Practical Tips for Machine Learning

Handling Messy Data with Confidence

Endri Raco

2024-09-28

Introduction to Real-World Data in ML

- Real-World Data vs. Synthetic Data: Explain the difference, emphasizing why real-world data is often messy, incomplete, and noisy.
- Challenges: Highlight typical problems like missing values, outliers, imbalanced data, high dimensionality, etc.
- Key ML Workflow Stages: Summarize the ML pipeline, including data collection, cleaning, feature engineering, model selection, and evaluation.

Data Collection and Understanding (15 min)

- Exploratory Data Analysis (EDA):
- Explain the importance of understanding your data's distribution, summary statistics, and relationships.
- Tools: Pandas, Matplotlib, Seaborn.
- Show a brief EDA example (e.g., correlation heatmap, histograms).
- Understanding Data Types: Distinguish between categorical, continuous, ordinal data and how each type requires different handling.

Handling Missing Data (10 min)

- Identify Missing Data: Briefly show techniques to find missing data (e.g., Pandas .isnull() function).
- Strategies:
  - Deletion: Dropping rows/columns with missing data.
  - Imputation: Filling missing values using statistical methods (mean, median) or predictive models (k-NN, regression imputation).
  - Special Cases: When missing data might carry meaning (missing not at random).

Dealing with Imbalanced Datasets (10 min)

- Understanding Class Imbalance: Explain the problem of imbalanced datasets in classification tasks (e.g., fraud detection, medical diagnoses).
  - Strategies:
- Resampling Techniques: Oversampling (SMOTE), undersampling.
- Algorithm-Level Solutions: Adjusting class weights, using algorithms like XGBoost or Random Forests.
- Evaluation Metrics: Using Precision, Recall, F1-score, and ROC-AUC instead of accuracy for imbalanced datasets.

Feature Engineering (15 min)

- Importance of Feature Engineering: Explain how domain knowledge can influence the model's performance and why well-engineered features can often be more impactful than complex models.
- Techniques:
- Transformations: Logarithmic or polynomial transformations to normalize skewed data.
- Encoding Categorical Variables: One-hot encoding, label encoding.
- Scaling and Normalization: Standardization (Z-score), Min-Max scaling.
- Feature Selection: Removing irrelevant or redundant features using techniques like correlation analysis, mutual information, or algorithms (e.g., Lasso).

Model Selection and Evaluation (10 min)

- Cross-Validation: Importance of using k-fold cross-validation to avoid overfitting and assess model performance on unseen data.
    - Choosing the Right Algorithm:
    - Simple vs. Complex Models: Start with simpler models (e.g., linear regression, decision trees) before moving to more complex ones (e.g., deep learning).
    - Hyperparameter Tuning: Use grid search or randomized search to find the best parameters.

Dealing with Overfitting and Underfitting (10 min)

- Overfitting: Explain why it happens (too complex models, too little data) and how to detect it (e.g., high training accuracy but low test accuracy).
  - Strategies:
  - Regularization: L1, L2 regularization.
  - Pruning Decision Trees: Reduce complexity to avoid overfitting.
  - Early Stopping: In neural networks, stop training when the model's performance on the validation set starts to deteriorate.
- Underfitting: Discuss cases where models are too simple and strategies to fix it (e.g., adding more features or increasing model complexity).

Model Interpretability and Explainability (10 min)

- Why Interpretability Matters: Especially in domains like healthcare and finance where understanding model decisions is crucial.
- Techniques:
- Feature Importance: Using models like Random Forests or SHAP (SHapley Additive exPlanations) to explain predictions.
- LIME (Local Interpretable Model-Agnostic Explanations): An approach to explain individual predictions in complex models.

Practical Example/Case Study (15 min)

- End-to-End ML Example: Walk through a practical case study, demonstrating:
- Loading data, performing EDA, handling missing values, feature engineering, training a model, and evaluating it.
- Example Dataset: Use a common open-source dataset (e.g., the Titanic dataset, a healthcare dataset) to showcase the process.

Conclusion and Q&A (5 min)

- Summary of Key Takeaways: Emphasize the importance of data preparation, understanding the problem domain, and iterating through different approaches.
- Best Practices: Encourage staying updated with new techniques, regularly cross-validating, and focusing on explainability when deploying models in real-world settings.
- *Q&A*: Open the floor for questions.