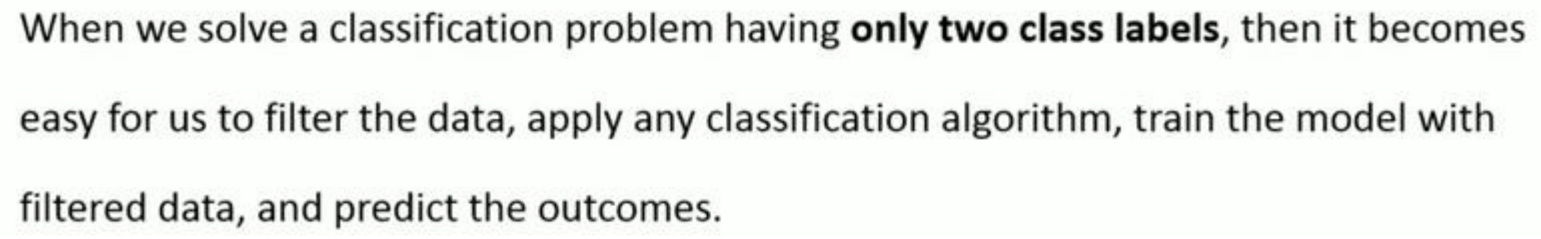
Multiclass classification

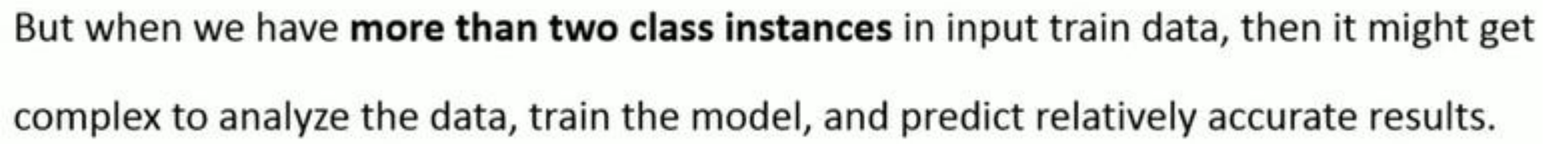
Dr. Parimala M

SCORE, VIT

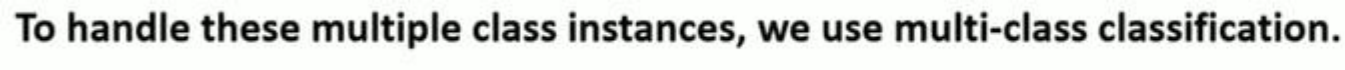
Multiclass classification



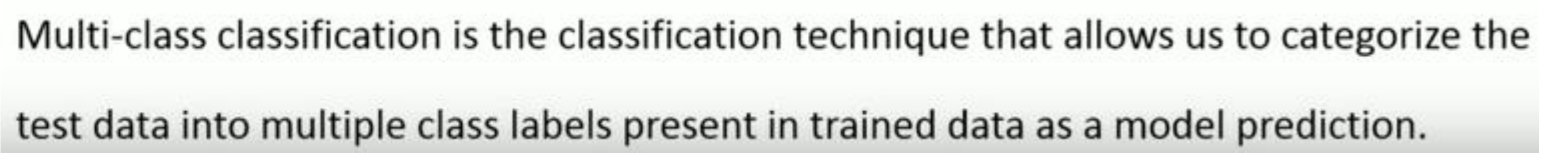
When we solve a classification problem having **only two class labels**, then it becomes easy for us to filter the data, apply any classification algorithm, train the model with filtered data, and predict the outcomes.



But when we have **more than two class instances** in input train data, then it might get complex to analyze the data, train the model, and predict relatively accurate results.

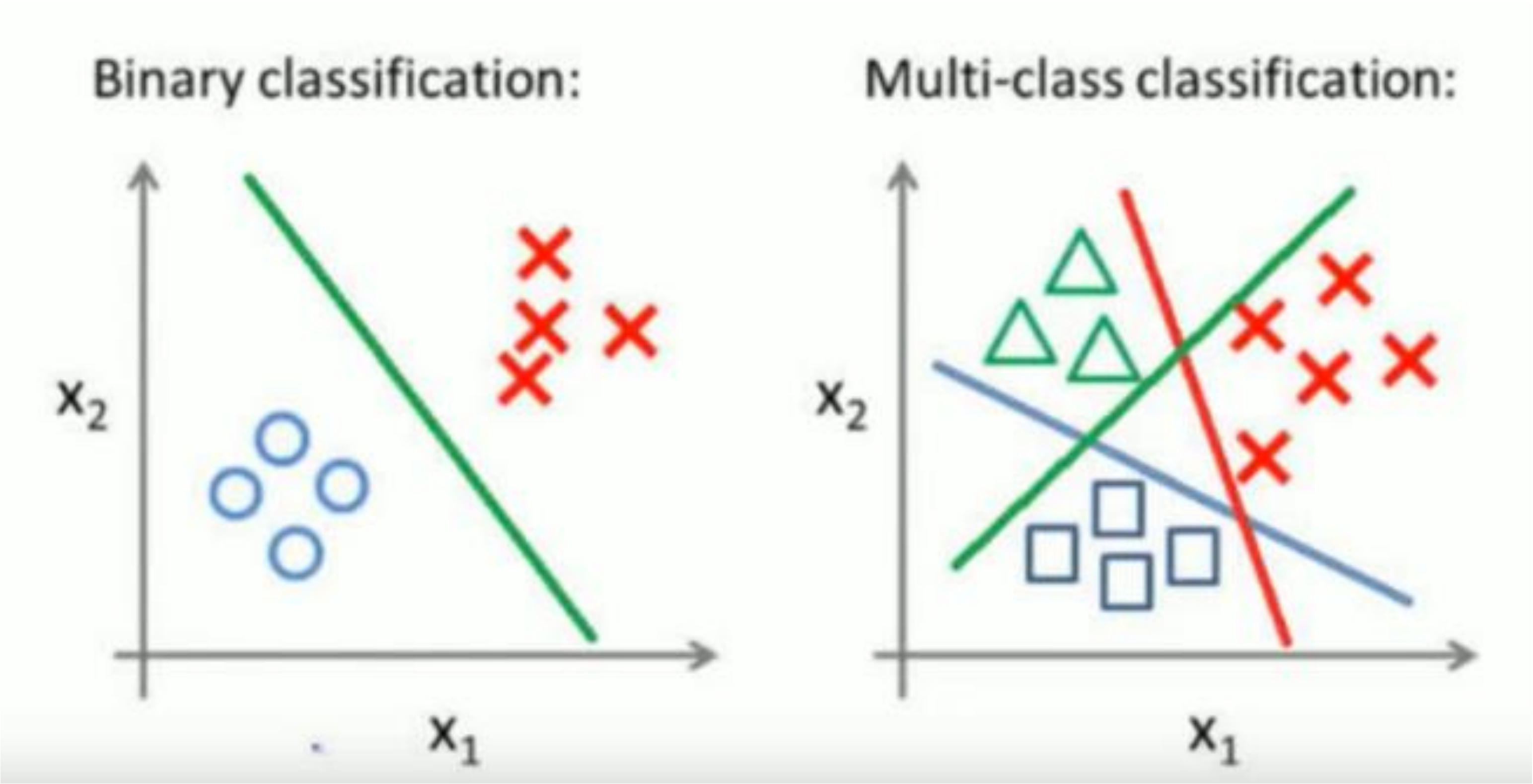


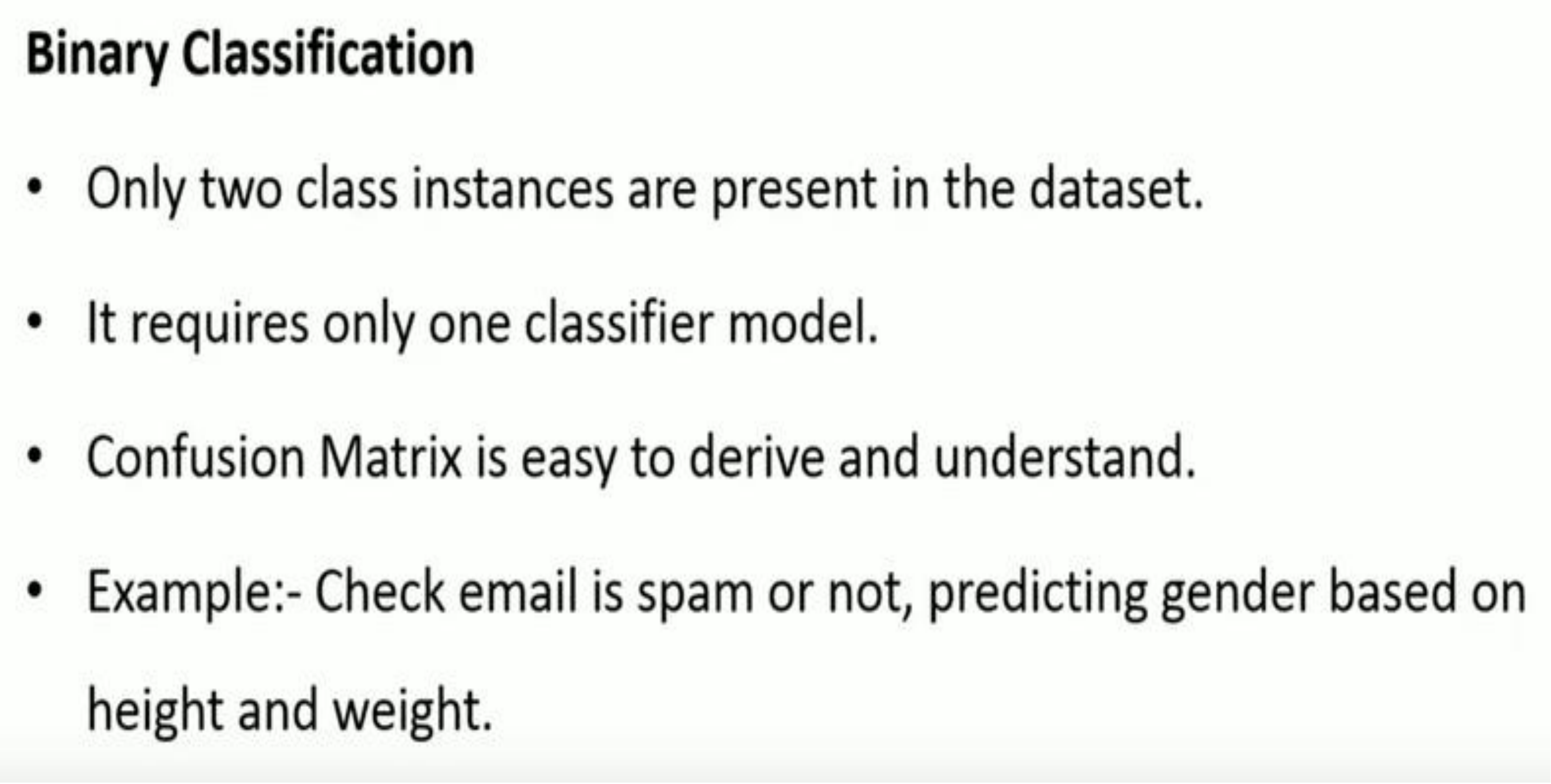
To handle these multiple class instances, we use multi-class classification.



Multi-class classification is the classification technique that allows us to categorize the test data into multiple class labels present in trained data as a model prediction.

Binary Vs Multiclass

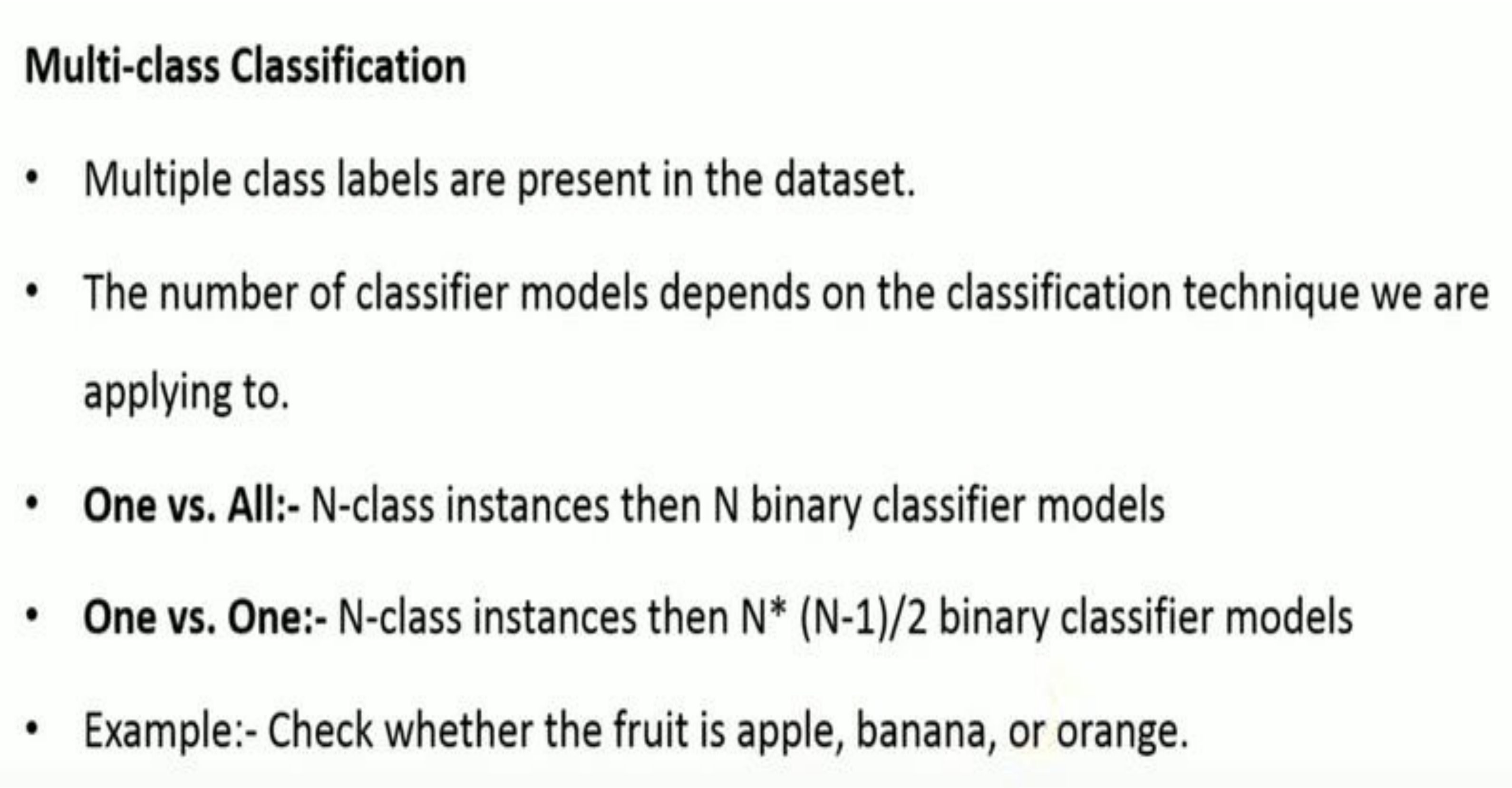




Binary Classification

- Only two class instances are present in the dataset.
- It requires only one classifier model.
- Confusion Matrix is easy to derive and understand.
- Example:- Check email is spam or not, predicting gender based on height and weight.

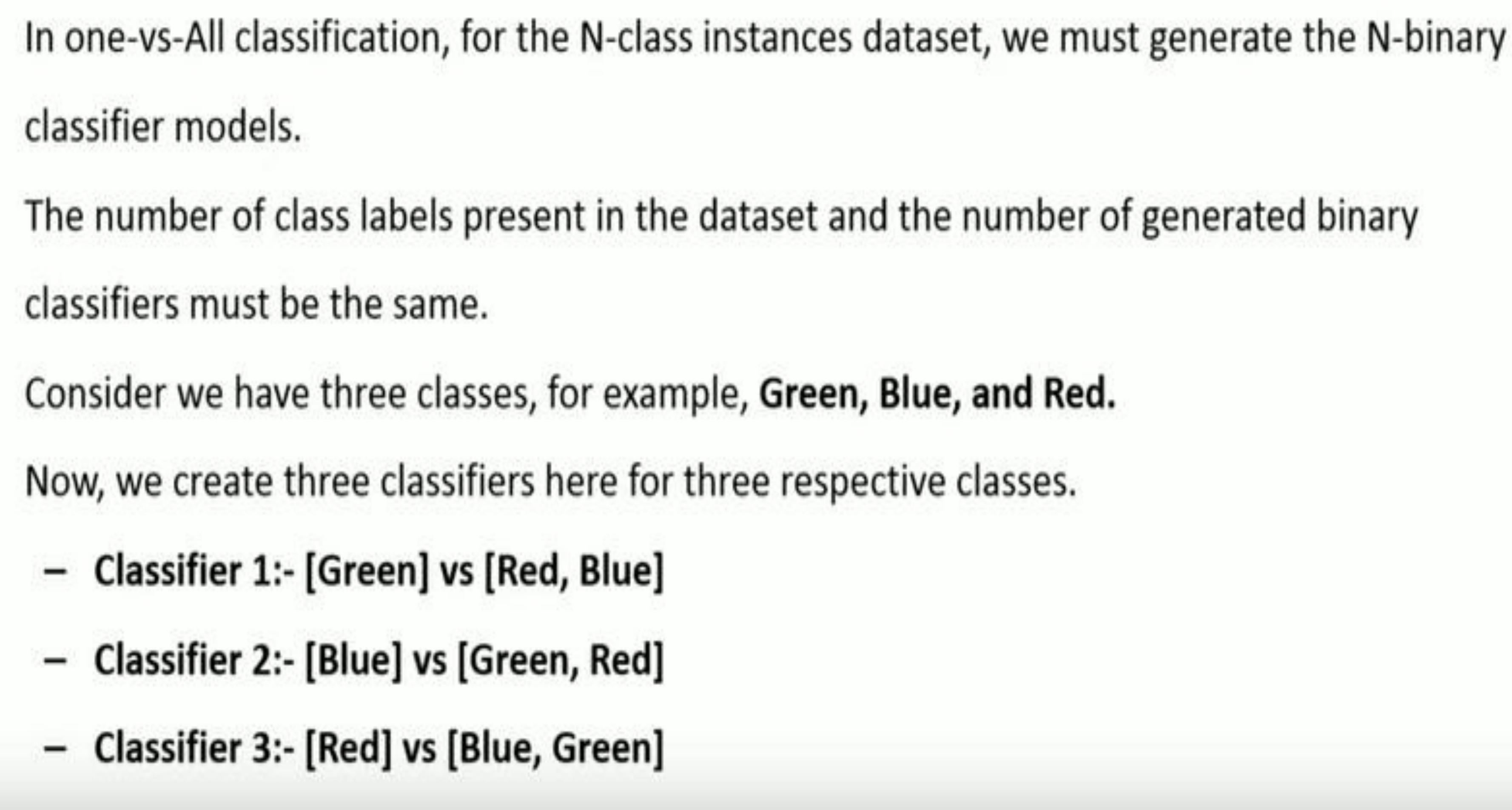
Multiclass



Multi-class Classification

- Multiple class labels are present in the dataset.
- The number of classifier models depends on the classification technique we are applying to.
- **One vs. All:-** N-class instances then N binary classifier models
- **One vs. One:-** N-class instances then N* (N-1)/2 binary classifier models
- Example:- Check whether the fruit is apple, banana, or orange.

One Vs All
Step:1 Create the classifiers



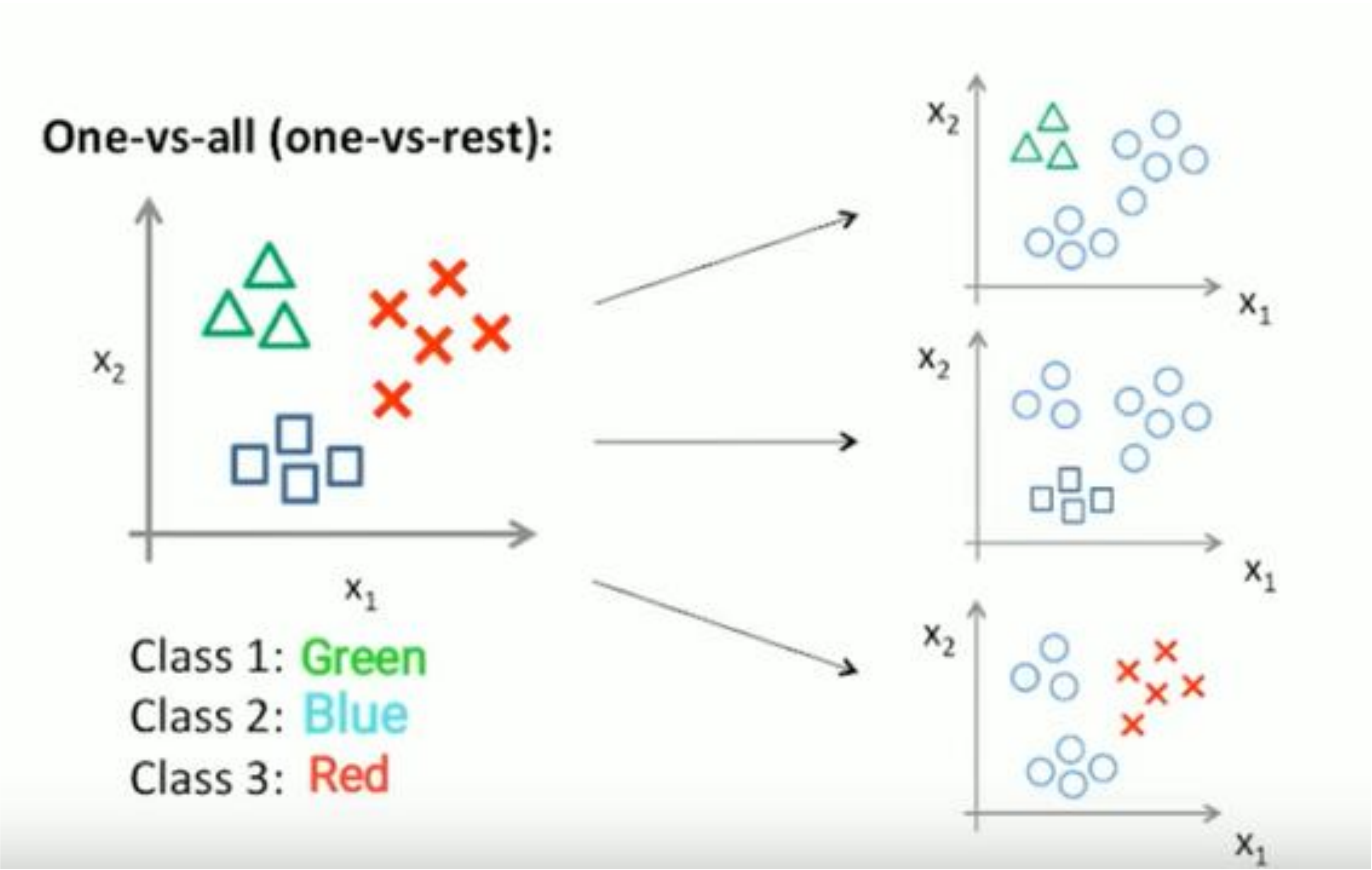
In one-vs-All classification, for the N-class instances dataset, we must generate the N-binary classifier models.

The number of class labels present in the dataset and the number of generated binary classifiers must be the same.

Consider we have three classes, for example, **Green, Blue, and Red.**

Now, we create three classifiers here for three respective classes.

- **Classifier 1:- [Green] vs [Red, Blue]**
- **Classifier 2:- [Blue] vs [Green, Red]**
- **Classifier 3:- [Red] vs [Blue, Green]**



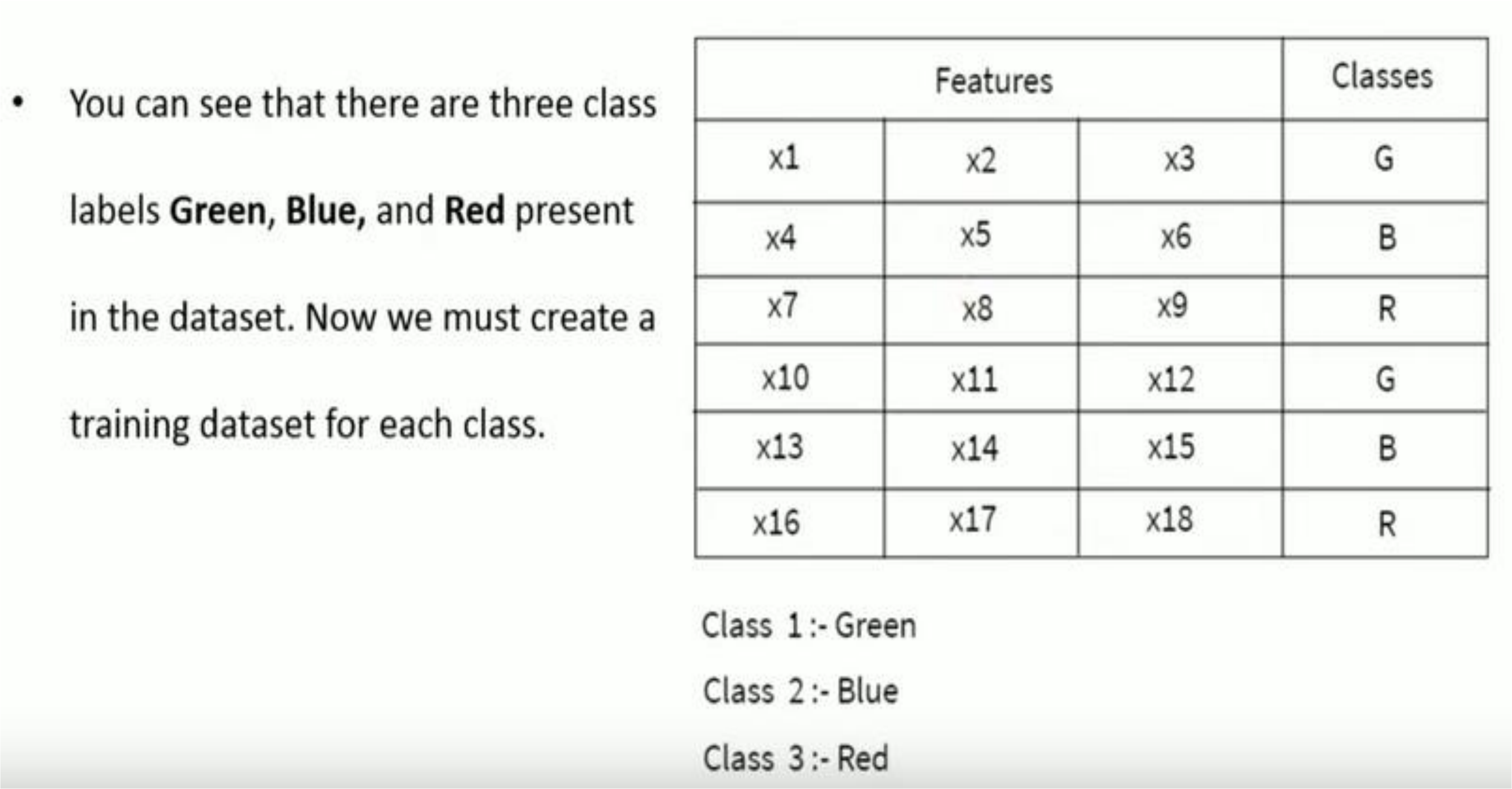
One-vs-all (one-vs-rest):

Class 1: Green

Class 2: Blue

Class 3: Red

Step:2 Create a training dataset for each class



- You can see that there are three class labels **Green**, **Blue,** and **Red** present in the dataset. Now we must create a training dataset for each class.

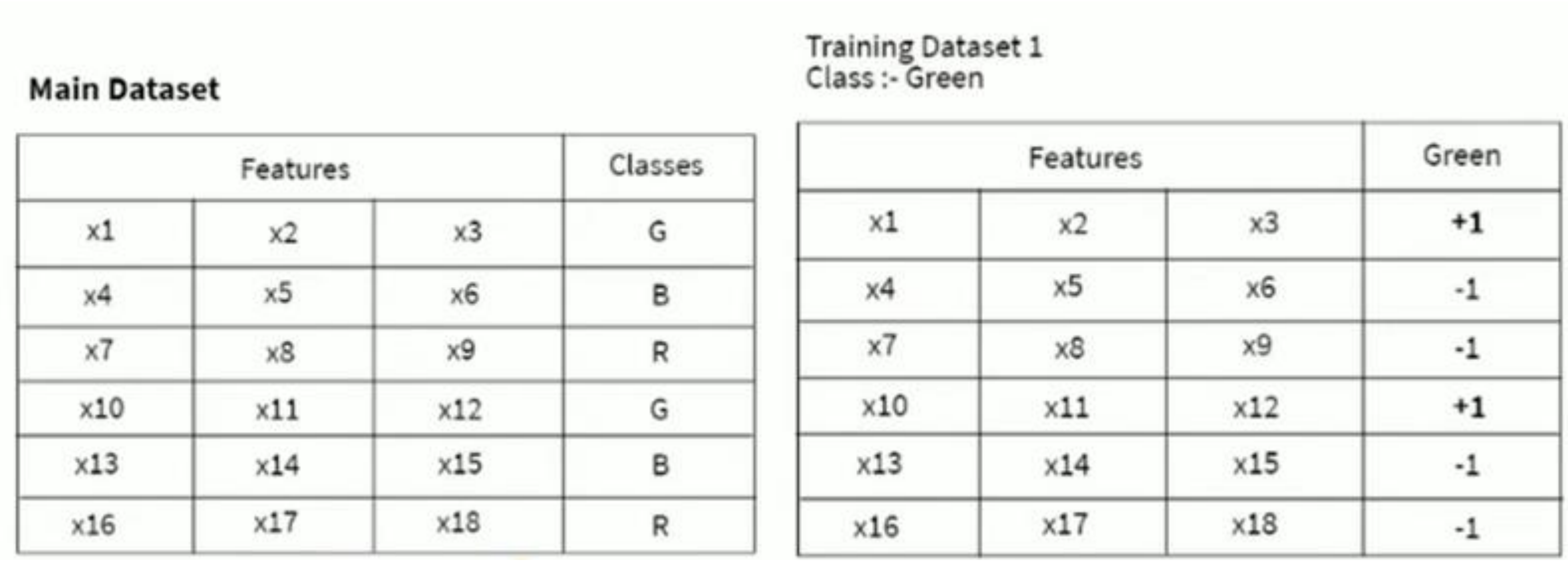
Features			Classes
x1	x2	x3	G
x4	x5	x6	B
x7	x8	x9	R
x10	x11	x12	G
x13	x14	x15	B
x16	x17	x18	R

Class 1 :- Green

Class 2 :- Blue

Class 3 :- Red

Training dataset for Green class

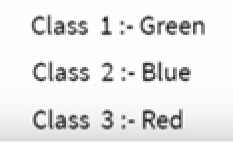


Main Dataset

Features			Classes
x1	x2	x3	G
x4	x5	x6	B
x7	x8	x9	R
x10	x11	x12	G
x13	x14	x15	B
x16	x17	x18	R

Training Dataset 1
Class :- Green

Features			Green
x1	x2	x3	**+1**
x4	x5	x6	-1
x7	x8	x9	-1
x10	x11	x12	**+1**
x13	x14	x15	-1
x16	x17	x18	-1

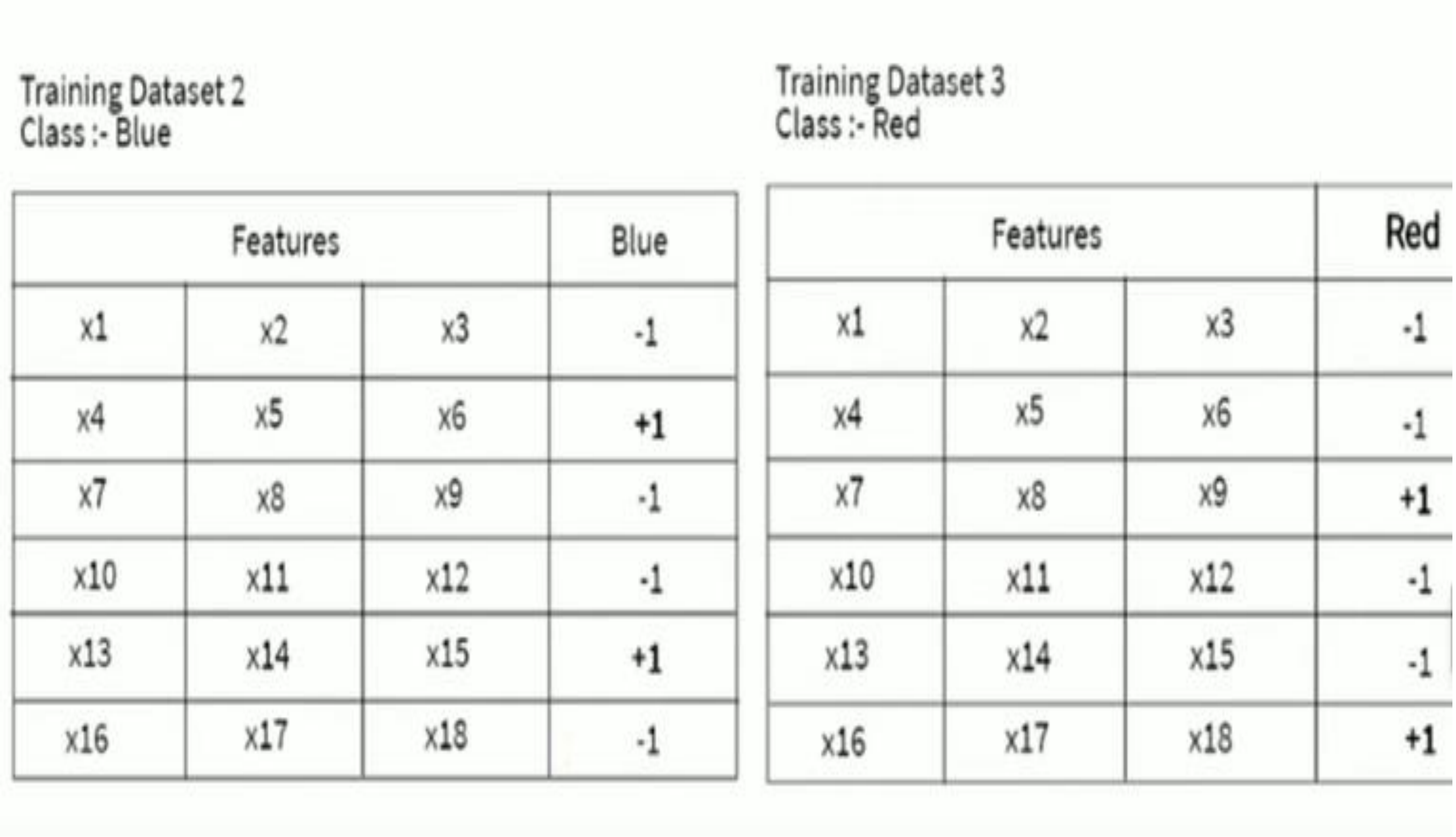


Class 1 :- Green

Class 2 :- Blue

Class 3 :- Red

Training dataset for blue & red class



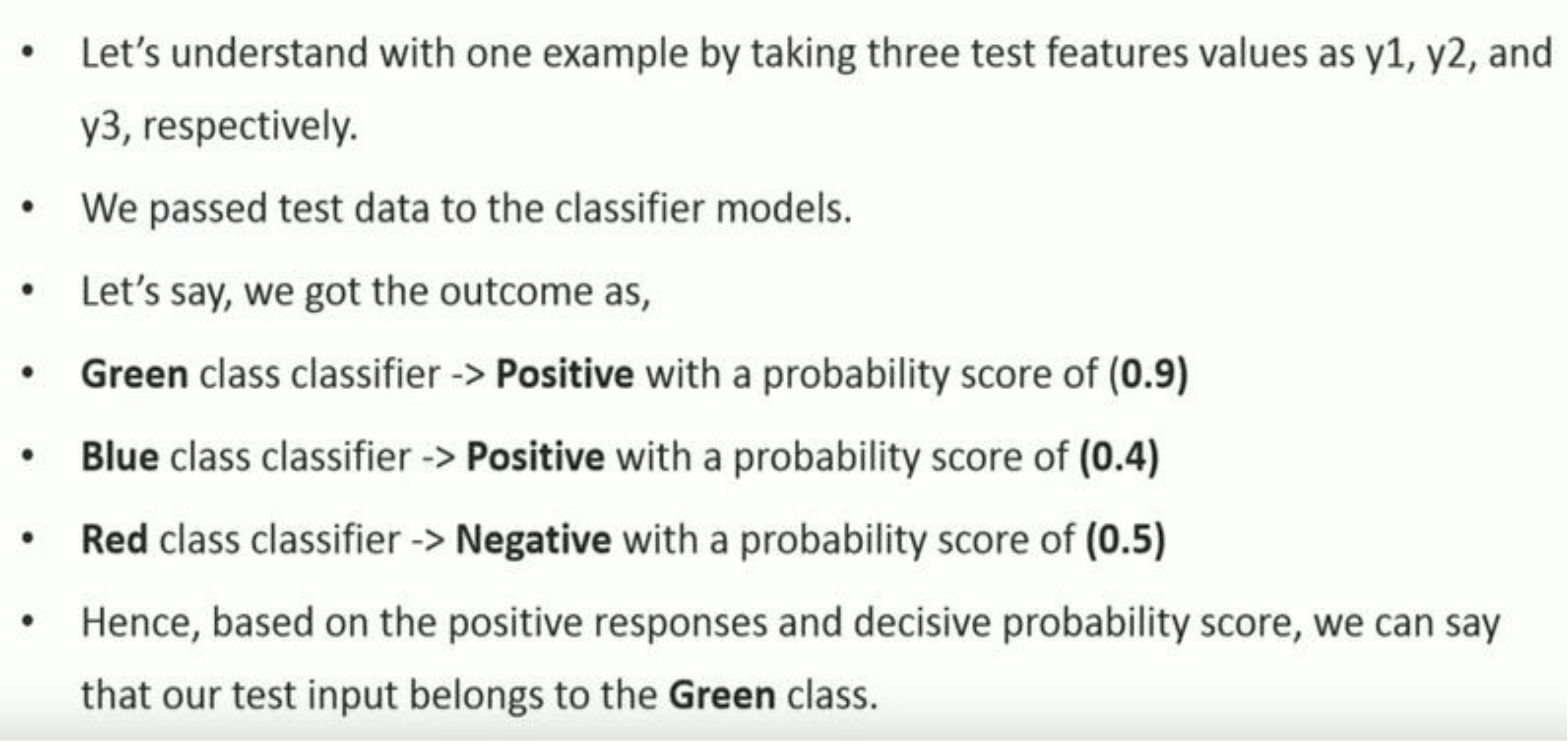
Training Dataset 2
Class :- Blue

Features			Blue
x1	x2	x3	-1
x4	x5	x6	**+1**
x7	x8	x9	-1
x10	x11	x12	-1
x13	x14	x15	**+1**
x16	x17	x18	-1

Training Dataset 3
Class :- Red

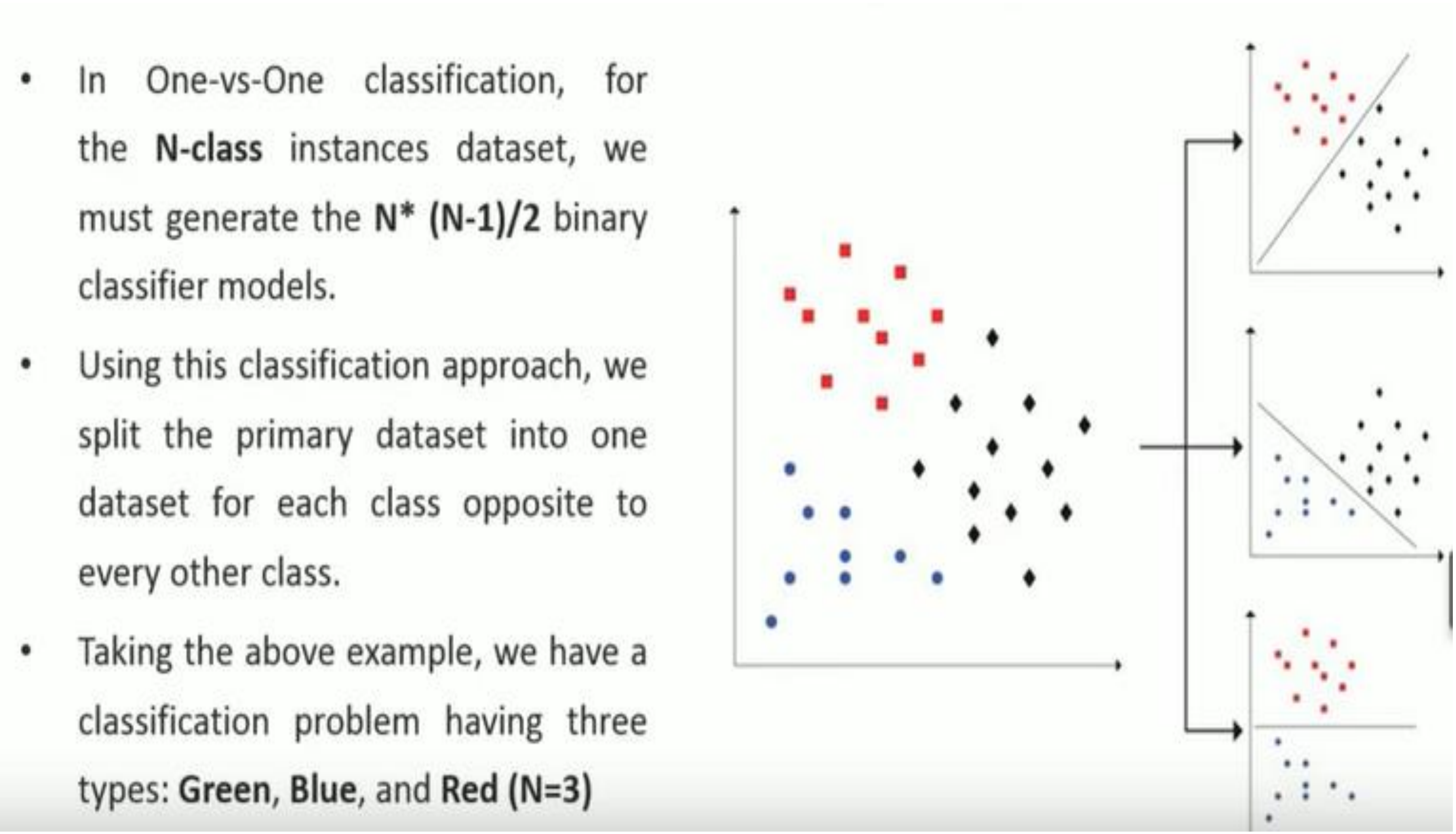
Features			Red
x1	x2	x3	-1
x4	x5	x6	-1
x7	x8	x9	**+1**
x10	x11	x12	-1
x13	x14	x15	-1
x16	x17	x18	**+1**

Step 3: Predict the test samples



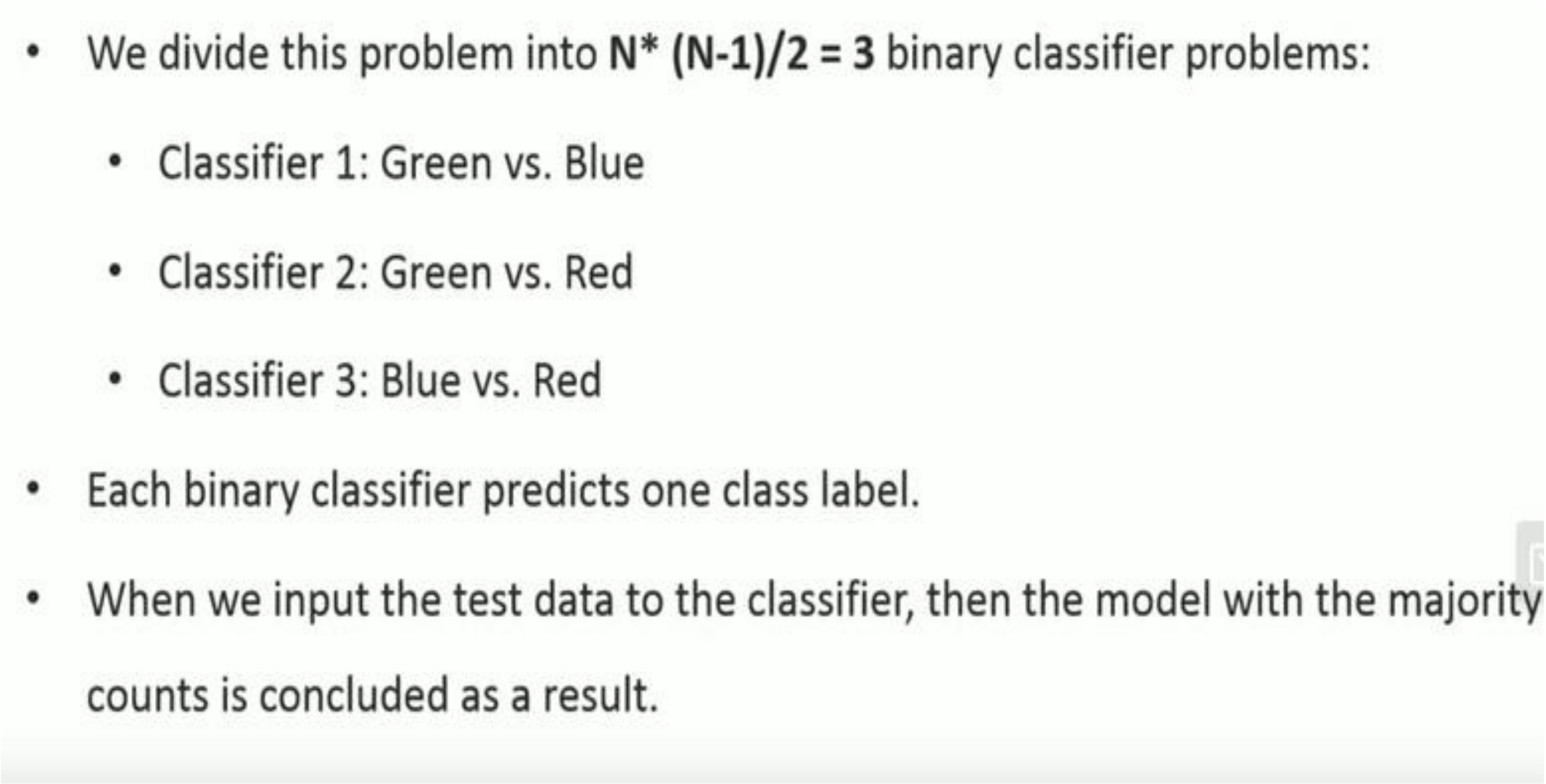
- Let’s understand with one example by taking three test features values as y1, y2, and y3, respectively.
- We passed test data to the classifier models.
- Let’s say, we got the outcome as,
- **Green** class classifier -> **Positive** with a probability score of (**0.9)**
- **Blue** class classifier -> **Positive** with a probability score of **(0.4)**
- **Red** class classifier -> **Negative** with a probability score of **(0.5)**
- Hence, based on the positive responses and decisive probability score, we can say that our test input belongs to the **Green** class.

One Vs One



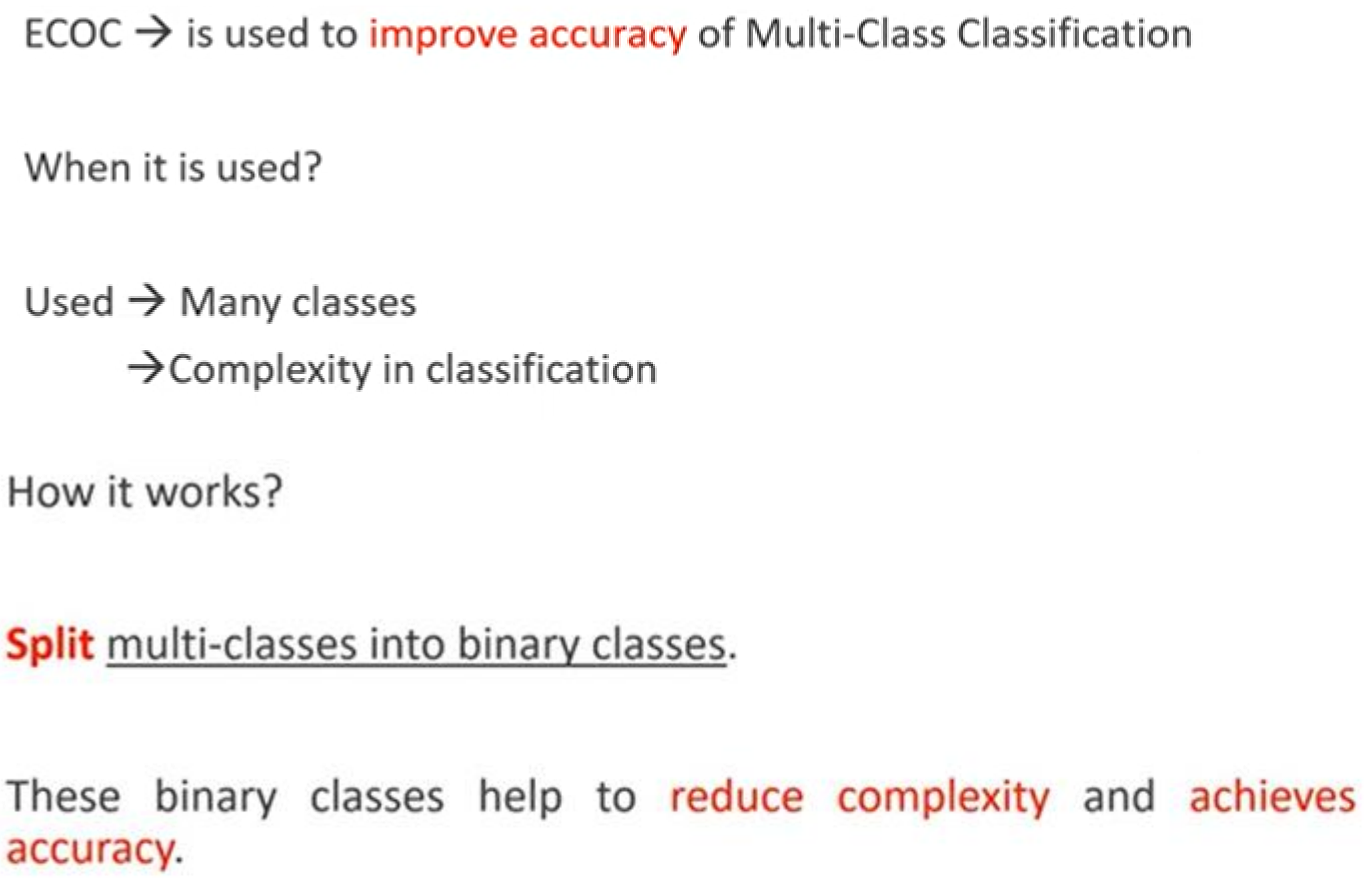
- In One-vs-One classification, for the **N-class** instances dataset, we must generate the **N* (N-1)/2** binary classifier models.
- Using this classification approach, we split the primary dataset into one dataset for each class opposite to every other class.
- Taking the above example, we have a classification problem having three types: **Green**, **Blue**, and **Red (N=3)**

Step:1-3



- We divide this problem into **N* (N-1)/2 = 3** binary classifier problems:
  - Classifier 1: Green vs. Blue
  - Classifier 2: Green vs. Red
  - Classifier 3: Blue vs. Red
- Each binary classifier predicts one class label.
- When we input the test data to the classifier, then the model with the majority counts is concluded as a result.

Error Correcting Output Code



ECOC → is used to improve accuracy of Multi-Class Classification

When it is used?

Used → Many classes

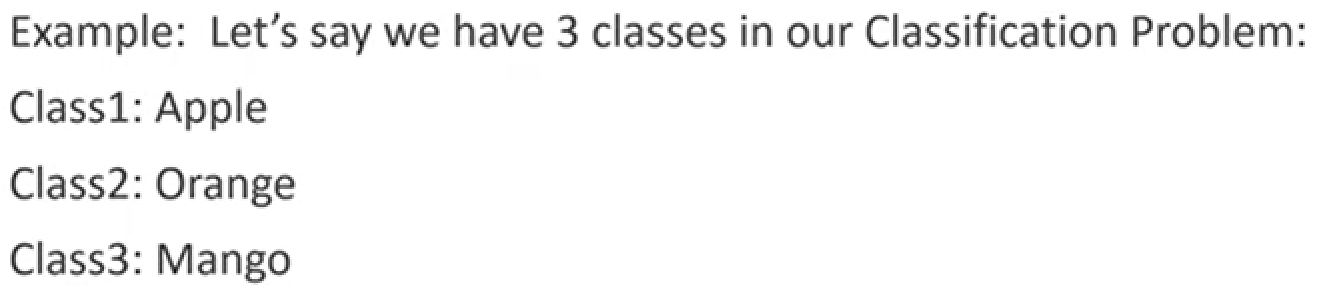
→Complexity in classification

How it works?

Split multi-classes into binary classes.

These binary classes help to reduce complexity and achieves accuracy.

One Vs ALL

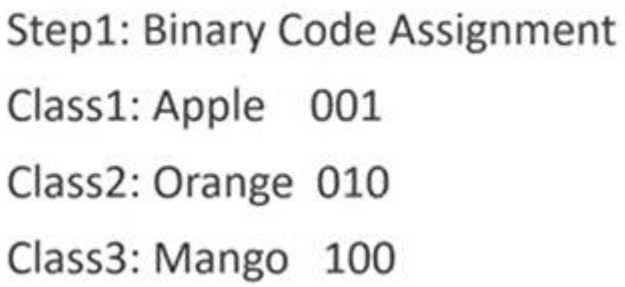


Example: Let’s say we have 3 classes in our Classification Problem:

Class1: Apple

Class2: Orange

Class3: Mango

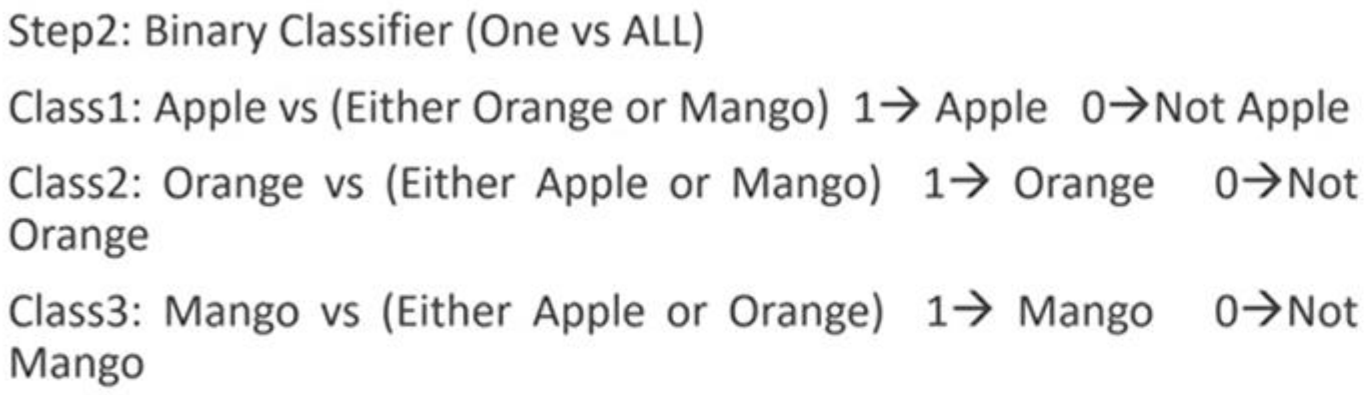


Step1: Binary Code Assignment

Class1: Apple 001

Class2: Orange 010

Class3: Mango 100

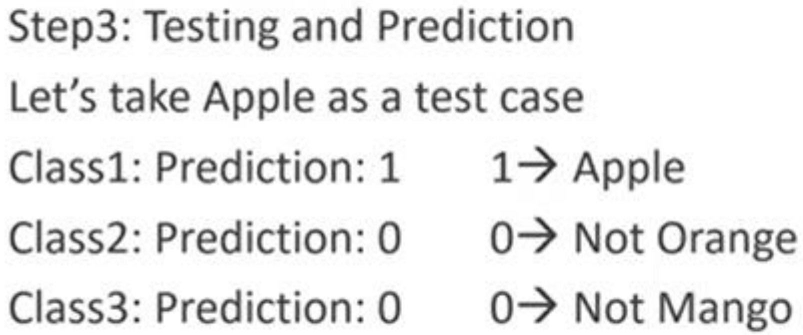


Step2: Binary Classifier (One vs ALL)

Class1: Apple vs (Either Orange or Mango) 1→ Apple 0→Not Apple

Class2: Orange vs (Either Apple or Mango) 1→ Orange 0→Not Orange

Class3: Mango vs (Either Apple or Orange) 1→ Mango 0→Not Mango



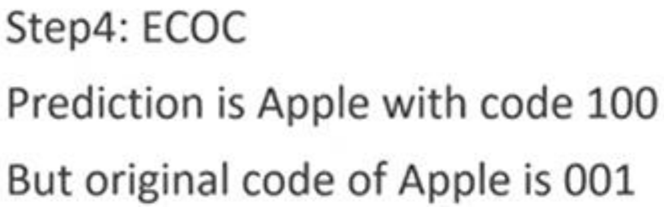
Step3: Testing and Prediction

Let's take Apple as a test case

Class1: Prediction: 1	1→ Apple
Class2: Prediction: 0	0→ Not Orange
Class3: Prediction: 0	0→ Not Mango

So from here it is derived that Prediction is Apple

With code 100



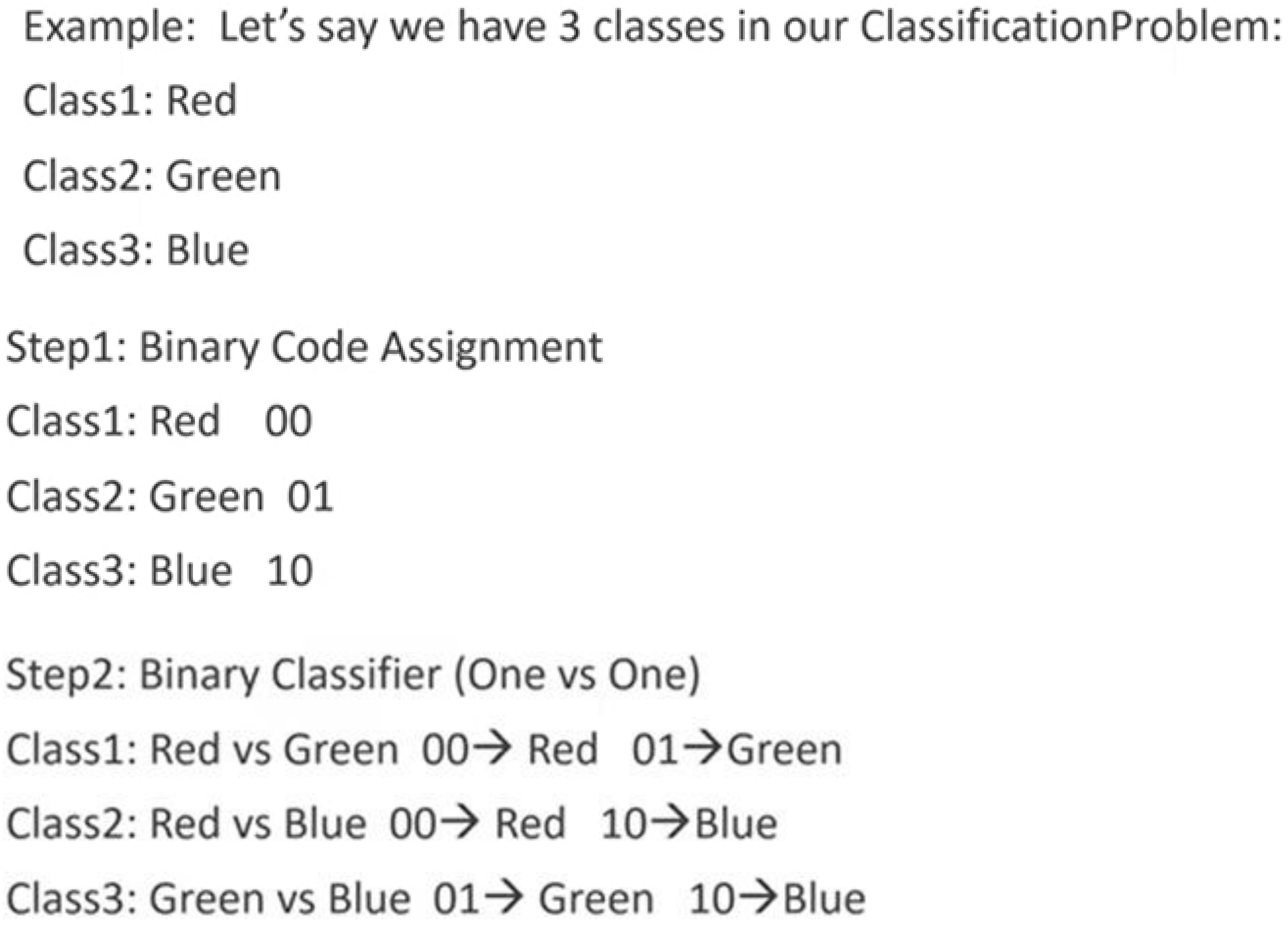
Step4: ECOC

Prediction is Apple with code 100

But original code of Apple is 001

So Flipping of bits in prediction is done and it is made 001

One Vs One



Example: Let’s say we have 3 classes in our ClassificationProblem:

Class1: Red

Class2: Green

Class3: Blue

Step1: Binary Code Assignment

Class1: Red 00

Class2: Green 01

Class3: Blue 10

Step2: Binary Classifier (One vs One)

Class1: Red vs Green 00→ Red 01→Green

Class2: Red vs Blue 00→ Red 10→Blue

Class3: Green vs Blue 01→ Green 10→Blue

Step3: Testing and Prediction

Let's take Red as a test case

Class1: Prediction: 00 00→ Red
Class2: Prediction: 10 10→ Blue
Class3: Prediction: 10 10→ Blue

So from here it is derived that Prediction is Blue

Step4: ECOC

Prediction is Blue with code 10

But original code of Red is 00

So Flipping of bits in prediction is done and it is made 00