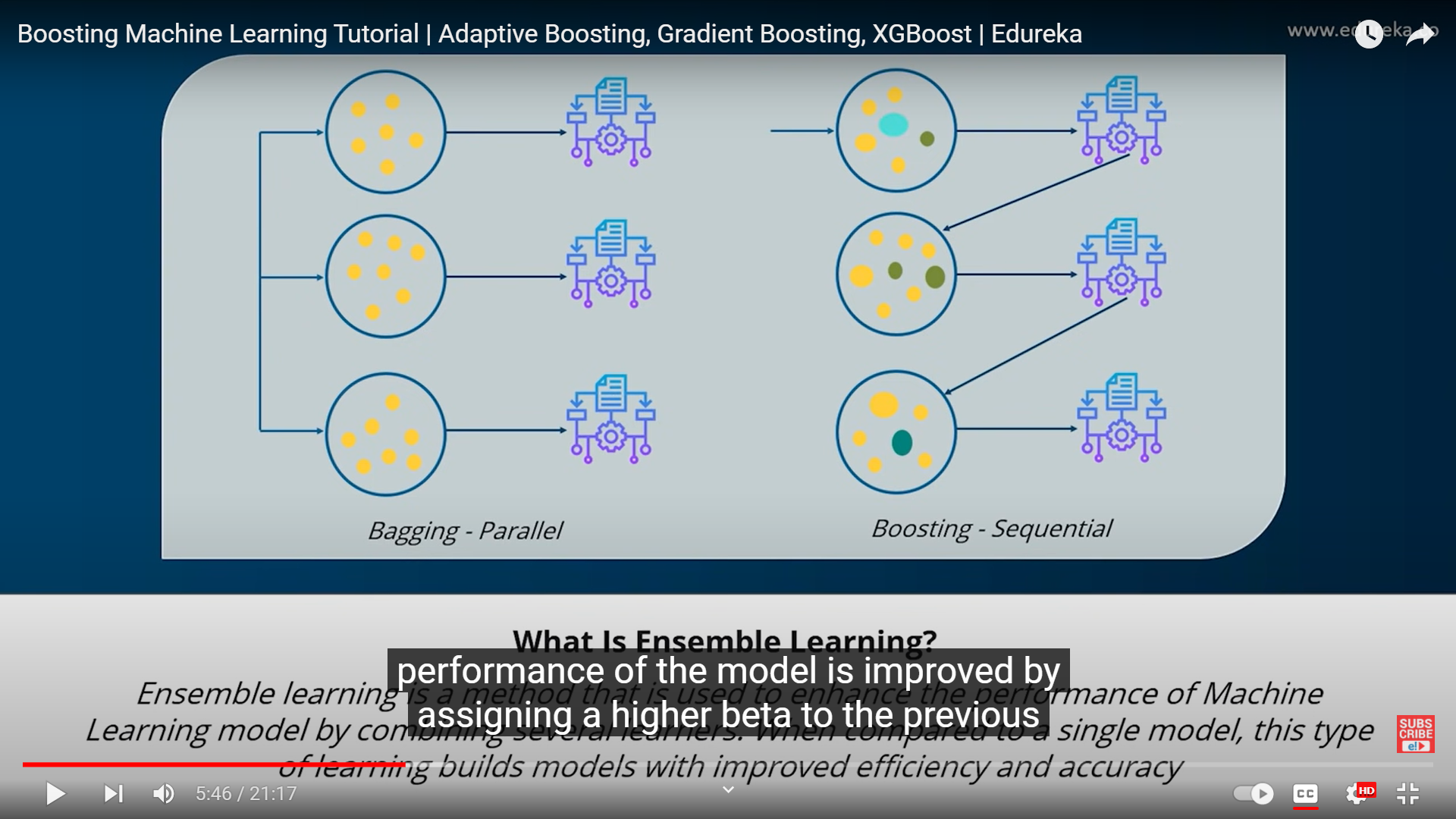
**Boosting**



**Ada Boost**

Example

|  |  |  |
| --- | --- | --- |
| Internal Memory | Camera pixels | Buy\_mobile |
| 4 | 14 | Yes |
| 8 | 13 | No |
| 8 | 14 | No |

Add weights

|  |  |  |  |
| --- | --- | --- | --- |
| Internal Memory | Camera pixels | Buy\_mobile | Weights |
| 4 | 14 | Yes | 1/3 |
| 8 | 13 | No | 1/3 |
| 8 | 14 | No | 1/3 |

Base model is been created ,trained and validated.If the model gives wrong predictions.

The misclassified record’s weights is been increased ,correct prediction record’s weights are decreased to normalize them.

Second model is created,trained and validated.The above steps are performed iteratively until correct predictions are obtained.

**Gradient Boosting**

|  |  |  |
| --- | --- | --- |
| Internal Memory | Camera pixels | Price |
| 4 | 14 | 10000 |
| 8 | 13 | 15000 |
| 8 | 14 | 17000 |

Aim:

1.Loss Function

2.Design Additive model

Step 1: Find the Average the target variable.

Average=14000

Step 2: Find the Residual (Predicted-Actual)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Internal Memory | Camera pixels | Price | Predicted (Average) | Residual |
| 4 | 14 | 10000 | 14000 | 4000 |
| 8 | 13 | 15000 | 14000 | -1000 |
| 8 | 14 | 17000 | 14000 | -3000 |

Step 3:Select a base learner model and predict the Residual

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Internal Memory | Camera pixels | Price | Predicted (Average) | Residual | Residual predicted |
| 4 | 14 | 10000 | 14000 | 4000 | -3500 |
| 8 | 13 | 15000 | 14000 | -1000 | -997 |
| 8 | 14 | 17000 | 14000 | -3000 | 3500 |

Update the default prediction:

Predicted+(learning rate \* residual\_predicted)

14000+(0.1\*-3.5) =13.996.5

Above steps will be repeated.

Final Prediction= Base Value+( learning \* 1st residual\_predicted) +( learning \* 2nd  residual\_predicted)+……….+ +( learning \* nth residual\_predicted)

**Shrinkage**

Shrinkage refers to the fact that the prediction of each tree in the ensemble is shrunk after it is multiplied by the learning rate (eta) which ranges between 0 to 1.

There is a trade-off between eta and number of estimators, decreasing learning rate needs to be compensated with increasing estimators in order to reach certain model performance.

Each tree predicts a label and final prediction is given by the formula, y(pred) = y1 + (eta ∗ r1) + (eta ∗ r2) + ....... + (eta ∗ rN)

