**Classification metrics**

Evaluation metrics for Classification

1. accuracy
2. Precision
3. Recall
4. F1-Score

When to use which metrics?

Based on given dataset, we have two options:

* **Balanced Dataset** (equal distribution of classes)

If you are using balanced dataset, use **accuracy** as a metric to judge the quality of the model

* **UnBalanced Dataset** (unequal distribution of classes)

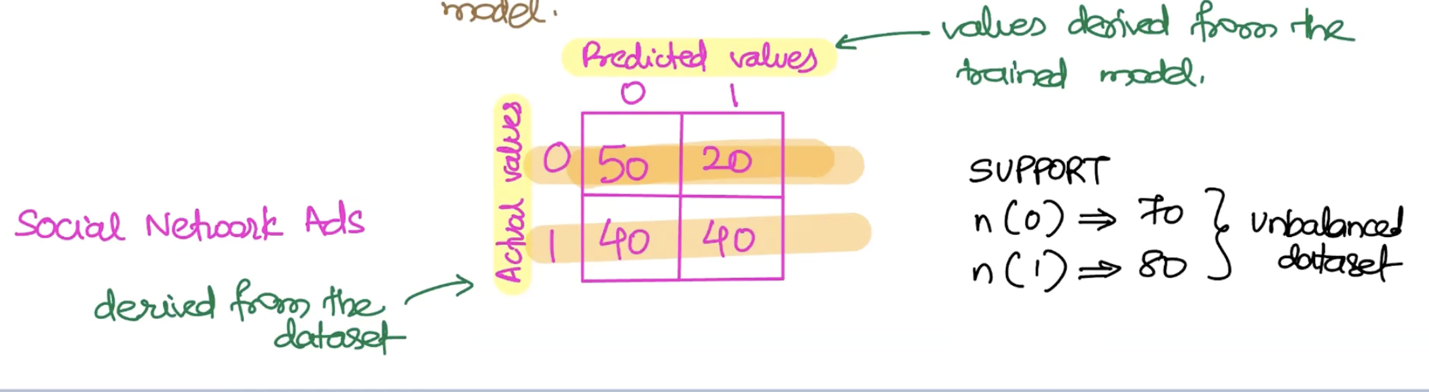
If you are using unbalanced dataset, you can use **Precision/Recall pai**r or **F1-Score**

How we identify mis-classification in a trained model?

To identify mis-classified data and To identify our model whether perform well or not, we need to use confusion matrix.

confusion matrix:

Evaluation matrix in table layout that helps visualizing the performance of your classification model.



In label 0, the model predicted 50 right and 20 wrong. Which means 20 of 70 for label 0 mis-classified.

In label 1, the mis-classification is 40.

Terminology

1. Support:

For a given label, how many numbers of record are available?

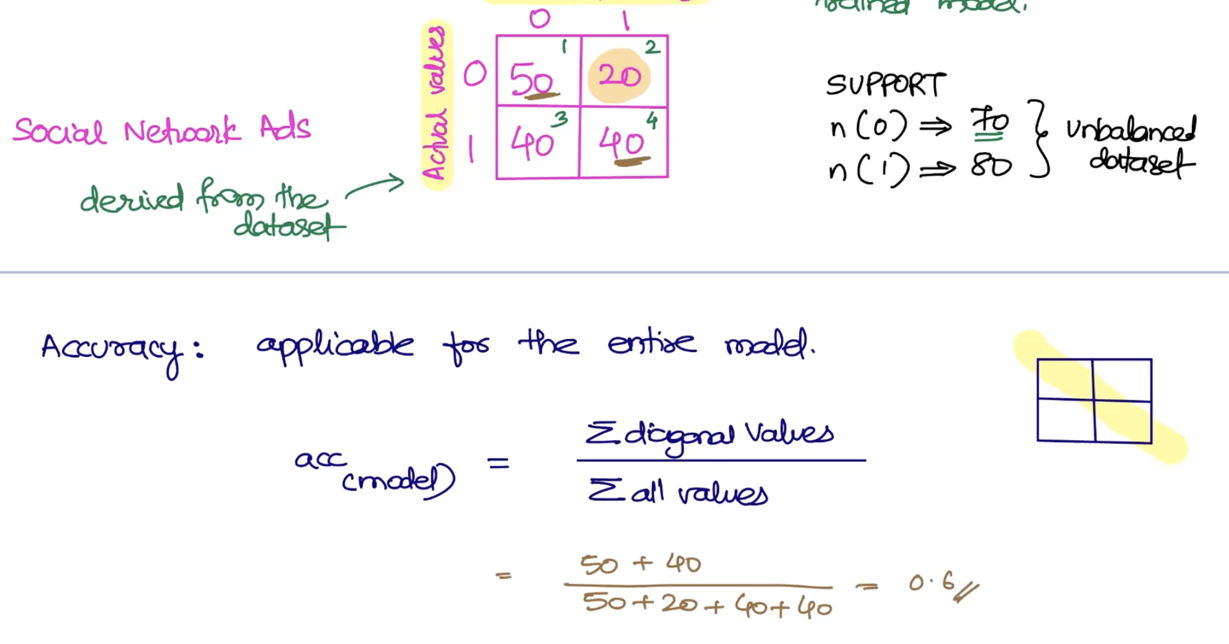
n(0) = numbers of record for zero ( in above table) 🡺 50 + 20 = 70

n(1) = numbers of record for one ( in above table) 🡺 40 + 40 = 80

So with look at supports numbers, we can define the dataset is balanced or unbalanced.

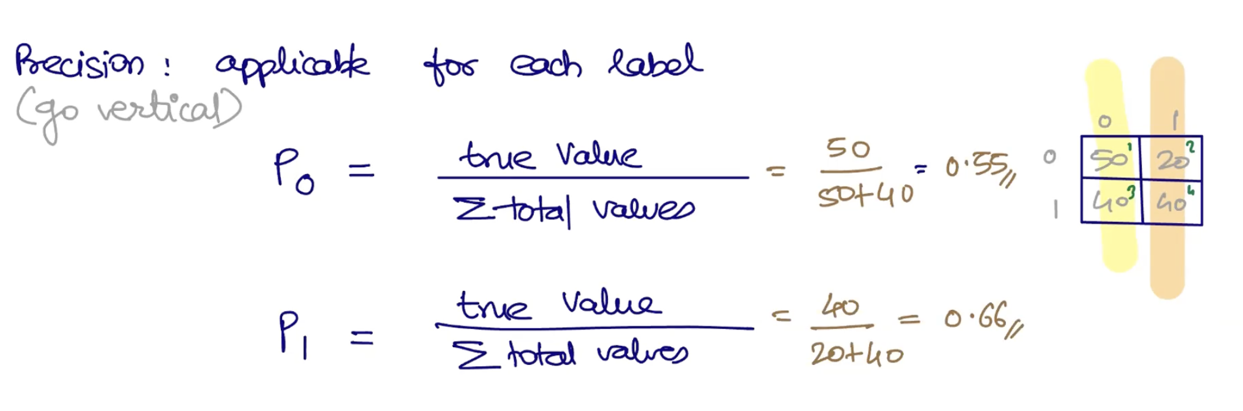
**Accuracy**:

applicable for the entire model:



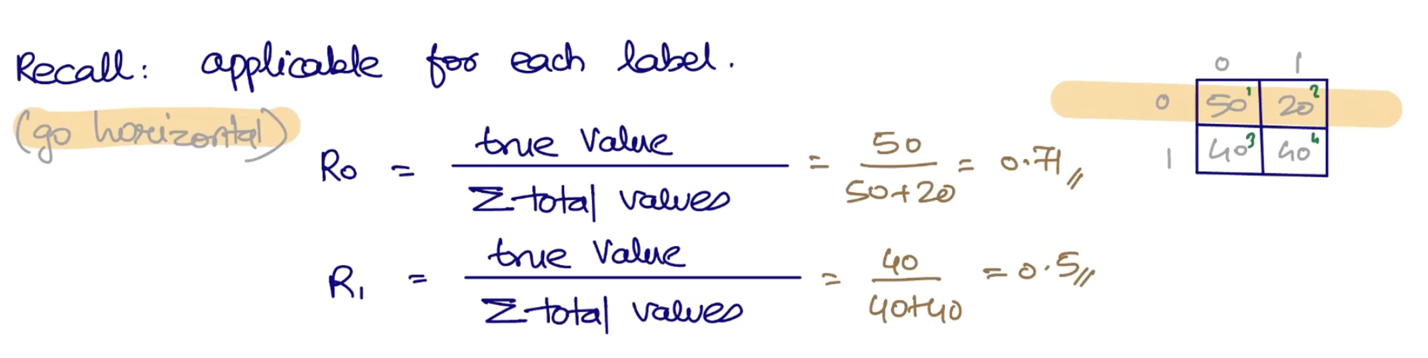
**Precision:**

applicable for each label:



**Recall:**

Applicable for each label



**Precision/Recall Pair:**

We had this use-case in classification topic:

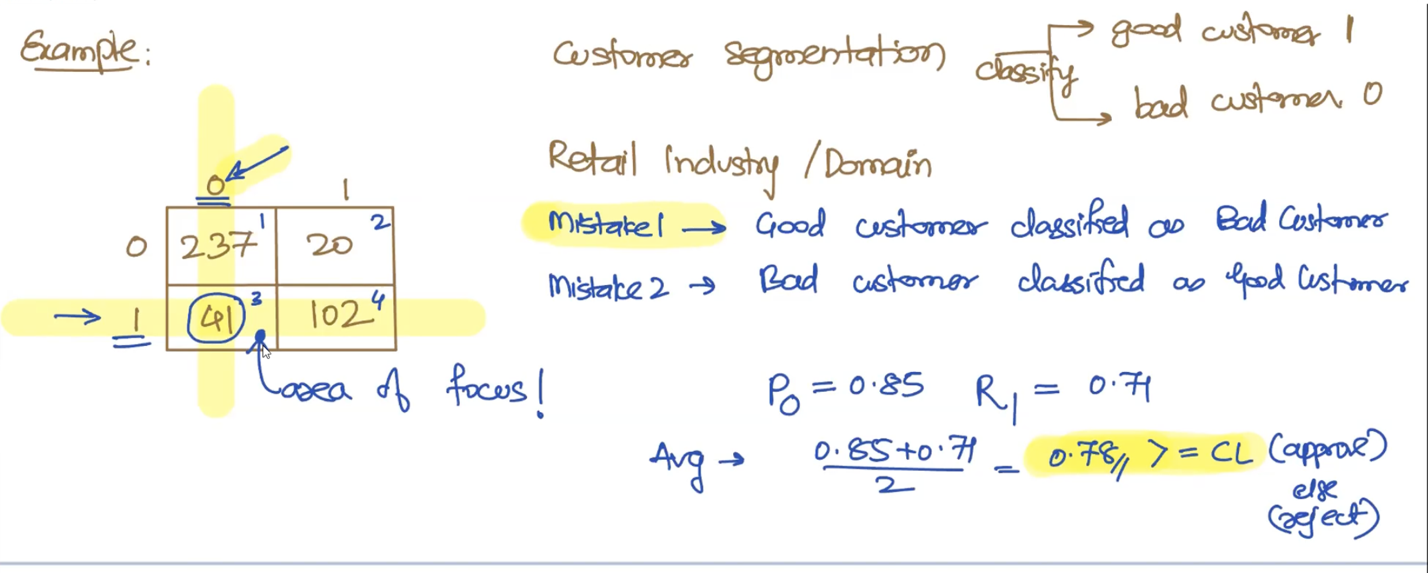
Use-case: An Online Shopping Mall has provided this dataset. Your job is to create a model that can predict whether the customer will shop or not based on customer's age and estimated salary.

|  |  |  |
| --- | --- | --- |
|  |  | data **=** pd**.**read\_csv('Social\_Network\_Ads.csv')  data**.**info()  <class 'pandas.core.frame.DataFrame'>  RangeIndex: 400 entries, 0 to 399  Data columns (total 5 columns):  # Column Non-Null Count Dtype  --- ------ -------------- -----  0 User ID 400 non-null int64  1 Gender 400 non-null object  2 Age 400 non-null int64  3 EstimatedSalary 400 non-null int64  4 Purchased 400 non-null int64  dtypes: int64(4), object(1) |
|  | Unbalanced dataset | data**.**Purchased**.**value\_counts()  0 257  1 143  Name: Purchased, dtype: int64 |
|  |  | **from** sklearn.metrics **import** confusion\_matrix  confusion\_matrix(label,finalModel**.**predict(features))  array([[237, 20],  [ 41, 102]]) |
|  |  | **from** sklearn.metrics **import** classification\_report  print(classification\_report(label,finalModel**.**predict(features)))  precision recall f1-score support  0 **0.85** 0.92 0.89 257  1 0.84 **0.71** 0.77 143  accuracy 0.85 400  macro avg 0.84 0.82 0.83 400  weighted avg 0.85 0.85 0.84 400 |

P0 is Precision for label zero and the value is: **0.85**

R1 is Recall for label One and the value is: **0.71**

And for Precision/Recall Pair, we take the average of P0 and R1.



When you are checking for generalization 🡺 Use accuracy

When you are checking the quality of the model for classification 🡺

Use **accuracy** for balanced dataset

Use **Precision/Recall Pair** for unbalanced

In Results:

1. Identify ideal SL value for project
2. Identify non-tolerable mistake (Classification use cases)

**Example 2:**

**Question:**

Use- case: model to predict whether the patient is sick or healthy?

SL = 0.05 CL = 1- SL 🡺 1- 0.05 = .95.

If metric > = CL:

accept the model

else:

reject the model

|  |  |  |
| --- | --- | --- |
|  | Healthy | Sick |
| Healthy | 50 | 10 |
| Sick | 5 | 100 |

**Answer:**

Is the given dataset is balanced or unbalanced dataset?

**Support:**

Num (Healthy)🡺 50+10 = 60

Num(Sick) 🡺 5+100 = 105

60 != 105 🡺 is unbalanced dataset. So we need to use Precision-Recall Pair or F1-score.

**Accuracy:**

applicable for the entire model:

50 + 100 / 50 + 10 + 5 + 100 = .90 🡺 Accuracy= .90, CL= .95 🡺 reject the model(accuracy < CL)

**Precision:**

applicable for each label (go vertical):

P(Healthy)🡺50/ 50+5 = .90

P(Sick) 🡺 100/10+100 = .90

**Recall:**

Applicable for each label(go Horizontal):

R(Healthy)🡺50/ 50+10 = .83

R(Sick) 🡺 100/5+100 = .95

**Precision-Recall Pair:**