**Model evaluation and selection**

**1. Introduction:**   
 Model evaluation is the process of assessing how well a machine learning model performs. Model selection involves choosing the best-performing model among several alternatives.

**Why Evaluate Models?**

* To ensure generalization to unseen data.
* To avoid overfitting/under fitting.
* To compare multiple algorithms.

**2. Holdout and Cross-Validation Methods**

### **a.** **Holdout Method**

### Split dataset into:

* + Training Set (e.g., 70–80%)
  + Test Set (e.g., 20–30%)
* Simple but may not be representative if the dataset is small.

### **b**. **K-Fold Cross-Validation**

### Data is split into k subsets.

* Model is trained k times, each time using a different fold as test data.
* Average performance across folds.

### **c. Stratified K-Fold**

### Ensures class distribution is maintained across folds.

* Useful in imbalanced classification problems.

### **d.** **Leave-One-Out Cross-Validation (LOOCV)**

### Each data point is used once as a test; all others as training.

* Best for small datasets; computationally expensive.

3. **Evaluation Metrics for Classification**

| **Metric** | **Formula** | **When to Use** |
| --- | --- | --- |
| **Accuracy** | (TP + TN) / (TP + TN + FP + FN) | Balanced datasets |
| **Precision** | TP / (TP + FP) | Minimize false positives |
| **Recall** | TP / (TP + FN) | Minimize false negatives |
| **F1 Score** | 2 × (Precision × Recall) / (Precision + Recall) | Balance precision and recall |
| **ROC Curve** | Graph of TPR (Recall) vs. FPR | Visual threshold performance |
| **AUC (Area under ROC)** | — | Higher is better |
| **Confusion Matrix** | 2×2 matrix of TP, FP, FN, TN | Visualize classification results |
|  |  |  |

4. **Bias-Variance Trade-Off**

* **High Bias**: Underfitting – model too simple.
* **High Variance**: Overfitting – model too complex.
* Goal: Find the sweet spot between bias and variance.

## 5. ****Model Selection Criteria****

## Best **cross-validation score**.

* **Consistency** across different random splits.
* **Interpretability** (if required).
* **Scalability** and **training time**.
* Use tools like **Grid Search**, **Random Search**, or **Bayesian Optimization**.

**TECHNIQUES TO IMPROVE CLASSIFICATION ACCURACY**

## ****1. Data Preprocessing****

## **Handle Missing Values**: Imputation or removal.

* **Remove Noise**: Apply filters or smoothing techniques.
* **Balance Class Distribution**:
  + **Oversampling** (e.g., SMOTE)
  + **Undersampling**
  + **Class weight adjustments**

## 2. ****Feature Engineering****

* **Feature Selection**: Remove irrelevant or redundant features.
  + Techniques: Mutual Information, Chi-square, Recursive Feature Elimination (RFE).
* **Feature Extraction**: PCA, LDA, or autoencoders.
* **Feature Scaling**: Normalize or standardize data (especially important for algorithms like SVM, KNN).

## 3. ****Model-Based Techniques****

* **Try Multiple Algorithms**: Compare decision trees, SVM, Random Forest, Gradient Boosting, etc.
* **Ensemble Methods**:
  + **Bagging** (e.g., Random Forest)
  + **Boosting** (e.g., XGBoost, AdaBoost, LightGBM)
  + **Stacking** (combine multiple models)

## 4. ****Hyperparameter Tuning****

* Use:
  + **Grid Search**
  + **Random Search**
  + **Bayesian Optimization**
* Tune key parameters such as:
  + Depth of trees
  + Learning rate
  + Regularization terms
  + Number of estimators

## 5. ****Regularization****

* Helps prevent overfitting:
  + **L1 Regularization (Lasso)**
  + **L2 Regularization (Ridge)**
* Applicable in models like logistic regression, linear regression, and neural networks.

## 6. ****Use of Advanced Models****

* Deep Learning (for large and complex datasets).
* Pretrained models (e.g., for NLP or image tasks).
* Transfer learning where applicable.

## 7. ****Error Analysis****

* Study misclassified samples.
* Find patterns in errors.
* May lead to new features or data preprocessing strategies.

## 8. ****Data Augmentation****

* Generate synthetic data to increase training samples.
* Common in image, audio, and text classification tasks.

## 9. ****Early Stopping (for iterative models)****

* Stop training when validation performance starts to degrade.
* Prevents overfitting.

## 10. ****Cross-Domain Knowledge****

* Incorporate domain expertise for better feature selection or interpretation.