Introduction to Model Training in Machine Learning

Understanding the Basics of Data Preparation, Model Training, and Evaluation

Sammie Omranian

Summer 2024

Model Training

- Data Preparation
- Model Selection
- Model Training
- Model Evaluation
- Hyperparameter Tuning

Data Preparation

Data preparation is a crucial step in the machine learning pipeline that significantly impacts the performance of the model.

- Data Collection
  - Data Sources: Databases, APIs, CSV files
- Data Cleaning
  - Cleaning steps such as Handling missing values, Removing duplicates, Addressing outliers
- Feature Engineering
  - Creating relevant features and transforming existing ones can reveal hidden patterns in the data, leading to better model performance. Like normalization and scaling.
- Data Splitting
  - Training, Testing, and Validation sets

Data Preparation - Importance of data preparation

Example:

Before cleaning

ID	Name	Age	Income	JoinDate
1	John	28	55000	12/01/2015
2	Alice		72000	15-03-2016
3	Bob	34	62000	2017/05/20
4	Charlie	29		2018.07.10

Steps:

- Handle Missing Values: Fill missing age with the mean, missing income with the median.
- Standardize Date Format: Convert all dates to YYYY-MM-DD format.
- Remove Duplicates: Ensure no duplicate rows are present.

After cleaning

ID	Name	Age	Income	JoinDate
1	John	28	55000	2015-12-01
2	Alice	31	72000	2016-03-15
3	Bob	34	62000	2017-05-20
4	Charlie	29	61000	2018-07-10

Model Selection

- Model selection is the process of choosing the most appropriate machine learning algorithm for your specific problem. The choice of model impacts the performance, accuracy, and interpretability of your results.

- Factors to Consider
  - Type of Problem:
    - Regression: Predicting a continuous output (e.g., house prices).
    - Classification: Predicting a categorical output (e.g., spam detection).

  - Size of Data:
    - Small Datasets: Simpler models like Linear Regression or Decision Trees may suffice.
    - Large Datasets: More complex models like Neural Networks or Ensemble Methods may be necessary.

Model Selection

- Feature Types:
  - Numerical: Algorithms like Linear Regression, Ridge, and Lasso work well.
  - Categorical: Algorithms like Decision Trees and Random Forests handle categorical data effectively.
- Model Interpretability:
  - High: Linear Regression, Decision Trees.
  - Low: Neural Networks, Ensemble Methods.
- Computational Resources:
  - Limited: Simpler models with less computational requirements (e.g., Logistic Regression).
  - Abundant: Complex models that require significant computational power (e.g., Deep Learning).

Model Training

- Model training is the process where a machine learning algorithm learns to make predictions or decisions based on data.

- Model Training Process

1. Feeding Data:
    - The training data, which consists of input features and corresponding target values, is fed into the machine learning algorithm.
    - Example: In supervised learning, the data is split into input variables (X) and output variable (y).

2. Adjusting Weights:
    - The model makes predictions and adjusts the weights (parameters) based on the difference between the predicted values and the actual target values.
    - Example: In Linear Regression, the weights (coefficients) are adjusted to minimize the difference between the predicted and actual values.

Model Training

3. Minimizing Loss Function:
    - The loss function measures the difference between the predicted and actual values. The goal of training is to minimize this loss function.
    - Common Loss Functions:
        - Mean Squared Error (MSE): Commonly used for regression problems.
        - Cross-Entropy Loss: Commonly used for classification problems.
    - Example: In Gradient Descent, the algorithm iteratively adjusts the weights to find the minimum value of the loss (cost) function.

Model Training - Overfitting and Underfitting

- Overfitting:
- When a model learns the training data too well, including the noise and outliers, leading to poor generalization on new data.
- How to detect it? High accuracy on training data but low accuracy on testing data.
- Solutions:
  - Simplify the Model: Reduce the complexity of the model like reducing the number of features (Feature Selection), or use simpler algorithms.
  - Regularization: Techniques like Lasso (L1) and Ridge (L2) regularization add a penalty for larger coefficients.
  - Cross-Validation: Use cross-validation techniques to ensure the model generalizes well.

Model Training - Overfitting and Underfitting

- Underfitting:
- When a model is too simple to capture the underlying patterns in the data, leading to poor performance on both training and testing data.

- How to detect it? Low accuracy on both training and testing data.

- Solutions:

  - Increase Model Complexity: Use a more complex model that can capture the underlying patterns.
  - Feature Engineering: Create new features or transform existing ones to provide more information to the model.

Model Evaluation

- Model Evaluation Metrics
  - Regression Metrics: MAE, MSE, RMSE, R-squared
  - Classification Metrics: Accuracy, Precision, Recall, F1-score

- **Regression Metrics**

  1. Mean Absolute Error (MAE)
  - The average of the absolute differences between the predicted values and the actual values.
  - Formula

$$MSE = \sum_{i=1}^{n} \frac{|y_i - y_i'|}{n}$$

  - MAE measures the average magnitude of the errors in a set of predictions.

Model Evaluation

2. Mean Squared Error (MSE)

The average of the squared differences between the predicted values and the actual values.

- Formula

$$MSE = \sum_{i=1}^{n} \frac{(y_i - y_i')^2}{n}$$

3. Root Mean Squared Error (RMSE)

The square root of the average of the squared differences between the predicted values and the actual values.

- **Formula**

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{\left(y_i - y_i'\right)^2}{n}}$$

Model Evaluation

3. R-squared (R^2)

The proportion of the variance in the dependent variable that is predictable from the independent variables.

Formula

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

R^2 indicates how well the model fits the data; values range from 0 to 1, with 1 indicating perfect fit.

We will talk about classification metrics when introducing classification problems.

Hyperparameter Tuning

- **Hyperparameters:** Parameters that are set before the learning process begins and are not learned from the data. Examples include the learning rate for training a neural network, the number of trees in a random forest, or the regularization parameter in a regression model.

- **Hyperparameter Tuning:** The process of finding the optimal set of hyperparameters that yield the best performance of the model on the validation data.

- Importance:
  - Model Performance: Hyperparameters can significantly impact the performance of machine learning models. Proper tuning can lead to substantial improvements in accuracy and generalization.

  - Prevent Overfitting/Underfitting: Properly tuned hyperparameters help in balancing the complexity of the model, avoiding both overfitting and underfitting.

  - Model Efficiency: Optimal hyperparameters can also reduce training time and computational cost.

Hyperparameter Tuning

- Grid Search:
  - An exhaustive search method that tries every possible combination of hyperparameters within the specified parameter grid.
  - Process: Define a grid of hyperparameter values and train the model on each combination, evaluating performance using cross-validation.
  - Pros: Comprehensive, finds the best combination of hyperparameters within the specified grid.
  - Cons: Computationally expensive, especially with large grids and complex models.