Machine Learning using Python

Training Content



**What is Machine Learning?**

Arthur Samuel, a pioneer in Artificial Intelligence (AI) defined machine learning as "field of study that gives computers the ability to learn without being explicitly programmed". The field of machine learning – a branch of AI is concerned with the question of how to construct computer programs that automatically improve with experience. In the recent years, many machine learning applications have been developed ranging from data mining that learn to detect fraudulent credit card transactions to information filtering systems that learn reading preferences, to autonomous vehicles that learn to drive on public roads, optical character recognition systems that identify and read the text and objects from images, medical diagnostics and many more. Many researchers also think it is the best way to make progress towards human-level AI.

“If a typical person can do a mental task with less than one second of thought, we can probably automate it using AI either now or in the near future.” - [Andrew Ng](https://hbr.org/search?term=andrew+ng), Chief Scientist of Baidu

**Overview of the Training Course**

This course provides a broad introduction to machine learning, data mining, and statistical pattern recognition. Topics include: (i) Supervised learning (ii) Unsupervised learning (iii) Best practices in machine learning. 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. You will spend 40% time on theory lectures and case-studies and 60% on hands on lab exercises. Lab exercises will be based on data from open data projects from government institutions or companies. Python will be the primary programming languages of choice for lab exercises and demos.

**Duration of the Training Course: 3 Days**

**Mode: Instructor led classroom based training, hands on oriented.**

**Prerequisite**

* **Basic knowledge of statistics, linear algebra are required to understand and implement some of the ML algorithms. You may revise the textbooks or online materials on these subjects.**
* Strong programming knowledge in Python is required to perform the exercises during the class.
* We strongly recommend to take one of the following online courses prior to attending the class. These course take as much 8 weeks to finish. So, you may need to plan ahead.
  + Machine Learning by Andrew Ng available in [coursera.org](http://coursera.org/)
  + Statistical Learning by Trevor Hastie and Rob Tibshirani available in [online.stanford.edu](http://online.stanford.edu/)
  + Analytics Edge available at [edx.org](http://edx.org/)

# Day 1: What is Machine Learning and Data Preparation and Exploration using Numpy, Pandas and Matplotlib

|  |  |
| --- | --- |
| Topic | Description |
| Overview of Data Science | * BI versus Data Science * Drivers of Big Data * Emerging Big Data Ecosystem and new Analytics * Examples of Big Data Analytics * Data Analytics Life Cycle * Common Tools for Model building |
| Introduction to Machine Learning | * What is machine learning and it use cases * Brief overview of descriptive statistics   + types of variable   + 5 point summary of numeric variables   + probability distribution of variables * Supervised and Unsupervised Learning * Types of machine learning – choice of algorithm   + Regression, Classification, Recommender   + Parametric vs Non-parametric * Machine Learning using Python |
| Introduction to Numpy | * Fast element-wise array functions using numpy arrays * Data processing using numpy arrays * Linear Algebra – matrix multiplication, decompositions/factorization determinants |
| Lab Exercises | * Working with multi-dimensional array using numpy * Matrix operation with numpy * Random walk |
| Introduction to Pandas | * Load data into Pandas Dataframe * Explore data in Dataframe   + Indexing, selection, filtering, sorting, and grouping   + Arithmetic operations * Handle Missing Data * Descriptive statistics   + Correlation and covariance   + Unique values, value count, distribution |
| Lab Exercises | * Analyzing the grouplens.org data using Pandas |
| Data Visualization using Matplotlib and Seaborn | * Figure and subplots   + Line plots   + Bar plots   + Histograms and density plots   + Scatter plots   + Step   + heat maps * Colors, styles, labels, legends * Using Seaborn for color pallets |

# Day 2: Regression Techniques for Predicting Numeric Outcomes

| Topic | Description |
| --- | --- |
| Linear Regression | Forecasting numeric values with linear regression   * Understanding the cost function and minimize cost function using gradience descent algorithm * Finding best fit lines with linear regression * Model evaluation metrics – R2, RMSE, MAE, F-stat * Shrinking coefficients to understand the data – Ridge regression, Lasso regression, Forward stage-wise regression, L1/L2 Penalty for regularization * Bias / variance trade off * Tree based regression (CART regression) |
| Lab Exercise | * Forecasting home price based on data from Kaggle.com challenge * Implement gradient descent algorithm using numpy |
| Model Tuning | * Combining transformers and estimators in a pipeline * Using k-fold cross validation to assess model performance * Debugging algorithms with learning and validation curve * Diagnosis bias and variance problems with learning curves * Fine tuning hyper parameters using grid search |
| Lab Exercises | * Tune models built in the previous exercises |

# Day 3: Classification Techniques for Predicting Categorical Outcomes

| Topic | Description |
| --- | --- |
| Classification Algorithm | Forecasting categorical values with classification   * Overview of Perceptron, Analine, Stochastic Gradient Descent, MiniBatch Gradient Descent for classification problems * Logistic Regression for binary and multi class classification   + One-vs-all patterns * Classification using decision trees, random forest * Classifying with KNN * Model evaluation using accuracy score, precision, recall, F-score, ROC plot, AUC, confusion matrix * Tune the hyper parameters using cross validation and grid search |
| Lab Exercise | * Predicting customer credit status customer profile * Reveal local attitudes from personal ads from craigslist.org RSS feed |

# Appendix: Lab Setup

For lab exercises each student needs a laptop or a desktop with standard configuration 8 GB+ RAM and i5 CPU. Students will require administrative access to install software. For the lab exercise student will require to install the following software. Internet access is required for downloading test datasets, sample code, browse and documentations. Access to USB pen drive is required to copy materials from training instructor’s machine.

* Python 3.5
* NumPy
* SciPy
* Pandas
* Jupyter
* Requests
* feedparser
* Matplotlib
* Seaborn
* Mlxtend
* TensorFlow
* Anaconda for python 3.5 (recommended)