*ELG5255 Applied Machine Learning*

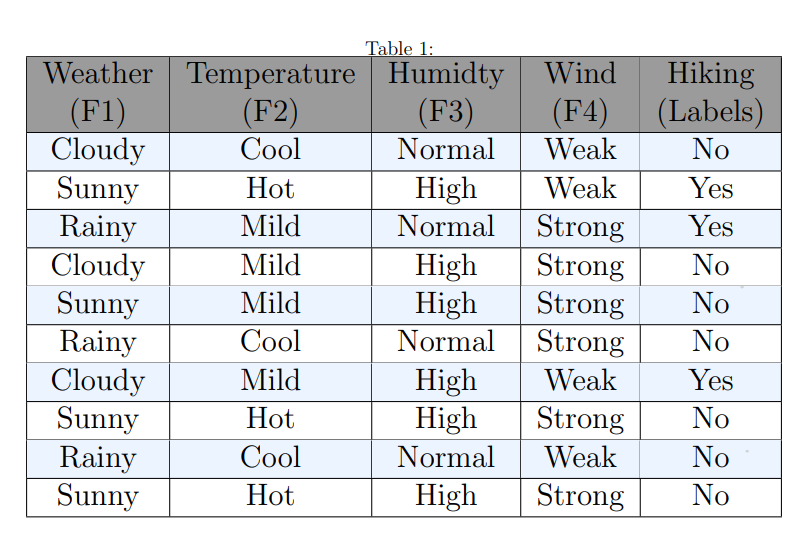
*Group Assignment #4*

*Group 4:*

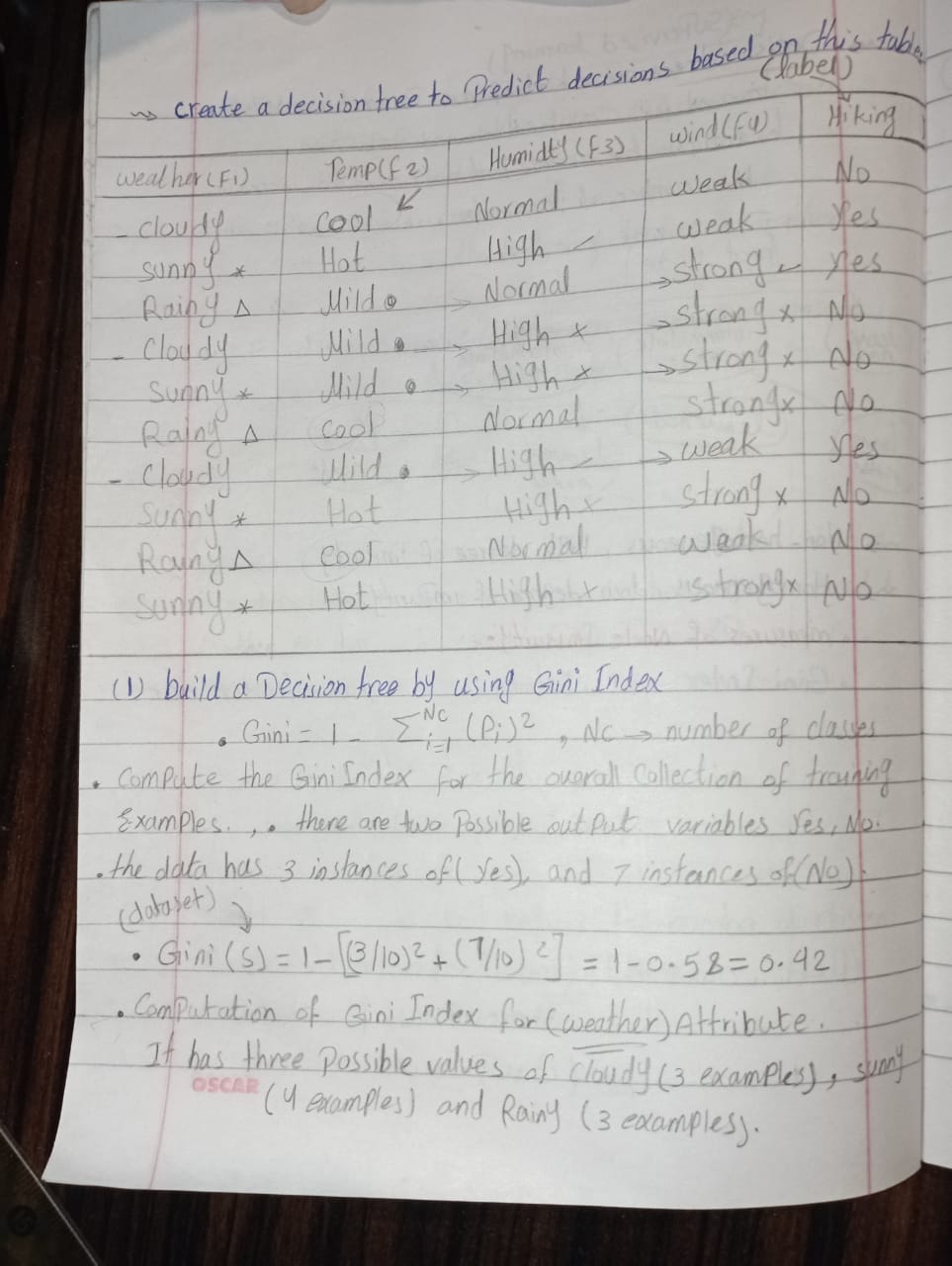
1. Yomna Mohamed Sayed Ahmed ID: 300327217
2. Khaled Mohamed Mohamed Mahmoud ID: 300327242
3. Marwa Hamdi Boraie Mahmoud ID: 300327267

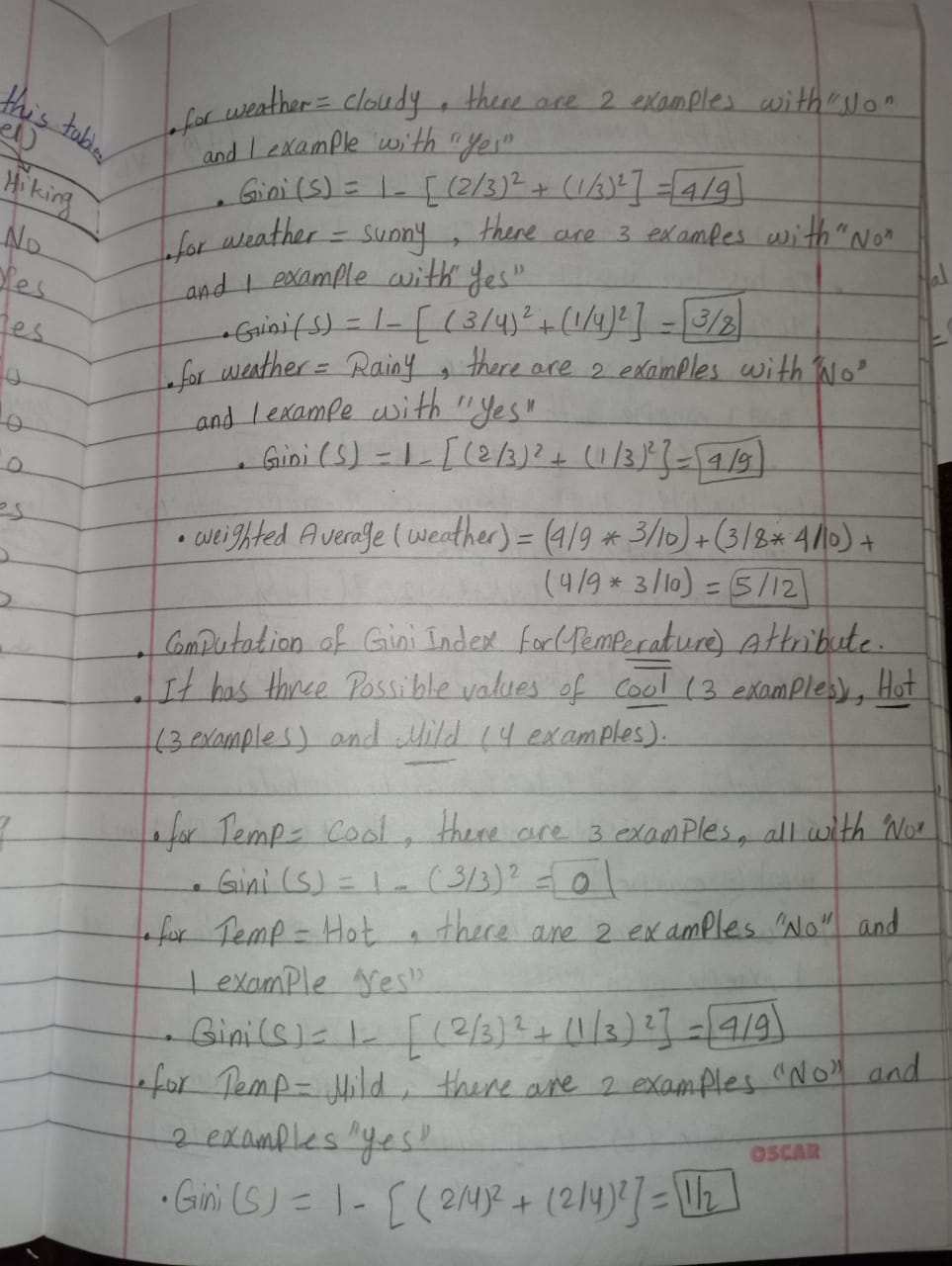
**Part 1:** *Numerical Questions*

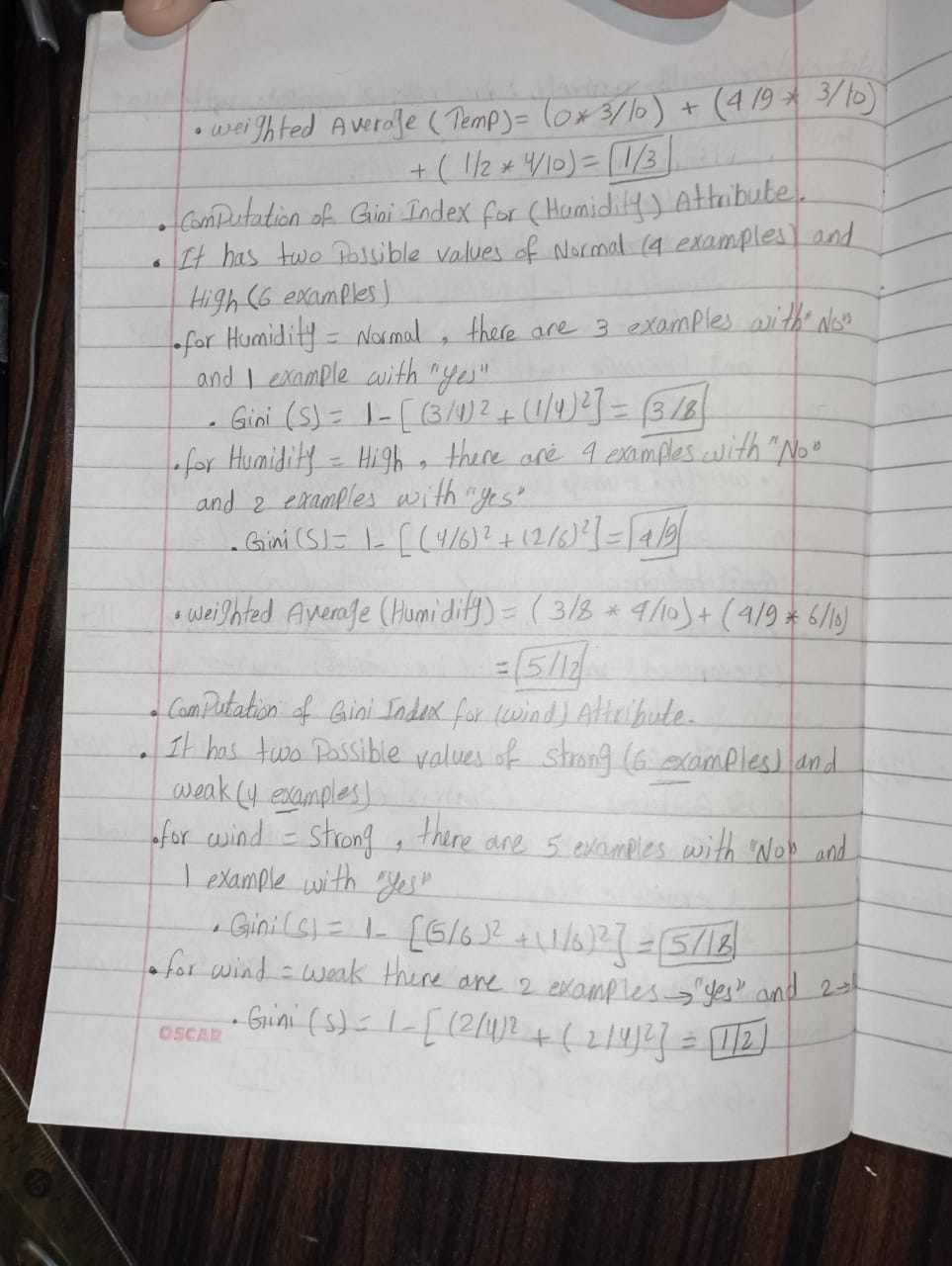
🡪Let’s assume that TAs would go hiking every weekend, and we would make final decisions (i.e., Yes/No) according to weather, temperature, humidity, and wind. Please create a decision tree to predict our decisions based on Table 1.

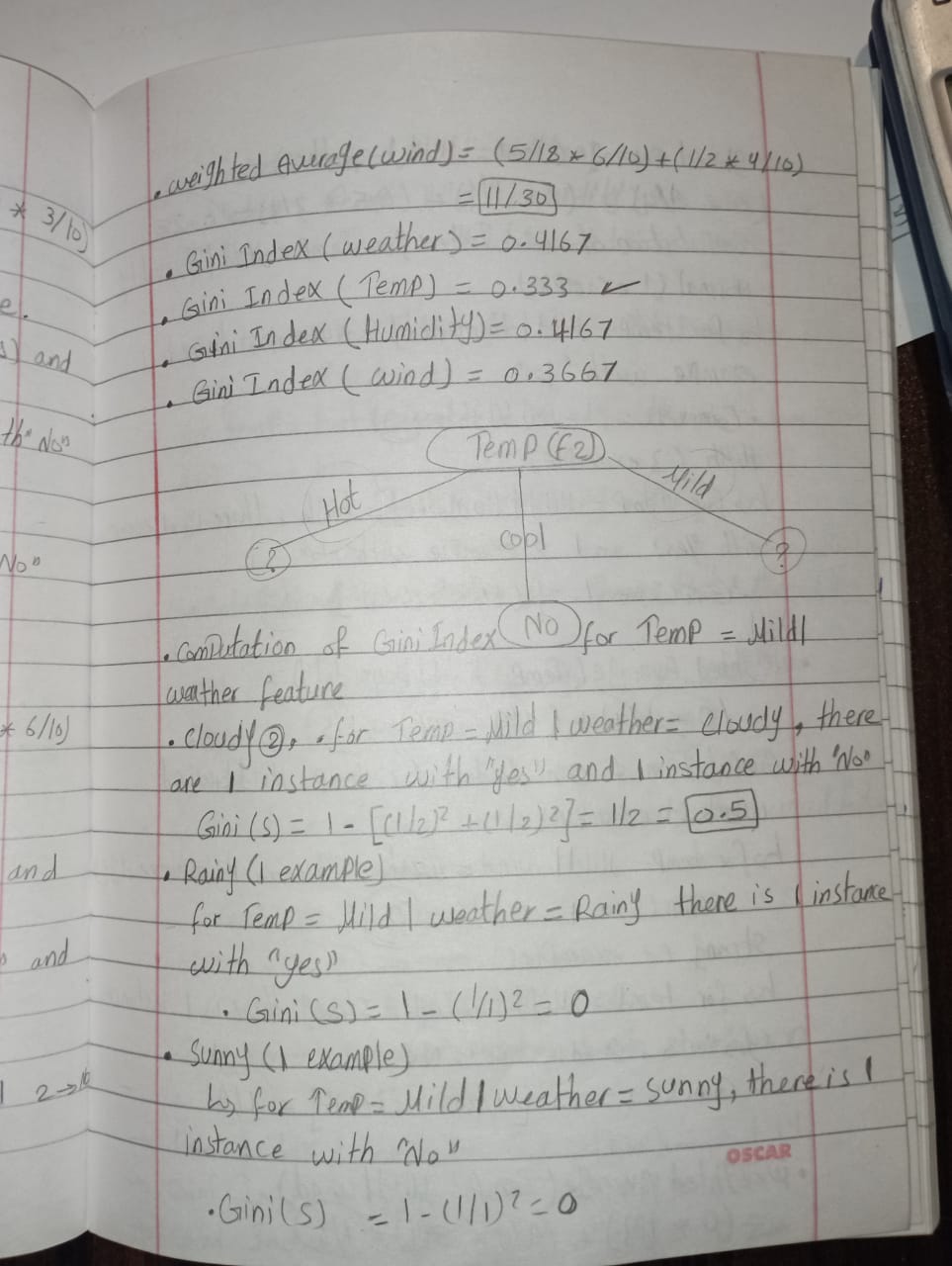
****

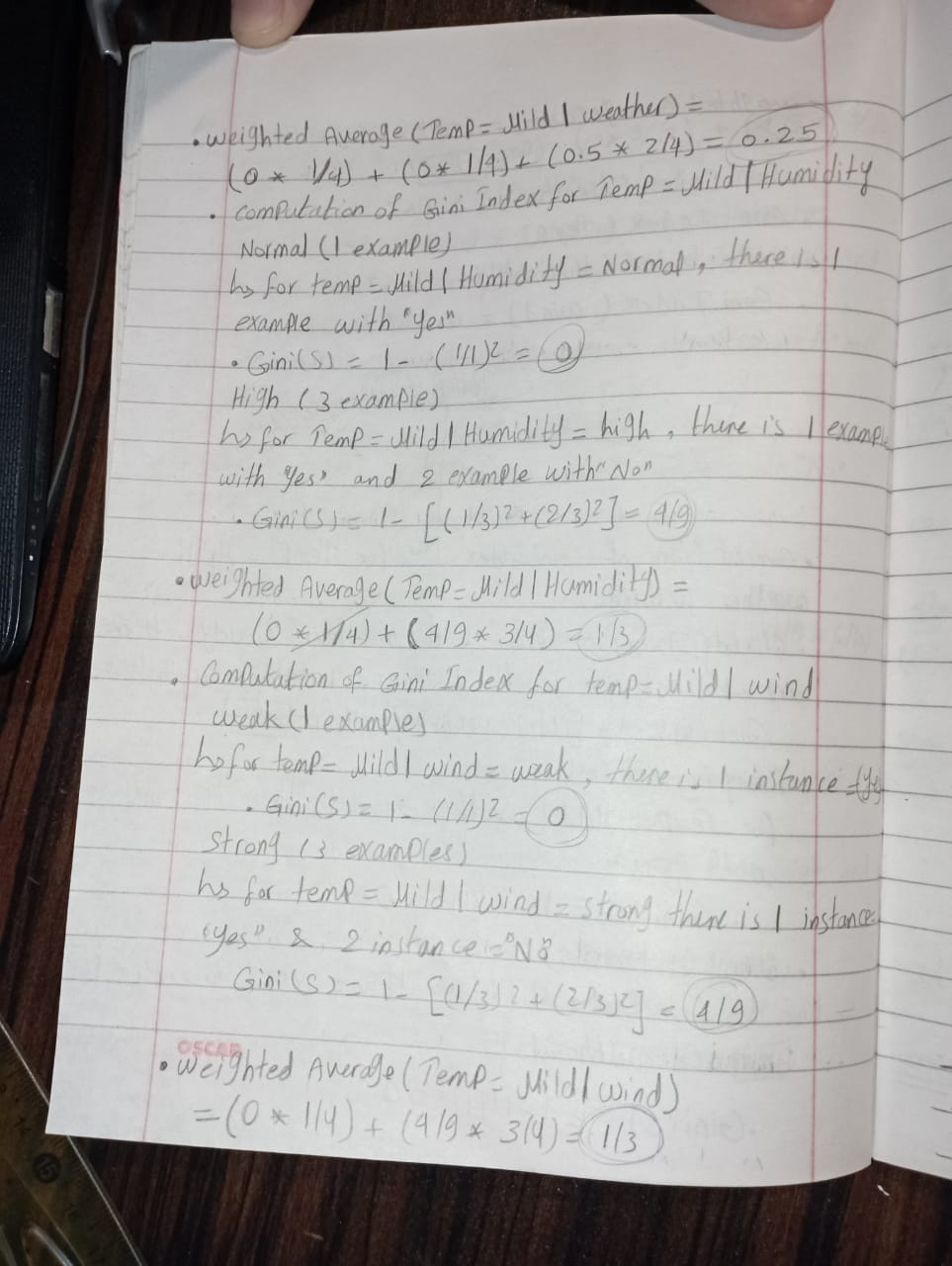
1. Please build a decision tree by using **Gini Index** (i.e., Gini = 1- , where NC is the number of classes).

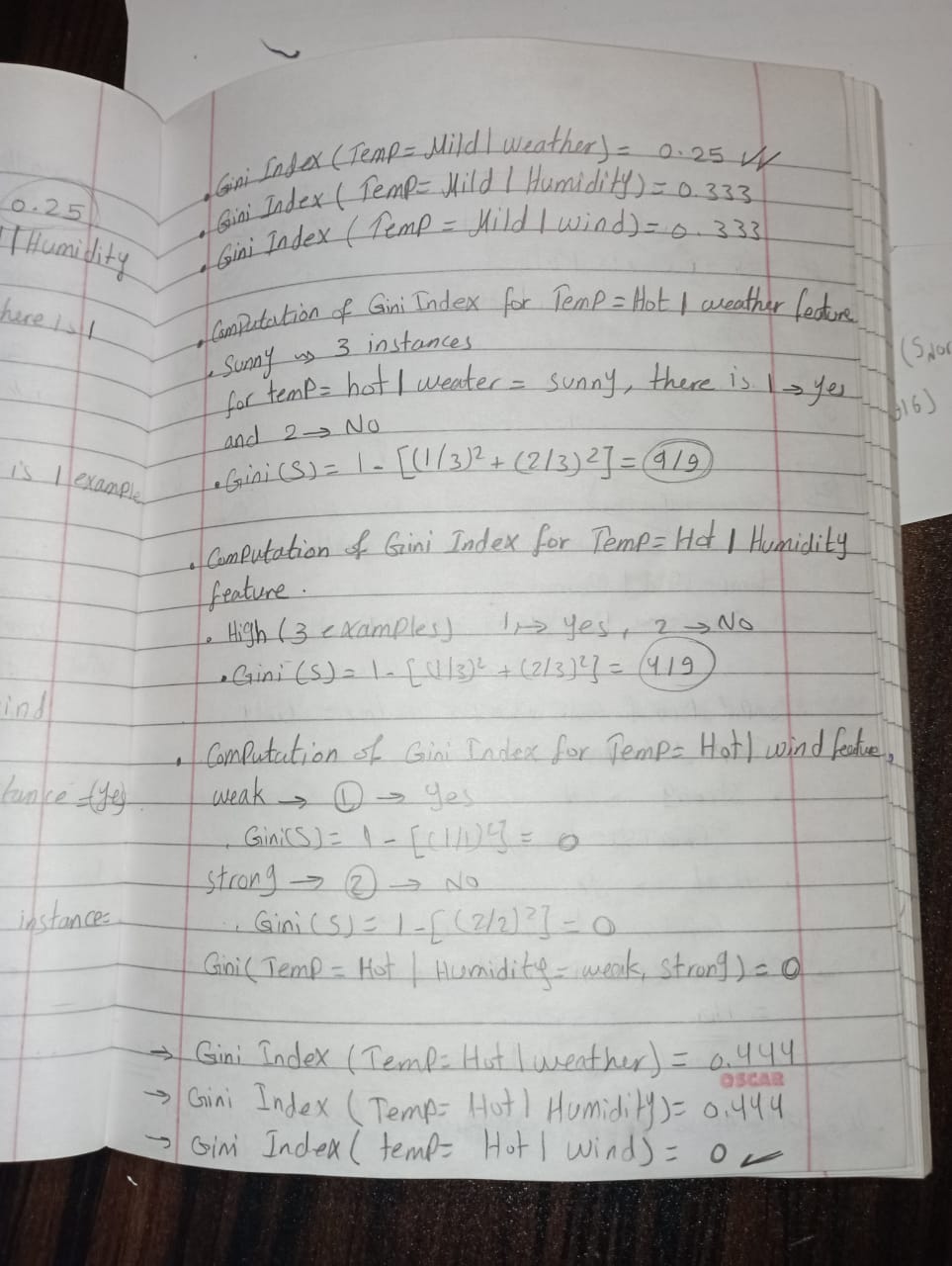


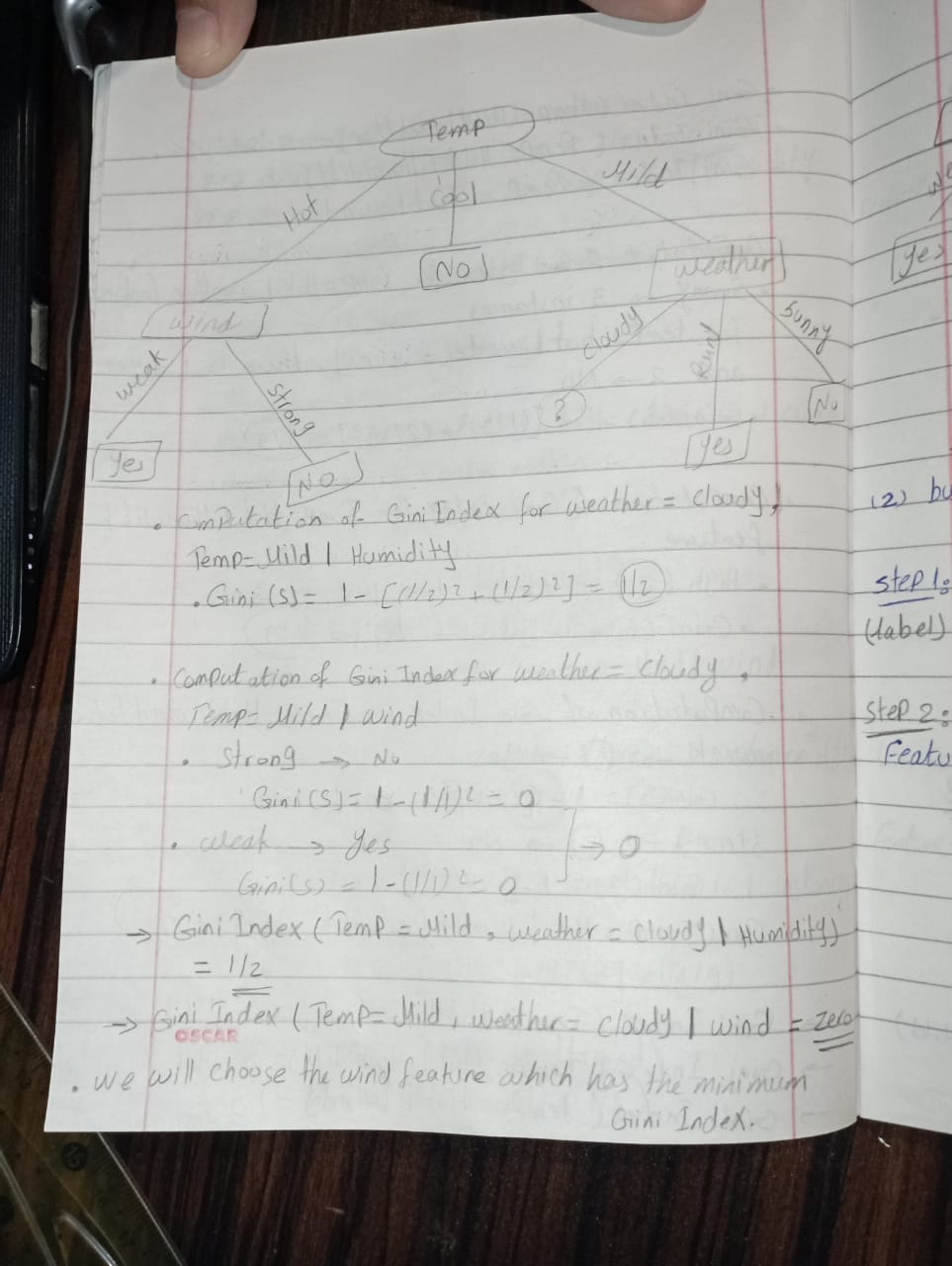


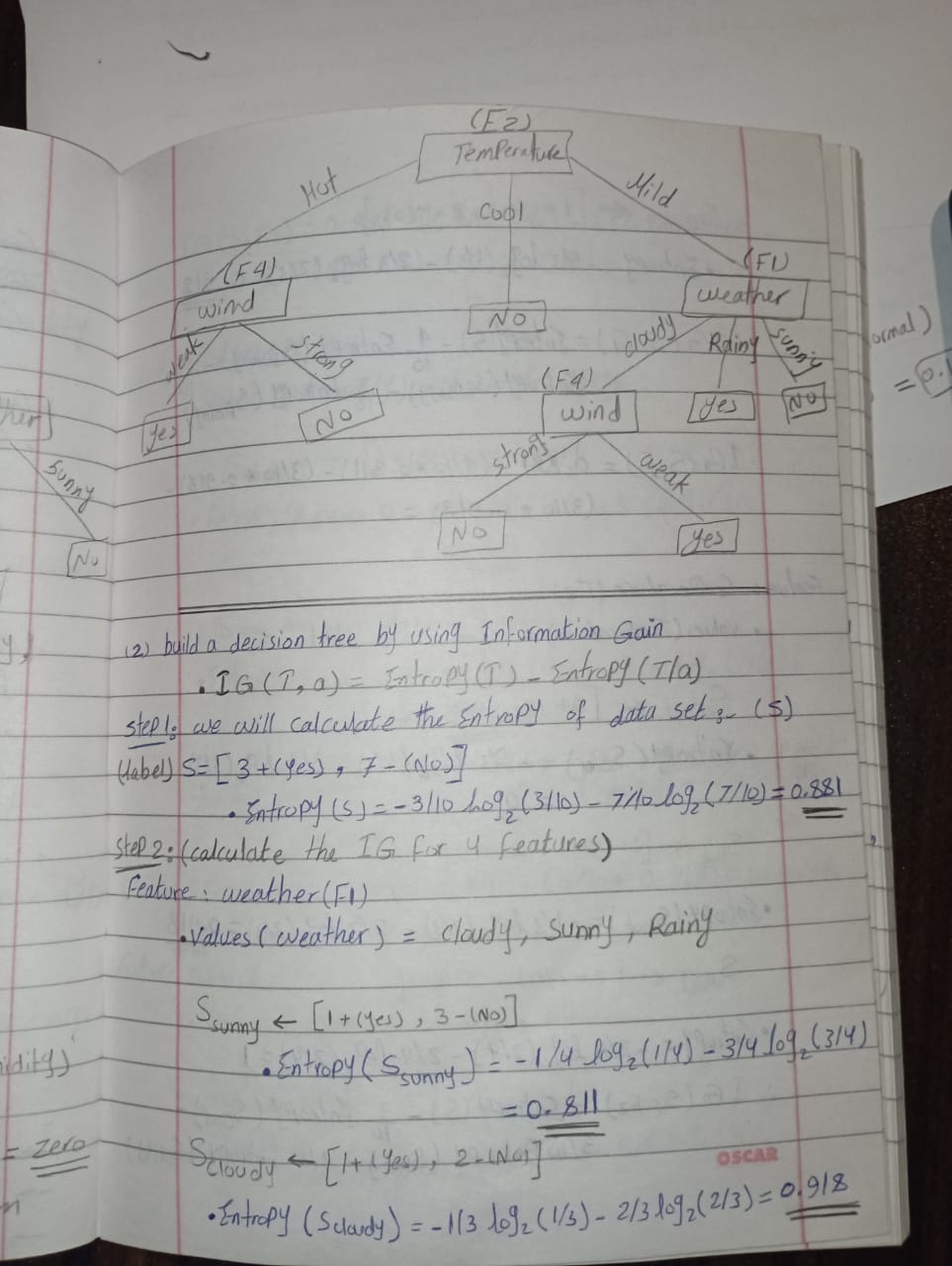




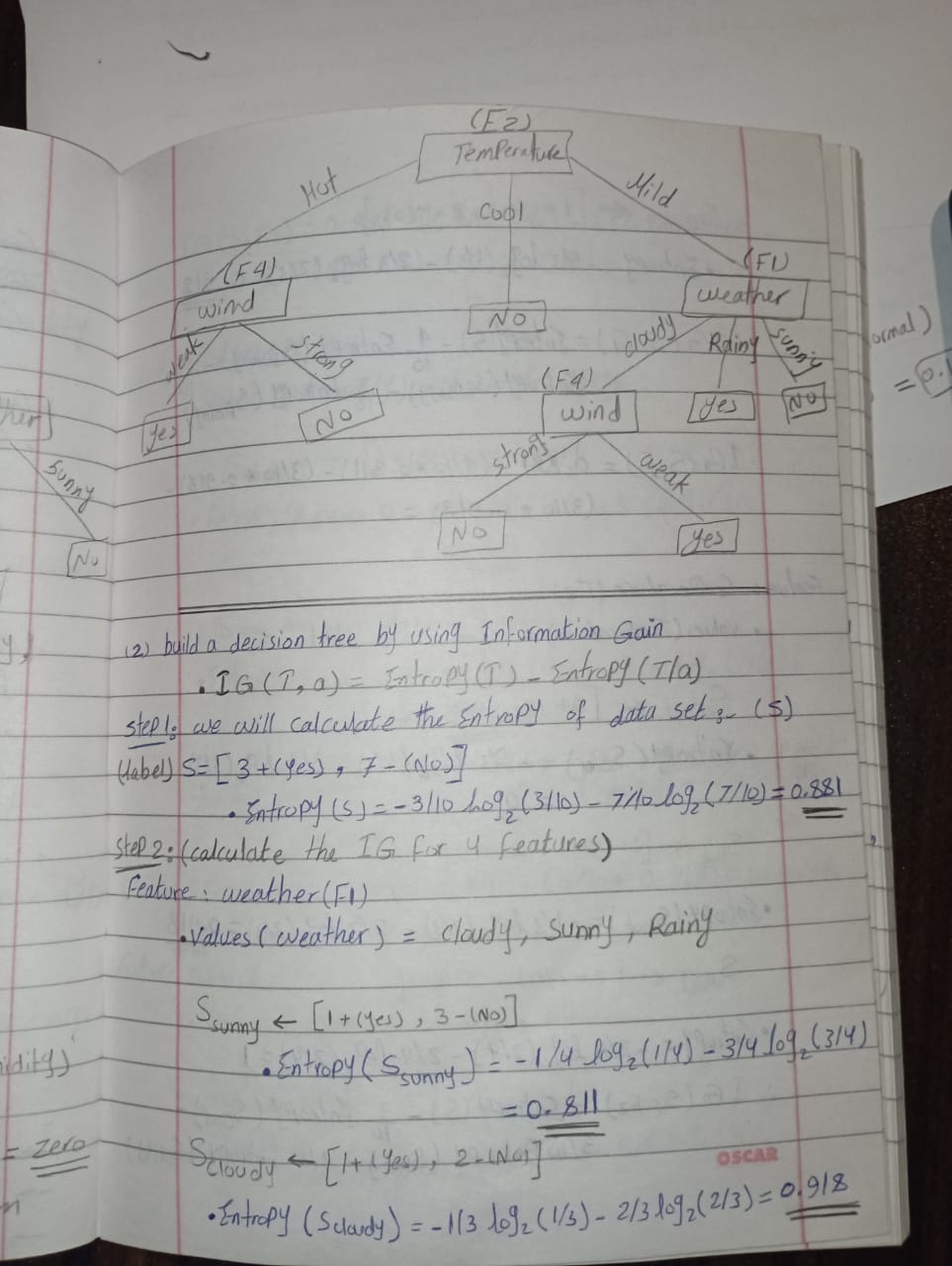


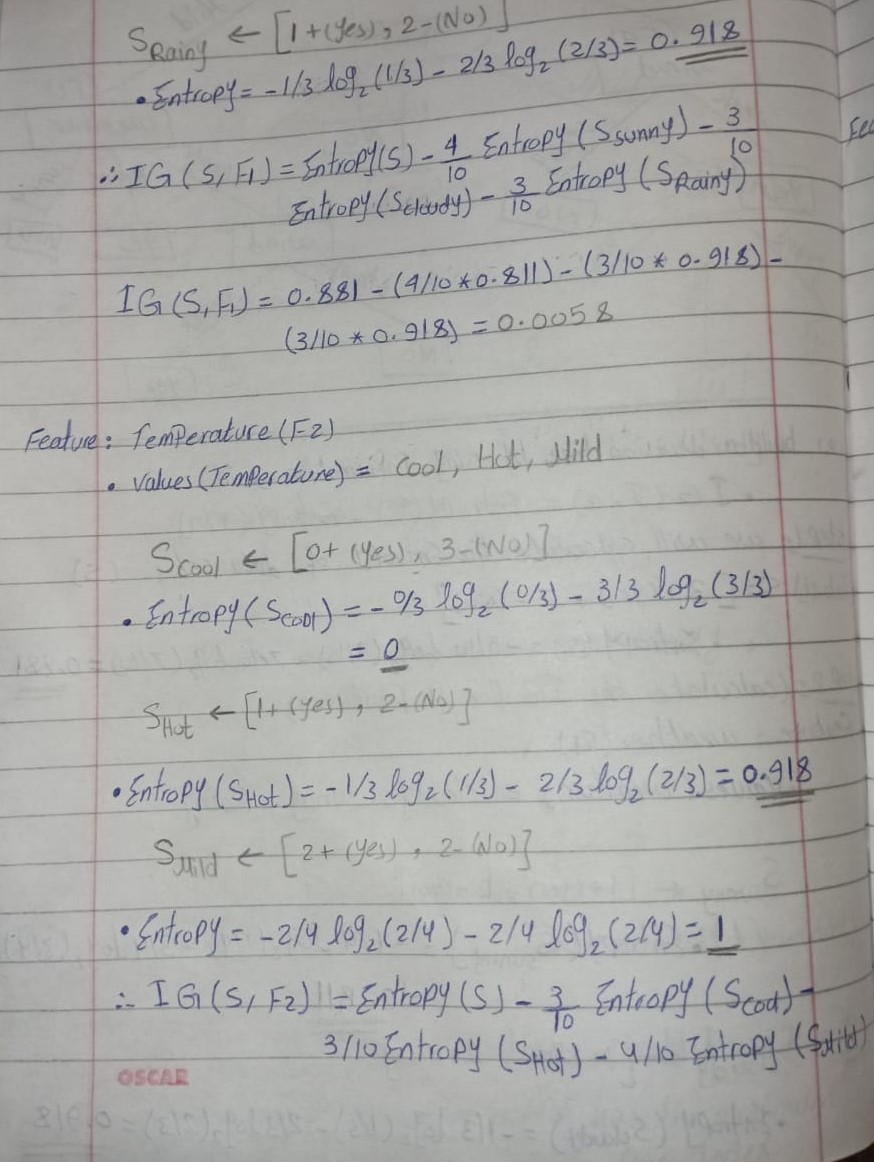


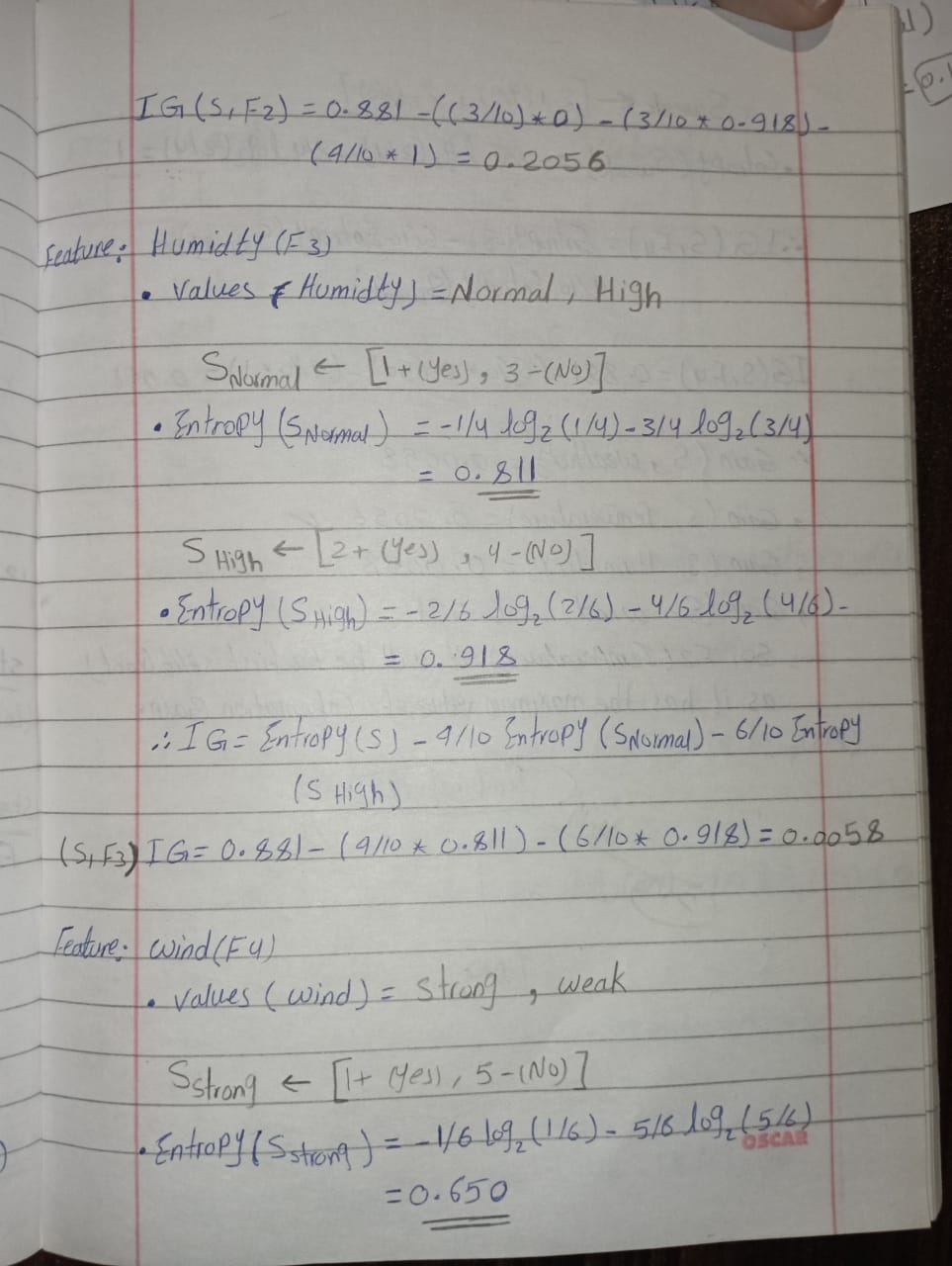


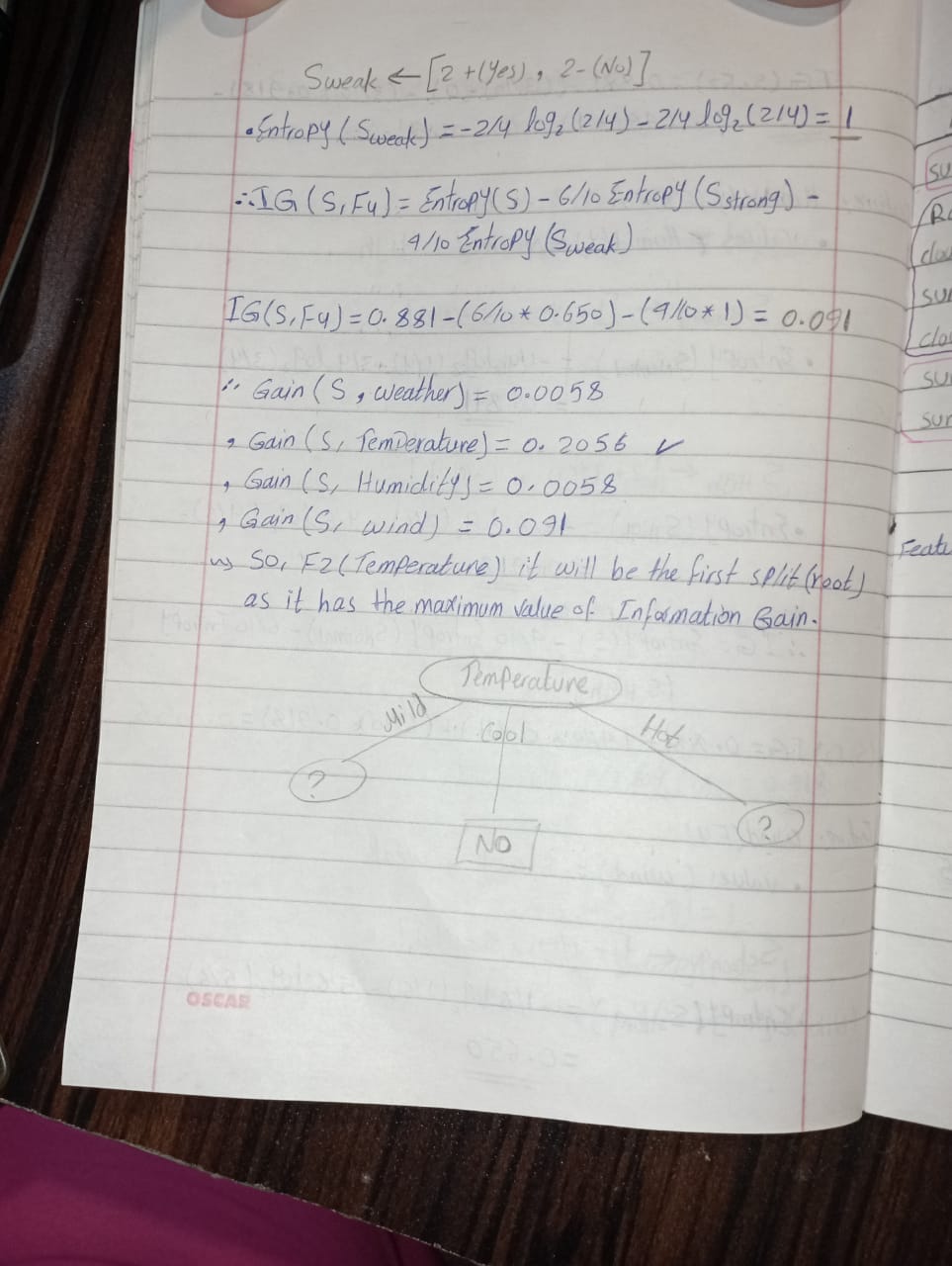


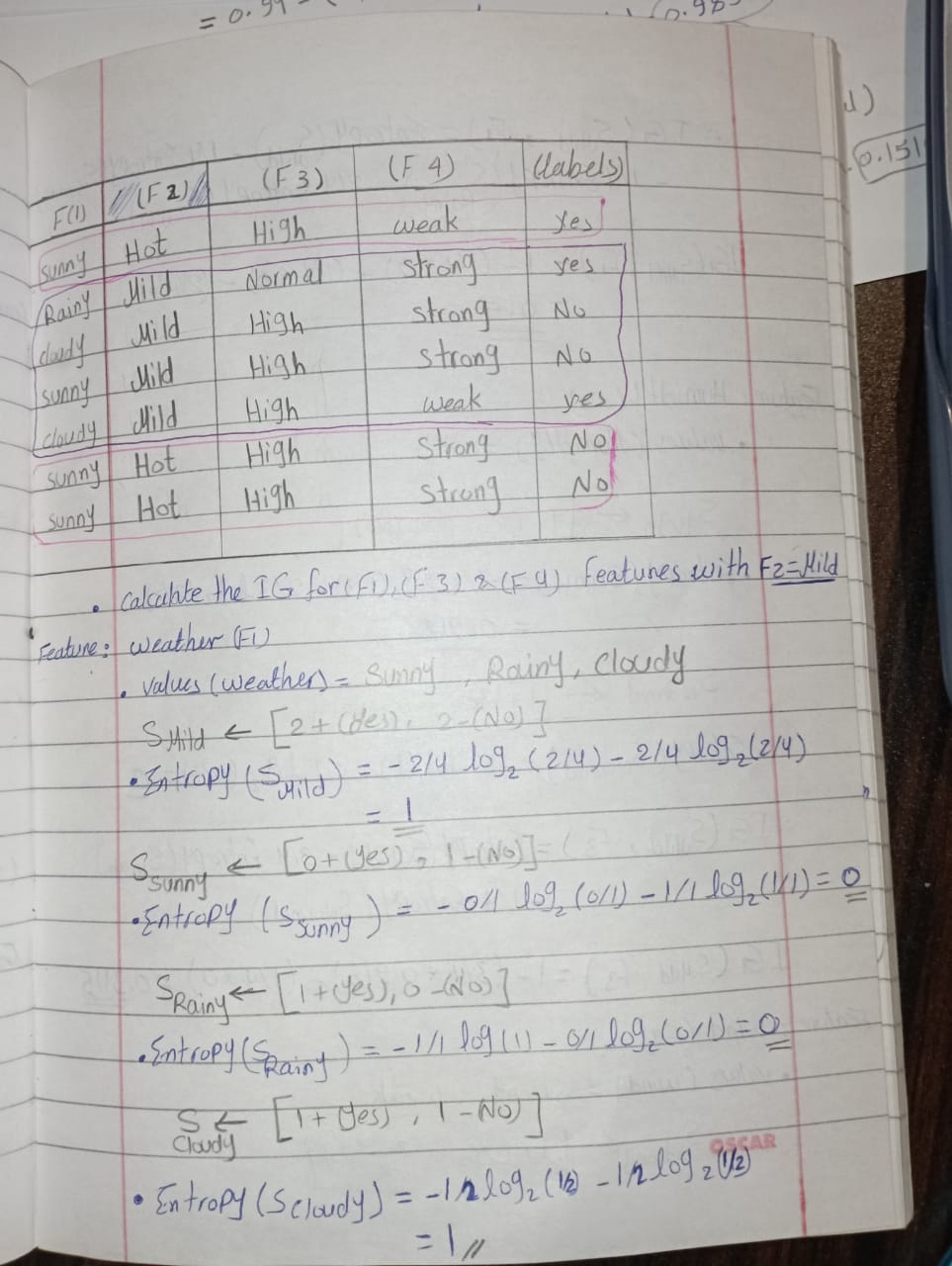
1. Please build a decision tree by using **Information Gain** (i.e., IG(T, a) = Entropy(T)− Entropy(T|a).

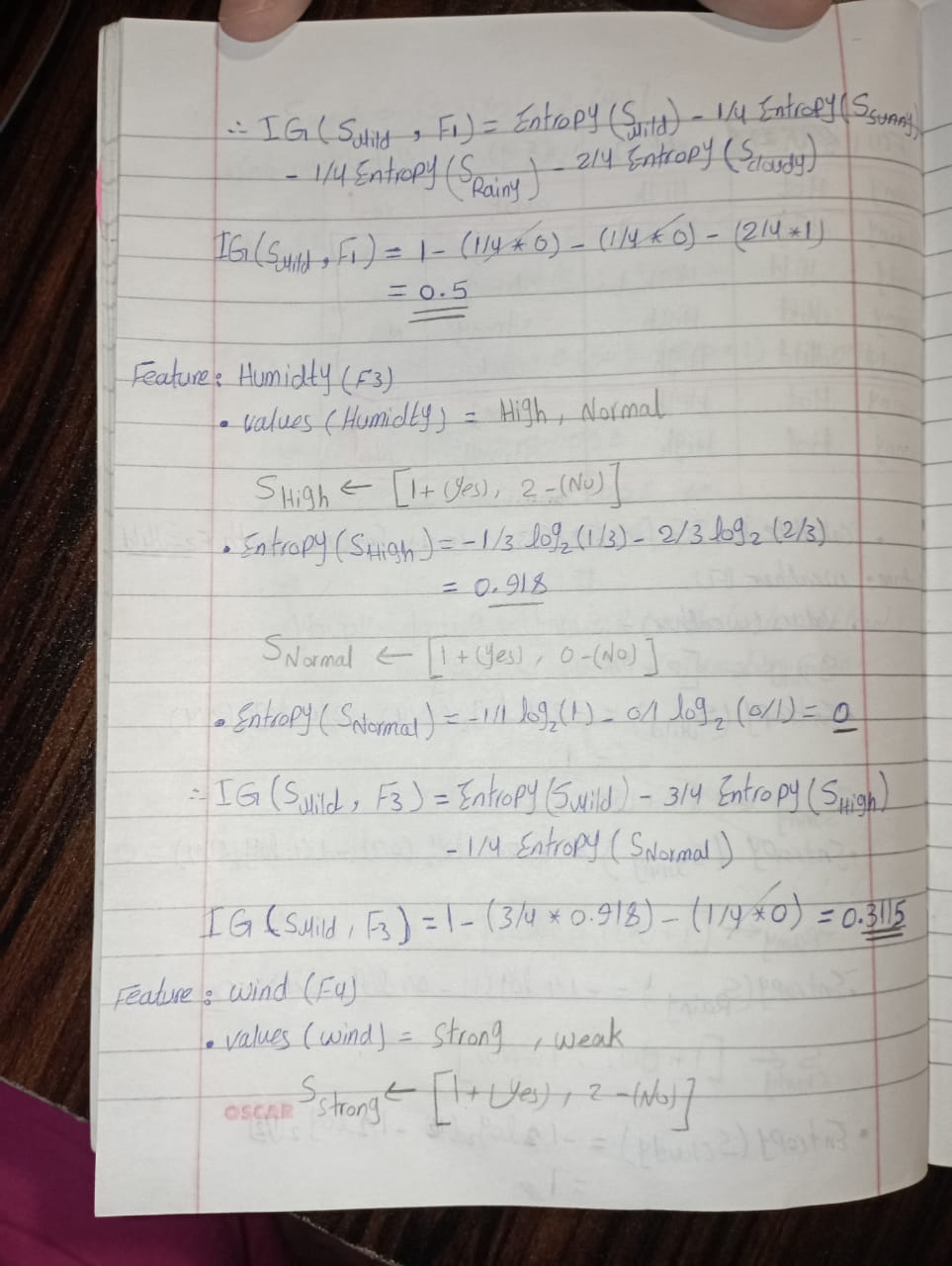


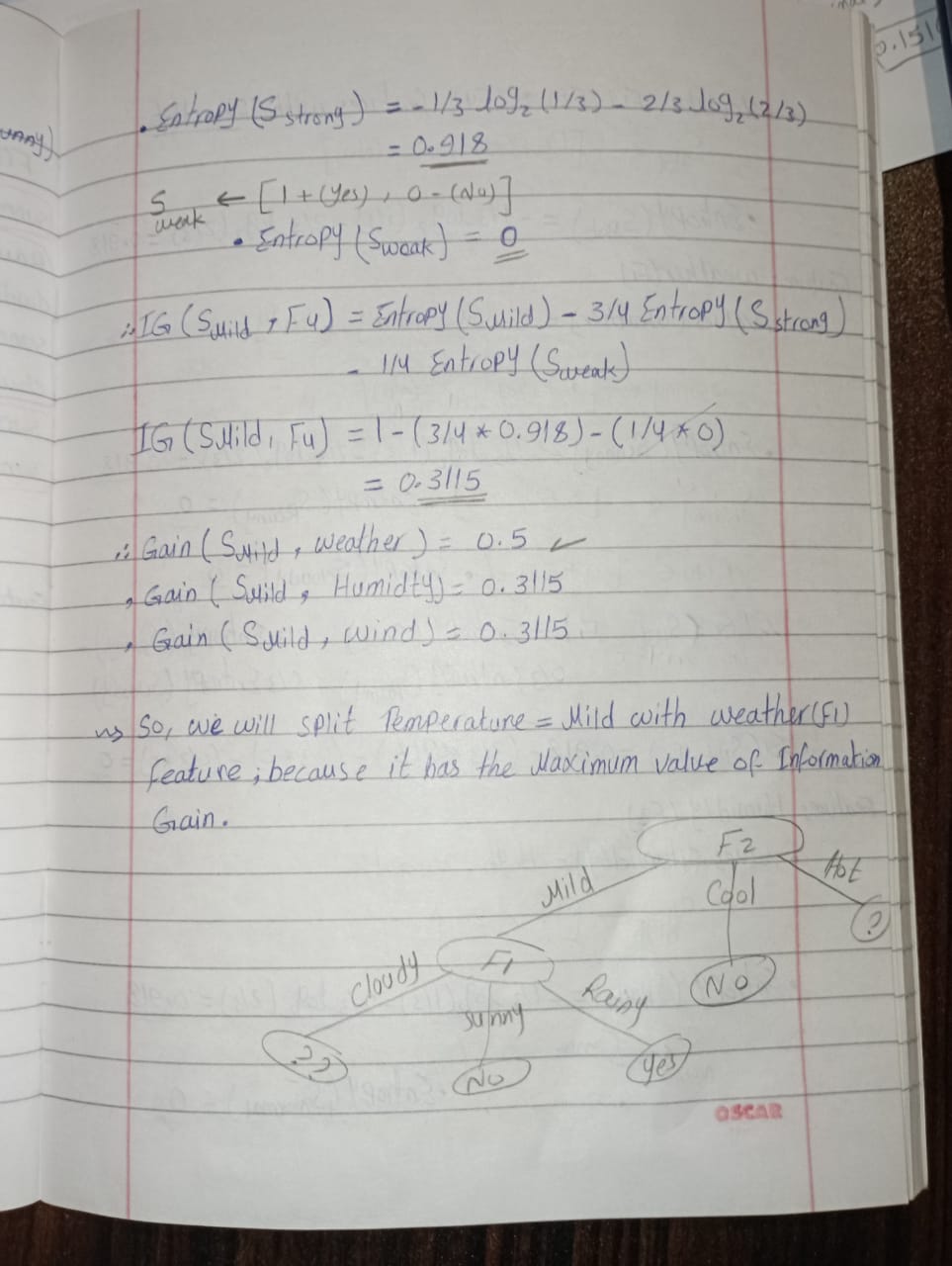


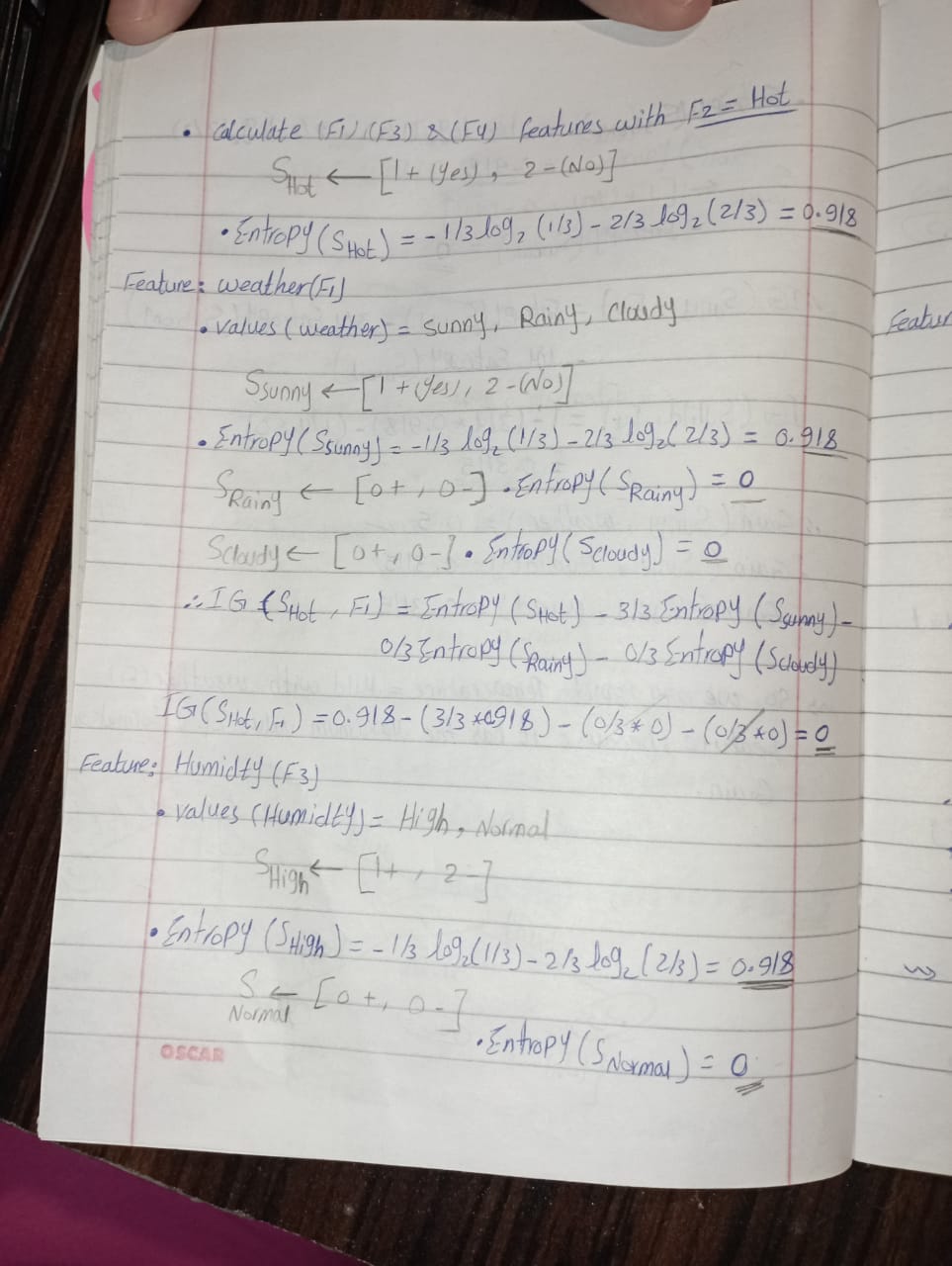


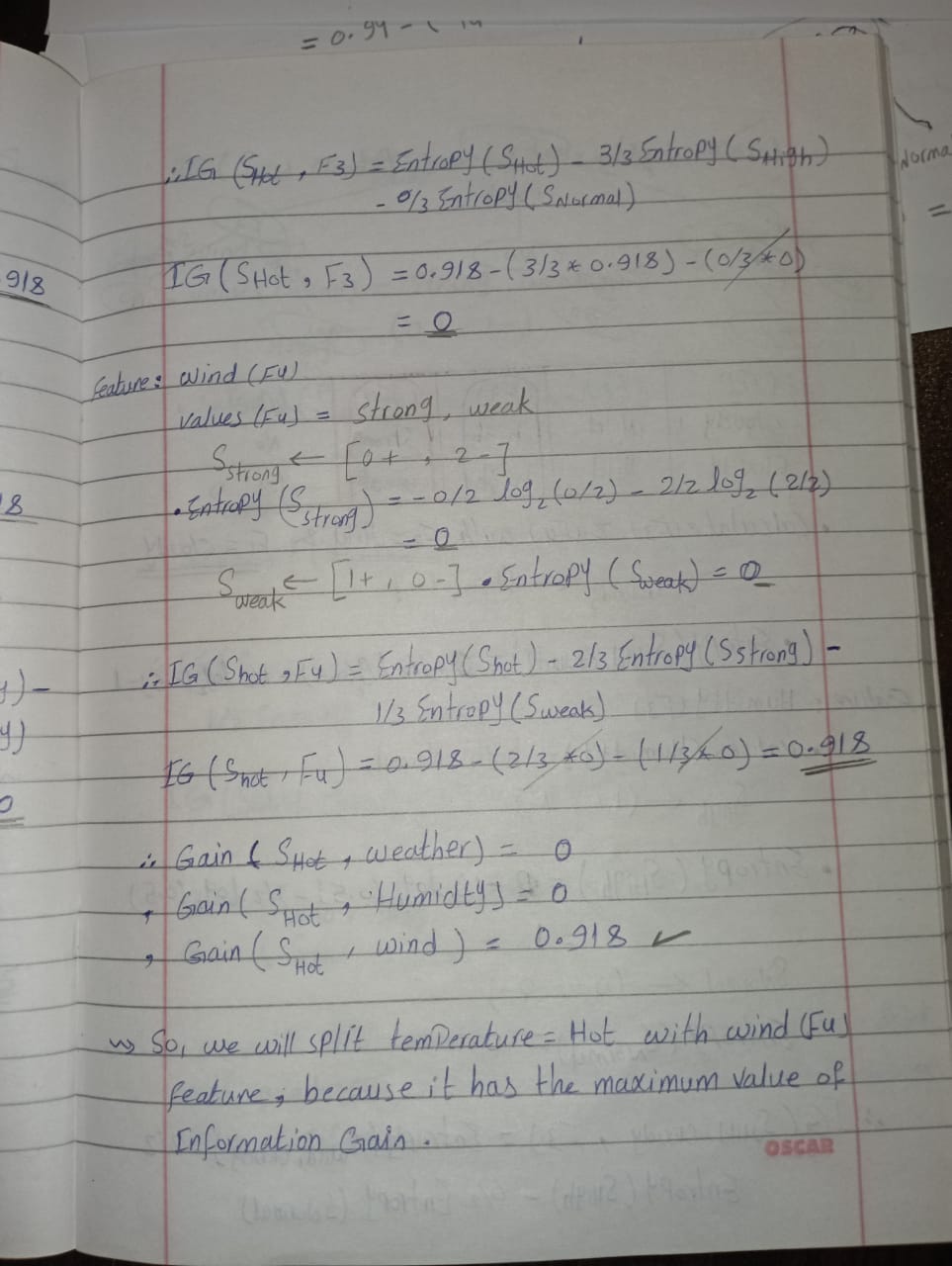


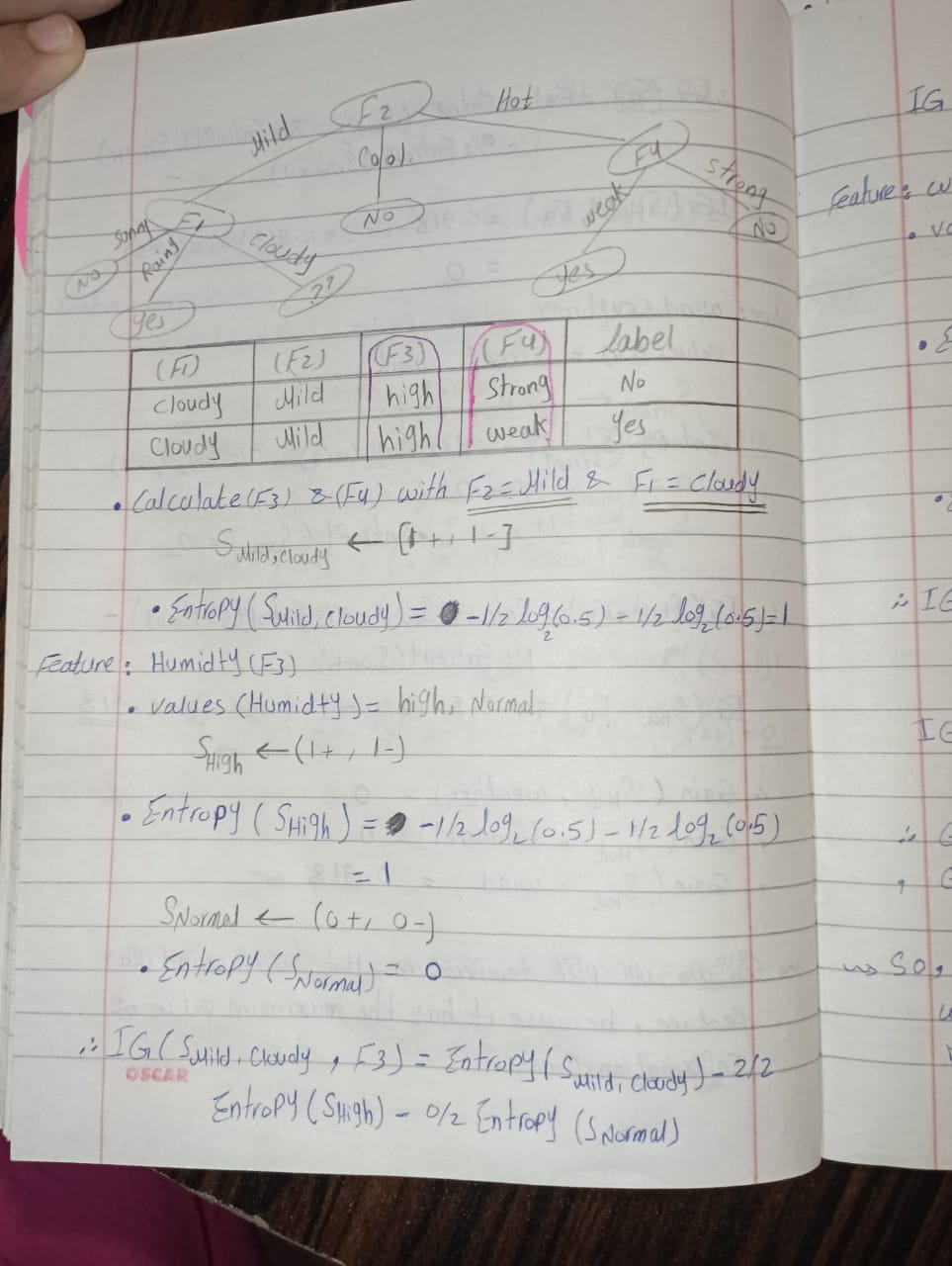


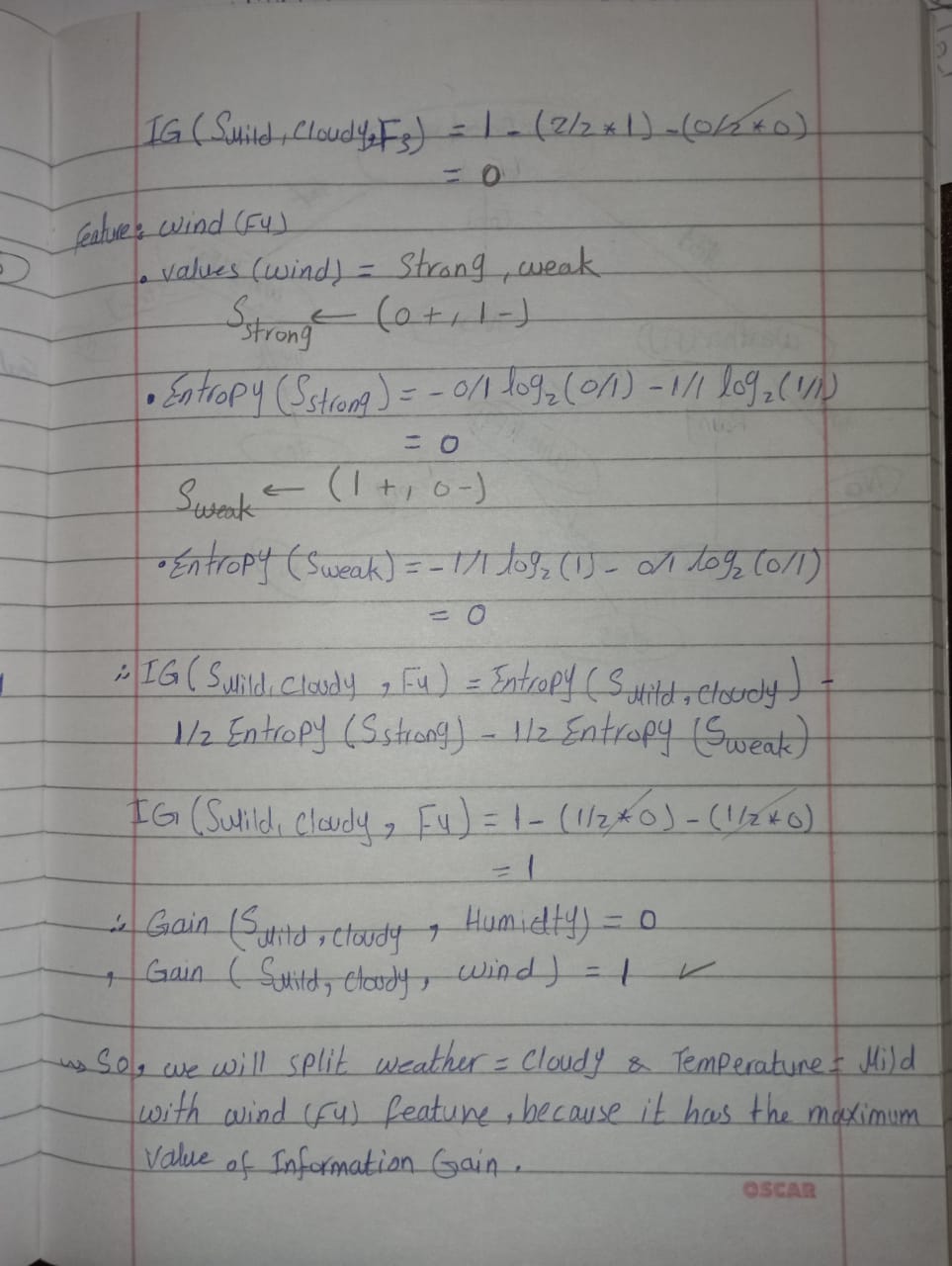


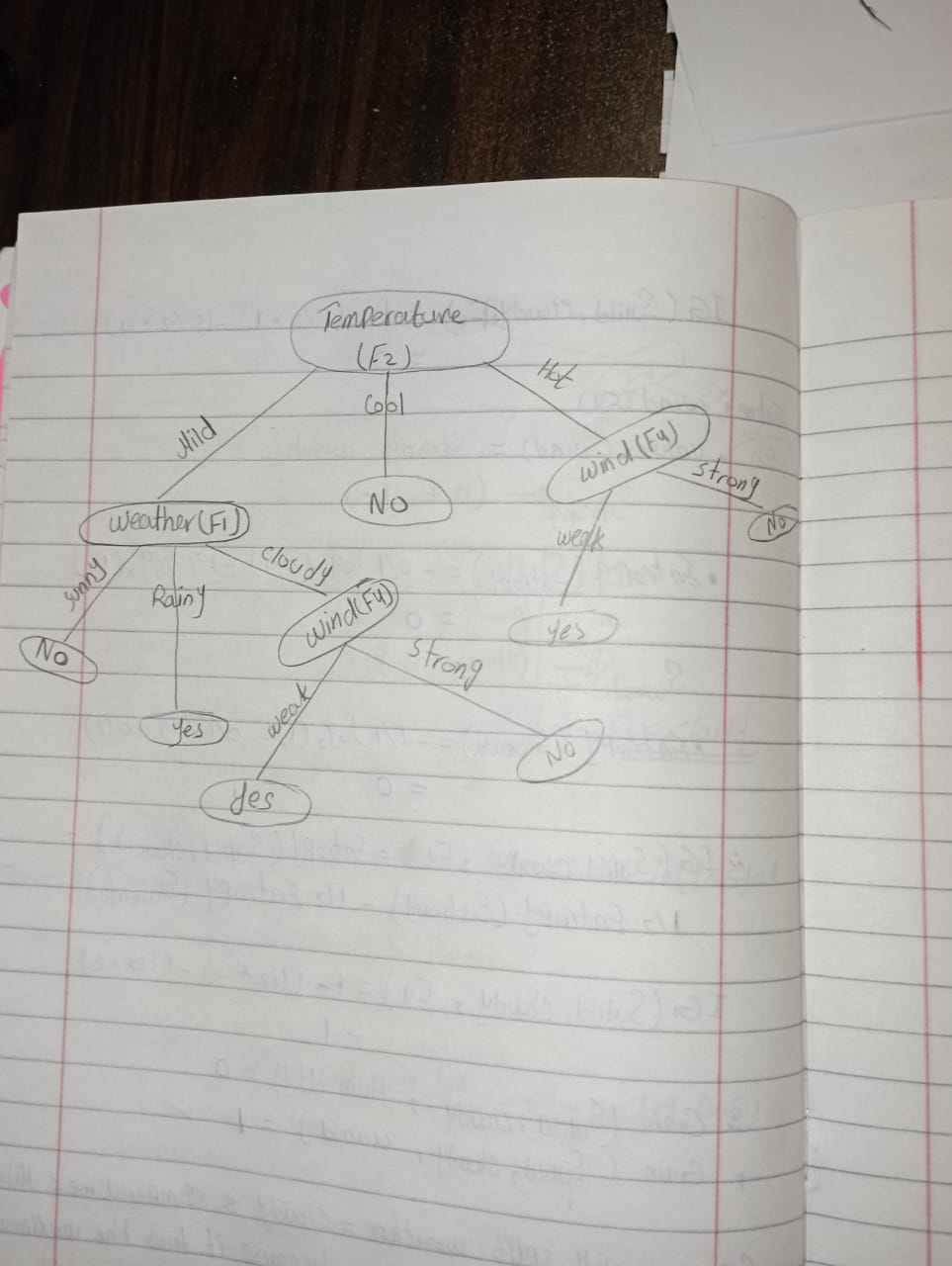










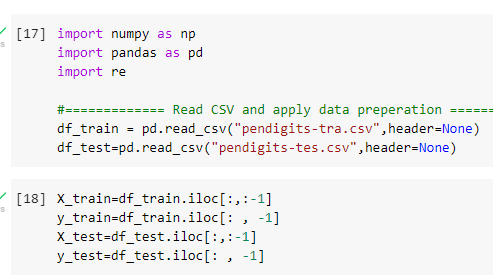


1. Please compare the advantages and disadvantages between Gini Index and Information Gain.

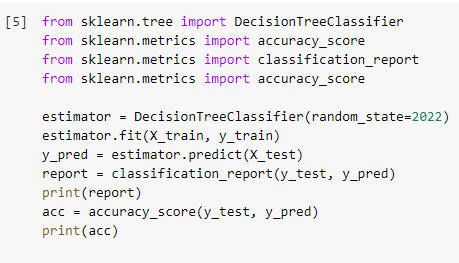
|  |  |  |
| --- | --- | --- |
|  | Advantages | Disadvantages |
| Gini Index | • favors larger partitions (distributions) and is very easy to implement and interpret.  • Modification of the information gain that reduces its bias.  • It deals with inequality, so it can judge the distribution pattern better. | • Sample Bias, the validity of the Gini index can be dependent on sample size.  • Data Inaccuracy, the Gini index is sometimes prone to random and systematic data errors; it can create problems with the index value.  • Degeneracy, in some exceptional cases, the Gini index value can be the same for different distributions. |
| Information Gain | • Information gain ratio biases the decision tree against considering attributes with a large number of distinct values. So, it solves the drawback of information gain namely, information gain applied to attributes that can take on a large number of distinct values might learn the training set too well.  • It creates a comprehensive analysis of consequences along each branch and identifies decision nodes that need further analysis. | • A notable problem occurs when information gain is applied to attributes that can take on a large number of distinct values (biased).  • Subsets are more likely to be pure if there are a large number of values (overfitting). |

**Part 2:** *Programming Questions*

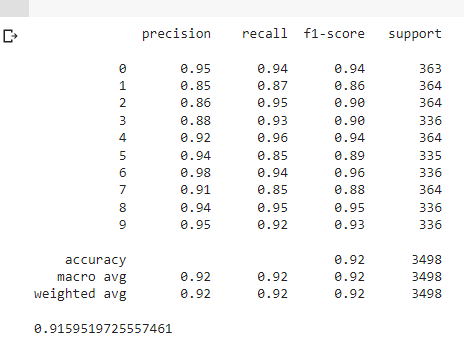
*2-*

*Reading the* Pen-Digits dataset 

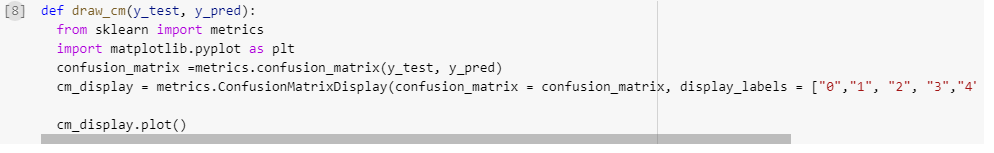
Apply decision tree to classify testing set



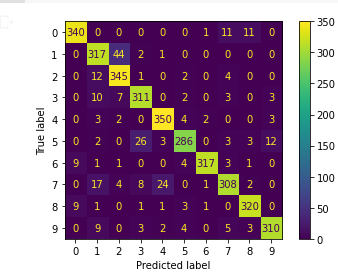
And this is the classification report



This function draw a confusion matrix

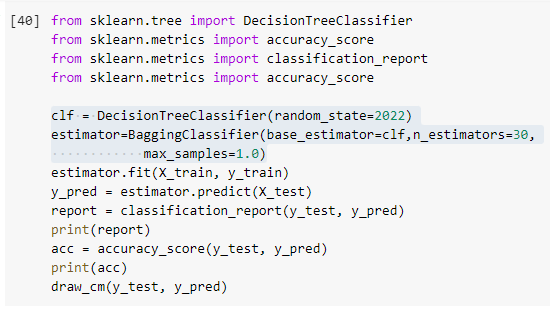


And this is the confusion matrix

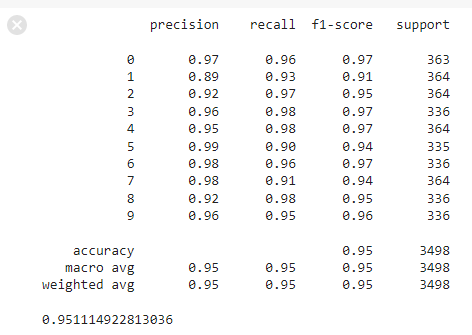


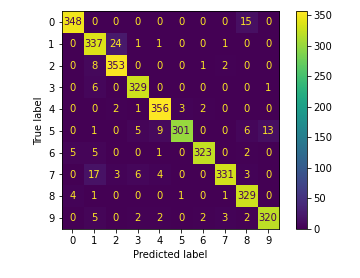
3-(a)

Bagging using Decision Tree

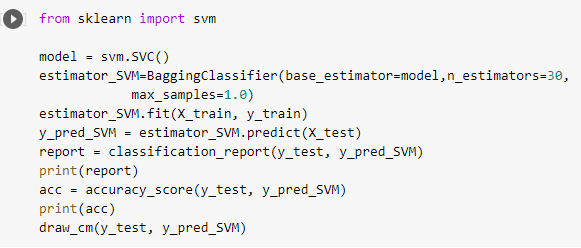


And this is the classification report for Decision Tree

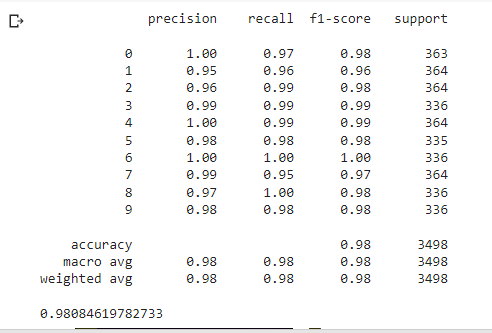
  
And this is the confusion matrix for Decision Tree



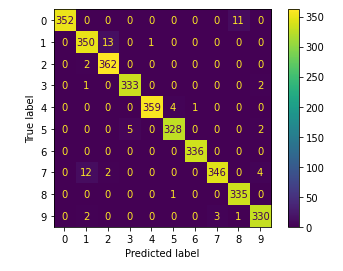
Bagging using SVM



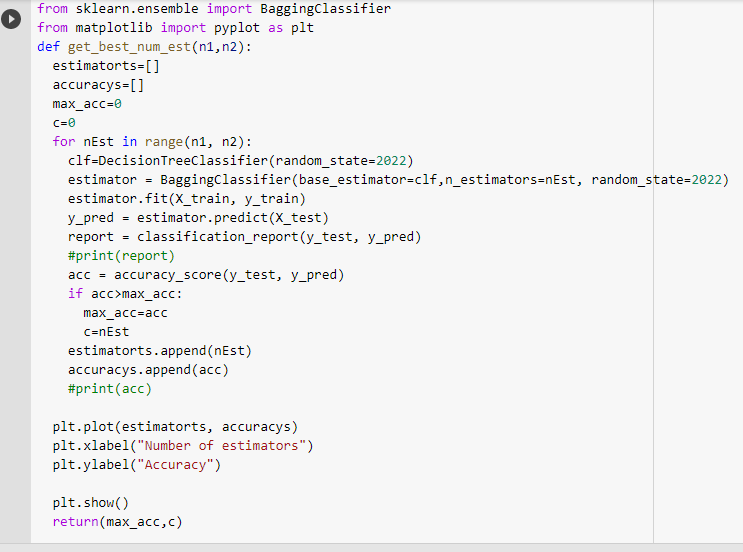
And this is the classification report for SVM



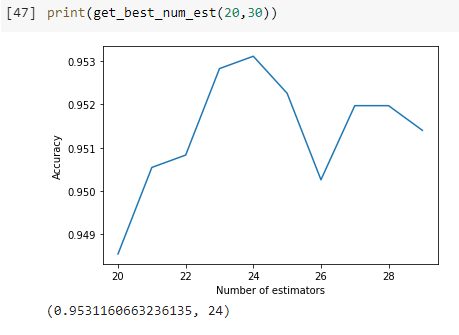
And this is the confusion matrix for SVM



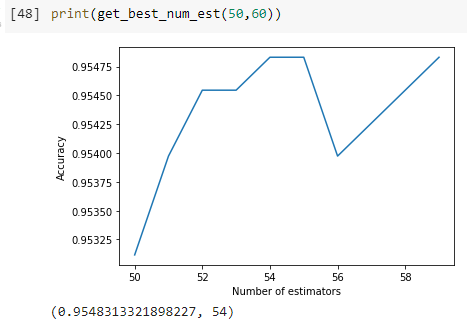
(b) Find the best number of estimators



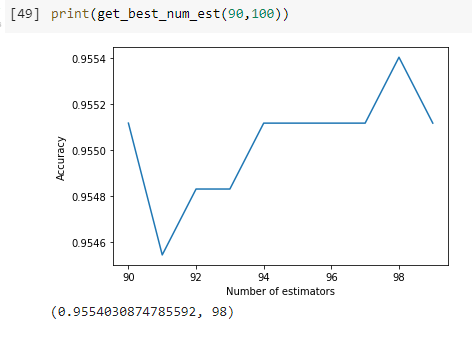
Get values [20,30] to the Estimators to find the best estimator that give us the highest accuracy



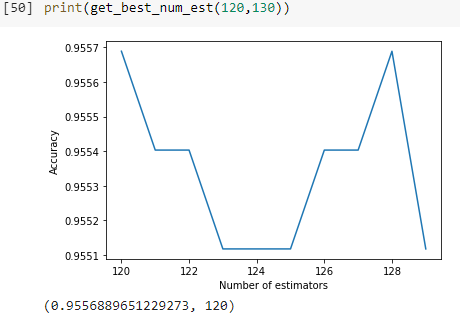
Get values [50,60] to the Estimators to find the best estimator that give us the highest accuracy



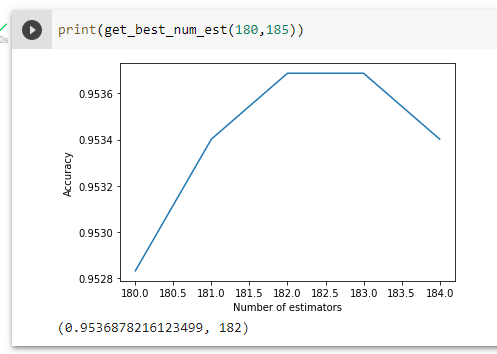
Get values [90,100] to the Estimators to find the best estimator that give us the highest accuracy



Get values [120,130] to the Estimators to find the best estimator that give us the highest accuracy



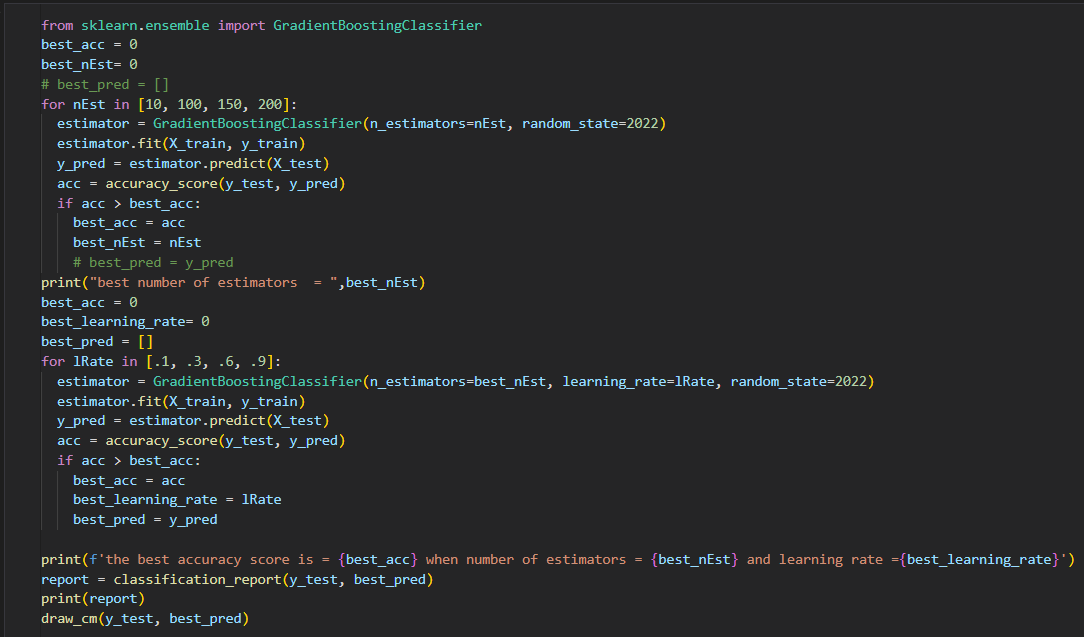
Get values [180,185] to the Estimators to find the best estimator that give us the highest accuracy

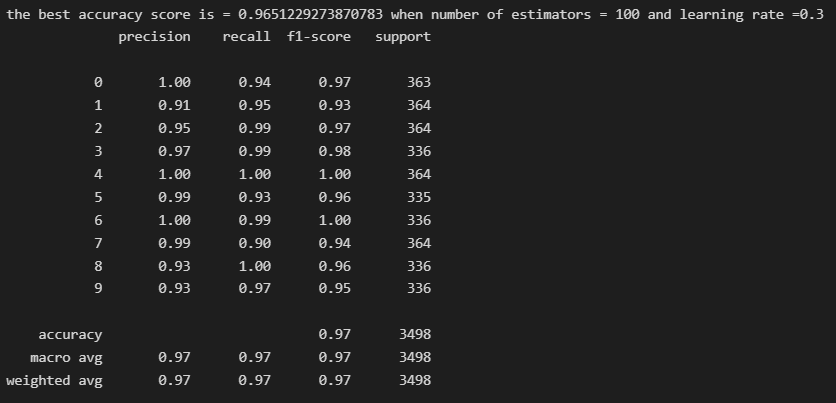


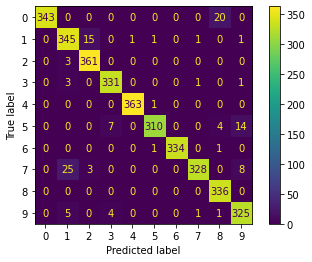
**4) Boosting:**

1. GradientBoosting classifier tuning number of estimator from [10, 100 , 150, 200] then the learning rate from [.1,.3,.6,.9]

We found the best accuracy = 0.965 when the number of estimators = 100 and the learning rate = 0.3

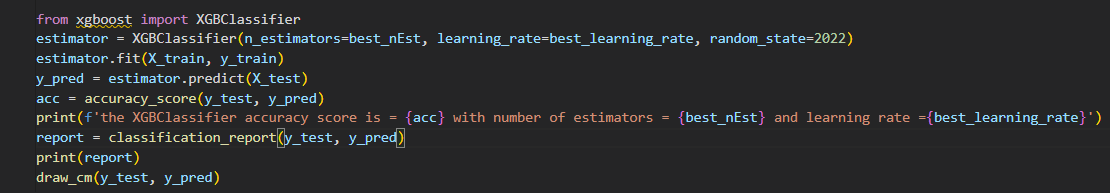


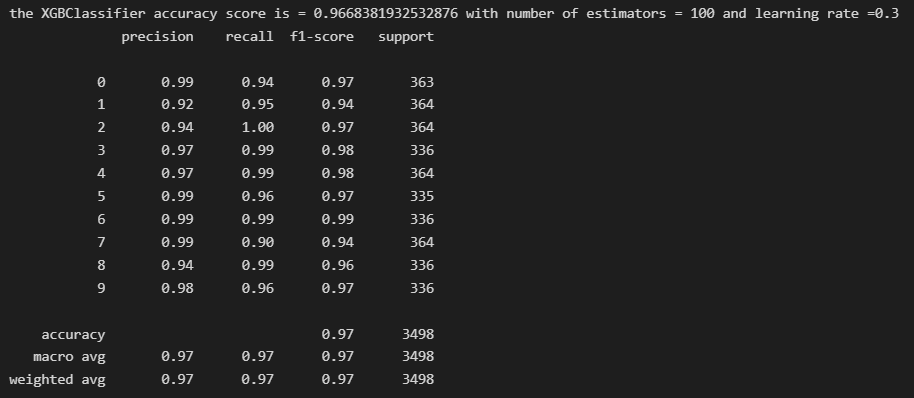


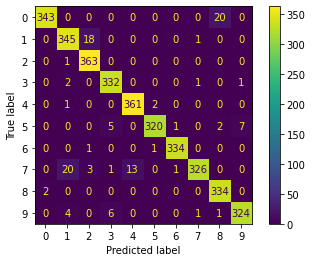


1. XGBoos Classifier with the same best hyper parameters (number of estimators =100 and learning rate = 0.3)

The accuracy = 0.9668







1. The accuracy for XGBoost = 96.68 which is slightly higher than the accuracy of the GradientBoosting = 96.51.

XGBoost has high predictive power and is almost 10 times faster than GradientBoosting

* GradientBoosting-> precision: there are more values = 1.0 for classes (1, 4, 6) which are higher than XGBoost but the low values in GradientBoosting = (0.91, 0.93, 0.93), meanwhile the values in XGBoost = (0.92, 0.94, 0.98) respectively for classes 1, 8 and 9.

On Average, they are equal for the precision, recall, and F1-score.

* It is much easier to use the accuracy metric for the comparison as here we have many classes (10 classes) to track the performance of the confusion matrix and comparing the performance between GradientBoosting and XGBoost for each class is really hard.
* Bagging is a parallel homogenous ensemble learning used to solve the high variance problem. Meanwhile, Boosting is a sequential homogeneous used to solve the high bias problem.
* Comparing the accuracies between Boosting (XGBoost) and bagging (decision tree) as the default base learner for XGBoost is the decision tree.
* Boosting (XGBoost): 96.68. Number\_estimators = 100
* Bagging (decision tree): 95.56. Number\_ estimators = 120

As we can see the boosting technique gives slightly higher accuracy over the test data. But if we could choose the right base learner we would reach a higher accuracy, as when we used the SVM model, the accuracy = 0.98