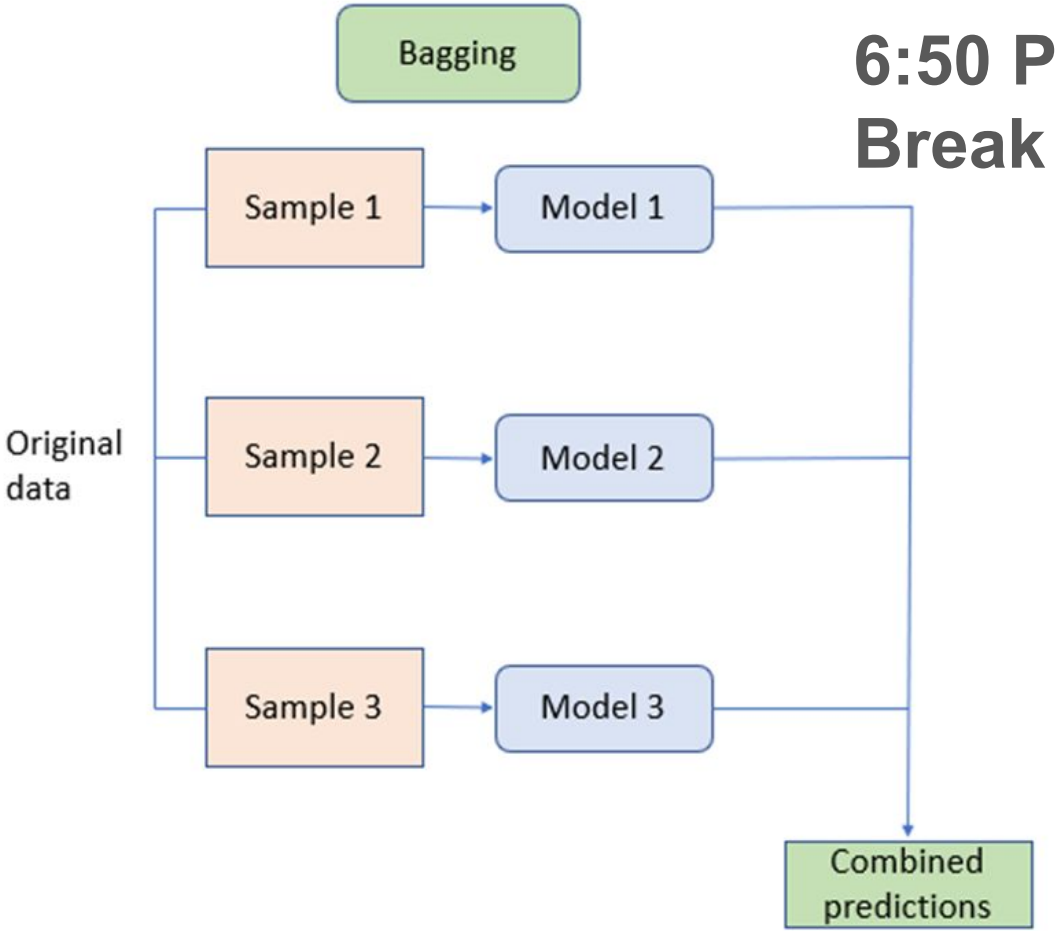
```
# Importing dependencies numpy and keras
import pandas as pd
from sklearn.ensemble import AdaBoostClassifier #For Classification
from sklearn.ensemble import AdaBoostRegressor #For Regression
from sklearn.tree import DecisionTreeClassifier

# reading the data
df = pd.read_csv('train-data.csv')

# first five rows of the data
print(df.head())

# shape of the data
print('\nShape : ', df.shape)

# separating the independent and dependent variables
# independent variable
x_train = df.drop('Survived', axis=1)
# dependent variable
y_train = df['Survived']

# Now we will use decision tree as a base estimator, you can use any ML learner as base es
dt = DecisionTreeClassifier()
clf = AdaBoostClassifier(n_estimators=100, base_estimator=dt,learning_rate=1)

# training the model
clf.fit(x_train,y_train)
```



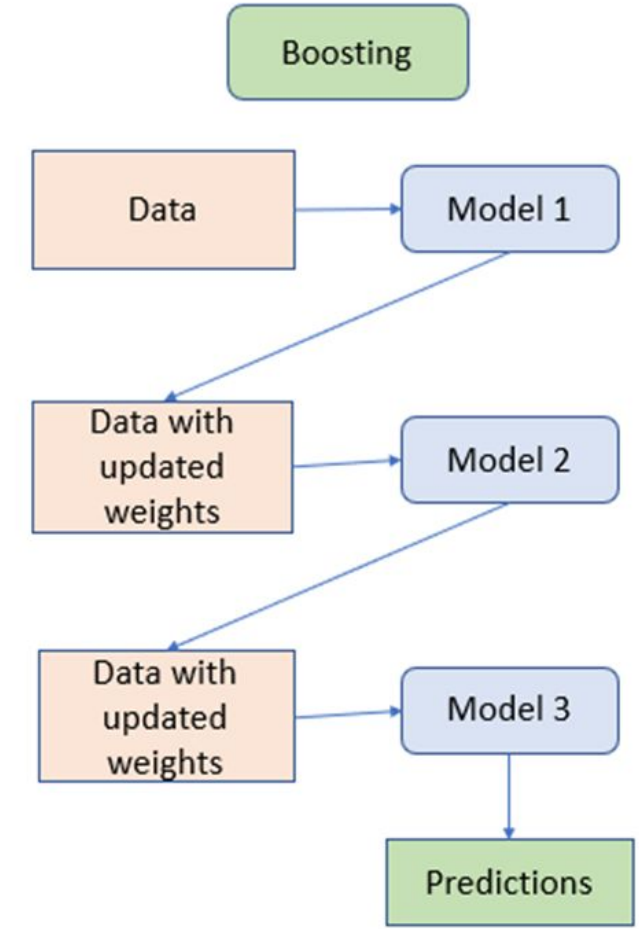
Classifier comparison — scikit-learn 1.3.0

- Decision tree:
  - Split data (nodes, branches , leafs)
  - اكثر عرضة لل **overfiting** > إذا كانت depth deep
  - تغيير بسيط بالداتا ممكن يغير كل ال tree
- Random forest
  - مجموعة من DT بتتدرب على عينات مختلفة من الداتا وبفيتشرز عشوائية والنتيجة بتطلع عن طريق تصويت
  - **decrease variance** < Bagging بالتصويت والتجميع
  - Ensemble learning
  - **هتحل مشكلة ال overfiting**
- Adaboost
  - بناء sequential لل decision stumps بسيط بحيث كل مودل جديد بصحح اخطاء السابق والنتيجة بتتحدد وزن ال rows samples حسب الاداء
  - الدقة اعلى من ال RF لانها **بتقلل bais** وبتركز على الخطأ لانها بتعطي اهمية واولوية اكثر للغلطان لحتى يتعلم منيح فهاد معناه انو الدقة هتزيد
  - حساسة لل outliers

6:50 PM Break



★ Bagging is used mainly in reducing variance and it is a parallel process



★ Boosting is used mainly in reducing bias and it is a sequential process