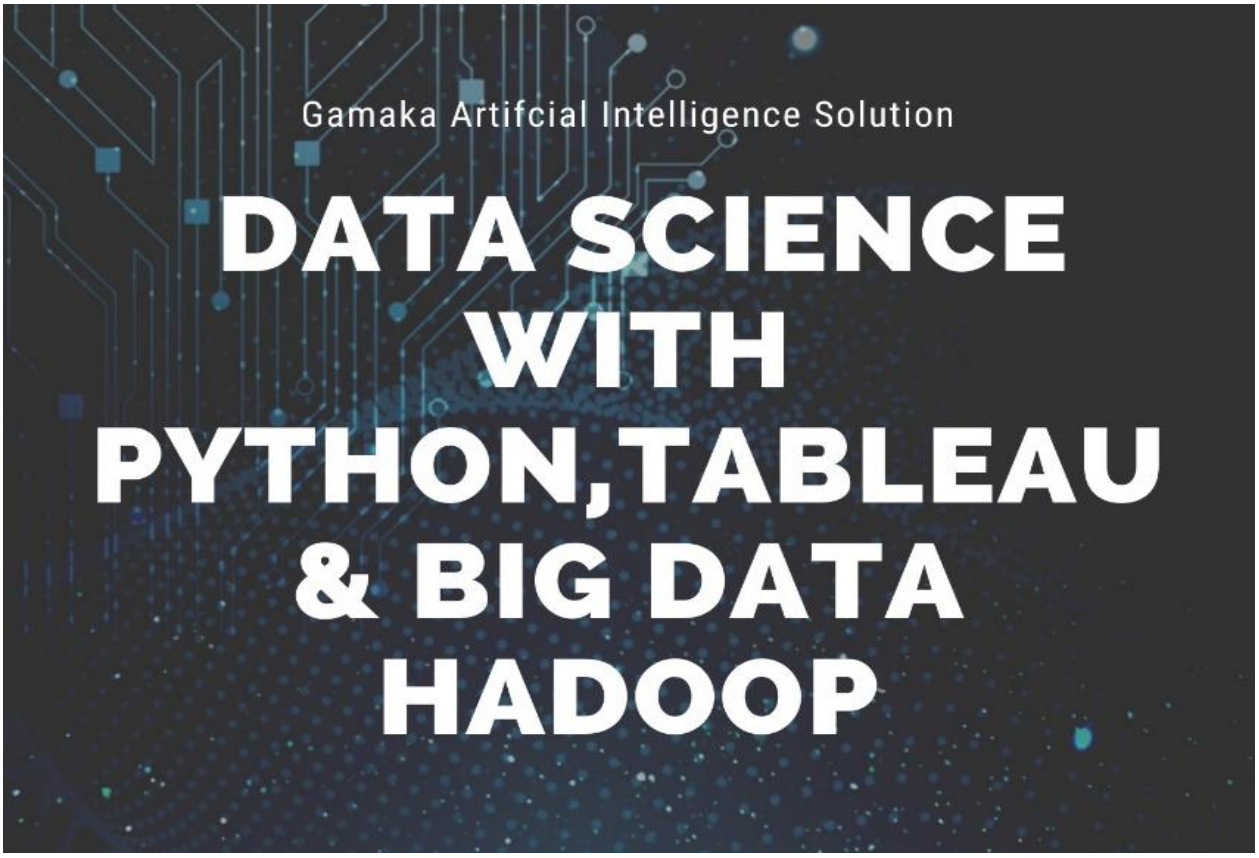
GAMAKA

Artificial Intelligence Solutions



Pune & Mumbai

WhatsApp: +91-7378493293
Phone: +91-7378483656

enquiry@gamakaai.com
gamakaai.com

Table of Contents

I. **Introduction** 2

- About US
- Target Audience
- Data Science Process
- Impact of Data Science

II. **Data Science with Python** 5

- Program Structure
- Projects/Case Studies
- Syllabus – Python(Basic & Advanced)
- Syllabus – Machine Learning
- Syllabus - Deep Learning with TensorFlow, NLP, Neural Networks

III. **Tableau** 16

- Why Tableau?
- Program Structure:
- Syllabus Tableau
- Assignments:
- Projects/Case Studies:

IV. **Big Data - Hadoop** 21

- Program Structure
- Syllabus - Bigdata & Hadoop
- Projects/Case Studies

V. **What You Get!!!** 33

- Course Completion Certificate
- Internship Certificate
- Advantages of joining GAMAKA AI

VI. **Struggling to Get a Job?** 36

- Industry Recruitment Challenge
- Strategies to get a job
- Trainer Role
- Our Students Placed Companies

Introduction

About US

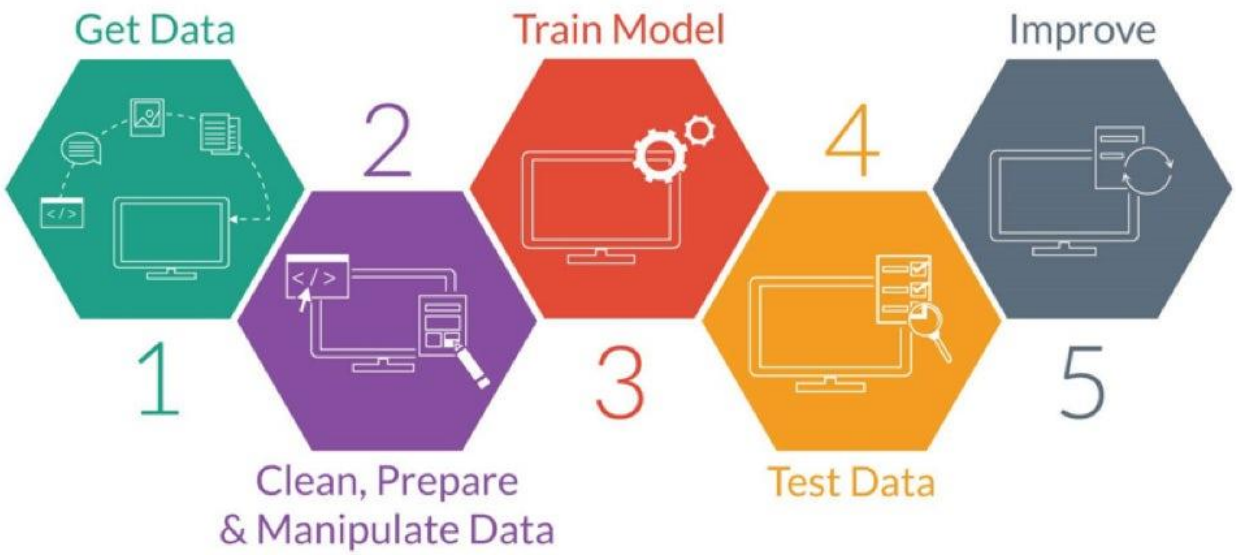
- Gamaka AI is a leading High-End Training on Emerging Technology and Placement company in India managed by IT veterans with more than a decade experience in leading MNC companies.
- We are known for our practical approach towards trainings that enable students to gain real-time exposure on competitive technologies. Trainings are offered by employees from MNCs to give a real corporate exposure.

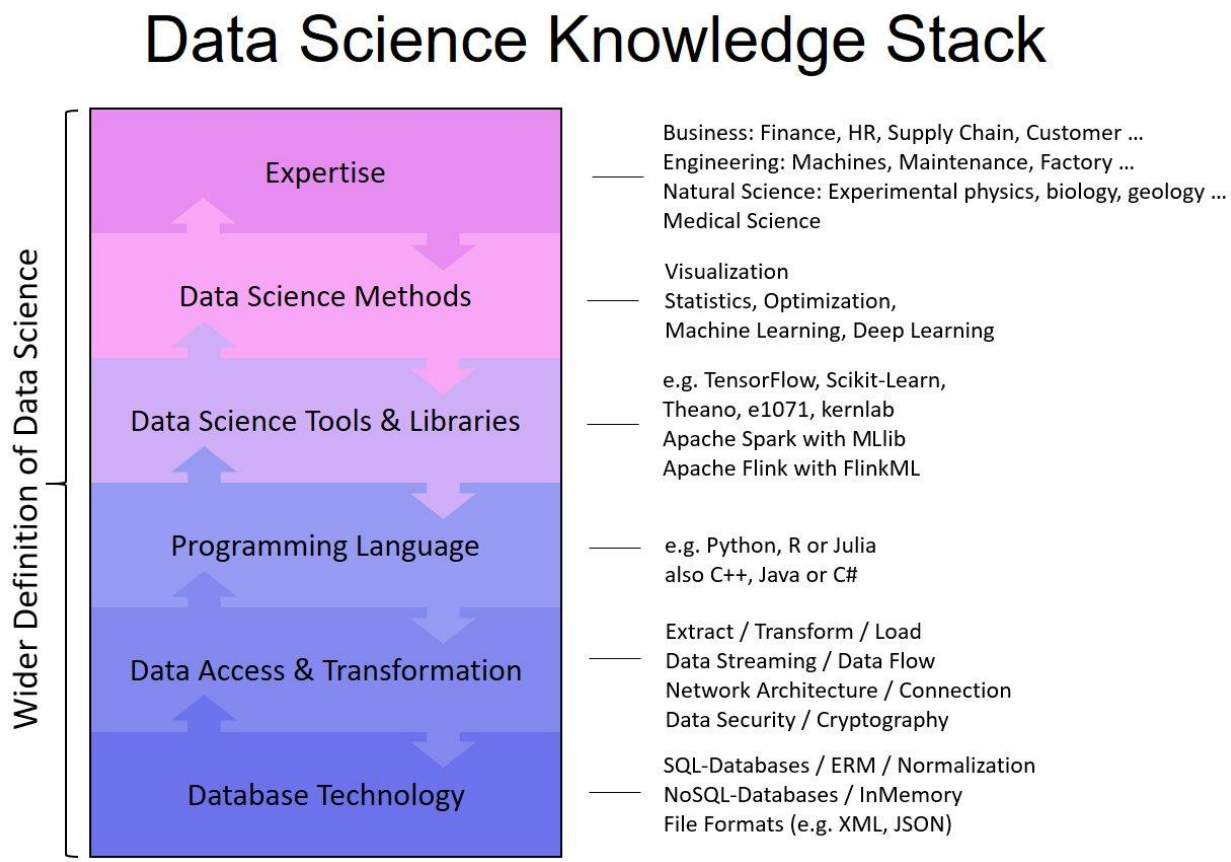
Target Audience

- Freshers from BCA, BCS, BE, BTech, MTech, MCA. MCS
- Final Year/Internship projects for BCA, BCS, BE, BTech, MTech, MCA. MCS
- Non-IT Professionals who've worked mostly with tools like Excel and want to learn how to use R for statistical analysis.
- Business Analyst
- IT Project Managers
- MBA Graduates or business professionals who are looking to move to a heavily quantitative role.
- Engineering Undergraduate/Graduate/Professionals who want to understand basic statistics and lay a foundation for a career in Data Science

No Prior Programming/Coding Skills Required

Data Science Process





Impact of Data Science

Data Science with Python

Program Structure

- Python – Basic & Advanced
- Mathematics & Statistics
- Machine Learning, Deep Learning
- TensorFlow, Kera's
- Computer Vision, NLP
- Data Science – Advanced
- 5 Live Projects / Case Studies
- Internship (Optional)

Note: Separate batch & extra sessions for NON-IT Professionals to build strong programming skills from scratch.

Duration: 3 Months / 90+ hours

Interview Preparation, Resume Building, GIT Profile, 100% Placement Assistance, Projects (part of internship)

Note: Separate batch & additional 1-month extra sessions for NON-IT Professionals to build strong programming skills from scratch.

Projects/Case Studies

- Forecasting Stock and Commodity Prices
- Build your own image recognition model with TensorFlow
- Customer Segmentation and Effective Cross Selling
- Predict fraud with data visualization & predictive modelling
- Chatbot Project using Microsoft Luis/Google Dialog flow/Amazon Lex.
- Deep Learning - Customer Feedback analysis using RNN LSTM.
- Deep Learning - Family member detection.
- Deep Learning - Industry financial growth prediction.
- Deep Learning - Speech recognition-based attendance system.
- Deep Learning - Vehicle Number plate detection and recognition system
- Forecasting Stock and Commodity Prices
- Build your own image recognition model with TensorFlow
- Web Scrapping - Web crawlers for image data sentiment analysis and product review sentiment analysis.
- Predict fraud with data visualization & predictive modelling
- Analyzing Movie Reviews Sentiment.
- Analyzing Music Trends and Recommendations
- Time Series - Arimam, Sarima, Auto Arima
- Time series using RNN LSTM Build your own Recommendation System
- Build your own Python predictive modelling, regression analysis & machine learning Model
- Football Players (Estimating Population Mean from a Sample)
- Election Polling (Estimating Population Proportion from a Sample)
- A Medical Study (Hypothesis Test for the Population Mean) Employee Behavior (Hypothesis Test for the Population Proportion)
- A/B Testing (Comparing the means of two populations
- Customer Analysis (Comparing the proportions of 2 populations)
- Predictive medicine: prognosis and diagnostic accuracy
- Virtual assistance for patients and customer support
- Analyzing Wine Types and Quality
- Creation of drugs - allows choosing, which experiments should be done and incorporates all the new information in a continuous learning loop
- Clustering algorithms for customer segmentation
- Discovering similarities across my Spotify music using data, clustering and visualization
- An End-to-End Project on Time Series Analysis and Forecasting with Python
- Using LSTMs to forecast time-series
- Evolution of a salesman: A complete genetic algorithm tutorial for Python
- A Machine Learning Approach—Building a Hotel Recommendation Engine
- How To Create Data Products That Are Magical Using Sequence-to-Sequence Models
- Deployment of all the project In cloudfoundary , AWS , AZURE and Google cloud platform.
- Deployment - Expose, api to web browser and mobile application retraining approach of Machine learning model.
- Deployment - Devops infrastructure for machine learning model.
- Deployment - AUTO ML, Prediction based on streaming data.

Syllabus – Python(Basic & Advanced)

Introduction & Setup

- What is Python and history of Python?
- Why Python and where to use it?
- Discussion about Python 2 and Python 3
- Set up Python environment for development
- Discuss about IDE's like IDLE, Pycharm and Enthought Canopy
- Discussion about unique feature of Python
- Introduction to Anaconda Distribution
- What is Anaconda Distribution?
- How to install Anaconda?
- conda repository
- Anaconda Navigator
- pip and conda to get new package
- pip and conda commands
- set Virtual

Scripting

- First "Hello World" Python Program
- Start programming on interactive shell.
- Using Variables, Keywords
- Interactive and Programming techniques
- Comments and document interlude in Python

Functional Programming

- Python Core Objects and built-in functions
- Number Object and operations
- String Object and Operations
- List Object and Operations
- Tuple Object and operations
- Dictionary Object and operations
- Set object and operations
- Boolean Object and None Object
- Different data Structures, data processing
- Map, Filter & Reduce
- List Comprehension
- Generators & Yields

Conditional Statements and Loops

- What are conditional statements?
- How to use the indentations for defining if, else, elif block
- What are loops?
- How to control the loops, infinite loops
- How to iterate through the various object
- Sequence and iterable objects

UDF Functions and Object Functions

- What are various type of functions
- Create UDF functions
- Parameterize UDF function, through named and unnamed parameters
- Defining and calling Function
- Anonymous Functions - Lambda Functions
- String Object functions
- List and Tuple Object functions
- Dictionary Object functions

File Handling with Python

- Process text files using Python
- Read/write and Append file object
- File object functions
- File pointer and seek the pointer
- Truncate the file content and append dataFile test operations using os.path

Packages & Modules

- Python inbuilt Modules
- os, sys, datetime, time, random, zip modules
- Create Python UDM – User Defined Modules
- Define PYTHONPATH
- Create Python Packages
- init File for package initialization

Exceptional Handing and Object Oriented Python

- Python Exceptions Handling
- What is Exception?
- Handling various exceptions using try....except...else
- Try-finally clause
- Argument of an Exception and create self exception class
- Python Standard Exceptions
- Raising an exceptions, User-Defined Exceptions
- Object oriented features
- Understand real world examples on OOP
- Implement Object oriented with Python
- Creating Classes and Objects, Destroying Objects
- Accessing attributes, Built-In Class Attributes
- Inheritance and Polymorphism
- Overriding Methods, Data Hiding\
- Overloading Operators

Advanced Topics

- Decorators
- Managed Attributes
- Unicode & Byte String
- Metaclasses
- Generators
- Descriptors

Debugging, Framework & Regular expression

- Debug Python programs using pdb debugger
- Pycharm Debugger
- Assert statement for debugging
- Testing with Python using UnitTest Framework
- What are regular expressions?
- The match and search Function
- Compile and matching
- Matching vs searching
- Search and Replace feature using RE
- Extended Regular Expressions
- Wildcard characters and work with them

Database interaction with Python

- Creating a Database with SQLite 3,
- CRUD Operations,
- Creating a Database Object.
- Python MySQL Database Access
- DML and DDL Operations with Databases
- Performing Transactions
- Handling Database Errors

Python Libraries

- Numpy
- SciPy
- Stats Model
- Pandas

Syllabus – Machine Learning

Duration: 40 Hours with hands on tutorials, 15 Case Studies with Internship

Python Environment Setup and Essentials Hadoop Fundamentals

- Anaconda Python Distribution – Windows, Mac OS, Linux
- Jupyter Notebook Installation
- Variable Assignment
- Understanding Data Types: Integer, Float, String, None, Boolean, Typecasting
- Tuples: Create, Access, and Slice
- Dicts: Create, View, Access, and Modify
- Studying Basic Operations: 'in', '+', '*

Computing with Python – NumPy and SciPy

- Mathematical Computing with Python - NumPy
- Understanding NumPy
- ndarray: Purpose, Properties, Types
- ndarray: Class and Attributes
- How to Access Array Elements?
- Indexing, Slicing, Iteration, Indexing with Boolean Arrays
- Studying Universal Functions
- What is Shape Manipulation?
- Linear Algebra
- Scientific Computing with Python – SciPy
- Understanding SciPy
- Studying SciPy Sub-packages
- Sub-Packages: Integration and Optimize
- Sub-Packages: Statistics, Weave, I O
- Linear Algebra

Data Manipulation with Python

- Data Manipulation and Machine Learning with Python
- Data Manipulation with Python – Pandas
- Understanding Pandas
- Defining Data Structures
- Data Operations and Data Standardization
- Pandas: File Read and Write Support
- SQL Operation
- Machine Learning with Python – Scikit
- Natural Language Processing with Scikit
- NLP Environment Setup & Applications
- NLP Sentence Analysis & Libraries
- Scikit – Built-in Modules & Feature Extraction
- Scikit – Grid Search & Parameters

Fundamentals of Machine Learning

- Overview & Terminologies
- What is Machine Learning?
- Why Learn?
- When is Learning required?
- Data Mining
- Application Areas and Roles
- Types of Machine Learning
- Supervised Learning
- Unsupervised Learning
- Reinforcement learning

Machine Learning Concepts & Terminologies

- Steps in developing a Machine Learning application
- Key tasks of Machine Learning
- Modelling Terminologies
- Learning a Class from Examples
- Probability and Inference
- PAC (Probably Approximately Correct) Learning
- Noise
- Noise and Model Complexity
- Triple Trade-Off
- Association Rules
- Association Measures
- Sample Algorithms

Simple Linear Regression

- Correlation
- Regression
- Model Assumptions
- Estimation Process
- Least Squares Method
- The Coefficient of Determination
- Correlation and Regression
- Simple Linear Regression Assignments

Multiple Regression Analysis

- Introduction
- Design Requirements
- Assumptions
- Independence
- Normality, Homoscedasticity, Linearity
- Multiple Regression
- Formal Statement of the Model
- Estimating parameters of the model
- F-test for the overall fit of the model
- Multiple regression model Building
- Selecting the best Regression equation
- Examples/Use Cases
- Interpreting the Final Model
- Multicollinearity and its Diagnostics
- Examples/Use Cases
- Qualitative Independent Variables
- Indicator variables
- Interpretation of Regression Coefficients
- Examples/Use Cases
- Regression Diagnostics and Residual Analysis
- Multiple Linear Regression Using R & Python
- Multiple Regression Assignment

Logistic Regression Analysis

- Theory Behind Logistic Regression
- Assessing the Model and Predictors
- When and Why do we Use Logistic Regression?
- Binary
- Multinomial
- Interpreting Logistic Regression
- Sample size requirements
- he logistic function & Interpretation
- Methods for including variables
- Computational method

Maximum Likelihood Estimation

- Bernoulli distribution
- Multinomial distribution
- Gaussian distribution
- Assessing the Model
- Assessing Changes in Models
- Assessing Predictors
- Methods of Regression
- Complete Separation
- Overdispersion
- MLE using Python

Decision Trees

- Understanding the Concept
- Internal decision nodes
- Terminal leaves.
- Tree induction: Construction of the tree
- Classification Trees
- Entropy
- Selecting Attribute
- Partially learned tree
- Overfitting
- Causes for over fitting
- Overfitting Prevention (Pruning) Methods
- Reduced Error Pruning

- Information Gain
- Decision trees - Advantages & Drawbacks
- Ensemble Models

Random Forests

- Introduction & Motivation
- Ensemble Methods - Bagging, Boosting & Random Forests
- Ensemble Classifiers
- Ensemble Models
- How random forests work?
- Gini Index
- Operation of Random Forest
- Random forest algorithm
- Common variables for random forests
- Random Forest – practical consideration
- Random Forest – Features, Advantages and Disadvantages
- Limitations of random forest
- Random Forest using Python

Support Vector Machine

- Problem Definition
- Separating Hyperplanes
- Linear separable case
- Formula for the Margin
- Finding the optimal hyperplane
- The optimization problem
- The Legrangin Dual Problem
- Importance of the Support Vectors
- VC dimension
- Non-linear SVM
- Mapping the data to higher dimension
- The Kernel Trick
- Important Kernel Issues
- Soft Margin
- The primal optimization problem
- The Dual Formulation
- The "C" Problem: Overfitting and Underfitting
- Model selection procedure
- SVM For Multi-class classification
- Applications of SVM
- Advantages & Drawbacks of SVM

Bayesian Theory

- Axioms of Probability Theory
- Conditional Probability
- Independence
- Joint Distribution
- Baye's Rule
- Bayesian Categorization
- Generative Probabilistic Models
- Naïve Bayes Generative Model
- Naïve Bayesian Categorization
- Example & Exercises
- Naïve Bayes Classifier using Python

K-Nearest Neighbor (K-NN)

- Non-parametric methods
- k-Nearest Neighbor Estimator
- How to Choose k or h
- Strengths and Weaknesses

Boosting

- Gradient Boosting
- Extreme Gradient Boosting
- ADA Boost

Dimensionality Reduction

- Principal Components Analysis (PCA)
- Singular Value Decomposition (SVD)
- Latent Dirichlet Analysis (LDA)
- Latent Dirichlet Analysis (LDA

K Means Clustering

- Parametric Methods Recap
- Clustering
- Direct Clustering Method
- Mixture densities
- Classes v/s Clusters
- Non-Hierarchical Clustering
- K-Means
- Distance Metrics
- K-Means Algorithm
- K-Means Objective
- Color Quantization
- Vector Quantization
- Encoding/Decoding
- Soft Clustering
- Expectation Maximization (EM)
- EM Algorithm
- Feature Selection vs Extraction
- Seed Choice
- Uses of Clustering
- Clustering as Pre-processing

Time Series

- The Art of Forecasting
- Forecasting Approaches
- Qualitative Forecasting Methods
- Quantitative Forecasting Methods
- Time Series & its Components
- Trend
- Cyclical
- Seasonal
- Irregular
- Smoothing Methods
- Moving Average Method
- Exponential Smoothing Method
- Forecast Effect of Smoothing Coefficient
- Linear Time-Series Forecasting Model
- Forecast using Trend Models
- The Linear Trend Model
- Time Series Plot
- Seasonality Plot
- Trend Analysis
- Quadratic Time-Series Forecasting Model

Quadratic Time-Series Model Relationships

- Quadratic Trend Model
- Exponential Time-Series Forecasting Model
- Exponential Weight
- Exponential Trend Model
- Autoregressive Modeling
- Time Series Data Plot
- Auto-correlation Plot
- Evaluating Forecasts
- Quantitative Forecasting Steps
- Forecasting Guidelines
- Pattern of Forecast Error
- Residual Analysis

Data Visualization and Web Scraping

- Data Visualization and Matplotlib
- Python Libraries
- Features of Matplotlib
- Line Properties Plot with (x, y)
- Set Axis, Labels, and Legend Properties
- Alpha and Annotation
- Multiple Plots and SubPlots
- Python Web Scraping and Data Science
- The Parser
- Searching & Modifying the Tree
- Printing, Formatting, Encoding

Syllabus - Deep Learning with TensorFlow, NLP, Neural Networks

Deep Learning Fundamentals

- Introduction to Deep Learning
- Historical Context
- Advances in Related Fields
- Pre-requisites
- Installing the Required Libraries
- Deep Learning Frameworks
- Introduction of each framework - TensorFlow, Theano, Keras, Torch, Caffe
- Architecture of each framework

Introduction to Keras

- Overview of Keras
- Installation Procedure
- - Dependencies
- - TensorFlow backend
- - Theano backend
- Guiding Principles
- - Modularity
- - Minimalism
- - Easy Extensibility
- - Work with Python

Stochastic Gradient Descent (SGD)

- Optimization Problems
- Method of Steepest Descent
- Batch, Stochastic (Single and Mini-batch) Descent
- - Batch
- - Stochastic Single Example
- - Stochastic Mini-batch
- - Batch vs. Stochastic
- Challenges with SGD
- - Local Minima
- - Saddle Points
- - Selecting the Learning Rate
- - Slow Progress in Narrow Valleys
- Algorithmic Variations on SGD
- - Momentum
- Nesterov Accelerated Gradient (NAS)
- - Annealing and Learning Rate Schedules
- - Adagrad
- - RMSProp
- - Adadelta
- - Adam
- Tricks and Tips for using SGD
- - Preprocessing Input Data
- - Choice of Activation Function
- - Preprocessing Target Value
- - Initializing Parameters
- - Shuffling Data
- - Batch Normalization
- - Early Stopping
- - Gradient Noise

Artificial and Conventional Neural Network

- Building an ANN
- Building Problem Description
- Evaluation the ANN
- Improving the ANN
- Tuning the ANN
- Conventional Neural Networks
- CNN Intuition
- Convolution Operation
- ReLU Layer
- Pooling and Flattening
- Full Connection
- Softmax and Cross-Entropy
- Building a CNN
- Evaluating the CNN
- Improving the CNN
- Tuning the CNN

Feed Forward Neural Networks

- Unit
- - Overall Structure of a Neural Network
- - Cross Entropy
- - Squared Error

- - Expressing the Neural Network in Vector Form
- - Evaluating the output of the Neural Network
- - Training the Neural Network
- Deriving Cost Functions using Maximum
- Likelihood
- - Binary Cross Entropy
- - Cross Entropy
- - Summary of Loss Functions
- Types of Units/Activation Functions/Layers
- - Linear Unit
- - Sigmoid Unit
- - Softmax Layer
- - Rectified Linear Unit (ReLU)
- - Hyperbolic Tangent

TensorFlow

- TensorFlow installation
- Introduction to TensorFlow
- TensorFlow APIs
- Tensors
- Importing TensorFlow
- Building & Running a computational graph
- Variables: Creation, Initialization, Saving, and
- Loading
- Tensor Ranks, Shapes, and Types
- Sharing Variables
- Reading Data
- Supervisor: Training Helper for Days-Long
- Trainings.
- TensorFlow Debugger (tfdbg) Command-LineInterface Tutorial: MNIST
- How to Use TensorFlow Debugger (tfdbg) with
- tf.contrib.learn
- Exporting and Importing a MetaGraph
- TensorFlow Version Semantics
- TensorFlow Data Versioning: GraphDefs and
- Checkpoints
- TensorBoard: Suite of visualization tools

Convolutional Neural Networks (CNN)

- Convolution Operation
- Pooling Operation
- Convolution-Detector-Pooling Building Block
- Convolution Variants
- Intuition behind CNNs

Recurrent Neural Networks (RNN)

- RNN Basics
- Training RNNs
- Bidirectional RNNs
- Gradient Explosion and Vanishing
- Gradient Clipping
- LSTM (Long Short-Term Memory) wit
- Time Series
- Case Study

Residual

- Autoencoders
- Custom Metrics
- Hyperparameter tuning
- GPU Programming in Cloud: Case Study
- Distributed TensorFlow

Self-Organizing Maps

- Self-Organizing Maps
- SOMs Intuition
- Plan of Attack
- Working of Self-Organizing Maps
- Revisiting K-Means
- K-Means Clustering
- Reading an Advanced SOM
- Building an SOM

Boltzmann Machines

- Energy-Based Models (EBM)
- Restricted Boltzmann Machine
- Exploring Contrastive Divergence
- Deep Belief Networks
- Deep Boltzmann Machines
- Building a Boltzmann Machine
- Installing Ubuntu on Windows
- Installing PyTorch

Tableau

Why Tableau?

Here are the top 5 reasons which tell us why a career in Tableau is the best career move at the moment.

1. Soaring Demand for Tableau Professionals

By 2020, the world is set to generate 50 times the amount of data as in 2011, according to a study by International Data Corporation (IDC). With this humongous amount of data and real business implications at play, business organizations across the world have the need for an easy to use tool to analyze data and derive actionable insights from it. Tableau helps organizations do exactly this! Thus, the popularity of Tableau – the four time leader in Gartner's magic quadrant – is expected to go through the roof.

2. Tableau Salary – Rewarding Tableau Career

Not only there is a great demand for Tableau experts, there are huge rewards on offer as well. Tableau professionals are getting paid the best salaries in the industry, with an average of $106,000. The average salaries too, are on an upward trend with the recent average salaries going up to as high as $158,000.

In India, it is an equally rewarding career with a median salary of above Rs. 5 Lakhs.

Tableau salary trends in India as well as the US are on a sharp rise and are pitted to increase even more in the near future.

3. Top Companies looking for Tableau Talent

A quick scan through the current job openings reveals that quite a few top companies are looking for Tableau talent. Some of these companies include: Facebook, Dell, Applied Systems, Booz Allen Hamilton, NetJets, University of California, Groupon, General Motors, Sony Electronics, Sunguard, Bank of America, KPMG, Verizon among others. So if you aspire to work for the big names in the industry, a career in Tableau is the way towards it.

4. Variety of Job Roles on Offer

The best thing about a Tableau career is you have a variety of job roles to choose from and at varioous levels in your career. Following are some of the the hottest job titles for Tableau professionals.

- Tableau Consultant
- Data Analyst
- Business Analyst
- Business Intelligence Analyst
- Business Intelligence Developer
- Business Intelligence Manager

5. Bright Future for Tableau

In 2015, Tableau was named as a "Leader" in the data visualization and business intelligence market for the 4th consecutive time by Gartner Research. Tableau is by far the market leader with respect to its competitors if we compare its "Ability to execute". Tableau is also a strong contender if you consider the "Completeness of Vision". It just goes to show that the future of Tableau is very bright and secure.

Program Structure

- Introduction
- Visualization Design and Data Types
- Tableau and Data Connections
- Data Organization
- Formatting and Annotations
- Chart Types
- Calculations
- Parameters
- Mapping and Locations
- Dashboards and Work Sharing

Duration: 2 Months / 60+ hours

Syllabus Tableau

Duration: 60+ Hours with hands on tutorials

Introduction

- What is Data Visualization?
- Scope of Data Visualization
- Tableau and its uses
- Scenario and Objectives
- Installation and Application
- Features and Architecture of Tableau
- Terminology and Definitions
- Tableau Work Space
- Files and Folders

Visualization Design and Data Types

- Defining Data
- Terminology of Data
- Types of Data
- Data Roles
- Dimension vs Measure
- Continuous vs Discrete
- Exporting Data
- Connecting Sheets
- Tableau Visualization Engine

Tableau and Data Connections

- Understanding Data Connections
- How to connect to Tableau Data Server?
- Data Connections: Joining and Blending
- Defining a Join
- Various Kinds of Join
- Usage of Join
- Custom SQL Enabled
- Data Blending and Tableau
- Usage of Data Blending
- Data Blending in Tableau
- What is Kerberos Authentication
- Working of Kerberos Authentication

- Right Outer Join

Data Organization

- Need to Organize Data
- How to Organize and Simplify Data
- What is Filtering
- How to Apply a Filter to a View?
- Filtering on Dimensions
- Totals and Sub totals
- Aggregating Measures
- Data Spotlighting
- Summary Card
- String Functions and Logical Functions
- What is Sorting
- How to Sort Data in Tableau
- Types of Sorting
- Combined Fields
- Group and Aliases
- Hierarchies
- Sets
- Tableau Bins
- Fixed Size and Variable Sized Bins
- Drilling
- Drilling Methods
- Aggregations

Formatting and Annotations

- Understanding Formatting and Annotations
- What is Spatial Analysis
- What is built-in Geocoding
- What is Custom Geocoding
- How to add Caption to Views?
- Adding Tooltips to Views
- Using Title Caption and Tooltip
- Formatting the Axes
- Edit Axis Option
- Formatting Window
- How to Format Mark Labels

Chart Types

- Objectives of Chart Types
- How to Use Dual Charts
- What is Dual Axis?
- Using Combination Charts
- How to Use Gantt Charts for Activity Tracking
- Using Motion Chart
- What are Box and Whisker Plots
- Using Reference Lines and Reference Bands
- What is Pareto Analysis
- What are Water Fall Charts
- How and What of Market Basket Analysis

Calculations

- Objectives of Calculations
- Strings Date Logical Calculation
- Arithmetic Calculations
- Aggregation Options
- Grand Totals and Sub-Totals
- Quick Table Calculations
- Custom Table Calculations
- Ad-hoc Analytics
- LOD Calculations
- Parallel Period
- Moving Averages
- Running totals
- Window Averages
- Trend Lines
- Predictive Models

Parameters, Mapping, and Locations

- What is a Parameter
- How to create a Parameter
- Parameter Controls
- What is Mapping
- Modifying Locations within Tableau
- Importing and Modifying Custom Geocoding
- Background Image
- Exploring Geographic Search
- Pan Zoom Lasso and Radial Selection

Dashboards and Work Sharing

- What is a Dashboard?
- How to build Dashboards
- How to build Interactive Dashboards
- What are Action Filters?
- How to create Story Boards
- Best Practices to create Dashboards
- Cover Pages
- Annotations
- Tool Tips and keyboard short cuts
- Sharing work
- Tableau Online
- Tableau Reader
- Tableau Public

Assignments:

01. Tableau Deep Dive: LOD – Introduction to Detail
02. Tableau Deep Dive: LOD – The Include Calculation
03. Tableau Deep Dive: LOD – The Exclude Calculation
04. Tableau Deep Dive: LOD – The Fixed Calculation
05. Tableau Deep Dive: LOD – LOD Calculations vs. Table Calculations
06. Tableau Deep Dive: Parameters – Parameter Overview
07. Tableau Deep Dive: Parameters – Parameter Properties
08. Tableau Deep Dive: Parameters – Filtering – Top N
09. Tableau Deep Dive: Parameters – Calculated Fields
10. Tableau Deep Dive: Parameters – Filtering Across Data Sources
11. Tableau Deep Dive: Parameters – Bins
12. Tableau Deep Dive: Parameters – Reference Lines
13. Tableau Deep Dive: Parameters – Table Calculations
14. Tableau Deep Dive: Sets – Introduction to Sets
15. Tableau Deep Dive: Sets – Constant Sets
16. Tableau Deep Dive: Sets – Computed Sets
17. Tableau Deep Dive: Sets – IN/OUT
18. Tableau Deep Dive: Sets – Combined Sets
19. Tableau Deep Dive: Sets – Calculated Fields
20. Tableau Deep Dive: Sets – Hierarchies
21. Tableau Deep Dive: Dates – Introduction to Dates
22. Tableau Deep Dive: Dates – Preparing Dates
23. Tableau Deep Dive: Dates – More Date Functions
24. Tableau Deep Dive: Dates – Exact Dates
25. Tableau Deep Dive: Dates – Custom Dates
26. Tableau Deep Dive: Dates – Rolling Dates
27. Tableau Deep Dive: Dates – Calendar Filters
28. Tableau Deep Dive: Dates – Week-by-Week Comparison
29. Tableau Deep Dive: Dashboard Design – Planning
30. Tableau Deep Dive: Dashboard Design – Layout & Structure
31. Tableau Deep Dive: Dashboard Design – Proof of Concept
32. Tableau Deep Dive: Dashboard Design – Adding Interactivity
33. Tableau Deep Dive: Dashboard Design – Visual Best Practices
34. Tableau Deep Dive: Dashboard Design – Optimization & Governance
35. Tableau Deep Dive: Dashboard Design – Publishing

36. Tableau Deep Dive: Table Calculations – Custom Sorts, Part One
37. Tableau Deep Dive: Table Calculations – Custom Sorts, Part Two
38. Tableau Deep Dive: Table Calculations – Custom Sorts, Part Three

Projects/Case Studies:

Any 2 Case Studies (T & C apply)

01. Ryan Soares explores how stay-at-home orders have affected community mobility trends
02. Iron Viz 2019 Agriculture Results
03. See how the 2019 Iron Viz Europe entries visualize energy and sustainability
04. Forecasting & Time Series with Tableau
05. TabPy - Integrating Tableau and Python

Big Data - Hadoop

Program Structure

- Chart Types
- Hadoop Fundamentals
- Hadoop Java API
- Pig Latin(basic & advanced)
- Hive (basic & advanced)
- HBASE & Zookeepar
- Sqoop
- Flume
- Oozie & Hue
- Yam Architecture
- Java for Hadoop
- Spark ecosystems & Big Data
- Scala & Spark, Spark Shell & PySpark
- RDD & Spark
- Spark Streaming
- Machine Learning & Mlib Spark SQL & GRAPHX

Duration: 2 Months / 60+ hours

Syllabus - Bigdata & Hadoop

Hadoop Fundamentals

- What is Bigdata?
- Evolution of Bigdata
- Types of Data and their Significance
- Need for Bigdata Analytics
- Why Bigdata with Hadoop?
- History of Hadoop
- Why Hadoop is in demand in market nowadays?
- Limitations of SQL based Tools
- Hadoop Nodes
- Hadoop Rack
- Hadoop Cluster
- Architecture of Hadoop
- Characteristics of Namenode
- Workaround with Datanodes
- Single node cluster and Multi node cluster setup
- Hadoop installation
- Introduction to Hadoop FS and Processing Environment's UIs
- How to read and write files
- Basic Unix commands for Hadoop
- Hadoop FS shell
- Hadoop releases practical
- Hadoop daemons practical
- Common Hadoop Shell Commands
- An Overview of Hadoop Administration
- How Hadoop is getting two categories Projects
- New projects on Hadoop

- Significance of JobTracker and Tasktrackers
- Hase co-ordination with JobTracker
- Secondary Namenode usage and Workaround
- Hadoop Releases and their Significance
- Introduction to Hadoop Release-1
- Hadoop Daemons in Hadoop Release-1
- Introduction to Hadoop Release-2
- Hadoop Daemons in Hadoop Release-2
- Hadoop Cluster Demo
- Hadoop 2.x Cluster Architecture
- A Typical Production Hadoop Cluster
- Hadoop Cluster Modes
- Hadoop 2.x Configuration Files
- Hadoop Storage – HDFS (Hadoop Distributed file system)
- Hadoop Processing Framework (Map Reduce / YARN)
- Alternates of Map Reduce
- Why NOSQL is in much demand instead of SQL
- Distributed warehouse for HDFS
- YARN Architecture
- Significance of Scalability of Operation
- Use cases where not to use Hadoop
- Use cases where Hadoop Is used Facebook,Twitter,Snapdeal, Flipkart

Working with Pig Latin - II (Advanced)

- Working with Binary Storage and Text Loader
- Bigdata Operations and Read write Analogy
- Filtering Datasets
- Filtering rows with specific condition
- Filtering rows with multiple conditions
- Filtering rows with String Based Conditions
- Sorting DataSets
- Sorting rows with Specific column or columns
- Multi level Sort
- Analogy of a Sort Operation
- Grouping Datasets and Co-grouping data
- Joining DataSets
- Types of Joins supported by Pig Latin
- Aggregate Operations like average, sum, min, max, count
- Flatten Operator
- Creating a UDF (USER DEFINED FUNCTION) using java
- Calling UDF from a Pig Scrip
- Data validation Scripts

Hadoop on Amazon Cloud

- Introduction to Cloud Infrastructure
- Amazon SaaS, Paas and IaaS
- Creating EC2 Instance for Processing
- Creating S3 Buckets
- Deploying Data on to the Cloud
- Choosing size of our instance
- Configuration of EMR Instance
- Creating a virtual cluster on Amazon
- Deploying project and getting stats

Flume

- Introduction to Flume
- Introduction to Apache Flume
- Flume Model
- Flume Goals
- Scalability in Flume
- Flume Data Integration
- Flume Installation on Single Node and Multinode Cluster
- Flume Architecture and various Components
- Data Sources: Types and Variants
- Data Target: Types and Variants
- Deploying an agent onto a single node cluster
- Problems associated with Flume
- Interview questions based on Flume

Yarn Architecture

- Introduction to YARN and MR2 daemons
- Active and Standby Namenodes
- Resource Manager and Application Master
- Node Manager
- Container Objects and Container
- Namenode Federation
- Cloudera Manager and Impala
- Load balancing in cluster with namenode federation
- Architectural differences between Hadoop 1.0 and 2.0

Scala and Spark

- Defining Scala
- Features of Scala
- Scala and Spark interdependency
- How to use Scala in other Frameworks
- What is Scala REPL
- Various Scala Operations
- Basic Data Types in Scala
- Understanding Basic Literals
- What are Operators
- Various Types of Operators
- What is Arithmetic Operator
- How to use Logical Operator
- Control Structures in Scala
- Functions and Procedures
- Anonymous Functions
- Objects and Classes
- Collections in Scala: Mutable vs Immutable Collection
- Array
- Array Buffer
- Map and Maps Operations
- Pattern Matching
- Tuples
- Lists
- Streams

Working with Key/Value Pairs

- Creating Pair RDDs
- Transformations on Pair RDDs
- Aggregations, Grouping Data, Joins, Sorting Data
- Data Partitioning
- Determining an RDDs Partitioner
- Operations that Benefit from Partitioning
- Operations that Affect Partitioning
- Loading and Saving Data
- File Formats: JSON, Comma-Separated and Tab-Separated Values
- File Formats: JSON, Comma-Separated and Tab-Separated Values
- File Formats: Sequence Files, Object Files
- Hadoop Input and Output Formats
- Filesystems: Local/Regular FS
- Amazon S3 and HDFS
- Databases: Java Database Connectivity
- Cassandra, HBase, ElasticSearch

Machine Learning with MLlib

- What is Machine Learning?
- System Requirements
- Machine Learning with Spark
- Spam Classification
- Data Types: Understand and working with Vectors
- Algorithms: Statistics, Classification, and Regression
- Mllib Clustering
- Mllib Collaborative Filtering
- Dimensionality Reduction
- Model Evaluation
- Configuring Algorithms
- Caching RDDs to Reuse
- Recognizing Sparsity
- Level of Parallelism
- Pipeline API

Hadoop Java API

- Hadoop Classes
- What is MapReduceBase?
- Mapper Class and its Methods
- What is Partitioner and types
- MapReduce Use Cases
- Traditional way VS MapReduce way
- Significance of MapReduce
- Hadoop 2. X MapReduce Architecture
- Hadoop 2. MapReduce Program
- Understanding Input Splits
- Relationship between Input Splits and HDFS Blocks
- MapReduce: Combiner & Partitioner
- Hadoop specific Data types
- Working on Unstructured Data Analytics
- What is an Iterator and its usage techniques
- Types of Mappers and Reducers
- What is Output collector and its Significance
- Workaround with Joining of datasets
- Complications with MapReduce
- Mapreduce Anatomy
- Anagram example,Teragen Example,Terasort Example
- WordCount Example
- Interview questions based on JAVA MapReduce
- Working with multiple mappers
- Working with weather data on multiple Data nodes in a Fully distributed Architecture
- Use Cases where MapReduce anatomy fails
- Advanced MapReduce
- Counters
- Distributed Cache
- MRunit
- Joins in MapReduce
- Reduce Side Join
- Replicated Join
- Composite Join
- Cartesian Product
- Custom Input Format
- Sequence Input Format
- XML File Parsing using MapReduce

Working with Hive

- Overview of Hive
- Background of Hive
- Hive VS Pig
- Installation and Configuration
- Interacting HDFS using HIVE
- Map Reduce Programs through HIVE
- Hive Architecture and Components
- Hive Commands
- Loading, Filtering, Grouping
- What is Meta Storage and Meta Store
- Derby Database
- HQL
- DDL, DML, and other Sub Languages of Hive
- Data types in Hive
- Partitions and Buckets
- Hive Tables: Managed and External
- Importing Data
- Querying Data
- Managing Outputs
- Hive Scipts
- Hive UDF
- Hive Operators
- Hive Joins, Unions, and Groups
- Sample Programs in Hive
- Alter and Delete in Hive
- Partition in Hive
- Indexing
- Industry Specific Configuration of Hive Parameters
- Authentication & Authorization
- Statistics with Hive
- Archiving in Hive
- Hands-on exercise

HBase & Zookeeper

- Introduction to HBase
- MapReduce Integration.

- HBase VS RDBMS
- HBase Components
- Hbase Architecture
- HBase Shell
- HBase Client API
- Data Loading Techniques
- Run Modes & Configuration
- HBase Cluster Deployment
- Regionservers and their implementation
- Client API's and their features
- How messaging system works
- Columns and column families
- Configuring hbase-site.xml
- Available Client
- Loading Hbase with semi-structured data
- Internal data storage in hbase
- Timestamps
- Creating table with column families
- HBase: Advanced Usage, Schema Design
- Load data from pig to hbase
- Zookeeper Data Model
- Zookeeper Service
- Challenges faced in Distributed Applications
- Coordination
- Znode
- Client API Functions
- Bulk Loading
- Receiving and Inserting Data
- Filters in HBase
- Sqoop architecture
- Data Import and export in SQOOP
- Deploying quorum and configuration throughout the Cluster

Oozie and Hue

- Introduction to Apache Oozie
- Oozie: Components
- Oozie: Workflow
- Scheduling with Oozie
- Hands-on Training on Oozie Workflow
- Oozie Coordinator
- Oozie Commands
- Oozie Web Console
- Oozie for MapReduce
- Hive in Oozie
- An Overview of Hue
- Hue in Real-time Scenarios
- Use Cases in Hue

Basics of JAVA for Hadoop

- The Java Virtual Machine
- Variables
- Data types
- Constructs: Conditional and Looping
- Types: Wrapper classes
- Object-Oriented JAVA
- Fields and Methods
- Constructors
- Overloading methods
- Garbage collection
- Nested classes
- Overriding methods
- Polymorphism
- Making methods and classes final
- Abstract classes and methods
- Interfaces
- Threads
- Classes
- The I/O Package
- JAVA Security

Programming Techniques in Scala

- What is a Class in Scala
- Understanding Getters and Setters
- What are Custom Getters and Setters
- General Properties of Getters
- What is an Auxiliary Constructor
- Defining Singletons
- What are Companion Objects
- How to extend a Class
- Overriding Methods
- Traits as Interfaces and Layered Traits

- What is a Primary Constructor

RDD and Spark

- Defining RDDs
- Transformations in RDD
- Various Actions in RDD
- Lazy Evaluations
- Passing Functions to Spark: Python, Scala, Java
- How to load data in RDD?
- How to save data through RDD?
- Scala RDD Extensions 00:00
- What are Double RDD Methods?
- RDD Methods
- Java Pair RDD Methods
- General RDD Methods
- Java RDD Methods
- Common Java RDD Methods
- Spark Java Function Classes
- Understanding Key-Value Pair RDD in Scala and Java
- MapReduce and RDD
- Spark and Hadoop Integration
- HDFS and Yarn

Spark SQL and GraphX

- Initializing Spark SQL
- SchemaRDDs
- Caching
- Loading and Saving Data
- Apache Hive, Parquet, JSON
- JDBC/ODBC Server
- Working with Beeline
- Spark SQL UDFs
- Hive UDFs
- What is a Graph-Parallel System?
- What are the Limitations of Graph-Parallel System?
- What is GraphX?
- What is Property Graph?
- What are Graph Operators?
- List of Operators
- Property and Structural Operators
- Subgraphs
- Join Operators
- How to create a Graph using GraphX
- Understanding Hive and Spark SQL
- An Insight into Spark SQL and SQL Context
- Hive and Spark SQL Integration
- Hive Queries through Spark
- Various Testing Tips in Scala
- Shared Variables
- Broadcast Variables
- Accumulators

Working with Pig Latin - (Fundamentals)

- Introduction to Pig Latin
- History and Evolution of Pig Latin
- Why Pig is used only with Bigdata
- MapReduce VS Pig
- Pig Architecture and Overview of Compiler and Execution Engine
- Programming Structure in Pig
- Pig Running Modes
- Pig Components
- Pig Execution
- Pig Release and Significance of Bugfixes
- Pig Specific Datatypes
- Complex Datatypes
- Bags, Tuples, Fields
- Joins and Cogroup
- Union
- Understanding Diagnostic Operators
- Specialized Joins in Pig
- Built in Functions
- Eval Function
- Load and Store Functions
- Math Function
- String Function
- Date Function
- Pig UDF
- Piggybank
- Parameter Substitution
- Pig Streaming

- Pig Specific Methods
- Comparison between Yahoo Pig & Facebook Hive
- Shell and Utility Commands
- Working with Grunt Shell
- Grunt commands: 17 in number
- Pig Latin: Relational Operators
- Pig Latin: File Loaders
- Pig Latin: Group Operator
- Cogroup Operator
- Pig Use Cases: Aviation and Healthcare
- Pig Data Input Techniques for flatfiles
- Flatfiles: Comma separated, Tab delimited, and fixed width
- Working with Schemaless Approach
- How to attach Schema to a file/table in Pig
- Schema referencing for similar Tables and Files
- Working with Delimiters

Advanced Hive

- Understanding Hive Releases
- Hive and OLTP
- OLAP in Hive
- Hive QL: Joining Tables
- Dynamic Partitioning & Bucketing
- Serialization and Deserialization
- Custom Map/Reduce Scripts
- Hive Indexes and Views
- Hive Query Optimizers
- Hive Architecture
- Understanding Thrift Server
- User Defined Functions
- Hue Interface for Hive
- Analyzing Data with Hive Script
- Difference between Hive and Impala
- UDFs in Hive
- Complex Use cases in Hive

Sqoop

- An Overview of Sqoop
- Sqoop Real-life Connect
- Sqoop and its Uses
- Advantages of Sqoop
- Sqoop Processing
- Sqoop Execution Process
- Importing Data Using Sqoop
- Sqoop Import Process
- How to Import data to Hive and HBase?
- How to Export Data from Hadoop using Sqoop?
- Sqoop Alternative
- Sqoop Connector

MongoDB

- Understanding MongoDB
- NoSQL Databases
- JSON and BSON
- Vertical and Horizontal Scaling
- Data Types
- MongoDB Tools
- Collection and Database
- Schema Design and Modeling
- CRUD Operations in MongoDB
- Indexing and Aggregation
- Replication and Sharding
- MongoDB Cluster Operations

Spark Ecosystem and BigData

- What and How of Distributed Systems
- What are New Generation Distributed Systems
- What is Big Data and its Limitations
- What are the Limitations of MapReduce in Hadoop
- Processing: Batch and Real-time Big Data Analytics
- Hadoop Ecosystem Overview
- Modes of Spark
- Overview of Spark on a cluster
- Spark Standalone Cluster
- Spark Web User Interface.
- Language Flexibility in Spark
- Architecture of Spark
- Spark and Big Data
- APIs in Spark
- Additional Benefits of Spark

- Understanding Apache Spark
- Evolution and Features of Spark
- Spark Ecosystem
- Various Tasks of Spark on a Cluster
- Apache Spark and Hadoop Ecosystem

Spark Shell and PySpark

- What is Spark Shell?
- How to create a Spark Context?
- How to load a file in Shell?
- Operations on files in Spark Shell
- Understanding SBT
- How to build a Spark Project with SBT?
- How to run Spark Project with SBT?
- What is a Local Mode?
- Spark Mode
- Distributed Persistence
- Built-in Libraries for Spark
- The PySpark Shell
- The PySpark Shell – Advanced
- Spark Tools
- PySpark Integration with Jupyter Notebook
- Case Study: Analyzing Airlines Data with PySpark

Spark Streaming

- What is Spark Streaming?
- Architecture and Abstraction of Spark Streaming
- Streaming Word Count
- What is a Micro Batch?
- Understanding DStreams
- Input DStreams and Receivers
- Basic and Advanced Sources
- Input Sources: Core Sources and Cluster Sizing
- Transformations in Spark Streaming
- Stateless and Stateful Transformations
- Transformations on DStreams
- Spark Streaming and Fault Tolerance
- Driver Fault Tolerance
- Worker Fault Tolerance
- Receiver Fault Tolerance
- Enabling Checkpointing
- Socket Stream and File Stream
- Stateful Operations
- Window Operations and its Types
- Join Operations-Stream-Dataset Joins
- Join Operations-Stream-Stream Joins
- Parallelism Level

Projects/Case Studies

Project 01: Working with MapReduce, Hive and Sqoop

Industry: General

Problem Statement: How to successfully import data using Sqoop into HDFS for data analysis

Topics: As part of this project, you will work on the various Hadoop components like MapReduce, Apache Hive and Apache Sqoop. You will have to work with Sqoop to import data from relational database management system like MySQL data into HDFS. You need to deploy Hive for summarizing data, querying and analysis. You have to convert SQL queries using HiveQL for deploying MapReduce on the transferred data. You will gain considerable proficiency in Hive and Sqoop after the completion of this project.

Highlights:
1.1 Sqoop data transfer from RDBMS to Hadoop
1.2 Coding in Hive Query Language
1.3 Data querying and analysis

Project 02: Work on MovieLens data for finding the top movies

Industry: Media and Entertainment

Problem Statement: How to create the top-ten-movies list using the MovieLens data

Topics: In this project you will work exclusively on data collected through MovieLens available rating data sets. The project involves writing MapReduce program to analyze the MovieLens data and creating the list of top ten movies. You will also work with Apache Pig and Apache Hive for working with distributed datasets and analyzing it.

Highlights:
2.1 MapReduce program for working on the data file
2.2 Apache Pig for analyzing data
2.3 Apache Hive data warehousing and querying

Project 03: Hadoop YARN Project; End-to-end PoC

Industry: Banking

Problem Statement: How to bring the daily data (incremental data) into the Hadoop Distributed File System

Topics: In this project, we have transaction data which is daily recorded/stored in the RDBMS. Now this data is transferred everyday into HDFS for further Big Data Analytics. You will work on live Hadoop YARN cluster. YARN is part of the Hadoop ecosystem that lets Hadoop to decouple from MapReduce and deploy more competitive processing and wider array of applications. You will work on the YARN central resource manager.

Highlights:
3.1 Using Sqoop commands to bring the data into HDFS
3.2 End-to-end flow of transaction data
3.3 Working with the data from HDFS

Project 04: Table Partitioning in Hive

Industry: Banking

Problem Statement: How to improve the query speed using Hive data partitioning

Topics: This project involves working with Hive table data partitioning. Ensuring the right partitioning helps to read the data, deploy it on the HDFS and run the MapReduce jobs at a much faster rate. Hive lets you partition data in multiple ways. This will give you hands-on experience in partitioning of Hive tables manually, deploying single SQL execution in dynamic partitioning and bucketing of data so as to break it into manageable chunks.

Highlights:
4.1 Manual Partitioning
4.2 Dynamic Partitioning
4.3 Bucketing

Project 05: Connecting Pentaho with Hadoop Ecosystem

Industry: Social Network

Problem Statement: How to deploy ETL for data analysis activities

Topics: This project lets you connect Pentaho with the Hadoop ecosystem. Pentaho works well with HDFS, HBase, Oozie and ZooKeeper. You will connect the Hadoop cluster with Pentaho data integration, analytics, Pentaho server and report designer. This project will give you complete working knowledge on the Pentaho ETL tool.

Highlights:
5.1 Working knowledge of ETL and Business Intelligence
5.2 Configuring Pentaho to work with Hadoop distribution
5.3 Loading, transforming and extracting data into Hadoop cluster

Project 06: Multi-node Cluster Setup

Industry: General

Problem Statement: How to setup a Hadoop real-time cluster on Amazon EC2

Topics: This is a project that gives you opportunity to work on real world Hadoop multi-node cluster setup in a distributed environment. You will get a complete demonstration of working with various Hadoop cluster master and slave nodes, installing Java as a prerequisite for running Hadoop, installation of Hadoop and mapping the nodes in the Hadoop cluster.

Highlights:
6.1 Hadoop installation and configuration
6.2 Running a Hadoop multi-node using a 4-node cluster on Amazon EC2
6.3 Deploying of MapReduce job on the Hadoop cluster

Project 07: Hadoop Testing Using MRUnit

Industry: General

Problem Statement: How to test MapReduce applications

Topics: In this project, you will gain proficiency in Hadoop MapReduce code testing using MRUnit. You will learn about real-world scenarios of deploying MRUnit, Mockito and PowerMock. This will give you hands-on experience in various testing tools for Hadoop MapReduce. After completion of this project you will be well-versed in test-driven development and will be able to write light-weight test units that work specifically on the Hadoop architecture.

Highlights:
7.1 Writing JUnit tests using MRUnit for MapReduce applications
7.2 Doing mock static methods using PowerMock and Mockito
7.3 MapReduce Driver for testing the map and reduce pair

Project 08: Hadoop Web Log Analytics

Industry: Internet Services

Problem Statement: How to derive insights from web log data

Topics: This project is involved with making sense of all the web log data in order to derive valuable insights from it. You will work with loading the server data onto a Hadoop cluster using various techniques. The web log data can include various URLs visited, cookie data, user demographics, location, date and time of web service access, etc. In this project, you will transport the data using Apache Flume or Kafka, workflow and data cleansing using MapReduce, Pig or Spark. The insight thus derived can be used for analyzing customer behavior and predict buying patterns.

Highlights:
8.1 Aggregation of log data
8.2 Apache Flume for data transportation
8.3 Processing of data and generating analytics

Project 09: Hadoop Maintenance

Industry: General

Problem Statement: How to administer a Hadoop cluster

Topics: This project is involved with working on the Hadoop cluster for maintaining and managing it. You will work on a number of important tasks that include recovering of data, recovering from failure, adding and removing of machines from the Hadoop cluster and onboarding of users on Hadoop.

Highlights:
9.1 Working with name node directory structure
9.2 Audit logging, data node block scanner and balancer
9.3 Failover, fencing, DISTCP and Hadoop file formats

Project 10: Twitter Sentiment Analysis

Industry: Social Media

Problem Statement: Find out what is the reaction of the people to the demonetization move by India by analyzing their tweets

Topics: This Project involves analyzing the tweets of people by going through what they are saying about the demonetization decision taken by the Indian government. Then you look for key phrases and words and analyze them using the dictionary and the value attributed to them based on the sentiment that they are conveying.

Highlights:
10.1 Download the tweets and load into Pig storage
10.2 Divide tweets into words to calculate sentiment
10.3 Rating the words from +5 to −5 on AFFIN dictionary
10.4 Filtering the tweets and analyzing sentiment

Project 11: Analyzing IPL T20 Cricket

Industry: Sports and Entertainment

Problem Statement: Analyze the entire cricket match and get answers to any question regarding the details of the match

Topics: This project involves working with the IPL dataset that has information regarding batting, bowling, runs scored, wickets taken and more. This dataset is taken as input, and then it is processed so that the entire match can be analyzed based on the user queries or needs.

Highlights:
11.1 Load the data into HDFS
11.2 Analyze the data using Apache Pig or Hive
11.3 Based on user queries give the right output

What You Get!!!

Course Completion Certificate

Will I get certified?

Upon successful completion of this data science course, you'll earn a Certificate. The certificate adds the required weight in any portfolio.

Internship Certificate

This certificate will be issued to those pursuing internships with our development team or clients with whom we have tie-ups. Data Science Internship gives opportunity to learn from professionals, gain practical experience in this field, and build a robust professional network.

GAMAKA

Artificial Intelligence Solutions

Office No 309, Paranjape – The Business Hub, Karve Road, Kothrud, Pune - 411038
Email: enquiry@gamakaai.com Cell: 91-7378483656. WhatsApp: 91-7378493293
www.gamakaai.com

03-Jun-2020

INTERNSHIP EXPERIENCE LETTER

This is to certify that Miss. Richa Bhat was working with Gamaka AI as Trainee Data Analyst on Internship.

Date of Joining	03 Feb 2020.
Date of leaving Service	29 May 2020.
Designation at the time of Leaving	Trainee Data Analyst

Scope of Work:
Worked as a Data Analyst in our IT development & consulting division.

Her job responsibilities were as follows:
- Application code design and development.
- Database query development

Tools & Technologies Used:
- Python 3.7, NumPy, SciPy, SciKit Learn, Panda, Matplotlib
- Mathematics, Statistics, Machine Learning – Supervised/Unsupervised
- Deep Learning – Neural Network, TensorFlow
- Tableau Desktop
- Big Hadoop 2

We found her sincere, hardworking & responsible.

We wish her all the success in her future endeavors.

Yours faithfully,
Sadeep Mane
Director

Note: The document does not carry signature due to COVID-19 situation

Advantages of joining GAMAKA AI

- Instructor led online classroom interactive sessions
- One-To-One online problem-solving sessions
- Complete Soft Copy of Notes & Latest Interview Preparation Set
- Trainers are working IT professional with top IT MNC's
- 100% Placement Assistance
- Resume Building & Mock Interview Sessions
- 100% Hands-on Training with Live Projects/Case Studies
- Internship & Course Completion Certificate
- 1 Year free subscriptions to Portal for Updated Guides, Notes, POC, Projects & Interview preparation set.
- Extensive training programs with Recorded Sessions
- 24*7 Support on enquiry@gamakaai.com

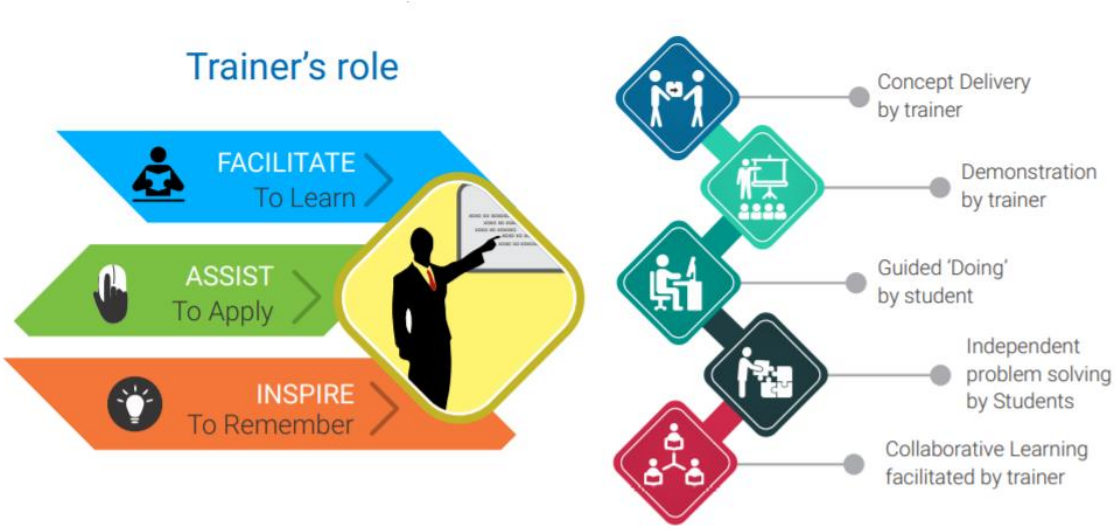
Struggling to Get a Job?

Industry Recruitment Challenge

Strategies to get a job

- Gain Industry Expertise, Internship Experience.
- Presentation skills & Grooming to face challenging interview
- Work on Industrial Projects/Case Studies
- Professional Resume & GIT Profile
- Interview Preparation with Mock Interviews
- Job Assistance & Placement

Trainer Role



Our Students Placed Companies