Evaluation Metrics for Regression and Classification

Teaching Notes & Interview FAQ

Sakir Mansuri

https://www.linkedin.com/in/sakirmansuri/

This document contains concise formulas, Python usage snippets, best-practice guidance, common pitfalls, and an extended interview FAQ (conceptual and practical).

Contents

1 Why do we need evaluation metrics? 4

2 Regression Metrics (continuous targets) 4

2.1 Mean Absolute Error (MAE) 4
2.2 Mean Squared Error (MSE) 4
2.3 Root Mean Squared Error (RMSE) 4
2.4 R^2 (Coefficient of Determination) 5
2.5 Adjusted R^2 5
2.6 MAPE (Mean Absolute Percentage Error) 5
2.7 MSLE / RMSLE (Mean Squared Logarithmic Error) 5

3 Classification Metrics (categorical targets) 6

3.1 Confusion matrix (binary) 6
3.2 Log Loss (Cross-Entropy) 6
3.3 ROC and AUC 6
3.4 Precision-Recall curve 7
3.5 Matthews Correlation Coefficient (MCC) 7
3.6 Cohen's Kappa 7

4 Quick Use-Case Table (when to use which metric) 8

5 Domain – Metric mapping 9

6 Metric limitations (handy reference) 9

7 Common pitfalls 10

8 Machine Learning Interview FAQ 10

8.1 Fundamentals & Concepts 10
8.2 Metrics-centered questions 10
8.3 Algorithms & Training 11
8.4 Model selection & validation 11

8.5 Production & business 12
8.6 Interpretability and fairness 12

9 Appendix: Full Interview FAQ (concise answers) 13
9.1 Fundamentals 13
9.2 Model evaluation & metrics 13
9.3 Algorithms deep dive 13
9.4 Model selection & tuning 14
9.5 Practical implementation 14
9.6 Advanced & business 14

1 Why do we need evaluation metrics?

Building a model is only the first step — evaluation metrics tell us *how well* the model performs relative to the task and business goals. Think of metrics as the model's report card: different metrics answer different questions and carry different trade-offs.

2 Regression Metrics (continuous targets)

Used when the prediction target is numeric (e.g., house price, temperature).

2.1 Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} \left|y_i - \hat{y}_i\right|$$

Meaning: average absolute error in the original units.
Python:

```
from sklearn.metrics import mean_absolute_error
mae = mean_absolute_error(y_true, y_pred)
```

2.2 Mean Squared Error (MSE)

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Penalizes larger errors more (quadratic).
Python:

```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_true, y_pred)
```

2.3 Root Mean Squared Error (RMSE)

$$\text{RMSE} = \sqrt{\text{MSE}}$$

Interpretable in the same units as the target.
Python:

```
rmse = mean_squared_error(y_true, y_pred, squared=False)
```

2.4 R^2 (Coefficient of Determination)

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Fraction of variance explained by the model. Can be negative if model is worse than predicting the mean.
Python:

```
from sklearn.metrics import r2_score
r2 = r2_score(y_true, y_pred)
```

2.5 Adjusted R^2

$$\text{Adjusted } R^2 = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1}$$

Where n is samples and p is number of predictors — penalizes unnecessary features.

2.6 MAPE (Mean Absolute Percentage Error)

$$\text{MAPE} = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

Gives percent error; **beware** $y_i = 0$ cases (undefined).
Python (naive):

```
import numpy as np
mape = np.mean(np.abs((y_true - y_pred) / y_true)) * 100
```

2.7 MSLE / RMSLE (Mean Squared Logarithmic Error)

$$\text{MSLE} = \frac{1}{n} \sum_{i=1}^{n} \big(\log(1 + y_i) - \log(1 + \hat{y}_i)\big)^2$$

Useful when relative differences matter (growth-like targets).
Python:

```
from sklearn.metrics import mean_squared_log_error
msle = mean_squared_log_error(y_true, y_pred)
```

3 Classification Metrics (categorical targets)

3.1 Confusion matrix (binary)

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

From this we derive:

- **Accuracy:** $\frac{TP+TN}{TP+TN+FP+FN}$ — overall correctness.
- **Precision:** $\frac{TP}{TP+FP}$ — of predicted positives, fraction correct.
- **Recall (Sensitivity):** $\frac{TP}{TP+FN}$ — of actual positives, fraction detected.
- **F1-score:** harmonic mean of precision and recall:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Specificity:** $\frac{TN}{TN+FP}$ — true negative rate.

3.2 Log Loss (Cross-Entropy)

For true label $y_i \in \{0,1\}$ and predicted probability $\hat{p}_i$:

$$\text{LogLoss} = -\frac{1}{n}\sum_{i=1}^{n}\Big[y_i \log(\hat{p}_i) + (1-y_i)\log(1-\hat{p}_i)\Big]$$

Penalizes confident wrong predictions heavily.
Python:

```
from sklearn.metrics import log_loss
loss = log_loss(y_true, y_prob)
```

3.3 ROC and AUC

ROC curve plots True Positive Rate (Recall) vs False Positive Rate (1 - Specificity). AUC is the area under ROC and measures ranking ability across thresholds.
Python:

```
from sklearn.metrics import roc_auc_score, roc_curve
auc = roc_auc_score(y_true, y_prob)
fpr, tpr, thresholds = roc_curve(y_true, y_prob)
```

3.4 Precision-Recall curve

Often more informative when classes are heavily imbalanced: shows precision vs recall at different thresholds.
Python:

```
from sklearn.metrics import precision_recall_curve
precisions, recalls, thresholds = precision_recall_curve(y_true, y_prob)
```

3.5 Matthews Correlation Coefficient (MCC)

A single-score measure balanced for all confusion-matrix cells:

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Good for imbalanced datasets.
Python:

```
from sklearn.metrics import matthews_corrcoef
mcc = matthews_corrcoef(y_true, y_pred)
```

3.6 Cohen's Kappa

Measures agreement between predictions and true labels while adjusting for chance agreement:

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

where p_o is observed agreement and p_e is expected agreement by chance.
Python:

```
from sklearn.metrics import cohen_kappa_score
kappa = cohen_kappa_score(y_true, y_pred)
```

4 Quick Use-Case Table (when to use which metric)

Metric	**When to Use / Notes**
MAE	Interpretability in original units; robust to outlier influence relative to RMSE.
MSE / RMSE	Penalizes large errors more (good when big mistakes are costly).
MAPE	Business
	MSLE / RMSLE
Growth-like targets; penalizes relative differences.	
R^2	Model explanatory power; can be negative and is sensitive to outliers.
Accuracy	Balanced datasets or when all errors cost equally.
Precision	When false positives are expensive (spam filters, fraud alerts).
Recall	When missing positives is costly (disease screening).
F1-Score	Balanced view when precision and recall both matter.
Log Loss	Probabilistic model calibration and confidence — use when probabilities matter.
MCC / Kappa	Imbalanced datasets — more reliable single-number score.
AUC	Ranking ability across thresholds; insensitive to calibration.

5 Domain – Metric mapping

Domain	Task	Best Metric(s) and rationale
Healthcare	Disease screening	Recall, F1, Precision-Recall curve (catch positives; low false negatives).
Finance	Fraud detection	Precision, Recall, F1, AUC (minimize false positives and false negatives; rank transactions).
E-commerce	Recommendation	Precision@K, Recall@K, MAP, AUC (top-K ranking quality).
Marketing	Conversion prediction	AUC, Log Loss, Precision-Recall (calibrated probabilities for targeting).
Forecasting (sales)	Demand forecasting	RMSE, MAPE (cost interpretable), prediction intervals.
NLP (classification)	Sentiment / NER	F1 (often class imbalance), token-level metrics for NER.
Manufacturing	Defect detection	Recall (don't miss defects), Precision (avoid false alarms).

6 Metric limitations (handy reference)

Metric	Limitations / Caveats
Accuracy	Misleading with imbalanced classes (high accuracy possible by predicting majority class).
MAPE	Undefined for zero actuals; biased when actuals are very small.
RMSE	Sensitive to outliers (may over-emphasize rare large errors).
R^2	Not meaningful for non-linear relationships without context; can be negative.
AUC	Does not reflect calibration; two models with equal AUC can have very different business impact.
Log Loss	Requires well-calibrated probabilities — penalizes overconfident wrong predictions.

7 Common pitfalls

- Using accuracy on imbalanced datasets without investigating class distribution.
- Relying solely on AUC when business cost depends on a specific threshold.
- Reporting only point metrics (report intervals or multiple metrics).
- Not checking calibration of predicted probabilities.
- Choosing metrics that are easy to compute rather than metrics aligned to business objectives.

8 Machine Learning Interview FAQ

These questions combine conceptual depth with short, interview-ready answers.

8.1 Fundamentals & Concepts

1. **What is Machine Learning vs traditional programming?**
Answer: Traditional programming: rules + data → output. ML: data + desired output → algorithm infers rules (model) to make predictions on new data.
2. **Types of ML?**
Answer: Supervised (labels), Unsupervised (no labels), Semi-supervised (mix), Reinforcement (agent + rewards).
3. **Regression vs Classification?**
Answer: Regression predicts continuous values; classification predicts discrete labels/classes.
4. **What is overfitting and how to detect it?**
Answer: Overfitting = model learns noise and performs well on train but poorly on unseen data. Detect via big gap between train/validation scores; use learning curves.
5. **Bias-variance tradeoff?**
Answer: Error decomposes into bias^2 + variance + irreducible error. Simpler models → high bias; complex models → high variance. Aim for balance.

8.2 Metrics-centered questions

6. **Why can R^2 be negative?**
Answer: If the model's SSE is greater than the variance of the data (predicting mean), the ratio exceeds 1 and $R^2 < 0$ — model worse than baseline.
7. **Why is RMSE usually greater than or equal to MAE?**
Answer: RMSE squares errors before averaging — it gives more weight to large errors. Mathematically RMS $\geq$ mean absolute by Cauchy-Schwarz.

8. **When is PR-curve preferred to ROC?**
   Answer: Use PR-curve for highly imbalanced datasets (focuses on positive class and shows precision vs recall).

9. **Why is F1 the harmonic mean (not arithmetic)?**
   Answer: Harmonic mean penalizes extreme values — a low precision or low recall drastically reduces F1, reflecting the need for balance.

10. **What is calibration and why does it matter?**
    Answer: Calibration means predicted probabilities match observed frequencies. Important when probabilities drive decisions (e.g., targeting customers).

8.3 Algorithms & Training

11. **Explain Linear vs Logistic Regression.**
    Answer: Linear predicts a continuous value using least squares. Logistic predicts probability using the logistic (sigmoid) on linear combination; trained by maximizing log-likelihood (cross-entropy).

12. **KNN pros and cons?**
    Answer: Pros: simple, non-parametric. Cons: expensive at inference, sensitive to scaling and irrelevant features.

13. **Decision Trees overfitting fixes?**
    Answer: Pre-pruning (max depth, min samples), post-pruning, use ensembles (bagging/boosting).

14. **Bagging vs Boosting?**
    Answer: Bagging (parallel, reduce variance) e.g. RandomForest. Boosting (sequential, reduce bias) e.g. XGBoost.

15. **When to use SVM over logistic regression?**
    Answer: SVM for small/medium high-dimensional data where margin matters, and when non-linear kernels help. Logistic gives calibrated probabilities and is faster for large data.

8.4 Model selection & validation

16. **Cross-validation types?**
    Answer: K-fold, stratified K-fold (maintain class balance), time-series CV (rolling windows), Leave-One-Out.

17. **How to handle imbalanced datasets?**
    Answer: Use appropriate metrics (precision/recall, PR-AUC), resampling (SMOTE, ADASYN), set class weights, threshold tuning, ensemble/cost-sensitive methods.

18. **Feature engineering basics?**
    Answer: Scaling, encoding (one-hot, target), interaction terms, domain-specific aggregations, datetime feature extraction, dimensionality reduction (PCA).

19. **Hyperparameter tuning approaches?**
    Answer: Grid search, random search, Bayesian optimization, cross-validated scoring (choose metric aligned with business).

8.5 Production & business

20. **How to explain model results to non-technical stakeholders?**
    Answer: Start with business impact, use analogies, show key metric(s), use visuals and concrete examples, state confidence/uncertainty and operational constraints.
21. **How to ensure model performs well in production?**
    Answer: Robust validation, feature availability checks, monitoring (drift, performance), A/B testing, fallback/default logic, model retraining plan.
22. **What is concept drift and how to detect it?**
    Answer: Distribution or relationship changes over time. Detect via changes in input distributions, target distributions, or performance degradation; handle via retraining or adaptive models.
23. **How to set a decision threshold?**
    Answer: Choose threshold to optimize business cost function (precision/recall trade-off), or pick using validation-based expected utility.

8.6 Interpretability and fairness

24. **How to explain model predictions?**
    Answer: Use feature importance (tree-based), SHAP values, LIME local explanations, partial dependence plots.
25. **How to assess fairness?**
    Answer: Check parity metrics (equalized odds, demographic parity), test for bias in datasets and predictions; mitigate via reweighting, adversarial debiasing or post-processing.

9 Appendix: Full Interview FAQ (concise answers)

This is a comprehensive Q&A list suitable for rapid interview revision.

9.1 Fundamentals

- **Q: What is ML and how is it different from programming?**
  A: (See earlier short answer) Data + outputs → model learns rules automatically.
- **Q: Types of ML?** A: Supervised, Unsupervised, Semi-supervised, Reinforcement.
- **Q: Regression vs Classification?** A: Regression = continuous target; Classification = discrete labels.

9.2 Model evaluation & metrics

- **Q: Explain confusion matrix and derive precision, recall, F1.** A: (Shown earlier).
- **Q: When use accuracy vs F1?** A: Accuracy for balanced classes; F1 for imbalanced or when both precision and recall matter.
- **Q: MAE vs MSE vs RMSE?** A: MAE linear, MSE quadratic, RMSE in units of target; MSE/RMSE penalize large errors more.
- **Q: How interpret R^2?** A: Fraction of variance explained; close to 1 = good explanatory, can be negative.

9.3 Algorithms deep dive

- **Q: Linear vs Logistic regression?** A: Linear outputs numeric; logistic outputs probability via sigmoid and uses cross-entropy.
- **Q: KNN working and issues?** A: Nearest neighbors voting/averaging; issues: scaling, speed, irrelevant features.
- **Q: Decision trees and overfitting solutions?** A: Depth limit, pruning, min samples, ensembles.
- **Q: Random Forest vs Gradient Boosting?** A: Bagging vs boosting; RF reduces variance; boosting reduces bias and often has higher accuracy but more tuning.
- **Q: SVM when to use?** A: High-dimensional small/medium data where margin helps; kernel trick for non-linear boundaries.

9.4 Model selection & tuning

- **Q: Bias-variance tradeoff?** A: (see earlier).
- **Q: Params vs hyperparams?** A: Params learned during training (weights). Hyperparams set by practitioner (regularization, k, depth).
- **Q: Regularization types?** A: L1 (Lasso) for sparsity, L2 (Ridge) for shrinkage, Elastic Net mix.
- **Q: How handle overfitting?** A: More data, regularization, CV, simpler models, ensembles, early stopping.

9.5 Practical implementation

- **Q: Data splitting?** A: Typical 70/15/15 or 60/20/20 (train/val/test).
- **Q: Cross-validation types?** A: K-fold, stratified, time-series CV, LOOCV.
- **Q: Handling imbalance?** A: Metrics, sampling, class weights, ensembles, thresholding.
- **Q: Feature engineering importance?** A: Often more impact than algorithm choice; scaling, encoding, creation, selection.

9.6 Advanced & business

- **Q: Ensemble methods?** A: Bagging (RF), Boosting (XGBoost), Stacking (meta-learner).
- **Q: XGBoost advantage?** A: Regularization, speed, handling missing data, feature importance.
- **Q: Gradient descent variants?** A: Batch, stochastic, mini-batch (tradeoff speed vs noise).
- **Q: How explain to stakeholders?** A: Start from business impact, avoid technical jargon, use visuals, give confidence measures.
- **Q: Production readiness?** A: Data checks, feature stability, monitoring, retraining plan, A/B testing.