* Classification *

⊙ Classification :-

> Categorize given data set into classes, can operate on both structured & unstructured data.

↳ Goal - Assign i/p data points to predefined classes.

↳ It is under Supervised Learning.

↳ classes - target, label / categories.

⊙ Learners in Classification Problems :-

① Lazy Learner -

↳ Firstly, store training dataset & wait until receive test dataset.

↳ Classification done on the basis of most related data stored in training dataset.

↳ Less time training & more time in prediction.

↳ Algorithms: KNN algo, Case-based reasoning.

② Eager Learner -

↳ Develop classification model based on training dataset, before receiving testing dataset.

↳ More time learning & less time in prediction.

↳ Algorithms - Decision Tree, Naive Bays, ANN.

Types of Classification Algorithms :-

↳ Supervised Learning Algos, used to categorize data based on i/p provided.

↳ Most common problems - speech recognition, Image recognition, text from handwriting.

↳ classification algos divided in 2 categories:

- Linear Modes -

  1. Logistic Regression
  2. Support Vector Machine

- Non-linear Models -

  1. Artificial Neural Network (ANN)
  2. Random Forest
  3. Decision Tree
  4. K-Nearest Neighbor (KNN)
  5. Naive Bayes
  6. Stochastic Gradient Descent

Terminologies used in Classification in ML -

1. Classifier - Model used to map i/p data to specific category.

2. Classification model - Model used to predict/ draw conclusion to i/p data given for training, predict class/ category for data.

3. Feature - Individual property of the dataset being observed.

④ Initialize — Assign classifier to be used for classification.

⑤ Train — Train the model using fit method. Using train_x & train_y dataset.

⑥ Predict — Use model to predict o/p for new set of i/p data.

⑦ Evaluate — Evaluation of performance of model.

• Types of Classification :-

1) Binary classification —

↳ Categorize i/p data to 1 of 2 possible classes or categories. Ex : True/ False, Yes/ No, 0/1.

↳ Example — • Email spam detection (spam or not).
• Churn prediction (churn / not)
• Conversion prediction (buy / not)
• Rain forecast (Yes/ No).

↳ 2 classes — 1 is normal state & other abnormal.

normal → Assigned 0

abnormal → Assigned 1.

↳ Ex : • Email spam — "spam" is abnormal state.
• Medical diagnosis — "cancer detected" abnormal.

↳ Algos used — ① Logistic Regression
② K-Nearest Neighbors
③ Decision Tree
④ Support Vector Machine
⑤ Naive Bayes

2) Multiclass Classification –

- Has more than 2 classes/ categories.
- Examples – • Face Classification.
  • Plant species classification.
  • Optical character recognition.
- Has range of classes to predict.
- Algos used – ① k-Nearest Neighbors.
  ② Decision Tree
  ③ Naive Bayes
  ④ Random Forest
  ⑤ Gradient Boosting
- Each sample is assigned to only 1 label / target.

3) Multi-Label classification —

- O/p have 2/ more class labels, where one/ more labels may be assigned to each example.
- Example – • Photo classification – Have many objects like – man, bicycle, apple, tree, banana.
- Multi-label versions of Algos Used –
  ① Multi-label Decision Tree
  ② Multi-label Random Forests
  ③ Multi-label Gradient Boosting

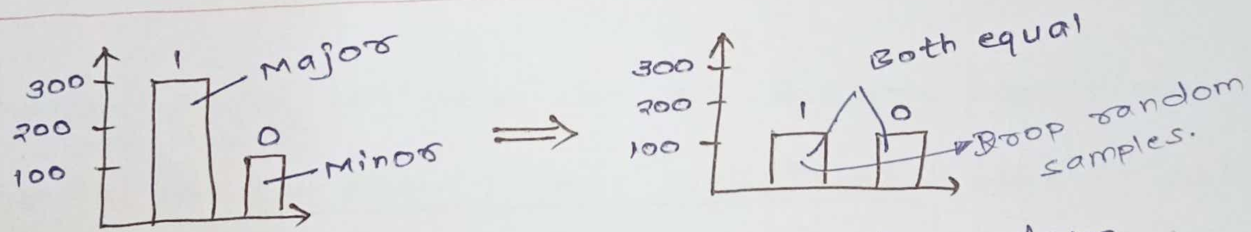
4) Imbalanced Classification –

- Num of examples in each class is unequally distributed. i.e. one type has ~~one~~ more examples than others.
- Handled using Random Forests.
- Typically are of binary classification, where normal class is in majority & abnormal minority.

- Techniques used for Balancing –
  1. under sampling
  2. Over sampling
  3. Bagging / Boosting

1. Under sampling –
   - Minority class remain as it is.
   - Reduce majority to match minority.

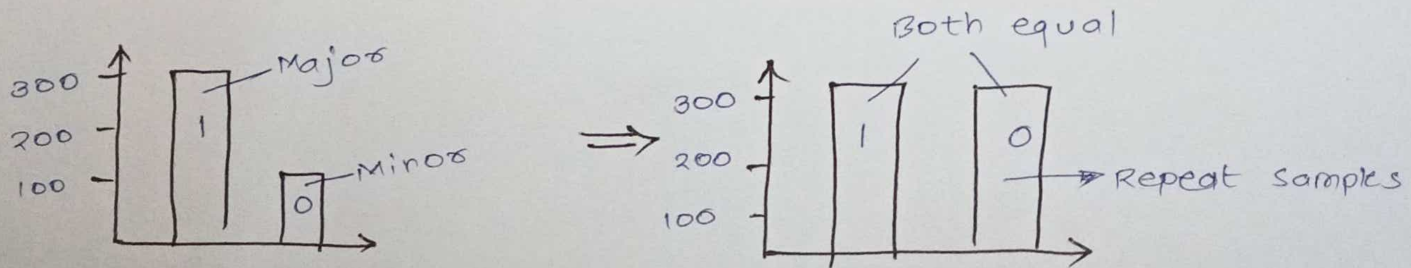
```
from imblearn.under_sampling import NearMiss
```



- Disadvantage – Possible to lose imp. data.

2. Over-Sampling – SMOT library used.
   - Repeat features to ↑ minority class to match with majority class.

```
from imblearn.over_sampling import SMOT
```



5) Hierarchical classification –

- Classes ordered in hierarchy / tree-like structure.
- Suitable for hierarchical relationship betn classes.
- Example: species classification in taxonomy – Kingdom, Phylum, Class, Order, Family, Genus & Species.

6) Ordinal Classification –

- Data where classes have natural ordering / ranking.
- Example – Movies rating: 1-star, 2-star etc.

7) Multiout Classification –

- Also k/a multiclass - multiout classification.
- Predict multiple o/p variables for each i/p data point.
- Each o/p variable can be binary / multiclass.
- It is extension of multilabel classification.

Model Evaluation Techniques for Classification :-

Used to find performance & efficiency of models.

1) Confusion Matrix :

- Detailed view of model's performance by showing counts of true +ve (TP), True -ve (TN), False +ve (FP) & False -ve (FN) predictions.
- Used to derive other metrics, like - Precision, Recall, F1-score & specificity.
- Useful for multi-class variables.

Example -

Y_actual	Y_pred
0	1
1	1
0	0
1	1
1	1
0	1
1	0

		Actual	
		True	False
Predicted	True	True +ve (TP) 3	False +ve (FP) 2
	False	False -ve (FN) 1	True -ve (TN) 1

2) Accuracy — Measure performance of the model.

- Measure how often model is correct.
- Ratio of total correct $Pred^n$ to total $Pred^n$.

$$\boxed{Accuracy = \frac{\overbrace{TP + TN}^{\text{correct Predictions}}}{\underbrace{TP + TN + FP + FN}_{\text{Total Predictions}}}}$$

$= \frac{4}{7} = \underline{\underline{0.57}}$ $\therefore$ 57% Accuracy.

③ Classification Error — Also k/a Misclassification Rate.

↳ Measure how often classifier is incorrect.

$$\boxed{\text{Error} = 1 - \text{accuracy}}$$

④ Precision — How accurate model's +ve predictions are.

↳ How many predicted True are correct.

↳ Ratio of True +ve to total no. of +ve.

$$\boxed{\text{Precision} = \frac{TP}{TP+FP}} = \frac{3}{3+2} = \frac{3}{5} = 0.60$$

∴ 60% Precision.

↳ Goal is to reduce FP.

⑤ Recall — How many actual True correctly predicted.

↳ Ratio of TP to sum (TP + FN).

$$\boxed{\text{Recall} = \frac{TP}{TP+FN}} = \frac{3}{4} = 0.75$$

∴ 75% Recall.

↳ Goal - Reduce FN. (Health related Problems).

- **Mail Classifier** -

Good { Actual - Spam, Pred - Spam } $\boxed{TP}$

May happen ok. { Actual - Spam, Pred - Not } $\boxed{FN}$

worst case { Actual - Not ~~Spam~~, Pred - ~~No~~ Spam } $\boxed{FP}$

	1	0
1		FP↓
0		

∴ Precision

- **Diabetes Classifier** -

Both Good { Actual - Yes, Pred - Yes $\boxed{TP}$; Actual - No, Pred - No $\boxed{TN}$ }

$\boxed{FP}$ Actual - No, Pred - Yes } May be okay.

$\boxed{FN}$ Actual - Yes, Pred - No } worst case.

	1	0
1		
0	FN↓	

∴ Recall.

(6) **F1- Score** — Evaluate overall performance of model.

Harmonic mean of Precision & Recall.

$$\boxed{F1.\ Score = \frac{2 * Precision * Recall}{Precision + Recall}}$$

$$F1 = \frac{2 * 60 * 75}{60+75} = 66.67$$

(7) **AUR- ROC Curve** —

- **ROC Curve** — Receiver Operating characteristics.
- **AUC** — Area Under Curve of the ROC Plot.

↳ Probability graph to show performance of classification model at different threshold levels.

↳ Curve plotted betn 2 params —

(1) TPR (True +ve Rate)

(2) FPR (False +ve Rate)

(1) **TPR** — Same as Recall. $\frac{TP}{TP+FN}$

(2) **FPR** — $\frac{FP}{FP+TN}$