Agoura Engineering Circle Introduction to Data Science Curriculum

Introduction to Data Science Using Python

Computers have been an integral part of our lives in the past few decades. Most of the applications, we use, have been "programmed" to do what they do. For example, take your iPhone or Android phone, the operating system behind your phone, is (for the most part) explicitly programmed to do what it does. However, in the last decade or two, we've been seeing a rise, in the kind of programs which don't have to be programmed to do their job. They instead "learn" or behave "intelligently". You may have heard of one or more of these: Chat bots, high frequency trading (buying and selling of stocks), crop weed detection, detecting patients with heart disease, Alpha Go (the program that defeated the best Go player!) or Alpha Zero (the chess program that learnt chess in one day) and self-driving cars. While these applications have been programmed, the programming is not the traditional kind where they program exactly what to do. They program a learning mechanism and it learns how to play or respond to the unique situations that arise during the course of running the program.

Imagine building a self-driving car and having to program exactly how to turn right at every right turn in the world. Sometimes, there may be people around the corner, it may be raining or another car might be close behind. Programming all these situations explicitly is almost impossible. Instead, these programs learn based on previously known "good" states and adapt to the new situations.

In this course, we'll learn to program an intelligent application, specifically, predicting the success of a movie. We'll do this as a four-part, 90-hour course consisting of 36 sessions:

1. **Python programming**
2. **Data Science using Numpy, Pandas and Machine Learning**

Project: Predicting success of a movie.

Course flow

We'll start out our journey by introducing basic programming concepts like branching, iteration, modular coding and data structures while solving mathematics problems that most students can understand but not necessarily solve by hand easily. This type of interdisciplinary learning helps in learning two things at the same time along with reinforcing any prior knowledge.

Once everyone has achieved some proficiency in programming,

Class Schedule

Alternate Saturdays 9:30am – 12:00pm (online).

Class	Homework Review	Quiz	Lecture & Class Work
Intro to Data Science	9: 30 – 10 AM	10- 10: 15 AM	10: 15 AM -12: 00 PM

Please note that if you miss two classes in a row, you'll not be able to follow the topics anymore. Students must login 10 min before start of the class.

Awards

At the end of each Semester (Fall, Spring), 3 students will receive the Star Award which is based on the cumulative score from quiz, home work and Final Exam.

Course Details

Pre-requisites: Good understanding of basic mathematical concepts (no higher than 8th grade level or taught in Senior Intermediate level at Agoura Math Circle). Also, students have to get their own desktops or laptops to program.

Course Registration & Website: Each student should register for the class using their own email ID (Not parent's email ID). All course communication, homework submission will be through the course website. Each student has to register for each one of the four sessions irrespective of whether they have previously registered or not. Preference will be given to students who have attended previous sessions.

Class workload: Apart from the 2.5 hours of class once every 2 weeks, students are expected to spend at least 1 hour every day of the week for a total work load of between 15 – 20 hours. If you can't make this commitment, please do not register. The course material to be covered is pretty heavy and if you fall behind, catching up is difficult.

Class style: Classes will be Skype based in an interactive manner involving discussions and coding either individually or as a team. Instead of striving towards finishing certain amount of material in each class, we'll work towards certain milestones which involve writing a few programs individually or as a team. During the courses, a textbook will be recommended for each part.

Homework Policy: After each class, homework assignments will be mailed out. They are due 11 days from that day i.e., Wednesday of the next week. Most of the homework problems are fairly challenging, especially to those without any previous experience in programming. Please feel free to discuss homework problems with friends, parents or anyone else. But the final submission should be yours. Any sort of plagiarism will not be tolerated.

Final Exam Policy: Each semester has a final exam worth 100 points. The test will be online for 3 hours.

What will you learn from this course?

1. Achieve decent proficiency in programming with Python
2. Setup a GitHub portfolio to show case your work
3. Improvement in report writing and presentation skills.
4. Team work and collaboration towards finishing a project in artificial intelligence using open-source libraries.
5. Increase in confidence to tackle problems in a logical and algorithmic fashion.

In order to come up with the list of topics in a manner that is comprehensive and meets our objectives, a few online resources including but not limited to Coursera, pandas have been used. While they were used to identify some topics, the material will be created by the developers of the course.

Syllabus

Fall Semester

Python programming

Module 1: Basic Programming: Variables and Control Flow

Session 1: Introduction to Python Programming

1. What is Python
2. Introduction to Variable
3. Operators
4. Conditional statements
    5. While Loop
    6. For Loop Introduction
7. Introduction to nested loops

Module 2: Data Types and Collection

Session 2: Data Types

1. What are the different types of data?
2. Scalar data types: int, str, float, bool, datetime

Collections in Python:

3. What are collections
4. Introduction to Lists
5. Accessing, adding and removing items.
6. List operations

Session 3: Multi-dimensional Collections:

1. Introduction to 2D lists
2. Problem solving using 2D lists
3. Nested lists
4. Filtering of lists
5. Sorting of lists

Session 4: Other types of collections

1. Immutable types: Tuples
2. Introduction to Sets

Wrapping Up

3. Key-Value Pairs or Dictionaries
4. Comparison of various collections
5. Introduction to file I/O

Module 3: Functions

Session 5: Functions

1. Introduction to functions
2. Calling functions and returning values
3. Introduction to calling functions with multiple values and getting multiple values back

Session 6: Function's continuation

1. Error handling
2. Understanding call by value vs. call by reference

Applications of functions and nuances

1. Map function
2. Introduction to shallow and deep copy

Session 7: Recursion

1. Introduction to recursion
2. Recursive function calls using global variables
3. Stack overflow and other issues with recursive calls

Module 4: Advanced Topics

Session 8: Classes

1. Introduction to classes
2. Object oriented programming principles overview
3. Inheritance explained
4. Working with classes

Session 9: Internet APIs

1. Introduction to JSON
2. Introduction to accessing data from other sources
3. Introduction to REST API calls
4. Integration and accessing data from either IMDB or Facebook or other popular sites

10. Final Exam/ Projects

Spring Semester

Data Science

Module 1: Overview Data Science and iPython

Session 1: Overview

1. Overview Data Science
2. Overview iPython
3. iPython Commands and Shortcut Keys
4. IPython and Shell Commands
5. Errors and Debugging
6. Profiling and Timing Code
7. Resource

Module 2: Introduction to Numpy

Session 2: Data types and Functions

1. Numpy Standards and data types
2. Numpy Arrays
3. Universal Functions
4. Aggregations

Session 3: Indexing and Arrays

1. Fancy Indexing
2. Sorting Arrays
3. Structured Data: NumPy's Structured Arrays

Module 3: Data Manipulation with Pandas.

Session 4: Overview Pandas/ Objects

1. Pandas Objects
2. Data Indexing and Selection
3. Operating on Data in Pandas
4. Handling Missing Data

Session 5: Datasets and Manipulation

1. Combining Datasets
2. Aggregation and Grouping
3. Pivot Tables
4. Vectorized String Operations
5. High-Performance Pandas

Module 4: Visualization with Matplotlib

Session 6: Overview Matplotlib

1. General Matplotlib Tips
2. Simple Line Plots
3. Simple Scatter Plots
4. Visualizing Errors
5. Visualizing a Three-Dimensional Function
6. Customizing Color bars

Session 7: Annotation and Visualization

1. Multiple Subplots
2. Text and Annotation
3. Customizing Ticks
4. Summary of Formatters and Locators
5. Three-Dimensional Contour Plots
6. Visualization with Seaborn

Module 5. Machine Learning

Session 8: Overview

1. Introducing Scikit-Learn
2. Application: Exploring Handwritten Digits
3. Hyperparameters and Model Validation
4. Feature Engineering
5. Bayesian Classification
6. Basis Function Regression
7. Regularization

Session 9: In-Depth Learning

1. In-Depth: Support Vector Machines
2. In-Depth: Decision Trees and Random Forests
3. In-Depth: Manifold Learning
4. In Depth: k-Means Clustering
5. In-Depth: Kernel Density Estimation
6. Application: A Face Detection Pipeline

Session 10: Final Exam