



AI for Python

Course #: PY-560 **Duration:** 4 days

Prerequisites

Students should have solid experience in writing programs using Python.

Details

Artificial Intelligence (AI) is the creation and study of “intelligent agents” – software devices that perceive their environment and take actions that maximize their chance of successfully achieving their goals.

Python is a high-level, interpreted, highly extensible, object-oriented language that consistently ranks as one of the most popular programming languages for working with AI. With its comprehensive standard library and a large community of extensions, it can be used to create a diverse array of types of programs.

This course will assist students in learning about which algorithms should be used in a given context, as well as teaching them how to create AI building blocks using standard data mining techniques, using examples gathered from real-world applications.

Software Needed

Python 3 or higher and Anaconda.

Outline

- **Introduction**
 - What is Artificial Intelligence?
 - Applications of AI
 - Branches of AI
 - Building Agents
 - Development Environments
- **Classification and Regression**
 - Supervised vs. Unsupervised Learning
 - What is Classification?
 - Preprocessing and Encoding
 - Types of Classifiers
 - What is Regression?
 - Building Regressors
- **Predictive Analytics**
 - What is Ensemble Learning?
 - Using Decision Trees
 - Random Forests
 - Finding Optimal Training Parameters

- Computing Relative Feature Importance
- **Pattern Detection and Unsupervised Learning**
 - What is Unsupervised Learning?
 - Clustering Data With K-Means
 - Estimating Clusters With Mean Shift
 - Gaussian Mixture Models
 - Affinity Propagation Models
- **Recommender Systems**
 - Building Recommender Systems
 - Creating a Training Pipeline
 - Extracting Nearest Neighbors
 - Computing Similarity Scores
 - Collaborative Filtering
- **Logic Programming**
 - What is Logic Programming?
 - Solving Problems With Logic Programming
 - Matching Mathematical Expressions
 - Validating Primes
- **Heuristic Searches**
 - Heuristic Search Techniques
 - Constraint Satisfaction Problems
 - Local Search Techniques
 - Solving Problems With Constraints
- **Genetic Algorithms**
 - Evolutionary and Genetic Algorithms
 - Fundamental Concepts
 - Generating a Bit Pattern
 - Visualizing the Evolution
 - Solving the Symbol Regression Problem
- **Building Games**
 - Using Search Algorithms in Games
 - Combinatorial Search
 - Minimax Algorithm
 - Alpha-Beta Pruning
 - Negamax Algorithm
 - Building Game Bots
- **Natural Language Processing**
 - Tokenizing Text Data
 - Converting Words to Base Forms
 - Dividing Text Into Chunks
 - Extracting Word Frequencies
 - Topic Modeling Using Latent Dirichlet Allocation
- **Probabilistic Reasoning**
 - Understanding Sequential Data
 - Slicing Time-Series Data
 - Extracting Statistics from Time-Series Data
 - Generating Data Using Hidden Markov Models
 - Identifying Alphabet Sequences
- **Speech Recognizers**
 - Working With Speech Signals
 - Visualizing Audio Signals
 - Transforming Audio Signals to the Frequency Domain
 - Generating Audio Signals
 - Synthesizing Tones
 - Extracting Speech Features
 - Recognizing Spoken Words
- **Object Detection and Tracking**
 - Frame Differencing
 - Tracking Objects Using Colorspaces

- Tracking Objects Using Background Subtraction
- Optical Flow Based Tracking
- **Artificial Neural Networks**
 - Building a Perceptron Based Classifier
 - Single Layer Neural Networks
 - Multilayer Neural Networks
 - Vector Quantizers
- **Reinforcement Learning**
 - Understanding the Premise
 - Reinforcement Learning vs. Supervised Learning
 - Building Blocks of Reinforcement Learning
 - Creating an Environment
 - Building a Learning Agent
- **Deep Learning and Convolutional Neural Networks**
 - What are Convolutional Neural Networks?
 - Architecture
 - Types of Layers
 - Building a Perceptron Based Linear Regressor