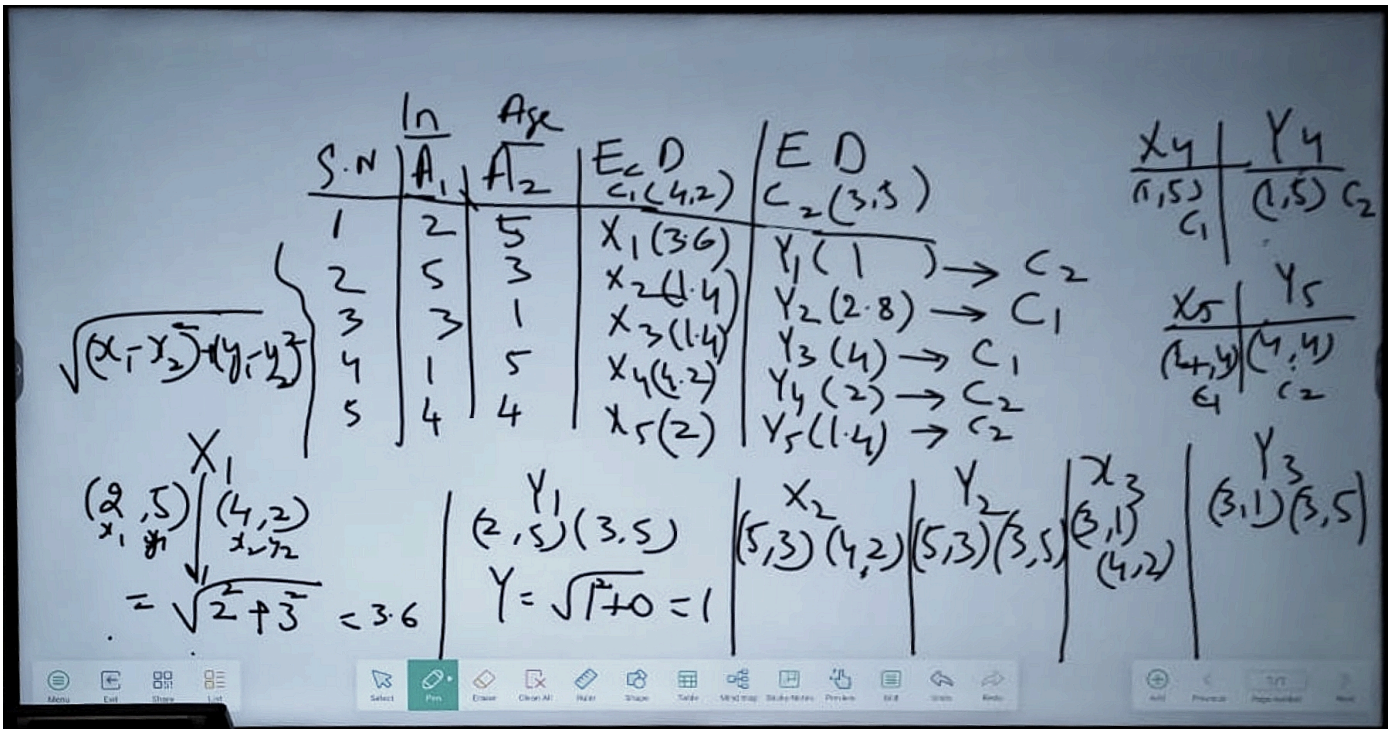
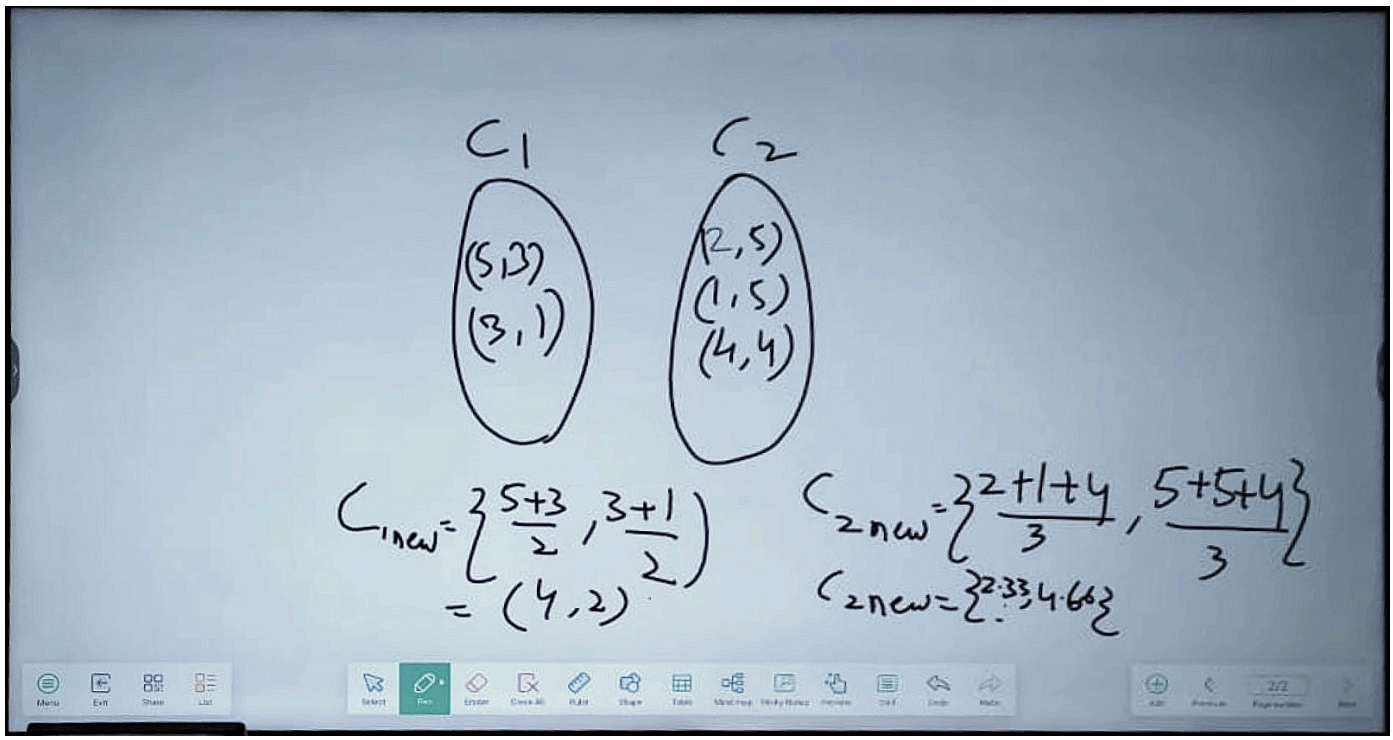
# Evaluate the Trained Model

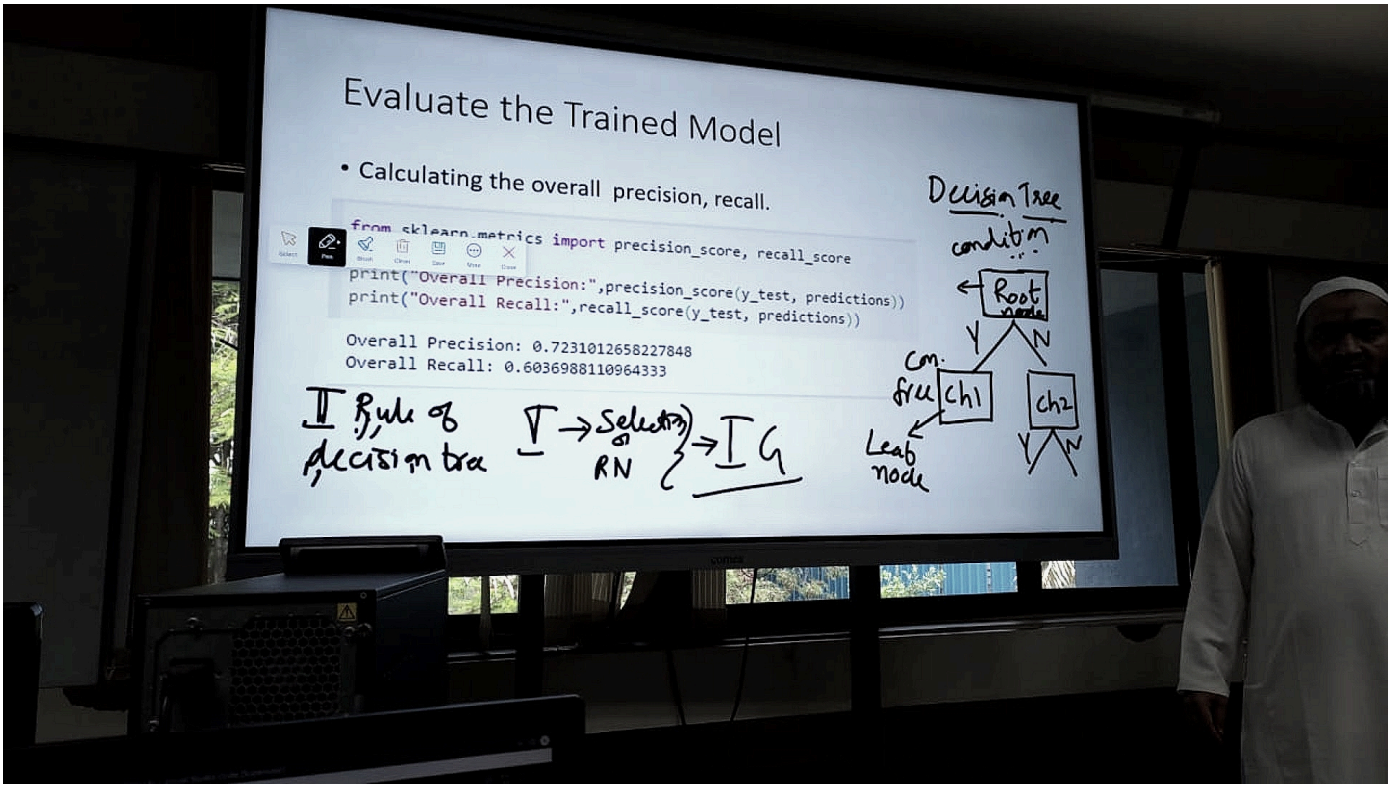
- Calculating the classification report to determine precision, recall F1-score etc..

```
from sklearn. metrics import classification_report

print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           0       0.81      0.88      0.85      2986
           1       0.72      0.60      0.66      1514

    accuracy                           0.79      4500
   macro avg       0.77      0.74      0.75      4500
weighted avg       0.78      0.79      0.78      4500
```


In
Ask
S.N
A1
A2
EcD C1(4,2)
ED C2(3,5)
1 2 5 X1(3.6) Y1(1) → C2
2 5 3 X2(4.4) Y2(2.8) → C1
3 3 1 X3(1.4) Y3(4) → C1
4 1 5 X4(4.2) Y4(2) → C2
5 1 4 X5(2) Y5(1.4) → C2
√(x1-x2)²+(y1-y2)²
X1
(2,5) (4,2)
x1 y1 x2 y2
= √2²+3² = 3.6
Y1
(2,5) (3.5)
Y = √1²+0 = 1
X2
(5,3) (4,2)
Y2
(5,3) (3,5)
X3
(3,1)
(4,2)
Y3
(3,1) (3,5)
X4 Y4
(1,5) (1,5)
C1 C2
X5 Y5
(4,4) (4,4)
C1 C2

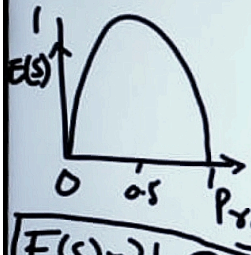

C1
C2
(5,3)
(3,1)
(2,5)
(1,5)
(4,4)
C1new = {5+3/2, 3+1/2}
= (4,2)
C2new = {2+1+4/3, 5+5+4/3}
C2new = {2.33, 4.66}


Evaluate the Trained Model
• Calculating the overall precision, recall.
from sklearn.metrics import precision_score, recall_score
print("Overall Precision:",precision_score(y_test, predictions))
print("Overall Recall:",recall_score(y_test, predictions))
Overall Precision: 0.7231012658227848
Overall Recall: 0.6036988110964333
Decision Tree
condition
Root node
Y
N
ch1
ch2
Leaf node
Rule of decision tree
Select RN
IG

# Example

- A researcher is trying to identify the root node to design a decision tree classifier to classify the students based on scores greater than or equal to 50% which is a pass and less than 50% which is a fail. The data is given in the below table.

| Student ID | No. of Assignments Completed | Number of Hours Studied | Attendance in % | Marks out of 100 | Target C |
|---|---|---|---|---|---|
| 1 | Less than 5 | Less than 3 hours | Less than 70% | 30 | F |
| 2 | Less than 5 | Less than 3 hours | Less than 70% | 45 | F |
| 3 | Less than 5 | More than 3 hours | Less than 70% | 50 | P |
| 4 | More than 5 | More than 3 hours | Less than 70% | 70 | P |
| 5 | More than 5 | More than 3 hours | More than 70% | 75 | P |
| 6 | More than 5 | More than 3 hours | More than 70% | 85 | P |




E(S)=1 @ P=0.5
0 @ P=0 & 1

$$E(S) = -\sum_{i=1}^{c} P_i \log P_i$$ , c = no. of classes