Evaluating ML Model

- Critical to evaluate how model perform on unseen data

Below steps are used for model evaluation

- Train the model using 80% of the dataset
- Keep 20% of the data as unseen test data
- Use the trained model to make predictions on test data
- Compare predictions with actual values
- Compute evaluation metrics to assess model performance

Regression

Academic Qualification	Experience Years	Company	Position	Salary
Bachelors	5	Google	Developer	1000
Masters	8	Microsoft	Data Engineer	1500
Masters	2	Google	Developer	800
Bachelors	1	Microsoft	Data Engineer	1100
Bachelors	2	Google	Developer	1400

Qualification	Experience	Company	Position	Salary	Salary Pred	Error
Bachelors	1	Microsoft	Data Engineer	1100	1000	100
Bachelors	2	Google	Developer	1400	1500	-100

Metrics

MSE (RMSE)

- Mean Squared Error
- Average Squared difference between predicted vs actual value
- Formulae: $\frac{\sum(ya - yp)^2}{n}$
- Sensitive to large error

MAE

- Mean absolute error
- Average of absolute error between predicted vs actual value
- Formulae: $\frac{\sum|y_a - yp|}{n}$
- If median is used instead of mean then its Median Absolute error

R – Squared

- Coefficient of determination
- Formulae: $1 - \frac{\sum(ya - \widehat{y_a})^2}{\sum(ya - yp)^2}$
- 1 ➔ Perfect prediction, 0 ➔ No better than mean, -1 ➔ Worse than mean

Classification

Temp (c)	Humidity pct	Wind Direction	Season	Cloud Cover pct	Rainfall
32	45	N	Summer	20	0
30	60	E	Monsoon	55	1
28	75	S	Monsoon	80	1
35	40	W	Summer	10	0
22	30	N	Monsoon	12	0
28	22	N	Summer	78	1
27	40	E	Monsoon	60	1
22	30	W	Summer	10	0

Temp	Humidity	Wind	Season	Cloud Cover	Rainfall	Rain Pred
22	30	N	Monsoon	12	0	1
28	22	N	Summer	78	1	0
27	40	E	Monsoon	60	1	1
22	30	W	Summer	10	0	0

Confusion Matrix

- Table that summarizes how well a classification model performs by comparing Actual vs Predicted label

- For binary classification

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Significance:

TP ➔ Model correctly predicted positive value

TN ➔ Model correctly predicted negative value

FP ➔ Model predicted positive but it was wrong (Type – I)

FN ➔ Model predicted negative but it was wrong (Type – II)

Classification

Rainfall	Rain Pred	Label
0	1	FP
1	0	FN
1	1	TP
0	0	FN
0	1	?
1	0	?
1	1	?
0	0	?
0	1	?
1	0	?
0	1	?
1	1	?

TP =	FN =
FP =	TN =

Other Metrics

- Accuracy
  - Overall Correctness
  - (TP + TN) / (TP + TN + FP + FN)
- Precision
  - Out of predicted positive how many were correct ?
  - TP / (TP + FP)
- Recall (Sensitivity)
  - Out of actual positive how many did we catch ?
  - TP / (TP + FN)
- Specificity
  - TN / (TN + FP)
- F1 Score
  - 2 * precision * recall / (precision + recall)

Why it matters

- Shows where the model is making error beyond the accuracy
- Accuracy itself is useless for imbalance class.
- Example: Disease test where 5% of patient are infected. Predicting everyone healthy gives 95% accuracy but misses all patients
- Helps in problem specific optimization:
  - Medical Diagnostic: Reduce **FN** is critical (Don't miss sick patient) [Recall]
  - Spam Detection: Reducing **FP** is critical (Don't block genuine mail) [Precision]
  - Credit Fraud: **FN** & **FP** critical (No loss vs customer card not block) [F1 Score]
  - Rain Prediction: **FN** (Farmer may lose crop) or **FP** (Trip Cancellation)
  - Security System for terrorist fact detection: **FN** critical