Practical ML (Machine Learning) – Five Day Course Outline

Overview

The practical machine learning course provides an introduction to the past present and future of machine learning.

Objective

The objective is to provide a level of comfort with the general concepts that are the foundation for machine learning and provide a hands-on experience that lays the foundation for further exploration. People who could not see practical applications in their work will now see how this could apply to their discipline.

Method

Build confidence by relating familiar skills, concepts and existing knowledge to new knowledge for machine learning. Let them echo back the meanings of the new terms often.

Review prerequisite knowledge of math and programing skills before diving into machine learning.

Do exercises that reinforce those skills and the new vocabulary.

Provide walk through of existing solutions to hard problems.

Pace

Pace is judged by the students with the least experience. Be sure to check in often with their work and progress. Offer advanced students extra credit features on each project.

If you get ahead of the schedule; then provide more time for Q and A and tease the next day with questions and examples that will make them eager for the next lesson when its scheduled.

If you fall behind schedule; then do walk throughs of the completed code examples and projects and then allow time for them to experiment and modify them rather than writing them from scratch. This will often be the case with inexperienced programmers in the group.

Day One

Python Review

Be sure to review terminology and concepts from variables to classes. Point out when there are many terms that mean the same thing. E.g.: identifier, parameter, argument, attribute, property; are all other names for a variable depending on context. Cover numeric data types. Differences between python 2 and 3 (We will be using 3) e,g Division, rounding and using *print* with parentheses.

Initial Pure Python Projects

Ask them not to use libraries in these review projects but instead use the simplest of python concepts. One of the points of the exercise is to learn that we can do things by hand, but we choose to use libraries for efficiency and reliability.

Statistics from scratch

Total

Write a function called *total* that takes a List/Array of numbers as an argument and returns the total sum. Call it and verify and print the results. Then do the same thing for *minimum*, *maximum*, *mean*, *mode*, and *median*.

Not everyone will complete all of these, but everyone should complete at least the first one. Do screen share help with anyone struggling and coach them through it publicly while everyone works as many more shy folks may be having the same struggles.

Random Numbers (Generate a data set with normal distribution) and Normal Distribution

Write a function named *roll* that takes a parameter called *sides* (defaults to 6) return generates a random number between 1 and sides (e.g. 6).

Remind them that this is open book and to use online resources to learn how to generate a random number.

Write a function named *roll_many* that take a parameter called *quantity* (defaults to 3) and calls roll that many times then returns the total. Return values with defaults will be in range 3-18 as if totaling thee dice.

Write a function named *generate* that takes a parameter called *epochs* (default to 100) and calls *roll_many* once for each epoch. Storing in a Dict/Hash the number of times each occurs. The keys will be 3-18 and resulting quantities should be higher in the middle near 10 and lower at the extremes of 3 and 18. Return the final populated Dict/Hash.

Write a function called *chart* that takes a parameter called data that is a Dict/Hash as returned from above that prints an ascii chart with hashes. E.g.:

```
 3|## (2)
 4|##### (4)
…
10|############ (13)
11|############# (14)
…
17|### (3)
18|# (1)
```

After later introducing PANDAs use this URL for extra reading.
https://jakevdp.github.io/PythonDataScienceHandbook/02.04-computation-on-arrays-aggregates.html

Linear Algebra

Describe how Vector is just another name for a List/Array,
Matrix is just a List of Lists (A two-dimensional Array),
And a Tensor is just an Array with three or more dimensions.

Review what a ***dot product*** is with short examples.

Write a function called dot that takes two lists called A and B as parameters and returns the dot product.

Extra Credit

Write a guessing game that guesses a number between 1 and 100 that a human has picked. The human will respond with one of three responses. H for higher, L for lower, or C for correct.

Ask everyone how they would approach the problem. We will call back to this when introducing Gradient Decent.

Outline

Day 1

Morning
- Introductions
- ML Overview, and What we and won't be covering
- Distinguish AI/ML and Data Science
- Brief History
- O.S.E.M.N. framework

Afternoon
- Hands On Intro to Kaggle
- Interactive Python and Advanced Features Review
- Your First Kaggle Project
- Math We Will Use, and Introduction of Linear Algebra

Day 2 – Process and Tools

Morning
- Data Science Process Review
- Data Manipulation Tools and Techniques
- NumPy
- PANDAS
- MatPlotLib

Afternoon
- Data Cleansing Exercise
- Data Exploration Exercise

- Individual Data Exploration Projects

Day 3 - Supervised Learning Algorithms

Morning - Supervised Learning Algorithms Pt 1
- Linear Regression
- Nearest Neighbor

Afternoon - Supervised Learning Algorithms Pt 2
- Decision Tree
- Random Forest
- Nearest Neighbor

Day 4 Unsupervised Learning Algorithms¡

Morning
- Dendrogram
- Dimensionality Reduction
- K-Means

Afternoon
- SVD
- PCA

Day 5 – Deep Learning

Morning
- Overview of Deep Learning
- Anatomy of simple Neural Network
- Hand on Neural Network Exercises - Perceptron

Afternoon
- Deep Learning with Advanced Neural Network Architectures
- Hand on project – Computer Vision
- GANs (Generative adversarial networks)
- Word2Vec
- Next Steps
- Final Q and A