Data Science Training

Course Duration: 6.5 months, Weekly 6 days, Each Class: 1 hr.
Trainer: D. Chaitanya (IIT Roorkee Alumni), 20+ yrs Exp.

a) **Python:** ~ 2 month

I) Complete Python

- Demo:
  i. Why Python?
  ii. Who Uses Python today
  iii. What can we do with Python
  iv. How Python Developed and Supported
  v. Python – Technical Strengths.
  vi. What next?

- Python Interpreter
- Program Execution – programmer's view, Python's view
- Installation
  i. Python
  ii. All Related Software: PyCharm, Anaconda
  iii. Setup, configure Python in Laptop
  iv. IDLE – UI, usage, features
- All Numeric types in Python, coding/Hands-on
- Python Variables, objects, References, Shared References, coding/Hands-on
- Garbage Collection of objects
- All built in types in python: Strings, Lists, Dictionaries, Tuples, sets, Files
- Python Statements - coding/Hands-on
- Assignments, Expressions and Prints
  i. if-else, if-elif-else, if-else ternary expression
  ii. while and for loops
  iii. Comprehensions vs regular
  iv. Parallel Traversals: map and zip functions
  v. Other important functions: range, len, enumerate

- Iterations and Comprehensions - coding/Hands-on
- Python online Documentation
- Python Functions – def, nested functions
- Variable Scopes – basics, LEGB rules, global, nonlocal

- Function Arguments - coding/Hands-on
  i. Arguments and shared references
  ii. Arguments passing basics
  iii. Arguments matching basics
  iv. Arguments matching syntax
  v. Multiple Results

- Advanced Function Concepts – Recursive functions, Attributes, Annotations, lambda
- Generators and comprehensions – Generator functions, yield, Generator expression
- Python Modules
  i. Definition, why modules?
  ii. Typical Python program architecture
  iii. Import statement - coding/Hands-on
  iv. How Import works: Find it, Compile it, Run it
  v. Standard library modules
  vi. __pycache__ folder for byte code files
  vii. Module search path

- Module coding Basics
  i. Module creation
  ii. import statement, from statement, from * statement
  iii. Module Namespaces, Namespace dictionaries: __dict__
  iv. Reloading modules

- Module Packages
  i. Package import basics
  ii. Why Package imports?
  iii. Relative import basics
  iv. Why relative imports?
  v. Package Namespaces

- Advanced Module Topics
  i. Data hiding in modules - coding/Hands-on
  ii. Mixed usage modes: __name__ and __main__ - coding/Hands-on
  iii. The as extension for import and from - coding/Hands-on

- Introduction to Python Classes - coding/Hands-on

i. Why Classes?
ii. Classes, constructors and Instances
iii. Method calls
iv. Attribute inheritance search
v. OOP is about code reuse
vi. Subclassing by Inheritance
vii. Polymorphism in Action
viii. Class vs instance attributes
ix. Storing objects in DB – Pickles & Shelves

- Coding with Classes - coding/Hands-on
  i. Abstract super classes
  ii. Nested classes
  iii. Classes vs Modules
  iv. Namespace dictionaries
  v. LEGB scopes rule revisited

- Operator Overloading - coding/Hands-on
  i. Constructors: __init__
  ii. Indexing, Slicing: __getitem__ and __setitem__
  iii. Attribute Access: __getattr__ and __setattr__
  iv. String Representation: __repr__ and __str__
  v. Right side and In-Place Uses: __radd__ and __iadd__
  vi. Call Expressions: __call__
  vii. Comparisons: __lt__, __gt__ and others
  viii. Boolean Tests: __bool__ and __len__
  ix. Destructors: __del__

- Special features of Classes - coding/Hands-on
  i. Inheritance: “IS-a” relationship
  ii. Composition: “HAS-a” relationship
  iii. Pseudo private class attributes
  iv. Bound and unbound method objects
  v. Class objects
  vi. “Mix-In” classes

- Advanced Class Topics - coding/Hands-on
  i. “New style” class model
  ii. Diamond inheritance change

iii. MRO: Method Resolution Order
iv. Slots: Attribute Declarations
v. Properties: Attribute Accessors
vi. Static and Class methods
vii. The “super” built-in function

- Exception Basics - coding/Hands-on
  i. Why Exceptions?
  ii. Default Exception handler
  iii. Catching Exceptions
  iv. Raising Exceptions

- Coding Exceptions - coding/Hands-on
  i. The try/except/else statement
  ii. try/finally statement
  iii. raise statement
  iv. assert statement
  v. with/as context managers
  vi. Nesting Exception Handlers

- Exception Objects
  i. Class based exceptions
  ii. Why Exception hierarchies?
  iii. Built-in Exception Classes
  iv. Custom Exceptions

II) <u>Regular Expressions with Python:</u>

- What are regular expressions?
- regex module in python
- The match Function
- The search Function
- Matching vs searching
- Search and Replace
- Meta characters, advanced patterns

III) <u>Data Libraries:</u>

- Introduction to numpy
- Creating arrays
- Indexing Arrays
- Array Transposition
- Universal Array Function
- Array Processing
- Array Input and Output
- Introduction to Pandas, Series, Dataframes
- Data reading with Pandas
- Data cleaning with Pandas
- Data wrangling with Pandas
- Data selection with Pandas
- Data extraction with Pandas
- Introduction to Matplotlib
- Data Visualization with matplotlib

b) **Mathematics:** ~ 0.5 month

- Linear Algebra
- Statistics:
- Probability:
- Differential Calculus:

c) **Machine Learning:** ~ 2 month

- Introduction to Machine Learning:
  i. What is Machine Learning
  ii. Why use Machine Learning
  iii. Examples of ML applications
  iv. Types of ML Systems.
  v. Supervised Learning
  vi. Unsupervised Learning
  vii. Batch vs Online Learning
  viii. Instance-based vs Model-based Learning
  ix. Challenges of Machine Learning
  x. Overfitting vs underfitting training data
  xi. All phases of End to End ML Project.

- Classification Models

i. Binary Classifier
ii. Performance Measures
    1. Accuracy
    2. Confusion Matrix
    3. Precision and Recall
    4. ROC Curve
iii. Multi Class Classification
iv. Error Analysis
v. Multi Label Classification
vi. Multi Output Classification

- Regression Models
  i. Linear Regression
  ii. Gradient Descent
      1. Batch Gradient Descent
      2. Stochastic Gradient Descent
      3. Mini-batch Gradient Descent
  iii. Polynomial Regression
  iv. Regularized Linear Models
      1. Ridge Regression
      2. Lasso Regression
      3. Early Stopping
  v. Logistic Regression

- Support Vector Machines
  i. Linear SVM Classification
  ii. Soft Margin vs Hard Margin Classification
  iii. Nonlinear SVM Classification
  iv. SVM Regression Models

- Decision Trees
  i. Introduction to Decision Tree
  ii. Training Decision Tree
  iii. Visualizing Decision Tree
  iv. Estimating Class Probabilities
  v. The CART Training Algorithm
  vi. Computational Complexity
  vii. Gini Impurity vs Entropy
  viii. Regularization of Hyperparameters
  ix. Regression, Instability

- Random Forests
  i. Ensemble Learning
  ii. Voting Classifiers
  iii. Bagging and Pasting
  iv. Bagging & Pasting in sci-kit Learn
  v. Out of bag evaluation
  vi. Random Patches and Random Subspaces
  vii. Random Forests
  viii. Extra Tree and Feature importance
  ix. Introduction to Boosting
  x. AdaBoost and Gradient Boost
  xi. Stacking

- Unsupervised Learning Techniques
  i. Clustering
  ii. K-Means Algorithm and Limits of K-Means
  iii. Preprocessing using Clustering
  iv. Semi-supervised Learning using Clustering
  v. DBSCAN

d) **Deep Learning:** ~ 1.5 month

- Introduction to Artificial Neural Networks with Keras
  i. Biological Neuron
  ii. The Perceptron
  iii. Multilayer Perceptron and Back propagation
  iv. Regression MLPs
  v. Classification MLPs
  vi. Implementing MLPs with Keras
  vii. Building an Image classifier using Sequential API
  viii. Building Regression MLP using sequential API
  ix. Building complex Models using Functional API
  x. Saving and Restoring Model
  xi. Fine tuning neural network hyper parameters
  xii. Number of Hidden layers
  xiii. Number of neurons per hidden layer
  xiv. Learning rate, Batch size and other hyper parameters

- Training Deep Neural Networks

i. Vanishing/Exploding Gradients problems.
    1. Glorot and He initialization
    2. Nonsaturating Activation Functions
    3. Batch Normalization
    4. Gradient Clipping
ii. Vanishing/Exploding Gradients problems.
    1. Transfer Learning
    2. Unsupervised Pretraining
iii. Faster optimizers
iv. Avoiding overfitting through Regularization
    1. I1 & l2 regularization
    2. Dropout
    3. MC (Monte Carlo) Dropout
    4. Max-Norm Regularization

- Loading and Preprocessing Data with Tensorflow
    i. Quick tour of Tensorflow
    ii. Tensors and Operations
    iii. Tensors vs numpy
    iv. The Data API
        1. Chaining Transformations
        2. Shuffling Data
        3. Preprocessing Data
        4. Prefetching Data
    v. Preprocessing Input features
        1. Encoding categorical Features using one hot vectors
        2. Encoding categorial features using Embeddings
        3. Keras Preprocessing Layers
    vi. TF Transform
    vii. TFDS Project

- Generative AI Overview
    i. Introduction to Generative AI
    ii. Architecture of Gen AI
    iii. Introduction to LLM (Large Language Models)
    iv. Foundational or Public LLMs
    v. Various components in LLMs
    vi. Character splitters
    vii. Embeddings and Vector stores
    viii. Significance of RAGs in LLMs

ix. Prompts in LLMs
x. Prompt Engineering