Machine Learning Internals

TRAINING

contents

Machine Learning Internals

TRAINING

course contents

- Introduction Day1
- Machine Learning for Predictive Data Analytics
- Data to Insights to Decisions
- Data Exploration
- Information-based Learning Day2
- Similarity-based Learning
- Probability-based Learning
- Error-based Learning
- Evaluation
- Software Tools Day3
- Linear regression
- Logistic regression
- Support Vector Machine
- Decision Trees
- Dimensionality Reduction. Day4
- Introduction to Deep Learning
- Deep Neural Network
- Clustering
- Gradient Boosting Techniques Day5
- Time Series Models

Description:

- Machine learning is the science of getting computers to act without being explicitly programmed. In the past decade, machine learning has given us self-driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human genome. Machine learning is so pervasive today that you probably use it dozens of times a day without knowing it. Many researchers also think it is the best way to make progress towards human-level AI. This trainig session provides a deep dive into machine learning, datamining, and statistical pattern recognition. Topics include: (i) Supervised learning (parametric/non-parametric algorithms, support vector machines, kernels, neural networks). (ii) Unsupervised learning (clustering, dimensionality reduction, recommender systems, deep learning). (iii) Best practices in machine learning (bias/variance theory; innovation process in machine learning and AI). The course will also draw from numerous case studies and applications, so that you'll also learn how to apply learning algorithms to building smart robots (perception, control), text understanding (web search, anti-spam), computer vision, medical informatics, audio, database mining, and other areas.

Key Skills:

- Measuring and Tuning performance of ML algorithms
- You'll learn about not only the theoretical underpinnings of learning, but also gain the practical know-how needed to quickly and powerfully apply these techniques to new problems
- Most effective machine learning techniques
- You will learn how to Prototype and then productionize
- Best practices in innovation as it pertains to machine learning and AI
- Use tools like Scikit for ML tasks

Prerequisites:

- Experience in Programming
- An understanding of Intro to Statistics would be helpful.
- A familiarity with Probability Theory, Calculus, Linear Algebra and Statistics is required
- Working Knowledge of Python

Instructional Method:

- This is an instructor led course provides lecture topics and the practical application of Machine Learning and the underlying technologies. It pictorially presents most concepts and there is a detailed case study that strings together the technologies, patterns and design.

Machine Learning Internals

- ***Introduction***
  - Model selection
  - Supervised learning
  - Discovering graph structure
  - Types of machine learning
  - Machine learning: what and why?
  - Parametric vs non-parametric models
  - No free lunch theorem
  - Linear regression
  - Some basic concepts in machine learning
  - Discovering clusters
  - Classification
  - Regression
  - Matrix completion
  - Logistic regression
  - Parametric models for classification and regression
  - The curse of dimensionality
  - Overfitting
  - Unsupervised learning
  - Discovering latent factors
  - A simple non-parametric classifier: K-nearest neighbors
- ***Machine Learning for Predictive Data Analytics***
  - Predictive Data Analytics Tools
  - How Does Machine Learning Work?
  - The Road Ahead
  - What Can Go Wrong with Machine Learning?
  - The Predictive Data Analytics Project Lifecycle: CRISP-DM
  - What Is Machine Learning?
  - What Is Predictive Data Analytics?
- ***Data to Insights to Decisions***

- Different Types of Data
- Different Types of Features
- Designing the Analytics Base Table
- Designing and Implementing Features
- Assessing Feasibility
- Converting Business Problems into Analytics Solutions
- Case Study: Motor Insurance Fraudmotor
- Implementing Features
- Handling Time

- ***Data Exploration***
  - Outliers
  - Handling Missing Values
  - Handling Outliers
  - Missing Values
  - Irregular Cardinality
  - Handling Data Quality Issues
  - The Data Quality Report
  - The Normal Distribution
  - Identifying Data Quality Issues
  - Getting to Know the Data
  - Advanced Data Exploration
    - Measuring Covariance and Correlation
    - Visualizing Relationships Between Features
    - Binning
    - Data Preparation
    - Normalization

- ***Information-based Learning***
  - Shannon’s Entropy Model
  - Handling Continuous Descriptive Features
  - Decision Trees
  - Predicting Continuous Targets
  - Extensions and Variations

  - Fundamentals
  - Information Gain
  - Big Idea
  - Standard Approach: The ID Algorithm
  - Tree Pruning
  - Alternative Feature Selection and Impurity Metrics
- ***Similarity-based Learning***
  - Standard Approach: The Nearest Neighbor Algorithm
  - Predicting Continuous Targets
  - Fundamentals
  - Other Measures of Similarity
  - Extensions and Variations
  - Data Normalization
  - Feature Space
  - Big Idea
  - Measuring Similarity Using Distance Metrics
  - Feature Selection
  - Handling Noisy Data
  - Efficient Memory Search
- ***Probability-based Learning***
  - Big Idea
  - Smoothing
  - Extensions and Variations
  - Bayes’ Theorem
  - Bayesian Networks
  - Continuous Features: Probability Density Functions
  - Continuous Features: Binning
  - Bayesian Prediction
  - Conditional Independence and Factorization
  - Fundamentals
  - Standard Approach: The Naive Bayes Model
- ***Error-based Learning***

  - Setting the Learning Rate Using Weight Decay
  - Error Surfaces
  - Multinomial Logistic Regression
  - Modeling Non-linear Relationships
  - Handling Categorical Descriptive Features
  - Interpreting Multivariable Linear Regression Models
  - Simple Linear Regression
  - Big Idea
  - Handling Categorical Target Features: Logistic Regression
  - Extensions and Variations
  - Fundamentals
  - Choosing Learning Rates and Initial Weights
  - Standard Approach: Multivariable Linear Regression with Gradient Descent
  - Gradient Descent
  - Multivariable Linear Regression
  - Measuring Error
- ***Evaluation***
  - Performance Measures: Prediction Scores
  - Designing Evaluation Experiments
  - Evaluating Models after Deployment
  - Performance Measures: Multinomial Targets
  - Extensions and Variations
  - Fundamentals
  - Performance Measures: Continuous Targets
  - Performance Measures: Categorical Targets
  - Big Idea
  - Standard Approach: Misclassification Rate on a Hold-out Test Set
- ***Software Tools***
  - scikit learn
  - Scikit Learn
    - Introduction to scikit learn

        - Types
        - Model persistence
        - setting up scikit learn
        - Machine learning: the problem setting
        - Learning and predicting
    - Tensorflow
        - Feeding Data to the Training Algorithm
        - Saving and Restoring Models
        - Introduction and setting up
        - Creating Your First Graph and Running It in a Session
        - Lifecycle of a Node Value
        - Managing Graphs

- ***Linear regression***
    - Regularization effects of big data
    - Bayesian inference when ?^2 is unknown *
    - Model specification
    - Numerically stable computation *
    - Computing the posterior
    - Geometric interpretation
    - Convexity
    - Connection with PCA *
    - Maximum likelihood estimation (least squares)
    - Bayesian linear regression
    - Derivation of the MLE
    - Computing the posterior predictive
    - EB for linear regression (evidence procedure)
    - Ridge regression
    - Basic idea
    - Robust linear regression *
    - Introduction
- ***Logistic regression***
    - Residual analysis (outlier detection) *

  - Generative vs discriminative classifier
  - Multi-class logistic regression
  - Online learning and regret minimization
  - Iteratively reweighted least squares (IRLS)
  - Quasi-Newton (variable metric) methods
  - Newton’s method
  - Bayesian logistic regression
  - A Bayesian view
  - Laplace approximation
  - l2 regularization
  - Gaussian approximation for logistic regression
  - Approximating the posterior predictive
  - Derivation of the BIC
  - Steepest descent
  - Introduction
  - MLE
  - Model specification
  - Online learning and stochastic optimization
  - Dealing with missing data
  - Fisher’s linear discriminant analysis (FLDA) *
  - Model fitting
  - Stochastic optimization and risk minimization
  - Pros and cons of each approach
  - The LMS algorithm
  - Logistic regression
  - The perceptron algorithm

- ***Support Vector Machine***
  - Linear SVM Classification
  - SVM Regression
  - Nonlinear SVM Classification
  - Under the Hood

  - Kernels

- Smoothing kernels
- Kernels for comparing documents
- The kernel trick
- SVMs for classification
- Kernelized ridge regression
- SVMs for regression
- Linear kernels
- Kernel machines
- Introduction
- Comparison of discriminative kernel methods
- Kernelized nearest neighbor classification
- Using kernels inside GLMs
- A probabilistic interpretation of SVMs
- Kernel functions
- RBF kernels
- Mercer (positive definite) kernels
- Kernel density estimation (KDE)
- Kernel PCA
- String kernels
- LVMs, RVMs, and other sparse vector machines
- Kernels for building generative models
- Choosing C
- Kernelized K-medoids clustering
- Pyramid match kernels
- Kernel regression
- Kernels derived from probabilistic generative models
- Locally weighted regression
- Summary of key points
- Support vector machines (SVMs)
- From KDE to KNN
- Matern kernels

- ***Decision Trees***

- Regularization Hyperparameters
- Regression
- Making Predictions
- Training and Visualizing a Decision Tree
- Gini Impurity or Entropy?
- Estimating Class Probabilities
- Decision Trees
- Instability
- Computational Complexity
- The CART Training Algorithm

- ***Dimensionality Reduction.***
  - SVD
  - PCA
  - Kernel PCA
  - LLE
  - Main Approaches for Dimensionality Reduction
  - The Curse of Dimensionality
  - CUR
- ***Introduction to Deep Learning***
  - Parameter Hyperspace
  - Minimizing Cost Entropy
  - Normalized Inputs And Initial Weights
  - Measuring Performance
  - Transition Into Practical Aspects Of Learning
  - Stochastic Gradient Descent
  - Training your Logistic Classifier
  - Transition: Overfitting -> Dataset Size
  - Momentum And Learning Rate Decay
  - Supervised Classification
  - Solving Problems
  - Lather Rinse Repeat
  - Optimizing A Logistic Classifier

  - Cross Entropy
  - What is Deep Learning
- ***Deep Neural Network***
  - "2-layer" neural network
  - Network Of ReLUs
  - Dropout
  - Intro to Deep Neural Network
  - No Neurons
  - Backprop
  - Regularization Intro
  - Linear Models Are Limited
  - The Chain Rule
  - Dropout Pt-2
  - Regularization
  - Training A Deep Learning Network
- ***Clustering***
  - Hierarchical risk parity
  - The expectation-maximization algorithm
  - k-Means clustering
  - Hierarchical DBSCAN
  - Gaussian mixture models
  - Density-based clustering
  - Visualization – dendrograms
  - Hierarchical clustering
  - DBSCAN
  - Evaluating cluster quality
- ***Gradient Boosting Techniques***
  - Adaptive boosting
    - AdaBoost with sklearn
    - The AdaBoost algorithm
  - Gradient boosting machines

        - How to use gradient boosting with sklearn
        - Shrinkage and learning rate
        - How to tune parameters with GridSearchCV
        - How to test on the holdout set
        - How to train and tune GBM models
        - Parameter impact on test scores
        - Ensemble size and early stopping
        - Subsampling and stochastic gradient boosting
    - Fast scalable GBM implementations
        - Randomized grid search
        - DART – dropout for trees
        - How algorithmic innovations drive performance
        - Second-order loss function approximation
        - Treatment of categorical features
        - How to create binary data formats
        - Depth-wise versus leaf-wise growth
        - Regularization
        - Additional features and optimizations
        - GPU-based training
        - Objectives and loss functions
        - How to tune hyperparameters
        - Simplified split-finding algorithms
        - Cross-validation results across models
        - How to use XGBoost, LightGBM, and CatBoost
        - Learning parameters
        - How to evaluate the results
- ***Time Series Models***
    - Multivariate time series models
        - Testing for cointegration
        - The vector autoregressive (VAR) model
        - Cointegration – time series with a common trend
        - How to use cointegration for a pairs-trading strategy

  - How to use the VAR model for macro fundamentals forecasts
  - Systems of equations
- Analytical tools for diagnostics and feature extraction
  - How to diagnose and achieve stationarity
  - How to compute rolling window statistics
  - How to decompose time series patterns
  - How to diagnose and address unit roots
  - Moving averages and exponential smoothing
  - How to apply time series transformations
  - Time series transformations
  - How to measure autocorrelation
  - Unit root tests
- Univariate time series models
  - Selecting the lag order
  - How to build ARIMA models and extensions
  - How to identify the number of lags
  - How to build autoregressive models
  - How to diagnose model fit
  - How to identify the number of AR and MA terms
  - Adding features – ARMAX
  - Adding seasonal differencing – SARIMAX
  - How to forecast macro fundamentals
  - How to build moving average models
  - Generalizing ARCH – the GARCH model
  - The autoregressive conditional heteroskedasticity (ARCH) model
  - The relationship between AR and MA models
  - How to identify the number of lags
  - How to build a volatility-forecasting model
  - How to use time series models to forecast volatility
- SOTA
  - CNN
  - ALSTM-FQN

- LSTM
- LSTM-FQN