

Office Telephone: +91 8660382715

E-mail: info@aas-ai.tech



Anagha Agile Systems

Data Science



Virtual Classroom Training



Audience (Who This Course is for)



★ This course is for IT PROFESSIONALS who want to broaden their practical knowledge of

- Data Science,
- Scientific Computing(Image Processing, Numerical Analysis),
- Data Analytics.
- Data Visualisation.
- Matplotlib

- ★ Classroom & Hands on Labs
- ★ SciKit learn based Training

Objectives



PARTICIPANTS

- Acquire the competencies and skills necessary for working in the field of Cloud Architecture
- Can develop in their current role, or qualify for a new role
- Benefit from a selected, well-structured and quality-checked digital learning curriculum
- Learn from experts with a high practical relevance
- Exchange information on challenges and best practices in communities
- Can apply the learning content directly in their everyday professional life
- Have the opportunity to develop further by completing a certificate or university degree

Prerequisites



This is a Advanced-level learning path and so no prior knowledge of scientific computing is required, however, general knowledge of computing would be highly beneficial. To get the most of out of this course, participants should have basic proficiency with networking technologies like programming. This course is ideal for solution developers as well as experienced senior professionals interested in applying scientific computing to data science, machine learning and data engineering. Students should have beginner level Linux and intermediate level Python skills. For your project in this course, you will build a big data engineering pipeline in a SciKit platform:Jupyter Notebooks.

Class Info:

Instructor: Jayaprakash

Office: Bangalore

Office Hours: Times & Days : 4pm to 8pm on meet.google.com

Office Telephone: +91 8660382715

E-mail: info@aas-ai.tech

Prerequisite for Windows IT Professionals are:

1. Intermediate skills of Python Programming and Understanding Engineering Maths
2. Good Familiarity with Windows 10.
3. Good Familiarity with Excel
5. Good Familiarity with Github & Visual Studio.



Class Meetings



This is a fully-online class. Weekly modules will be released each Tuesday during the course. These modules will present the topics for that week in a series of live videos, screencasts, and slides. There will be short exercises within the modules to reinforce learning and give you a chance to practice what you've just learned.

Optional sections will be held during the week that are designed to help you with the implementing the work in the course as well as address more doubts on concepts taught. Sections will be held "live" via a conferencing platform, and they will also be recorded for later viewing. Training Coordinators will notify students when they will be holding sections, and will let students know the format of sections.

Class Schedule:

Section Schedule:

Monday, 10:30am - 4:30pm, hosted by Jayaprakash S T
Wednesday, 10:30am - 4:30pm, hosted by Jayaprakash S T
Friday, 10:30am - 4:30pm, hosted by Jayaprakash S T

Schedule
Monday - 8:30 - 18:00
Saturday 8:30 - 13:30

Lab Schedule:

Tuesday, 2:30pm - 5:30pm, hosted by AAS-AI Technical team
Thursday, 2:30pm - 5:30pm, hosted by AAS-AI Technical team
Saturday, 2:30pm - 5:30pm, hosted by AAS-AI Technical team

Additional Expenses:

Texts:



Recordings, Relevant reading materials
Book on Linux Administration,
Dockers, Cloud Computing and
Python Programming

Technology:

Web server Hosting charges
Cloud Subscription and
Project Expenses





Activities, Assignments, Final Project

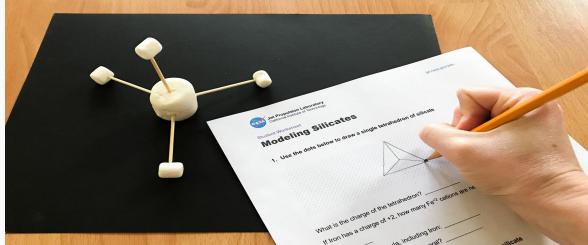


There are 3 types of exercises you will encounter in this course:

Practice Activities. These are smaller in-module exercises designed to give you a chance to work on what you've just learned or to answer some basic questions on what you've just learned. They are designed to be "low-stakes", but are important for the learning process. These in-module exercises will be 15% of your final grade. The completion of these activities are due before the end of the week. The activities must be completed by 11:59pm the following Wednesday.

Assignments. These are more significant exercises given each week that will be submitted and graded. The due date will be listed on each assignment and on the syllabus page. Generally, they will be due by 11:59pm the following Wednesday. These weekly exercises will be 50% of your final grade.

Final Project. The final project is to plan, design, build, and publish a small-scale web site. We will work on this throughout the term in stages. You will have 2 smaller assignments where you ideate and plan your site, and a final assignment where you submit your final project. As part of the work towards the final project, there will be opportunities to get feedback from your teaching assistants. The project will be 35% of your grade.

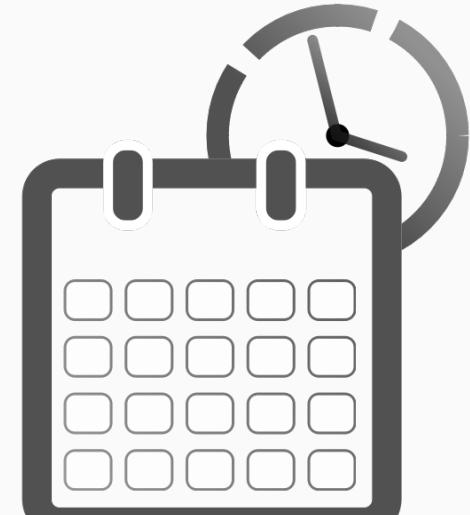


Course Duration

The Duration of the Course is 4 weeks or 240 hrs with 60% of the course is hands-on experience.

The course materials are provided are soft copies of slides and notes included.

All the queries, lab assignments will be hosted on Anagha Agile Systems website & can be accessed after authentication.



Student Duties

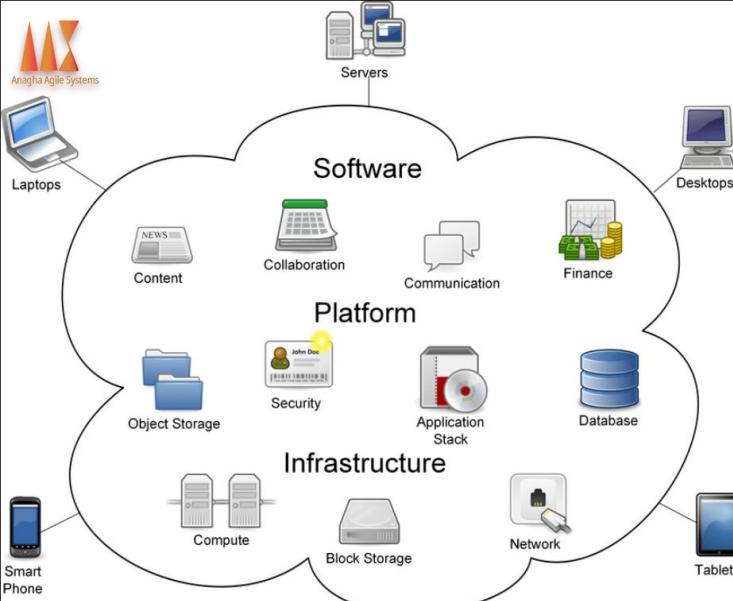
The AAS-AI Faculty Staff recommends that you include the following in your syllabus:

- Contact information for the instructor(s) and TA(s), including email, office hours/location, etc.
- Course description that explains the course's scope and purpose, format (e.g., lecture, recitation, lab, studio), relevance to students' academic/professional goals, and major deliverables (e.g., project, presentation, paper, exhibit)
- Statement of assumed/required prior knowledge
- List of learning resources and whether they are required or supplementary (e.g., textbooks, software, etc.)
- Course-level learning objectives that articulate what students should be able to do by the end of the course
- Description of major assessments and how they contribute to the final grade
- Statements and policies that clearly communicate your expectations regarding
 - Attendance
 - Academic integrity, including collaboration and plagiarism
 - Late/make-up work
 - Accommodations for students with disabilities
- Statement encouraging student wellness
- Expected semester schedule
- Diversity statement



**A CULTURE
OF SUPPORT**





Scientific Computing Overview

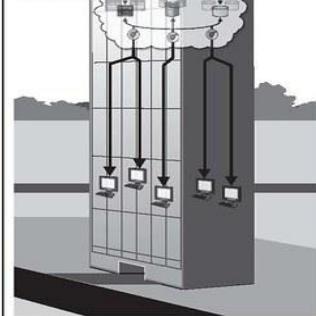
Module 01:

Introduction to Scientific Computing Virtual Classroom

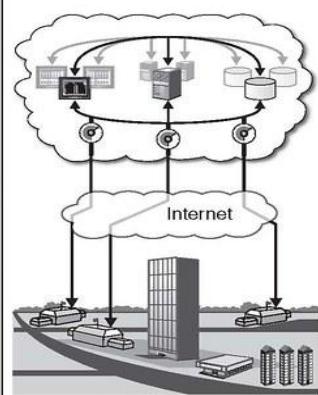
01: Introduction to Python and Scientific Computing.

1.1 Overview of Python features.

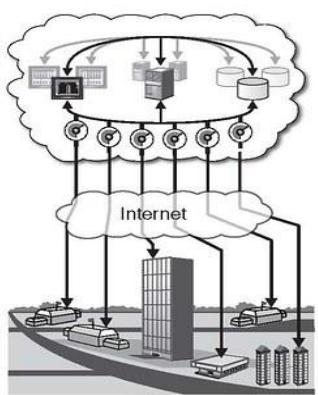
1.2 Overview of SciKit-learn



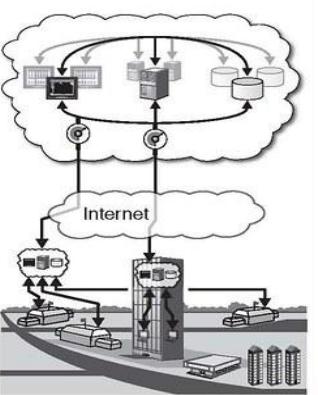
Private cloud
is operated solely for an organization and the cloud may be on or off the premises.



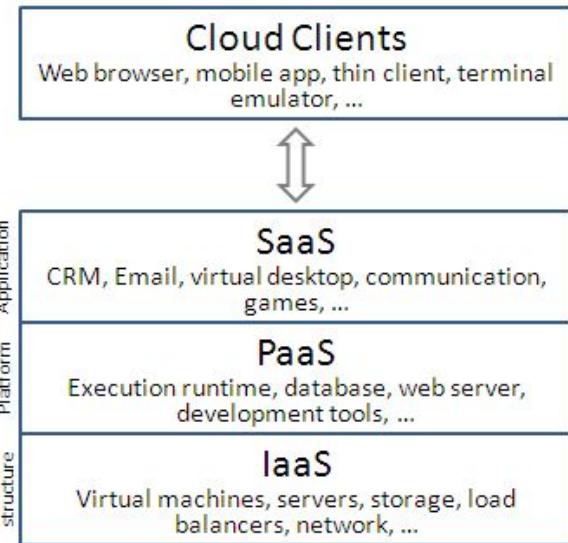
Community cloud
is shared by several organizations and supports a specific community of customers that have similar information technology requirements.



Public cloud
has an infrastructure that is made available to the general public or large industry group.



Hybrid cloud
has an infrastructure that is composed of two or more clouds that remain unique entities but are bound together by standardized or proprietary technology.



Numpy

Module 02:

2 NumPy arrays: tips and tricks:

2.1: NumPy Basics

2.1.1: NumPy – Working with Numerical Arrays



Introduction to Numpy

Module 02:

- 2.1: Numpy Basics
 - 2.1.2: Introduction to NumPy
 - 2.1.3: Motivation:NumPy is fast!
 - 2.1.4: N-dimensional Arrays
- 2.2: NumPy Array Construction and Indexing
 - 2.2.1: Array Construction Routines
 - 2.2.2: Array Indexing
- 2.3: NumPy Array Math and Universal Functions
 - 2.3.1: Array Math and Universal Functions
- 2.4: NumPy Broadcasting
 - 2.4.1: Broadcasting
- 2.5: NumPy Advanced Indexing – Memory Views and Copies
 - 2.5.1: Advanced Indexing – Memory Views and Copies
 - 2.5.2: Fancy Indexing
 - 2.5.3: Boolean Masks for Indexing
- 2.6: Random Number Generators
 - 2.6.1: Random Number Generators
- 2.7: Reshaping NumPy Arrays
 - 2.7.1: Reshaping Arrays
- 2.8: NumPy Comparison Operators and Masks
 - 2.8.1: Comparison Operators and Masks
- 2.9: Linear Algebra with NumPy
 - 2.9.1: Linear Algebra with NumPy Arrays



Linear Algebra

Module 03:

3.1 Linear Algebra with Numpy.

3.1.1 NumPy Arrays

3.1.2 Array Attributes

3.1.3 Matrix Operations and Functions

3.1.4 Arithmetic Operations

3.1.5 Matrix Multiplication

3.1.6 Matrix Powers

3.1.7 Transpose

3.1.8 Inverse

3.1.9 Trace

3.1.10 Norm

3.1.11 Determinant

3.1.12 Dot Product

3.2 Solving Linear Systems

3.2.1 Linear Systems

3.2.2 Gaussian elimination

3.2.3 Elementary Row Operations

3.2.4 Implementation

3.2.5 Examples

3.2.6 Find the Inverse

3.2.7 Solve a System

3.2.8 `scipy.linalg.solve`

3.3 Eigenvalues and Eigenvectors

3.3.1 Definition

3.3.2 `scipy.linalg.eig`

3.3.3 Examples

3.3.3.1 Symmetric Matrices

3.3.3.2 Diagonalization

3.3.3.3 Matrix Powers



Interpolation

Module 04:

- 4.1 Interpolation (`scipy.interpolate`)
 - 4.1.1 Univariate interpolation
 - 4.1.2 Multivariate interpolation
 - 4.1.3 1-D Splines
 - 4.1.4 2-D Splines
 - 4.1.5 Additional tools



Optimization and fitting techniques

Module 05:

- 5.1 Fitting data
- 5.2 Kwarg optimization wrapper
- 5.3 Large-scale bundle adjustment in scipy
- 5.4 Least squares circle
- 5.5 Linear regression
- 5.6 OLS
- 5.7 Optimization and fit demo
- 5.8 Optimization demo
- 5.9 RANSAC
- 5.10 Robust nonlinear regression in scipy



Ordinary differential equations

Module 06:

- 6.1 Coupled spring-mass system
- 6.2 Korteweg de Vries equation
- 6.3 Matplotlib: lotka volterra tutorial
- 6.4 Modeling a Zombie Apocalypse
- 6.5 Solving a discrete boundary-value problem in scipy
- 6.6 Theoretical ecology: Hastings and Powell

Root finding

Module 07:

- 7.1 Function intersections
- 7.2 Spherical Bessel Zeros



Data Rebinning

Module 08:

8.1 Data rebinning :

Examples of rebinning data to produce smaller arrays with and without interpolation.

8.2 Rebinning FUNCTION



Simple Plotting

Module 09:

- 9.1 Histograms
- 9.2 Matplotlib: animations
- 9.3 Matplotlib: arrows
- 9.4 Matplotlib: bar charts
- 9.5 Matplotlib: custom log labels
- 9.6 Matplotlib: hint on diagrams
- 9.7 Matplotlib: legend
- 9.8 Matplotlib: maps
- 9.9 Matplotlib: multicolored line
- 9.10 Matplotlib: multiline plots
- 9.11 Matplotlib: plotting values with masked arrays
- 9.12 Matplotlib: shaded regions
- 9.13 Matplotlib: sigmoidal functions
- 9.14 Matplotlib: thick axes
- 9.15 Matplotlib: transformations
- 9.16 Matplotlib: treemap
- 9.17 Matplotlib: unfilled histograms



Convex-Hull

Module 10:

- 10.1 Finding the Convex Hull of a 2-D Dataset
- 10.2 Finding the minimum point in the convex hull of a finite set of points



Kalman filtering

Module 11:

11.1 Kalman Filtering Example from the Welch & Bishop Introduction to the Kalman Filter.



Communication Theory Example of BPSK simulation.

Module 12:

12.1 Communication theory examples

12.1 These two examples illustrate simple simulation of a digital BPSK modulated communication system

[<https://scipy.github.io/old-wiki/pages/Cookbook/CommTheory.html>]



Signal Smoothing

Module 13:

13.0 Smoothing a signal Performing smoothing of 1D and 2D signals by convolving them with a window.

- 13.1 Signal Processing (scipy.signal)
<https://scipy-cookbook.readthedocs.io/items/SignalSmooth.html>
<https://docs.scipy.org/doc/scipy/reference/tutorial/signal.html>
- 13.2 bspline(x, n)
B-spline basis function of order n.
- 13.4 cubic(x)
A cubic B-spline.
- 13.6 quadratic(x)
A quadratic B-spline.
- 13.8 gauss_spline(x, n)
Gaussian approximation to B-spline basis function of order n.
- 13.11 cspline1d(signal[, lamb])
Compute cubic spline coefficients for rank-1 array.
- 13.12 qspline1d(signal[, lamb])
Compute quadratic spline coefficients for rank-1 array.
- 13.14 cspline2d(input[, lambda, precision])
Coefficients for 2-D cubic (3rd order) B-spline.
- 13.16 qspline2d(input[, lambda, precision])
Coefficients for 2-D quadratic (2nd order) B-spline:
- 13.18 cspline1d_eval(cj, newx[, dx, xo])
Evaluate a cubic spline at the new set of points.
- 13.20 qspline1d_eval(cj, newx[, dx, xo])
Evaluate a quadratic spline at the new set of points.
- 13.22 spline_filter(lin[, lmbda])
Smoothing spline (cubic) filtering of a rank-2 array.



Nearest-neighbor queries/ KD Tree Searching

Module 14:

- 14.0 https://scipy-cookbook.readthedocs.io/items/KDTree_example.html
 - 14.1 <https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.KDTree.html>
 - 14.2 Spatial algorithms and data structures (scipy.spatial)
 - 14.3 `KDTree(data[, leafsize, compact_nodes, ...])`
 - 14.4 kd-tree for quick nearest-neighbor lookup.
 - 14.5 `cKDTree(data[, leafsize, compact_nodes, ...])`
 - 14.6 kd-tree for quick nearest-neighbor lookup
 - 14.7 `Rectangle(maxes, mins)`
 - 14.8 Hyperrectangle class.
- <https://docs.scipy.org/doc/scipy/reference/spatial.html>



Butterworth Bandpass Filter

Module 15:

15.1 Create and apply a Butterworth bandpass filter.

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.buttap.html#scipy.signal.buttap>
<https://scipy-cookbook.readthedocs.io/items/ButterworthBandpass.html>

15.2 Butterworth Bandpass: This cookbook recipe demonstrates the use of `scipy.signal.butter` to create a bandpass Butterworth filter.



FIR Filter Design,

Module 16:

16.1 FIR Filter Design, Design a lowpass FIR filter using the window method.

<https://scipy-cookbook.readthedocs.io/items/FIRFilter.html>

<https://scipy-cookbook.readthedocs.io/items/ApplyFIRFilter.html>

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.firwin.html>

https://mpastell.com/pweave/_downloads/FIR_design_rst.html OR

<https://mpastell.com/2010/01/18/fir-with-scipy/>



Multithreading

Module 17:

17.1 Easy multithreading for embarrassingly parallel multidimensional space using kd trees.
<https://scipy-cookbook.readthedocs.io/items/Multithreading.html>



KD Tree

Module 18:

18.1 KDTree Searching multidimensional space using kd-trees.
https://scipy-cookbook.readthedocs.io/items/KDTree_example.html



Particle Filter

Module 19:

19.1 Particle Filter A simple particle filter algorithm for tracking objects in a video sequence.
<https://scipy-cookbook.readthedocs.io/items/ParticleFilter.html>



Brownian Motion

Module 20:

20.1 Brownian Motion Compute Brownian motion (i.e. the Wiener process).
<https://scipy-cookbook.readthedocs.io/items/BrownianMotion.html>



Correlated Random Samples

Module 21:

21.1 Correlated Random Samples Generate correlated normally distributed random samples.
<https://scipy-cookbook.readthedocs.io/items/CorrelatedRandomSamples.html>



Large Markov Chains

Module 22:

22.1 Large Markov Chains Find the stationary distribution of a large Markov chain; the M/M/1 tandem queue
https://scipy-cookbook.readthedocs.io/items/Solving_Large_Markov_Chains.html



Watershed algorithm

Module 23:

23.1 Watershed algorithm Apply the watershed algorithm in order to split an array into distinct components (e.g. for the segmentation of an image into objects).
<https://scipy-cookbook.readthedocs.io/items/Watershed.html>



Linear Classification

Module 24:

24.1 Linear Classification Fisher's discriminant function and Probabilistic Generative model
<https://scipy-cookbook.readthedocs.io/items/LinearClassification.html>



Eye Diagram

Module 25:

25.1 Plot an eye diagram using numpy and matplotlib.

<https://scipy-cookbook.readthedocs.io/items/EyeDiagram.html>

<https://dspillustrations.com/pages/posts/misc/eye-diagram-examples.html>

<https://github.com/WarrenWeckesser/eyediagram>

https://en.wikipedia.org/wiki/Eye_pattern



Scipy

Module 26:

- 26.1 What is SciPy?
- 26.2 Why Use SciPy?
- 26.3 Which Language is SciPy Written in?
- 26.4 Where is the SciPy Codebase?



Top level SciPy

Module 27:

	Name	Description
27.1	cluster	Clustering algorithms
27.2	constants	Physical and mathematical constants
27.3	fftpack	Fast Fourier Transform routines
27.4	integrate	Integration and ordinary differential equation solvers
27.5	interpolate	Interpolation and smoothing splines
27.6	io	Input and Output
27.7	linalg	Linear algebra
27.8	ndimage	N-dimensional image processing
27.9	odr	Orthogonal distance regression
27.10	optimize	Optimization and root-finding routines
27.11	signal	Signal processing
27.12	sparse	Sparse matrices and associated routines
27.13	spatial	Spatial data structures and algorithms
27.14	special	Special functions
27.15	stats	Statistical distributions and functions



SciPy for Linear Algebra

Module 28:

- 28.1 `scipy.linalg` vs `numpy.linalg`
- 28.2 `numpy.matrix` vs 2-D `numpy.ndarray`
- 28.3 Basic routines
- 28.4 Decompositions
- 28.5 Matrix functions
- 28.6 Special matrices



SciPy for Numerical Analysis

Module 29:

- 29.1 Numerical Integration Problem Statement
- 29.2 Riemann's Integral
- 29.3 Trapezoid Rule
- 29.4 Simpson's Rule
- 29.5 Computing Integrals in Python



SciPy for Signal Processing

Module 30:

- 30.1 The Basics of Waves
- 30.2 Discrete Fourier Transform (DFT)
- 30.3 Fast Fourier Transform (FFT)
- 30.4 FFT in Python
- 30.5 Summary and Problems



SciPy for Data Mining

Module 31:

- 31.1 Overview of clustering methods
- 31.2 K-means
- 31.3 Affinity Propagation
- 31.4 Mean Shift
- 31.5 Spectral clustering
- 31.6 Hierarchical clustering
- 31.7 DBSCAN
- 31.8 OPTICS
- 31.9 BIRCH
- 31.1 Clustering performance evaluation



SciPy for Computational Geometry

Module 32:

- 32.1 Jeff Erickson's notes on convex hulls in the plane.
- 32.2 The Kirkpatrick-Seidel convex hull algorithm: the article.
- 32.3 For arrangements of lines, read this and David Mount's notes (pages 82 on).
- 32.4 David Kirkpatrick's point location algorithm.
- 32.5 Jeff Erickson's notes on Tamal Dey's proof.
- 32.6 Welcome to scikit-geometry docs's documentation!
- 32.7 Contents:
 - 32.7.1 Introduction
 - 32.7.2 Basic Geometric Types
 - 32.7.3 Finding intersections
 - 32.7.4 Drawing functions
 - 32.7.5 Further functionality
 - 32.7.6 Computing a Convex Hull
 - 32.7.7 Arrangements and Visibility
 - 32.7.8 Finding things in the arrangement
 - 32.7.9 Voronoi Diagrams
- 32.8 Polygons:
 - 32.8.1 Polygons
 - 32.8.2 Polygon with Holes
 - 32.8.3 Minkowski Sum of 2 Polygons
 - 32.8.4 Polygon Set
 - 32.8.5 Polygon Simplification
 - 32.8.6 Simplification Modes
 - 32.8.7 Simplification and Topology
 - 32.8.8 Polygonal skeletons (straight skeletons)
- 32.9 Reference:
 - Geometric Primitives
 - Two-dimensional primitives
 - Three-dimensional primitives
- 32.1 Indices and tables
- Index
- Module Index
- Search Page

Part II: DATA ANALYTICS



Bayesian Data Science by Simulation

Module 33:



Bayesian Data Science by Probabilistic Programming

Module 34:

- 34.1 Two Group Comparisons: Drug effect on IQ
- 34.2 Multi-Group Comparisons: Multiple ways of sterilizing phones
- 34.3 Two Group Comparisons: Darwin's Finches
- 34.4 Hierarchical Modelling: Baseball
- 34.5 Hierarchical Modelling: Darwin's Finches
- 34.6 Bayesian Curve Regression: Identifying Radioactive Element

Part III: Advanced Data Analytics



Spatial Data Structures and Algorithms

Module 35:



Multidimensional Image Processing Functions

Module 36:



Multidimensional Image Processing Functions

Module 37:



File I/O

Module 38:

- 38.1 MATLAB files
- 38.2 IDL files
- 38.3 Matrix Market files
- 38.4 Wav sound files (`scipy.io.wavfile`)
- 38.5 Arff files (`scipy.io.arff`)
- 38.6 Netcdf



pandas

Module 39:

- 39.1 How do I use the pandas library to read data into Python?
- 39.2 How do I use the seaborn library to visualize data?
- 39.3 What is linear regression, and how does it work?
- 39.4 How do I train and interpret a linear regression model in scikit-learn?
- 39.5 What are some evaluation metrics for regression problems?
- 39.6 How do I choose which features to include in my model?



Scikit-Learn

Module 40:

40.1 What is Scikit-Learn?

Scikit-learn provides tools for:

- Regression, including Linear and Logistic Regression
- Classification, including K-Nearest Neighbors
- Model selection
- Clustering, including K-Means and K-Means++
- Preprocessing, including Min-Max Normalization

40.2 Getting started with Scikit-learn

40.3 Datasets and import sklearn

40.4 Generate synthetic regression data

40.5 Data Preprocessing

40.6 Scikit-learn Linear Regression: implement an algorithm

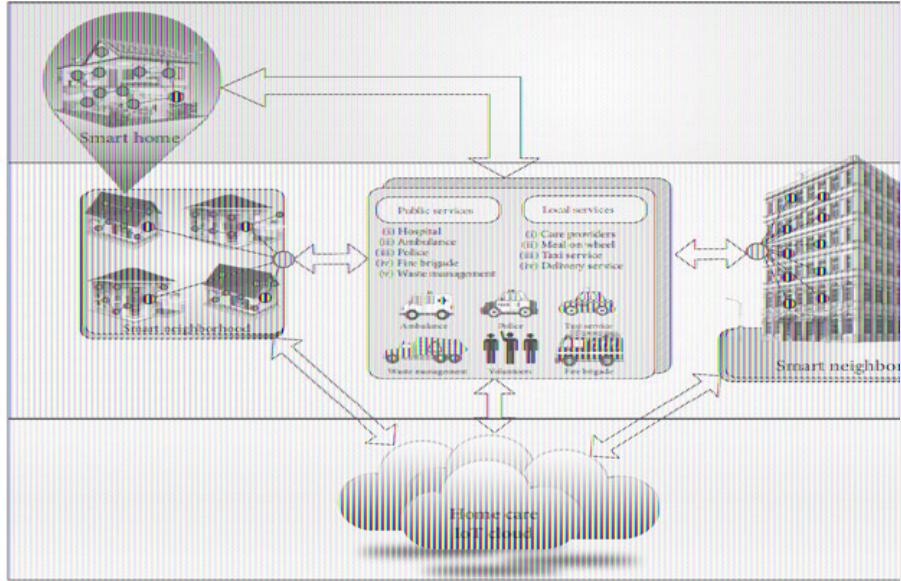
You've just taken your first steps to master Scikit-Learn. Today, we covered the purpose of Sklearn, how to import or generate sample data, how to scale our data, and how to implement the popular linear regression algorithm.

As you continue your Scikit-learn journey, here are some next algorithms and topics to learn:

- Support Vector machine
- Random Forest
- Naive Bayes model
- Unsupervised learning
- Deep learning
- Logistic regression

To help you get started in Machine Learning, AAS-AI has created the course Hands-on Machine Learning & Deep Learning with Pytorch Framework. With in-depth explanations of all the Pytorch basics and popular Deep learning algorithms, this course will give you, everything you need to learn about Machine Learning & Deep Learning in one place.

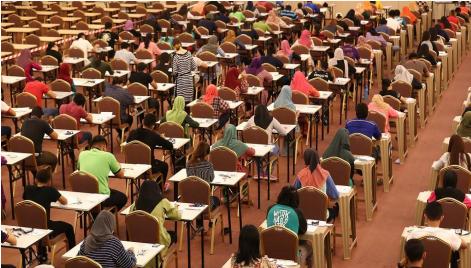
What Next?.....



Data Science Trends



1. Multi Choice
Questions



3. Oral Interview



Q & A on Project Presentation

20 mins

180 mins

30 mins

30 mins

20 mins

2. Project hands on Lab



4. Project Presentation



Conclusion

Let's wind up with different IT positions you can take up after this course.

1. **Data Analyst** With so much data in the world, and with data as such a constant in all of our lives, there is a need for professionals who can analyze that data and suss out the human element. The job of Data Analysts is to analyze this data, and determine what it tells us about the sample/population using that data.
2. **Data Engineering** In the world of big data, every job works together in order to understand the data collected and make the best use of it for the business. The role of a data engineer is to prepare the infrastructure like algorithms that data scientists and data analysts to analyze in order to enable the business to make the best decisions based on their findings.
3. **Data Product Owner** The role of Data Product Owners is to ensure value comes out of all the data projects of their companies. They need to develop a vision of what the data science team wants to achieve, together with a strategy to get there. He is someone who develops and maintains a vision and plan for that data and oversees the execution of that plan.
4. **Data Scientist** Those who practice data science are called data scientists, and they combine a range of skills to analyze data collected from the web, smartphones, customers, sensors, and other sources to derive actionable insights. "The ability to take data — to be able to understand it, to process it, to extract value from it, to visualise it, to communicate it — that's going to be a hugely important skill in the next decades." si done by DATA SCIENTIST

Feedback

Please contact us at info@aas-ai.tech if you have any questions, or want to share your thoughts about the exam.



This course serves as a requirement to pursue other higher data science certifications.

- Data Science Introduction
- Evolution of Data Science
- Organization Data Science Stack
- Roles in Data Science
- Business Analytics vs Data Science
- Machine Learning in Data Science
- Business Analytics Classification
- Statistics – Exploratory Analysis
- Statistics – Hypothesis Testing
- Data Science Industry Use cases

Course Prerequisite.....