#UpSkillWithKalpesh

Day 17

Data Science Unlocked

From Zero to Data Hero

Machine Learning
Part-2 : More Basics

Kalpesh Pathade
@DataSimplified

Machine Learning Part 2: More Basics

Type	@DataSimplified

These are just introduction notes. All these topics will be covered in very well detail along with code in upcoming notes.

I. Introduction to Testing and Validating, Hyperparameter Tuning, and Data Mismatch

Machine learning models require rigorous testing, validation, and tuning to ensure optimal performance. This document provides an in-depth discussion on three critical aspects:

- **Testing and Validating Models** – Ensuring that models generalize well and do not overfit or underfit.
- **Hyperparameter Tuning and Model Selection** – Optimizing the model's hyperparameters for better accuracy and efficiency.
- **Data Mismatch** – Understanding and mitigating issues when training and real-world data differ.

II. Testing and Validating

21 Importance of Testing and Validation

Testing and validation help assess a model's performance on unseen data. Without proper validation, models may memorize training data instead of learning general patterns, leading to overfitting.

22 Splitting Data for Validation

22.1 Standard Splitting Ratios

- **Train-Test Split**: Typically, 80% of data is used for training and 20% for testing.
- **Train-Validation-Test Split**:
  - Training Set: 60-70%
  - Validation Set: 10-20%
  - Test Set: 20-30%

22.2 Splitting Large Datasets

For extremely large datasets, a smaller portion of data can be used for validation and testing:

- **98-1-1 Split**: 98% training, 1% validation, 1% testing (suitable for datasets with millions of samples).

23 Cross-Validation Techniques

23.1 K-Fold Cross-Validation

- The dataset is divided into K folds (e.g., 5 or 10).
- The model is trained on K-1 folds and tested on the remaining fold.
- The process repeats K times, and results are averaged.

23.2 Stratified K-Fold Cross-Validation

- Ensures class distribution remains the same across all folds.
- Useful for imbalanced classification problems.

2.3.3 Leave-One-Out Cross-Validation (LOO-CV)

- Uses every sample as a test set once while training on the rest.
- Computationally expensive but provides an unbiased estimate.

2.4 Model Evaluation Metrics

2.4.1 Regression Models

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- R^2 Score

2.4.2 Classification Models

- Accuracy
- Precision, Recall, F1-Score
- ROC-AUC Score
- Confusion Matrix

2.4.3 Clustering Models

- Silhouette Score
- Davies-Bouldin Index
- Adjusted Rand Index

III. Hyperparameter Tuning and Model Selection

3.1 Hyperparameter vs. Parameter

- **Parameters**: Learned from data (e.g., weights in a neural network).

- **Hyperparameters**: Set before training (e.g., learning rate, number of layers in a neural network).

3.2 Hyperparameter Tuning Techniques

3.2.1 Grid Search

- Exhaustively searches all possible hyperparameter combinations.
- Computationally expensive.

3.2.2 Random Search

- Randomly samples hyperparameters from a given range.
- Faster than Grid Search.

3.2.3 Bayesian Optimization

- Uses previous evaluations to predict the best hyperparameter values.
- More efficient than Grid and Random Search.

3.2.4 Automated Hyperparameter Tuning

- Uses tools like **Optuna**, **Hyperopt**, or **AutoML**.
- Reduces manual effort in hyperparameter selection.

3.3 Model Selection

- Choosing the best model based on validation metrics.
- Comparing multiple models (e.g., Decision Tree vs. Random Forest).
- Ensuring the model generalizes well to new data.

IV. Data Mismatch

4.1 What is Data Mismatch?

Data mismatch occurs when the training data distribution differs from real-world data, leading to poor model performance.

4.2 Causes of Data Mismatch

- **Domain Shift**: Training data is collected from a different source than real-world data.
- **Feature Distribution Shift**: The statistical properties of input features change over time.
- **Sampling Bias**: The training data is not representative of the target population.
- **Data Quality Issues**: Missing or noisy data in real-world scenarios.

4.3 Handling Data Mismatch

4.3.1 Further Splitting Data

- Instead of a single train-test split, data can be divided into multiple sets:
  - **Training Set**: Used for initial model training.
  - **Validation Set**: Used for hyperparameter tuning.
  - **Real-World Test Set**: Collected separately from real-world scenarios.
  - **Continuous Monitoring Set**: Used for real-time tracking of model performance.

4.3.2 Collecting More Representative Data

- Ensuring data is sampled from diverse environments.
- Using domain adaptation techniques to fine-tune the model.

4.3.3 Data Augmentation

- Generating synthetic data to increase variability.
- Useful for handling class imbalances.

4.3.4 Transfer Learning

- Using pre-trained models and fine-tuning them on new data.
- Reduces data mismatch when limited real-world data is available.