Introduction to Machine Learning and Classification Algorithms

July 20, 2025

Introduction to Machine Learning

Machine learning is a subset of artificial intelligence (AI) that enables computers to learn from and make decisions based on data without explicit programming. It revolves around the idea of building algorithms that can automatically detect patterns in data and make predictions or decisions based on those patterns.

- Supervised learning
- Unsupervised learning
- Reinforcement learning

Types of Machine Learning

Machine learning algorithms are typically categorized into three main types:

- **Supervised Learning**: The model is trained on labeled data and learns to map inputs to outputs.
- **Unsupervised Learning**: The model works with unlabeled data to find hidden patterns.
- **Reinforcement Learning**: The model learns by interacting with an environment and receiving feedback in the form of rewards or punishments.

Applications of Machine Learning

Machine learning is used in a variety of fields:

- **Healthcare**: Disease prediction, medical imaging, personalized treatment plans.
- **Finance**: Fraud detection, algorithmic trading, risk management.
- **Retail**: Recommendation systems, demand forecasting, customer segmentation.
- **Self-Driving Cars**: Object detection, decision-making, path planning.

Classification Algorithms Overview

Classification is a type of supervised learning where the goal is to predict categorical labels. Common classification algorithms include:

- Logistic Regression
- Decision Trees
- Random Forests
- k-Nearest Neighbors (k-NN)
- Support Vector Machines (SVM)
- Naive Bayes

Performance Metrics in Classification

To evaluate the performance of a classifier, we use several metrics:

- **Accuracy**: The percentage of correct predictions.
- **Precision**: The proportion of positive results that are actually correct.
- **Recall**: The proportion of actual positive cases that were identified correctly.
- **F1-Score**: The harmonic mean of precision and recall.
- **AUC-ROC**: Measures the trade-off between true positive rate and false positive rate.

Logistic Regression

Logistic Regression is used for binary classification problems. The output is modeled using a sigmoid function to predict probabilities that a sample belongs to a particular class.

- Mathematical model: $P(y = 1|X) = \frac{1}{1+e^{-(w^T X+b)}}$
- It is a linear model, but with a non-linear output (via sigmoid function).
- Often used in cases where the output is binary (e.g., spam vs. not spam).

Logistic Regression - Example

- Given a feature X, logistic regression calculates the probability of belonging to a particular class.
- For multi-class problems, softmax regression (a generalization of logistic regression) is used.
- Example: Predicting whether an email is spam or not based on features like word frequency and sender information.

Decision Trees

A Decision Tree is a non-linear model used for both classification and regression tasks. It works by splitting data at each node based on feature values, creating a tree structure.

- The tree is built by selecting the feature that maximizes the information gain (or minimizes the Gini impurity).
- It's easy to interpret and visualize.
- Overfitting is a common issue in decision trees, especially with deep trees.

Decision Trees - Example

In this example, the decision tree might classify whether a customer will buy a product based on features like age, income, and browsing history.

- The tree splits on features that provide the most significant reduction in uncertainty.
- Leaves of the tree represent class labels (e.g., 'Buy' or 'Don't Buy').

Random Forests

A Random Forest is an ensemble method that creates multiple decision trees and combines their outputs to make a final prediction.

- Reduces overfitting compared to a single decision tree.
- Each tree is trained on a random subset of the data (bootstrapping).
- Predictions are made through majority voting (for classification) or averaging (for regression).

k-Nearest Neighbors (k-NN)

k-NN is a non-parametric method used for classification and regression. It works by finding the 'k' nearest data points in the feature space and predicting the class based on the majority vote of these neighbors.

- The distance metric (e.g., Euclidean distance) is used to identify the nearest neighbors.
- Sensitive to the choice of k and the scale of the data.
- Works well for problems with high-dimensional feature spaces.

k-Nearest Neighbors - Example

- Suppose you want to classify a new data point based on features like height and weight.
- The classifier looks at the 'k' nearest data points and assigns the most frequent class label.

Support Vector Machines (SVM)

SVM is a powerful classification algorithm that finds the hyperplane that best separates the data into two classes. The goal is to maximize the margin between the classes while minimizing classification errors.

- The algorithm relies on support vectors (the data points closest to the hyperplane).
- Non-linear SVMs use the kernel trick to map data into higher-dimensional spaces.
- SVMs are effective in high-dimensional spaces and are robust to overfitting.

Support Vector Machines - Example

- Suppose you have two classes: “Cats” and “Dogs”, represented by points in a 2D feature space.
- The SVM algorithm finds the best hyperplane (or line in 2D) that maximizes the margin between the two classes.

Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem. It assumes that features are conditionally independent given the class label.

- Often used for text classification (e.g., spam filtering).
- Simple and computationally efficient.
- Despite the "naive" assumption, it often works surprisingly well in practice.

Naive Bayes Example

- Given features: {Age, Income, Purchase} for customer classification.
- Example dataset:
  - {30, 50k, Yes}
  - {40, 60k, No}
  - {25, 55k, Yes}
- Compute conditional probabilities using Bayes' Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- Calculate probabilities for each class (e.g., "Yes" or "No") and assign the class with the higher probability.
- Assumes feature independence, simplifying calculation.

Naive Bayes Example

- Given features: {Age, Income, Purchase} for customer classification.
- Example dataset:
  - {30, 50k, Yes}
  - {40, 60k, No}
  - {25, 55k, Yes}
- Compute conditional probabilities using Bayes' Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- Calculate probabilities for each class (e.g., "Yes" or "No") and assign the class with the higher probability.
- Assumes feature independence, simplifying calculation.

Hyperparameter Tuning

- Hyperparameters: Parameters that control the learning process (e.g., learning rate, tree depth).
- **Grid Search:** Exhaustively searches through hyperparameter space.
- **Random Search:** Randomly selects combinations of hyperparameters.
- **Bayesian Optimization:** Uses probabilistic models to select hyperparameters.
- **Cross-validation:** Helps estimate model performance during tuning.
- **Example for Decision Tree:** Vary max depth and min samples per leaf.

Cross-Validation

- Cross-validation: Technique to assess the model's performance.
- **K-fold Cross-Validation:**
  - Split data into K subsets.
  - Train model on K-1 subsets, test on the remaining subset.
  - Repeat for each subset, average the performance metrics.
- **Stratified Cross-Validation:** Ensures balanced class distribution across folds.
- Reduces bias, provides a better estimate of model performance.

Feature Engineering

- **Feature Selection:** Identify and use the most important features.
- **Techniques:**
  - Filter methods (e.g., Chi-square, correlation)
  - Wrapper methods (e.g., Recursive Feature Elimination)
  - Embedded methods (e.g., Lasso, decision trees)
- **Feature Transformation:** Modify features to improve model performance.
  - Scaling: Normalize/standardize data.
  - Encoding: Convert categorical features (e.g., One-hot encoding).
- **Dimensionality Reduction:** Techniques like PCA to reduce features while preserving data variability.

Real-World Example: Self-Driving Cars

- Machine learning algorithms are used in self-driving cars for:
  - **Object Detection:** Identifying pedestrians, cyclists, and other vehicles.
  - **Path Planning:** Deciding the best route while avoiding obstacles.
  - **Decision Making:** Real-time decision-making to navigate complex traffic scenarios.
- **Algorithms Used:**
  - Convolutional Neural Networks (CNN) for image recognition.
  - Reinforcement Learning for optimal driving policies.
  - Random Forest for classifying road signs and conditions.

Real-World Example: Healthcare Diagnostics

- ML algorithms are widely used for diagnosing diseases:
  - **Cancer Detection:** Predictive models trained on medical imaging to identify tumors.
  - **Heart Disease:** Algorithms analyze patient data to predict heart conditions.
  - **Predicting Outcomes:** ML models are used to predict recovery rates and hospital readmissions.
- **Algorithms Used:**
  - Neural Networks for image analysis (X-ray, MRI scans).
  - Support Vector Machines (SVM) for classifying diseases.
  - Decision Trees for patient risk stratification.

Challenges in Machine Learning

- **Data Quality:** Missing or noisy data can significantly impact model accuracy.
- **Overfitting:** Models that perform well on training data but poorly on unseen data.
- **Model Interpretability:** Complex models like deep learning can be difficult to interpret.
- **Bias:** Models may inherit biases from the data, leading to unfair predictions.
- **Computational Cost:** Some ML models require significant computational resources.

Thank you for your attention!

Feel free to ask any questions or reach out for more information.