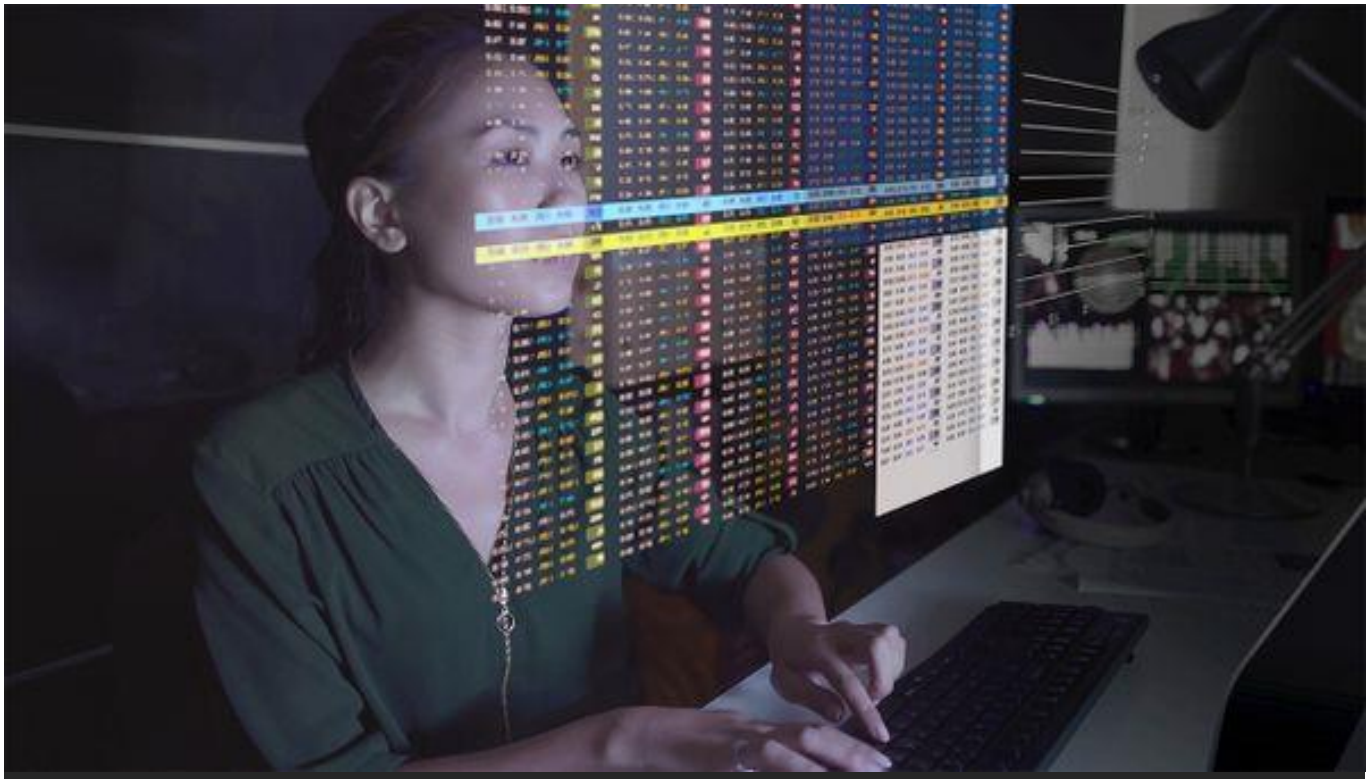
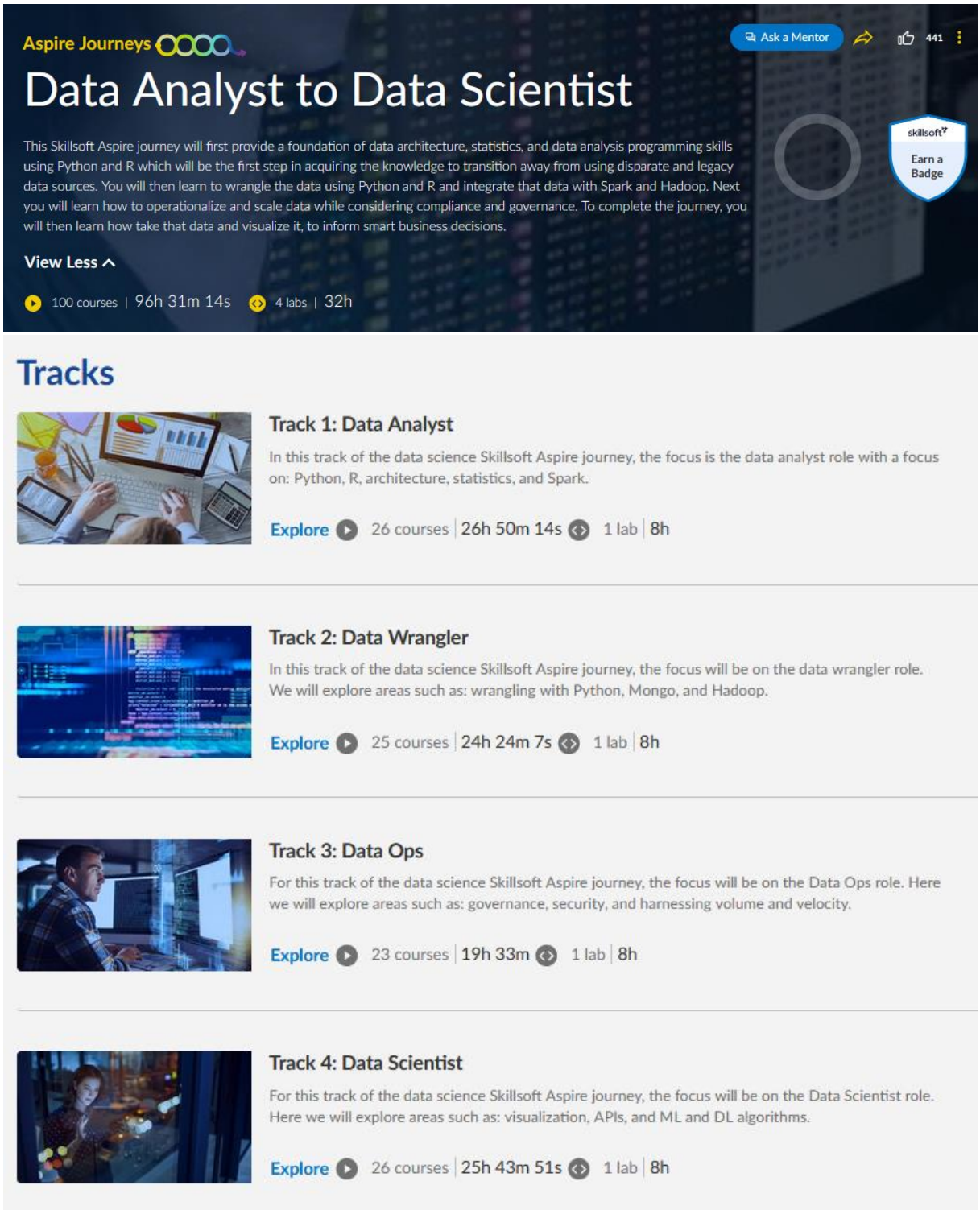


Data Analyst to Data Scientist

SKILLSOFT ASPIRE JOURNEY

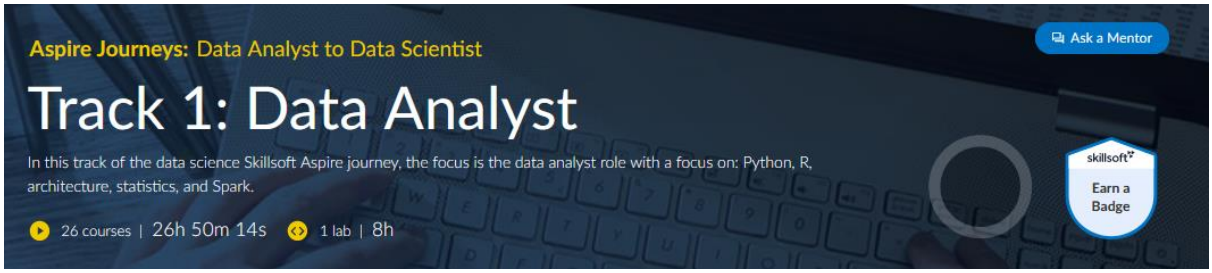
skillsoft | percipio



Prerequisites

We recommend the following prerequisite skills:

- Familiarity with Python and R programming
- Comfortable working with data with applications such as Excel
- Basic math and statistics skills
- Familiarity with cloud systems, such as Azure and AWS



Data Architecture Getting Started

Objectives:

- identify the relationship between data, information, and analytics
- recognize PII, PHI, and common data privacy regulations
- list the six phases of the data lifecycle
- compare and contrast SQL and NoSQL database solutions
- use Visual Paradigm to create a relational database ERD
- deploy Microsoft SQL Server in the Amazon Web Services cloud
- deploy DynamoDB in the Amazon Web Services cloud
- define what big data is and how it is managed
- recognize the relationship between data and how it is governed
- distinguish among the various types of data architectures, including the TOGAF enterprise architecture
- describe how organizations can derive value from data they already have
- implement effective data management solutions

Data Engineering Getting Started

Objectives:

- describe distributed systems from a data perspective
- identify the differences between batch and in-memory processing
- describe NoSQL stores and how they are used
- identify different tools available for data management
- describe the ETL process and different tools available
- use Talend Open Studio to showcase the ETL concept
- describe and create a data model
- describe the hierarchy of needs when working with AI and machine learning
- describe and create a data partition
- identify data engineering best practices
- describe data reporting tools
- create a data model

Python - Introduction to NumPy for Multi-dimensional Data

Objectives:

- identify the applications of NumPy
- install NumPy and learn how to create basic NumPy arrays
- create specialized NumPy arrays
- describe how arrays of different shapes and sizes can be displayed
- explore the different mathematical operations available when working with arrays
- work with functions which apply to each element of an array
- retrieve specific parts of an array using row and column indices
- describe the options available when iterating over 1-dimensional and multi-dimensional arrays
- perform reshape operations on arrays to visualize its contents in different ways
- utilize NumPy to perform basic array manipulation

Python - Advanced Operations with NumPy Arrays

Objectives:

- identify different ways in which arrays can be split up
- describe how grayscale and color images can be represented as multi-dimensional arrays
- perform some basic image manipulation after converting images to arrays
- create a view into a NumPy array and learn of the relationship between views and their base arrays
- compare deep copies of arrays with views and know when to use each of them
- use fancy indexing with arrays using an index mask
- use fancy indexing to analyze real-world data
- apply boolean masks to access array elements which fulfil a specific condition
- use structured arrays in order to store heterogeneous data
- describe how operations can be performed between arrays of mismatched shapes using broadcasting
- perform operations between arrays of mismatched shapes by applying broadcasting rules
- utilize NumPy to perform multi-dimensional array operations

Python - Introduction to Pandas and DataFrames

Objectives:

- understand the various applications of Pandas and why it is a building block in the field of data science
- install Pandas and create a Pandas Series
- work with Pandas Series by accessing elements using the default and a custom index
- define a Pandas DataFrame and describe how data can be stored and accessed in these data structures
- initialize and populate a simple Pandas DataFrame
- load data into a DataFrame from a CSV file
- edit individual cells and entire rows and columns in a Pandas DataFrame
- access specific rows and columns of a Pandas DataFrame using the index and labels
- access parts of a Pandas DataFrame based on specific conditions
- describe the concept of hierarchical index or multi-index and why can be useful
- re-orient a DataFrame as a pivot table to better visualize data
- apply a multi-index to a DataFrame and reshape it using the stack and melt operations
- work with Pandas for basic tabular data manipulation

Python - Manipulating & Analyzing Data in Pandas DataFrames

Objectives:

- learn how to iterate over a DataFrame's rows and columns
- export the contents of a DataFrame into files of various formats
- describe and apply the different techniques involved in handling datasets where some information is missing
- describe and apply the different techniques involved in handling datasets where some information is missing
- implement a hierarchical index and access the DataFrame's contents based on that index
- combine two similar DataFrames using the concat operation
- apply a join operation on two related but dissimilar DataFrames using the merge function
- load data into a Pandas DataFrame from a table in a relational database
- use Pandas for advanced tabular data manipulation

R Data Structures

Objectives:

- create vectors in R
- manipulate R vectors
- sort R vectors
- use lists in R
- create matrices in R
- perform matrix operations in R
- create factors in R
- create data frames in R
- perform data frame operations in R
- create and use a data frame

Importing & Exporting Data using R

Objectives:

- read data from a CSV formatted text file
- read data from an Excel spreadsheet
- read tabular data from a HTML file
- export tabular data from R to a CSV file
- export tabular data from R to an Excel spreadsheet
- export tabular data from R to an HTML table
- read data from an HTML table and export to CSV

Data Exploration using R

Objectives:

- use the dplyr library to load data frames
- select subsets of data using dplyr
- filter tabular data using dplyr
- perform multiple operations using the pipe operator
- create new columns using the mutate method
- summarize data using summary functions
- use the dplyr join functions to combine data
- use the group_by method from the dplyr library
- query data using various dplyr library functions

R Regression Methods

Objectives:

- perform the preparatory steps needed to create a linear model
- create a linear regression model using the lm method in R
- extract the results of a linear regression
- test the predict method on a linear model
- perform the preparatory steps needed to create a logistic model
- apply the glm method on a logistic regression problem
- create a linear regression model and use the predict method

R Classification & Clustering

Objectives:

- perform the preparatory steps needed to create a classification and decision tree
- use the rpart library to build a decision tree
- use the ctree library to build a decision tree
- perform the preparatory steps needed to carry out clustering
- use the k-means clustering method
- use hierarchical clustering with the hclust and cutree methods
- apply a decision tree method to a classification problem

Simple Descriptive Statistics

Objectives:

- enumerate objectives of descriptive and inferential statistics and distinguish between the two
- enumerate objectives of population and sample and distinguish between the two
- enumerate objectives of probability and non-probability sampling and distinguish between the two
- define the mean of a dataset and enumerate its properties
- define the median and mode of a dataset and enumerate their properties
- define the range of a dataset and enumerate its properties
- define the inter-quartile range of a dataset and enumerate its properties
- define the variance and standard deviation of a dataset and enumerate their properties
- differentiate between inferential and descriptive statistics, enumerate the two most important types of descriptive statistics, and define the formula for standard deviation

Common Approaches to Sampling Data

Objectives:

- describe important terms associated with the sampling process
- define sampling bias and describe problems caused by this phenomenon
- define simple random sampling and enumerate its properties
- define systematic random sampling and differentiate it from simple random sampling
- define stratified random sampling and differentiate it from simple and systematic random sampling
- define non-probability sampling and enumerate some non-probability sampling techniques
- define the two properties of probability sampling, enumerate three types of probability sampling, and list two types of non-probability sampling

Inferential Statistics

Objectives:

- draw the shape of a Gaussian distribution and enumerate its defining properties
- enumerate the steps involved in hypothesis testing and define the null and alternative hypotheses
- describe the role of test statistic and p-value in accepting or rejecting a null hypothesis
- enumerate types and uses of t-tests in hypothesis testing
- outline the significance of skewness and kurtosis and define the skewness and kurtosis of a normally distributed random variable
- calculate the autocorrelation of a time series
- define linear regression
- interpret the R-squared of a regression and identify overfitting
- differentiate between null and alternative hypotheses, enumerate four use cases for t-tests, and calculate the correlation of time series with itself

Apache Spark Getting Started

Objectives:

- recognize where Spark fits in with Hadoop and its components
- describe Spark RDDs and their characteristics, including what makes them resilient and distributed
- identify the types of operations which are permitted on an RDD and describe how RDD transformations are lazily evaluated
- distinguish between RDDs and DataFrames and describe the relationship between the two
- list the crucial components of Spark and the relationships between them and recognize the functions of the Spark Session, Master and Worker nodes
- install PySpark and initialize a Spark Context
- create and load data into an RDD
- initialize a Spark DataFrame from the contents of an RDD
- work with Spark DataFrames containing both primitive and structured data types
- define the contents of a DataFrame using the SQLContext
- apply the map() function on an RDD to configure a DataFrame with column headers
- retrieve required data from within a DataFrame and define and apply transformations on a DataFrame
- convert Spark DataFrames to Pandas DataFrames and vice versa
- describe basic Spark concepts

Hadoop & MapReduce Getting Started

Objectives:

- describe what big data is and list the various sources and characteristics of data available today
- recognize the challenges involved in processing big data and the options available to address them such as vertical and horizontal scaling
- specify the role of Hadoop in processing big data and describe the function of its components such as HDFS, MapReduce, and YARN
- identify the purpose and describe the workings of Hadoop's MapReduce framework to process data in parallel on a cluster of machines
- recall the steps involved in building a MapReduce application and the specific workings of the Map phase in processing each row of data in the input file
- recognize the functions of the Shuffle and Reduce phases in sorting and interpreting the output of the Map phase to produce a meaningful output
- recognize the techniques related to scaling data processing tasks, working with clusters, and MapReduce and identify the Hadoop components and their functions

Developing a Basic MapReduce Hadoop Application

Objectives:

- create and configure a Hadoop cluster on the Google Cloud Platform using its Cloud Dataproc service
- work with the YARN Cluster Manager and HDFS NameNode web applications that come packaged with Hadoop
- use Maven to create a new Java project for the MapReduce application
- develop a Mapper for the word frequency application that includes the logic to parse one line of the input file and produce a collection of keys and values as output
- create a Reducer for the application that will collect the Mapper output and calculate the word frequencies in the input text file
- specify the configurations of the MapReduce applications in the Driver program and the project's pom.xml file
- build the MapReduce word frequency application using Maven to produce a jar file and then prepare for execution from the master node of the Hadoop cluster
- run the application and examine the outputs generated to get the word frequencies in the input text document
- idenfity the apps packaged with Hadoop and the purposes they serve and recall the classes/methods used in the Map and Reduce phases of a MapReduce application

Hadoop HDFS Getting Started

Objectives:

- recognize the need to process massive datasets at scale
- describe the benefits of horizontal scaling for processing big data and the challenges of this approach
- recall the features of a distributed cluster which address the challenges of horizontal scaling
- identify the features of HDFS which enables large datasets to be distributed across a cluster
- describe the simple and high-availability architectures of HDFS and the implementations for each of them
- identify the role of Hadoop's MapReduce in processing chunks of big datasets in parallel
- recognize the role of the YARN resource negotiator in enabling Map and Reduce operations to execute on a cluster
- describe the steps involved in resource allocation and job execution for operations on a Hadoop cluster
- recall how Apache Zookeeper enables the HDFS NameNode and YARN ResourceManager to run in high-availability mode
- identify various technologies which integrate with Hadoop and simplify the task of big data processing
- recognize the key features of distributed clusters, HDFS, and the input outs of the Map and Reduce phases

Introduction to the Shell for Hadoop HDFS

Objectives:

- provision a Hadoop cluster on the cloud using the Google Cloud Platform's Dataproc service
- identify the various GCP services used by Dataproc when provisioning a cluster
- list the metrics available on the YARN Cluster Manager app and recognize how it can be useful to monitor job executions
- recall the details and metrics of HDFS available on the NameNode web app and how it can be used to browse the file system
- identify the tools of the Hadoop ecosystem which are packaged with Hadoop and recall how they can be accessed
- configure HDFS using the hdfs-site.xml file and identify the properties which can be set from it
- compare the hadoop fs and hdfs dfs shells and recognize their similarities to Linux shells
- explore apps for Hadoop, configure HDFS, work with HDFS shells

Working with Files in Hadoop HDFS

Objectives:

- identify the different ways to use the ls and mkdir commands to explore and create directories on HDFS
- transfer files from your local file system to HDFS using the copyFromLocal command
- copy files from your local file system to HDFS using the put command
- transfer files from HDFS to your local file system using the copyToLocal command
- use the get and getmerge functions to retrieve one or multiple files from HDFS
- work with the appendToFile and rm commands on the hdfs dfs shell
- utilize HDFS commands to work with and manipulate files using the HDFS shell

Hadoop HDFS File Permissions

Objectives:

- count the number of files and view their sizes on HDFS using the count and du commands
- configure and view permissions for individual files and directories using the getfacl and chmod commands
- define and set permissions for the entire contents of a directory with the chmod command
- write a simple bash script
- automate the transfer of all the files in a directory on your local file system over the HDFS with a shell script
- identify the data and metrics available on the HDFS NameNode UI and work with its file system explorer
- delete a Google Cloud Dataproc cluster and all of its associated resources
- work with file-permissioning in HDFS and recognize the data and metrics available in the NameNode UI

Data Silos, Lakes, & Streams Introduction

Objectives:

- recall the characteristics and drawbacks of data silos
- specify what a data lake enables
- recognize the advantages of using data lakes to store data
- describe the architecture of a data lake and identify challenges in its design
- recall the characteristics of a data warehouse
- specify the differences between data warehouses and data lakes
- distinguish between batch and streaming data and recognize the Stream-First Architecture
- describe how data can be moved from on-premise to the AWS cloud platform
- recognize the technologies used to build data lakes on AWS
- describe various use cases and architectures of working with data lakes on AWS
- recall characteristics of data silos, data lakes, and data streams

Data Lakes on AWS

Objectives:

- configure a custom role with specific permissions on AWS
- create an S3 bucket and upload files
- recognize the different operations that can be performed using the AWS Glue console
- create metadata tables in Glue using the web console
- perform queries on the Glue data catalog using Athena
- perform data crawling on S3 to automatically detect schemas
- execute queries on data in crawled tables
- perform crawling operations with multiple files in the same path
- merge data stored in multiple files in the same folder path
- merge data when files have the exact same schema
- recall the roles and features of the different AWS services used in the data lake architecture

Data Lake Sources, Visualizations, & ETL Operations

- Objectives:
- configure a Redshift cluster to store data
- load data into a Redshift cluster from S3 buckets
- configure a JDBC connection on Glue to the Redshift cluster
- crawl data on a Redshift cluster using a Glue crawler
- crawl data stored in a DynamoDB table
- configure the Amazon QuickSight business intelligence tool to visualize data
- build charts and dashboards in QuickSight
- define a job in Glue to perform ETL operations
- run ETL scripts using Glue
- perform ETL operations in Glue to backup data originally stored in Redshift
- perform ETL operations in Glue to backup data originally stored in DynamoDB
- recall how to use AWS services for visualizations and ETL

Applied Data Analysis

Objectives:

- install and configure Python using Anaconda
- install and configure R using Anaconda
- use Jupyter notebook to explore data
- read data from files and write data to files using the Python Pandas library
- import and export data in R
- recognize and deal with missing data in R
- use the Dplyr package in R to transform data
- work with the Python data analysis library NumPy
- work with the Python data analysis library Pandas
- perform exploratory data analysis in R using mean, median, and mode
- use the Python data analysis library Pandas to analyze data
- use the ggplot2 library to visualize data using R
- use Pandas built-in data visualization tools to visualize data using Python
- perform data analysis using R and Python

Analyzing Data with Python

Objectives:

- In this Skillsoft Aspire lab, you will practice performing data analysis tasks using Python by configuring VSCode, loading data from SQLite into Pandas, grouping data and using box plots. Then, test your data science skills by answering assessment questions after using Python to calculate frequency distribution, measures of center, and coefficient of dispersion.

This lab provides access to several tools commonly used in data science, including:

- VS Code
- Anaconda
- Jupyter Notebook + JupyterHub
- Pandas, NumPy, SiPy
- Seaborn Library
- Spyder IDE

Final Exam: Data Analyst

Objectives:

- build and run the application and confirm the output using HDFS from both the command line and the web application
- compare and contrast SQL and NoSQL database solutions
- configure a JDBC connection on Glue to the Redshift cluster
- configure and view permissions for individual files and directories using the getfacl and chmod commands
- configure HDFS using the hdfs-site.xml file and identify the properties which can be set from it
- crawl data stored in a DynamoDB table
- create and configure a Hadoop cluster on the Google Cloud Platform using its Cloud Dataproc service
- create and configure simple graphs with lines and markers using the Matplotlib data visualization library
- create and load data into an RDD
- Create data frames in R
- create matrices in R
- create vectors in R
- define linear regression
- define the contents of a DataFrame using the SQLContext
- define the inter-quartile range of a dataset and enumerate its properties
- Define the mean of a dataset and enumerate its properties
- delete a Google Cloud Dataproc cluster and all of its associated resources
- deploy DynamoDB in the Amazon Web Services cloud
- describe and apply the different techniques involved in handling datasets where some information is missing
- describe NoSQL Stores and how they are used
- describe the concept of hierarchical index or multi-index and why can be useful
- describe the ETL process and different tools available
- describe the options available when iterating over 1-dimensional and multi-dimensional arrays
- draw the shape of a Gaussian distribution and enumerate its defining properties
- edit individual cells and entire rows and columns in a Pandas DataFrame
- execute the application and verify that the filtering has worked correctly; examine the job and the output files using the YARN Cluster Manager and HDFS NameNode web UIs
- explain the concept of hierarchical index or multi-index and why can be useful
- export the contents of a DataFrame into files of various formats
- export the contents of a DataFrame into files of various formats
- identify different tools available for data management
- identify the various GCP services used by Dataproc when provisioning a cluster
- import and export data in R
- initialize a Spark DataFrame from the contents of an RDD
- install Pandas and create a Pandas Series
- list the six phases of the data lifecycle
- load data into a Redshift cluster from S3 buckets
- read data from an Excel spreadsheet
- read data from files and write data to files using the Python Pandas library
- recall how Apache Zookeeper enables the HDFS NameNode and YARN ResourceManager to run in high-availability mode
- recall the steps involved in building a MapReduce application and the specific workings of the Map phase in processing each row of data in the input file

- recognize and deal with missing data in R
- recognize the challenges involved in processing big data and the options available to address them such as vertical and horizontal scaling
- retrieve specific parts of an array using row and column indices
- run ETL scripts using Glue
- run the application and examine the outputs generated to get the word frequencies in the input text document
- set up a JDBC connection on Glue to the Redshift cluster
- specify the configurations of the MapReduce applications in the Driver program and the project's pom.xml file
- standardize a distribution to express its values as z-scores and use Pandas to generate a correlation and covariance matrix for your dataset
- transfer files from your local file system to HDFS using the copyFromLocal command
- use fancy indexing with arrays using an index mask
- use NumPy to compute statistics such as the mean and median on your data
- use NumPy to compute the correlation and covariance of two distributions and visualize their relationship with scatterplots
- use the dplyr library to load data frames
- use the get and getmerge functions to retrieve one or multiple files from HDFS
- use the ggplot2 library to visualize data using R
- use the NumPy library to manipulate arrays and the Pandas library to load and analyze a dataset
- using the independent t-test and with a related sample using a paired t-test using the SciPy library
- using the mutate method
- work with the YARN Cluster Manager and HDFS NameNode web applications that come packaged with Hadoop
- write a simple bash script

Python - Using Pandas to Work with Series & DataFrames

Objectives:

- install and work with Pandas
- create and configure Pandas Series objects
- perform data wrangling operations on Series objects
- use appending and sorting operations on Series objects
- create and configure Pandas DataFrame objects
- perform indexing operations on DataFrames
- identify and troubleshoot missing data
- work with aggregations on columns
- perform statistical operations on DataFrames
- recall basic concepts and instantiate Series and DataFrame objects

Python - Using Pandas for Visualizations and Time-Series Data

Objectives:

- load and explore the dataset used for visualization
- plot pie charts, box plots, and scatter plots using Pandas
- identify and work with time-series data
- calculate deltas and percentage returns in stock prices
- define time deltas and date ranges in Pandas
- clean missing data in mismatched DataFrames
- identify string data stored in DataFrames
- perform advanced manipulations on string data
- change column values by applying functions
- transform data with user-defined functions
- transform all columns in a DataFrame
- recall how to plot visuals and transform column values

Python - Pandas Advanced Features

Objectives:

- perform grouping and aggregations on data
- work with multiple, hierarchical indexes
- specify grouping and aggregations with multiple indexes
- perform general user-defined aggregations
- extract subsets of data using filtering
- identify kinds of masking operations
- troubleshoot data with duplicates
- identify how categorical data differs from continuous
- perform filtering operations on categorical data
- recognize default and custom indexes and reindex DataFrames
- perform filtering operations, drop duplicate data, and work with categories

Cleaning Data in R

Objectives:

- recognize types of unclean data
- recognize criteria for ensuring data quality
- fetch a JSON document over HTTP and load it using dplyr
- load multiple sheets from an Excel document
- handle common errors encountered when reading CSV data
- read data from a relational database using a SQL query
- combine two related datasets using a join operation
- reshape tabular data by spreading values from rows to columns
- apply a summary function using dplyr
- use mean imputation to replace missing values
- use a regular expression to extract data into a new column
- practice applying data wrangling functions using R

Technology Landscape & Tools for Data Management

Objectives:

- describe the concept and characteristics of the current technology landscape from the data perspective as well as the tools involved
- describe the comparative benefits of essential data management tools
- recognize the need for machine learning in modern data analytics
- list the various prominent tools and frameworks that can be used to implement machine learning
- work with scikit-learn to implement machine learning
- recognize the capabilities provided by Python and R in the data management cycle
- specify the capabilities and benefits provided by the implementation of machine learning in the cloud
- explore essential data management tools and implement machine learning with scikit-learn

Machine Learning & Deep Learning Tools in the Cloud

Objectives:

- recognize the capabilities of Microsoft machine learning tools
- recognize the machine learning tools provided by AWS for data analysis
- specify Spark's machine leaning capabilities and the features of PySpark
- list frameworks that can be used to implement deep learning such as Keras, TensorFlow, Caffe, and PyTorch
- implement deep learning using Keras
- list tools that can be used to implement data mining and analytics and their features
- demonstrate the capabilities of building and processing data pipeline with Knime
- set up Keras, implement a deep learning algorithm, and build data pipelines using KNIME

Data Wrangling with Trifacta

Objectives:

- remove units from data and covert to another format
- change date formats to the ISO 8601 standard
- create filters based on existing column data
- replace values based on a particular criteria
- count the number of matches for values in data
- split columns based on a pattern
- merge multiple columns into one
- create a new column from extracted data
- apply a group by transformation to aggregate with a conditional value
- apply a number of transforms to reshape data
- join two data sets into one using a join operation
- apply data wrangling functions using Trifacta

MongoDB Querying

Objectives:

- configure and test PyMongo in a Python program
- work with MongoDB document structure
- perform create, read, update, and delete operations on a MongoDB document
- work with MongoDB document ObjectIDs and Timestamps
- use the find operation to select documents from a collection
- specify the fields to be returned from the find operation
- use the comparison query operators to match criteria
- apply the $exists and $type elements to a query
- use the $regex operator to query documents
- use the $size and $all operators to query array fields
- perform a text search query on string content
- use the mongoimport tool to import from JSON and CSV
- use the mongoexport tool to export data from MongoDB to JSON and CSV
- combine a number of different operators to get a result from MongoDB

MongoDB Aggregation

Objectives:

- recognize the structure of aggregate operations in MongoDB
- use the $group operator to perform a computation using an accumulator operator
- use the $match operator to filter data in an aggregation query
- use the $project operator to specify fields in an aggregation query
- use the $limit and $sort operators in an aggregation pipeline
- use the $unwind operator to expand an array field
- use the $lookup operator to perform a join
- use createIndex to build an index on a collection
- use a geospacial index for a geoSearch operation
- implement a multi-stage aggregation pipeline

•

Getting Started with Hive

Objectives:

- define what a data warehouse is and identify its characteristics
- describe the functions served by relational databases and the features they offer
- distinguish between Online Transaction Processing and Online Analytical Processing and identify the specific problems they are meant to solve
- identify where Hive fits in the Hadoop ecosystem and how it simplifies working with Hadoop
- describe the architecture of Hive and the functions served by HiveServer and the Metastore
- identify the services and features offered by AWS, Azure, and GCP to run Hadoop and Hive on their infrastructure
- describe the different primitive and complex data types available in Hive
- compare managed and external tables in Hive and how they relate to the underlying data
- contrast OLTP and OLAP systems, identify major components of Hadoop, explore Hive benefits for data analysis

Loading & Querying Data with Hive

Objectives:

- use the Google Cloud Platform's Dataproc service to provision a Hadoop cluster
- define and create a simple table in Hive using the Beeline client
- load a few rows of data into a table and query it with simple select statements
- run Hive queries from the shell of a host where a Hive client is installed
- define and run a join query involving two related tables
- describe the structure of the Hive Metastore on the Hadoop Distributed File System (HDFS)
- create, load data into, and query an external table in Hive and contrast it with a Hive-managed table
- use the alter table statement to change the definition of a Hive table
- work with temporary tables that are only valid for a single Hive session and recognize how they differ from regular tables
- populate Hive tables with data in files on both HDFS and the file system of the Hive client
- load data into multiple tables from the contents of another table
- use the Hadoop shell to execute Hive query scripts and work with Hive tables

Viewing & Querying Complex Data with Hive

Objectives:

- load and access data in the form of arrays
- work with data in the form of key-value pairs - map data structures in Hive
- define and use structured data in the form of Hive struct types
- transform complex data types to a tabular format to facilitate analysis using the explode and posexplode functions
- combine the results of the explode function with other columns of a table to generate a lateral view
- flatten multi-dimensional data structures by chaining lateral views
- use the UNION and UNION ALL operations on table data and distinguish between the two
- search for values in the results of a subquery using the IN and EXIST clauses
- create and load data into tables efficiently by including these operations in a single query
- define and work with views in Hive to simplify querying and control access to data
- perform queries and utilize views on complex data types available in Hive

Optimizing Query Executions with Hive

Objectives:

- recognize how Hive translates queries to Hadoop MapReduce operations
- identify the different options available in Hive to optimize query execution
- recall how partitioning of a dataset can help queries run efficiently and identify the types of partitioning available in Hive
- specify how bucketing improves query performance and compare it with partitioning a dataset
- identify how to join tables in Hive to ensure the best performance of your query
- work with techniques to improve performance and work with partitioning, bucketing and structured queries

Using Hive to Optimize Query Executions with Partitioning

Objectives:

- use the Google Cloud Platform's Dataproc service to provision a Hadoop cluster. Not required if you already have a Hadoop environment set up with Hive
- define a table which will contain data partitioned based on the value in one of its columns
- insert data into partitions of a Hive table and explore the partition and its data on HDFS
- load data into table partitions from files
- create and populate partitions in an external table
- alter the definition of a partition to modify its contents
- define and work with dynamic partitions on your Hive tables
- configure a table to use more than one column to define partitions and explore the partition on HDFS
- use partitioning to boost query performance in HDFS

Bucketing & Window Functions with Hive

Objectives:

- implement bucketing for a Hive table and explore the structure of the table and bucket on HDFS
- apply both bucketing and partitioning for a table and describe the structure of such a table on HDFS
- extract further performance from Hive queries by sorting the contents of buckets
- work with samples of a Hive table by dividing it into buckets
- perform join operations on three or more tables by chaining the joins
- implement a window function to calculate running totals on an ordered dataset
- apply a window function within a partition of your dataset
- apply bucketing of Hive tables to boost query performance and to use window functions

Filtering Data Using Hadoop MapReduce

Objectives:

- create a new project and code up the Mapper for an application to count the number of passengers in each class of the Titanic in the input dataset
- develop a Reducer and Driver for the application to generate the final passenger counts in each class of the Titanic
- build the project using Maven and run it on the Hadoop master node to check that the output correctly shows the numbers in each passenger class
- apply MapReduce to filter through only the surviving passengers on the Titanic from the input dataset
- execute the application and verify that the filtering has worked correctly; examine the job and the output files using the YARN Cluster Manager and HDFS NameNode web UIs
- use MapReduce to obtain a distinct set of the cuisines offered by the restaurants in a dataset
- build and run the application and confirm the output using HDFS from both the command line and the web application
- identify configuration functions used to customize a MapReduce and recognize the types of input and output when null values are transmitted from the Mapper to the Reducer

Hadoop MapReduce Applications With Combiners

Objectives:

- recognize the need for combiners to optimize the execution of a MapReduce application by minimizing data transfers within a cluster
- recall the steps involved in processing data in a MapReduce application
- describe the working of a Combiner in performing a partial reduction of the data that is output from the Mapper
- configure a Combiner to optimize a MapReduce application that calculates an average value
- use Maven to create a new project for a MapReduce application and plan out the Map and Reduce phases by examining the auto prices dataset
- develop the Mapper and Reducer for the application that will calculate the average price for each make of automobile in the input dataset
- create the driver program for the MapReduce application
- run the MapReduce application and check the output to get the average price for each automobile make
- code up a Combiner for the MapReduce application and configure the Driver to use it for a partial reduction on the Mapper nodes of the cluster
- fix the bug in the previous application by defining a type that represents both the aggregate price and count of automobiles that can be used to correctly calculate the average price
- compare the output of the modified application with the previous buggy version and verify that the average prices for the vehicles are being calculated correctly
- identify the shortcomings of regular MapReduce operations which are addressed by Combiners, and how Combiners differ from Reducers

Advanced Operations Using Hadoop MapReduce

Objectives:

- define a vehicle type that can be used to represent automobiles to be stored in a Java PriorityQueue
- configure a Mapper to use a PriorityQueue to store the five most expensive vehicles it has processed from the dataset
- use a PriorityQueue in the Reducer of the application to receive the five most expensive automobiles from each mapper and write the top 5 vehicles overall to the output
- execute the application and examine the output on HDFS to confirm that the five most expensive automobiles have been written out
- define the Mapper for a MapReduce application to build an inverted index from a set of text files
- configure the Reducer and the Driver for the inverted index application
- run the application and examine the inverted index on HDFS
- recognize the data structures and configurations involved when extracting the top N values from a data set

Data Analysis Using the Spark DataFrame API

Objectives:

- recognize the features that make Spark 2.x versions significantly faster than Spark 1.x
- specify the reasons for using shared variables in your Spark application and distinguish between the two options available for sharing variables
- create a Spark DataFrame from the contents of a CSV file and apply some simple transformations on the DataFrame
- define a transformation to view a random sample of data from a large DataFrame
- apply grouping and aggregation operations on a DataFrame to analyze categories of data in a dataset
- use Matplotlib to visualize the contents of a Spark DataFrame
- perform operations to prepare your dataset for analysis by trimming unnecessary columns and rows containing missing data
- define and apply a generic transformation on a DataFrame
- apply complex transformations on a DataFrame to extract meaningful information from a dataset
- work with broadcast variables and perform a join operation with a DataFrame that has been broadcast
- use a Spark accumulator as a counter
- store the contents of a DataFrame in a text file for archiving or sharing
- define and work with a custom accumulator to count a vector of values
- perform different join operations on Spark DataFrames to combine data from multiple sources
- analyze data using the DataFrame API

Data Analysis using Spark SQL

Objectives:

- recall the different stages involved in optimizing any query or method call on the contents of a Spark DataFrame
- create views out of a Spark DataFrame's contents and run queries against them
- trim and clean a DataFrame before a view is created as a precursor to running SQL queries on it
- perform an analysis of data by running different kinds of SQL queries, including grouping and aggregations
- recognize how Spark DataFrames infer the schema of data loaded into them and configure a DataFrame with an explicitly defined schema
- define what a window is in the context of Spark DataFrames and when they can be used
- create and analyze categories of data in a dataset using Windows
- analyze data using Spark SQL

Data Lake Framework & Design Implementation

Objectives:

- describe the architectural differences between data lakes and data warehouses
- identify the features data lakes provide as a part of the enterprise architecture
- recognize how to use data lakes to democratize data
- identify the design considerations for data lakes
- describe the architecture of AWS data lakes and their essential components
- implement data lakes using AWS
- recognize the prominent architectural styles used when implementing data lakes on-premises and on multiple cloud platforms
- list the various frameworks that can be used to process data from data lakes
- compare data lakes and data warehouses, specify data lake design patterns, and implement data lakes using AWS

Data Lake Architectures & Data Management Principles

Objectives:

- implement Lambda and Kappa architectures to manage real-time big data
- identify the benefits of adopting Zaloni data lake reference architecture
- describe data ingestion approaches and compare Avro and Parquet file format benefits
- demonstrate how to ingest data using Sqoop
- describe the data processing strategies provided by MapReduce V2, Hive, Pig, and Yam for processing data with data lakes
- recognize how to derive value from data lakes and describe the benefits of critical roles
- describe the steps involved in the data life cycle and the significance of archival policies
- implement an archival policy to transition between S3 and Glacier, depending on adopted policies
- ingest data using Sqoop and implement an archival policy to transition from S3 to adopted policies

Data Architecture Deep Dive - Design & Implementation

Objectives:

- describe data complexity management strategies
- recognize data modeling techniques and describe data modeling processes
- list prominent distributed data models and their associative implementation benefits
- describe data partitioning methods and data partitioning implementation criteria
- install MongoDB and implement data partitioning using MongoDB
- identify important components of a hybrid data architecture
- demonstrate how to implement directed acyclic graphs using Elasticsearch
- describe CAP theorems and their implementation approaches
- compare the differences between batch and streaming data
- recognize the read and write optimizations in MongoDB
- implement serverless architecture with Lambda and data partitioning using MongoDB

Data Architecture Deep Dive - Microservices & Serverless Computing

Objectives:

- describe data pattern implementation in microservices
- describe the beneficial features of serverless and Lambda architectures
- demonstrate how to implement Lambda architecture in AWS
- manage resources with the implementation of clusters
- describe data architecture implementations and their advantages
- specify the steps involved in discovering and deriving value from data in existing datasets
- classify the different types of data risks that need to be managed when implementing data modeling and design
- specify the steps involved in building a successful data POC
- recall the beneficial features of Lambda and serverless architecture and specify the essential processes of discovering data

Data Wrangling with Python

Objectives:

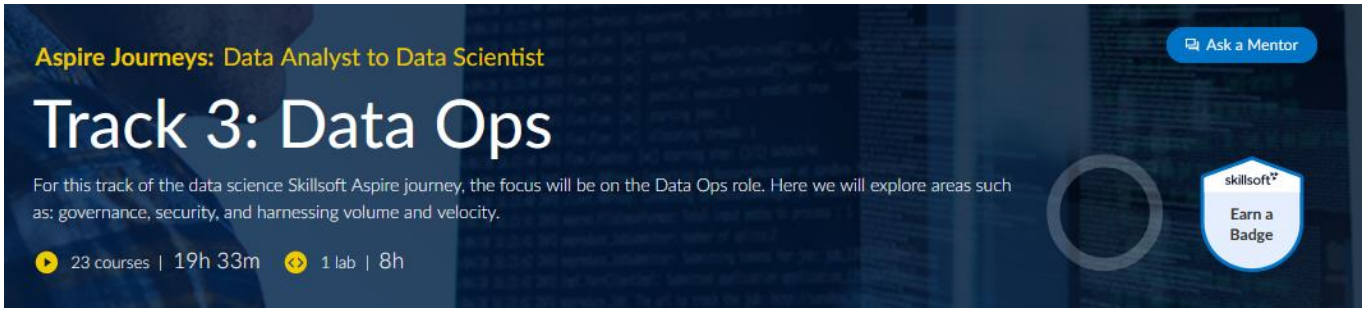
- In this Skillsoft Aspire lab for the Data Wrangler track of the Data Science journey, you will perform data wrangling tasks including using a Pandas DataFrame to convert multiple Excel sheets to separate JSON documents, extract a table from an HTML file, use mean substitution and convert dates within a DataFrame.

Final Exam: Data Wrangler

Objectives:

- apply a group by transformation to aggregate with a conditional value
- apply grouping and aggregation operations on a DataFrame to analyze categories of data in a dataset
- build and run the application and confirm the output using HDFS from both the command line and the web application
- change column values by applying functions
- change date formats to the ISO 8601 standard
- code up a Combiner for the MapReduce application and configure the Driver to use it for a partial reduction on the Mapper nodes of the cluster
- compare managed and external tables in Hive and how they relate to the underlying data
- configure and test PyMongo in a Python program
- configure the Reducer and the Driver for the inverted index application
- create and analyze categories of data in a dataset using Windows
- Create and configure Pandas dataFrame objects
- Create and configure pandas series object
- create and instantiate a directed acyclic graph in Airflow
- create a Spark DataFrame from the contents of a CSV file and apply some simple transformations on the DataFrame
- create the driver program for the MapReduce application
- define and run a join query involving two related tables
- define a vehicle type that can be used to represent automobiles to be stored in a Java PriorityQueue
- define the Mapper for a MapReduce application to build an inverted index from a set of text files
- define what a window is in the context of Spark DataFrames and when they can be used
- demonstrate how to ingest data using Sqoop
- describe data ingestion approaches and compare Avro and Parquet file format benefits
- describe the beneficial features that we can achieve using serverless and lambda architectures
- describe the data processing strategies provided by MapReduce V2, Hive, Pig, and Yam for processing data with data lakes
- describe the different primitive and complex data types available in Hive
- extract subsets of data using filtering
- flatten multi-dimensional data structures by chaining lateral views
- handle common errors encountered when reading CSV data
- identify and troubleshoot missing data
- identify and work with time-series data
- identify kinds of masking operations
- implement a multi-stage aggregation pipeline
- implement data lakes using AWS
- implement deep learning using Keras
- install MongoDB and implement data partitioning using MongoDB
- list the prominent distributed data models along with their associative implementation benefits
- list the various frameworks that can be used to process data from data lakes
- load a few rows of data into a table and query it with simple select statements
- load multiple sheets from an Excel document
- perform create, read, update, and delete operations on a MongoDB document
- perform statistical operations on DataFrames
- plot pie charts, box plots, and scatter plots using Pandas
- recall the prominent data pattern implementation in microservices
- recognize the capabilities of Microsoft machine learning tools
- recognize the machine learning tools provided by AWS for data analysis
- recognize the read and write optimizations in MongoDB

- setup and install Apache Airflow
- split columns based on a pattern
- test Airflow tasks using the airflow command line utility
- trim and clean a DataFrame before a view is created as a precursor to running SQL queries on it
- use a regular expression to extract data into a new column
- use a Spark accumulator as a counter
- use createIndex to build an index on a collection
- use Maven to create a new project for a MapReduce application and plan out the Map and Reduce phases by examining the auto prices dataset
- use the alter table statement to change the definition of a Hive table
- use the find operation to select documents from a collection
- use the mongoexport tool to export data from MongoDB to JSON and CSV
- use the mongoimport tool to import from JSON and CSV
- use the UNION and UNION ALL operations on table data and distinguish between the two
- work with data in the form of key-value pairs - map data structures in Hive
- work with scikit-learn to implement machine learning



Data Science Tools

Objectives:

- describe what a data science platform is
- describe the challenges of deploying data science tools
- identify some considerations for data science tools
- identify and describe each step of a data science workflow
- describe different uses for data science analytic tools
- describe different uses for data science visualization tools
- describe different uses for data science database tools
- list the benefits of deploying cloud-based tools
- list the challenges of deploying cloud-based tools
- describe what DevOps is and some of the common functionalities
- describe DevOps for data science
- identify different uses of data science tools

Delivering Dashboards: Management Patterns

Objectives:

- recognize the various types of visualizations that can be used to build concise dashboards
- specify the different types of dashboards and with their associated features and benefits
- describe the different types of data that are used in analysis and types of visualizations that can be created from the data
- identify the essential components that are involved in building a productive dashboard
- recall the best practices for building a productive dashboard
- create dashboards using ELK
- create dashboards using PowerBI
- specify criteria to consider selecting appropriate charts
- recognize the critical benefits provided by leaderboards and scorecards
- list the prominent types of scorecards
- create dashboards using PowerBI and ELK

Delivering Dashboards: Exploration & Analytics

Objectives:

- identify the data exploration capabilities provided by charts
- list prominent tools that can be used to implement charts
- demonstrate how to create bar and line charts using Kibana
- create dashboards using Kibana
- share dashboards using Kibana
- create charts and dashboards using Tableau
- create charts and dashboards using Qlikview
- build dashboards with real-time data updates
- describe and use dashboard design patterns
- create monitoring dashboards using ELK
- create dashboards using Kibana, Tableau, and Qlikview

Cloud Data Architecture: Cloud Architecture & Containerization

Objectives:

- recognize the impact of implementing containerization on cloud hosting environments
- recall the benefits of container implementation and the role of cloud container services
- describe the concept of serverless computing and its benefits
- describe approaches of implementing DevOps in the cloud
- implement OpsWorks on AWS using Puppet
- classify storage from the perspective of capacity and data access technologies
- specify the benefits of implementing machine learning, deep learning, and artificial intelligence in the cloud
- recognize the impact of cloud technology on BI analytics
- recall container and cloud storage types, container and serverless computing benefits, and advantages of implementing cloud-based BI analytics

Data Compliance Issues & Strategies

Objectives:

- describe the issues surrounding data compliance
- describe the common compliance standards that companies should be familiar with including GDPR, HIPPA, and PCI DSS
- describe the importance of knowing global standards
- identify the risks associated with not knowing relevant company standards
- identify the myths and facts of data compliance
- identify how corporate end-users can be educated about data compliance
- identify how management can be educated about data compliance
- identify the benefits of rolling out a successful data compliance program
- identify the elements of a successful data compliance strategy
- describe how to build a compliance strategy
- describe procedures for internal and external reporting and other responses to data breaches
- list regulations covering data protection, explain big data's popularity, list myths about data compliance, and explain the benefits of a data compliance program

Implementing Governance Strategies

Objectives:

- describe the concept of governance and how it applies to big data
- describe why we need data governance
- discuss the five main requirements for data governance
- define the differences between big data and traditional data paradigms
- identify the types of data that need to be governed
- identify the stakeholders that need to be part of a data governance program
- recognize the impact of how cloud technologies affect data governance
- specify how to design a data governance process
- describe how to manage a data governance strategy
- describe how to monitor a data governance strategy
- describe how to maintain a data governance strategy
- define the importance of big data, why data requires governance, the benefits of the cloud, and non-IT team requirements

Data Access & Governance Policies: Data Access Governance & IAM

Objectives:

- discuss how data access governance identifies and protects digital assets through policies
- list examples of standard security accreditations related to the protection of sensitive data
- provide examples of security controls related to data accessibility
- discuss how DLP, user awareness and training, applying updates, encryption, and malware scanning can minimize data breaches
- map HR job roles to IT system and data permissions
- set Windows NTFS file system permissions in accordance with the principle of least privilege
- identify the role IAM plays in a data governance framework
- use the AWS console to create IAM users and groups
- use the AWS console to assign permissions policies to IAM groups
- mitigate data breach events by identifying weaknesses
- fulfill organizational and regulatory data security requirements
- implement effective security controls to protect data

Data Access & Governance Policies: Data Classification, Encryption, & Monitoring

Objectives:

- recognize the importance of data classification
- use Microsoft File Server Resource Manager to set file classification values
- recall methods of encrypting sensitive data
- enable Microsoft BitLocker to protect data at rest
- configure and test Microsoft VPN to protect data in motion
- use Microsoft System Center Configuration Manager to view managed device security compliance
- identify the relevance of tracking data access trends
- identify how data access can be monitored through SIEM and reports
- recognize how logging and auditing feed into data analytics
- enable filtered logs in the Windows Event Viewer
- configure file system object auditing using Group Policy
- use encryption to protect data and monitor data access

Streaming Data Architectures: An Introduction to Streaming Data in Spark

Objectives:

- recognize the differences between batch and streaming data and the types of streaming data sources
- list the steps in involved in processing streaming data, the transformation of streams, and the materialization of the results of the transformation
- describe how the use of a message transport decouples a streaming application from the sources of streaming data
- describe the techniques used in Spark 1.x to work with streaming data and how it contrasts with processing batch data
- recall how structured streaming in Spark 2.x is able to ease the task of stream processing for the app developer
- compare how streaming processing works in both Spark 1.x and 2.x
- recognize how triggers can be set up to periodically process streaming data and describe the various output modes available to publish the results of stream processing
- recognize the key aspects of working with structured streaming in Spark

Streaming Data Architectures: Processing Streaming Data with Spark

Objectives:

- install the latest available version of PySpark
- configure a streaming data source using Netcat and write an application to process the stream
- describe the effects of using the Update mode for the output of your stream processing application
- write an application to listen for new files being added to a directory and process them as soon as they come in
- compare the Append output to the Update mode and distinguish between the two
- develop applications that limit the files processed in each trigger and use Spark's Complete mode for the output
- perform aggregation operations on streaming data using the DataFrame API
- work with Spark SQL in order to process streaming data using SQL queries
- define and apply standard, re-usable transformations for streaming data
- recall they key ways to use Spark for streaming data and explore the ways to process streams and generate output

Scalable Data Architectures: Getting Started

Objectives:

- recognize the need to scale architectures to keep up with the needs for storage and processing of big data
- identify the characteristics of data warehouses that make them ideally suited to the task of big data analysis and processing
- distinguish between relational databases and data warehouses
- recognize the specific characteristics of systems meant for online transaction processing and online analytical processing and how data warehouses are an example of OLAP systems
- identify the various components of data warehouses that enable them to work with varied sources, extract and transform big data, and generate reports of analysis operations efficiently
- recall the features of Amazon Redshift that enable big data to be processed at scale
- list the features of data warehouses and contrast them with those of relational databases, and contrast the two options available to scale compute capacity

Scalable Data Architectures: Using Amazon Redshift

Objectives:

- use the Amazon Redshift Quick Launch feature to provision a data warehouse on Amazon Web Services
- define additional configuration options when provisioning a Redshift cluster by using the default cluster
- recognize the various tool configuration options available for a Redshift cluster and use the metrics available to optimize a cluster configuration
- create an IAM role on AWS that includes the necessary permissions to interact with the Redshift and S3 services
- provision an IAM user that can be used to connect to and interact with AWS using the CLI
- install the AWS command line interface and use it to create and delete Redshift clusters
- use the Redshift Query Editor to create tables, load data, and run queries
- recall the features of Amazon Redshift and the commands and configurations needed to work with Redshift using the CLI

Scalable Data Architectures: Using Amazon Redshift & QuickSight

Objectives:

- use the AWS console to load datasets to Amazon S3 and then load that data into a table provisioned on a Redshift cluster
- run queries on data in a Redshift cluster and use the query evaluation feature to analyze the query execution metrics
- work with the SQL Workbench client to connect to and query data in a Redshift cluster
- disable automated snapshots for a Redshift cluster and configure a table to be excluded from snapshots
- recover an individual table from the snapshot of an entire cluster
- add more nodes to a Redshift cluster
- scale up each individual node of a Redshift cluster and scale down the number of nodes
- create a security group rule to enable access from Amazon's QuickSight servers to a Redshift cluster
- configure Amazon QuickSight to load data from a table in a Redshift cluster for analysis
- use the QuickSight dashboard to generate a time series plot to visualize sales at a retailer over time
- configure snapshots of Redshift clusters and recall the steps involved in analyzing data in Redshift using QuickSight

Building Data Pipelines

Objectives:

- describe data pipelines and automation
- build a traditional ETL pipeline with batch processing
- build a ETL pipeline with stream processing
- setup and install Apache Airflow
- describe the key concepts of Apache Airflow
- create and instantiate a directed acyclic graph in Airflow
- use tasks and include arguments in Airflow
- use dependencies in Airflow
- build an ETL pipeline with Airflow
- build an automated pipeline without using ETL
- test Airflow tasks using the airflow command line utility
- use Apache Airflow to create a data pipeline

Data Pipeline: Process Implementation Using Tableau & AWS

Objectives:

- describe data pipeline and its features and list the steps involved in building one
- recognize the processes involved in building data pipelines
- identify the different stages of a data pipeline
- list various technologies that can be used to implement a data pipeline
- list various data sources that are involved in the data pipeline transformation phases
- define scheduled data pipelines and list all the associated components, tasks, and attempts
- install the Tableau server and command line utilities
- build data pipelines using the Tableau command line utilities
- demonstrate the steps involved in building data pipelines on AWS
- install Tableau command line utilities, build a pipeline with Tableau command line utilities, and build data pipelines on AWS

Data Pipeline: Using Frameworks for Advanced Data Management

Objectives:

- recognize the features of Celery and Luigi that can be used to set up data pipelines
- implement Python Luigi in order to set up data pipelines
- list Dask task scheduling and big data collection features
- implement Dask arrays in order to manage NumPy APIs
- list frameworks that can be used to implement data exploration and visualization in data pipelines
- integrate Spark and Tableau to manage data pipelines
- use Python to build visualizations for streaming data
- recognize the data pipeline building capabilities provided by Kafka, Spark, and PySpark
- set up Luigi to implement data pipelines, integrate Spark and Tableau for data pipeline management, and build visualizations for data pipelines using Python

Data Sources: Integration from the Edge

Objectives:

- recognize required elements for deploying IoT solutions
- describe the prominent service categories of IoT solutions
- recognize the capabilities provided by IoT solutions and the maturity models of IoT solutions
- list the critical design principles that need to be implemented when building IoT solutions
- describe the cloud architectures of IoT from the perspective of Microsoft Azure, AWS, and GCP
- compare the features and capabilities provided by the MQTT and XXMP protocols for IoT solutions
- identify key features and applications that can be implemented using IoT controllers
- recognize the concept of IoT data management and the applied lifecycle of IoT data
- list the essential security techniques that can be implemented to secure IoT solutions
- generate weather data streams and connect web applications to AWS IoT

Data Sources: Implementing Edge Data on the Cloud

Objectives:

- identify the approaches and the steps involved in setting up AWS IoT Greengrass
- recognize the essential components of GCP IoT Edge
- connect a web application to AWS IoT using MQTT over WebSockets
- demonstrate the essential approaches of using IoT Device Simulator
- generate streams of weather data using the MQTT messaging protocol
- create a device type, a user, and a device using IoT Device Simulator

Securing Big Data Streams

Objectives:

- understand the main security concerns related to big data
- understand key security concerns related to streaming data
- understand key security concerns related to NoSQL databases
- understand key security risks associated with distributed processing frameworks
- understand key concerns and flaws related to data mining and analytics
- understand risks related to end-point devices such as devices on the Internet of Things
- understand some of the key ways that big data security concerns are addressed
- understand how data streams are secured
- understand how to deploy a VPN using Azure to secure data in motion
- understand how end-point devices are secured using validation and filtering
- understand how to use encryption to secure data at rest
- recognize how big data and streaming data are secured

Harnessing Data Volume & Velocity: Turning Big Data into Smart Data

Objectives:

- recognize the differences between big data and smart data from the perspectives of volume, variety, velocity, and veracity
- specify the smart data capabilities for machine learning and artificial intelligence
- recognize how to turn big data to smart data and how to use data volumes
- list the applications of smart data and smart process
- recall use cases for smart data application
- recognize the life cycle of smart data and the associated impacts and benefits
- identify the steps involved in transforming big data to smart data using k-NN
- describe the various smart data solution implementation frameworks
- recall how to turn smart data to business using data sharing and algorithms
- recognize how to implement clustering on smart data
- integrate smart data and describe its impact on the optimization of data strategy
- list the frameworks for smart data and specify the algorithms for smart data transition

Data Rollbacks: Transaction Rollbacks & Their Impact

Objectives:

- describe the concept and characteristics of rollback process and its impact on transactions
- recognize the various states of transactions
- list the prominent types of transactions along with their essential features (distributed and compensating transactions)
- implement SQL transaction management with commit, savepoint, and release savepoint
- recall the various transaction log operations and their characteristics (transaction recovery and transaction replication)
- recognize the deadlock management capabilities and features provided by SQL Server using Lock Monitor and Trace
- list SQL Server rollback mechanisms
- use SQL Server to rollback databases to a specific point in time
- implement transaction management and rollbacks using SQL Server

Data Rollbacks: Transaction Management & Rollbacks in NoSQL

Objectives:

- compare the transaction management architecture and capabilities of NoSQL and SQL
- recognize the transaction management capabilities of MongoDB and the impacts on consistency and availability
- implement multi-document transaction management using Replica set in MongoDB
- list essential SQL Server change data capture features
- recognize the features of change streams in MongoDB
- demonstrate how to create change streams to enable real-time data change streaming for applications using MongoDB
- compare the transaction management architecture and capabilities of NoSQL and SQL

Final Exam: Data Ops

Objectives:

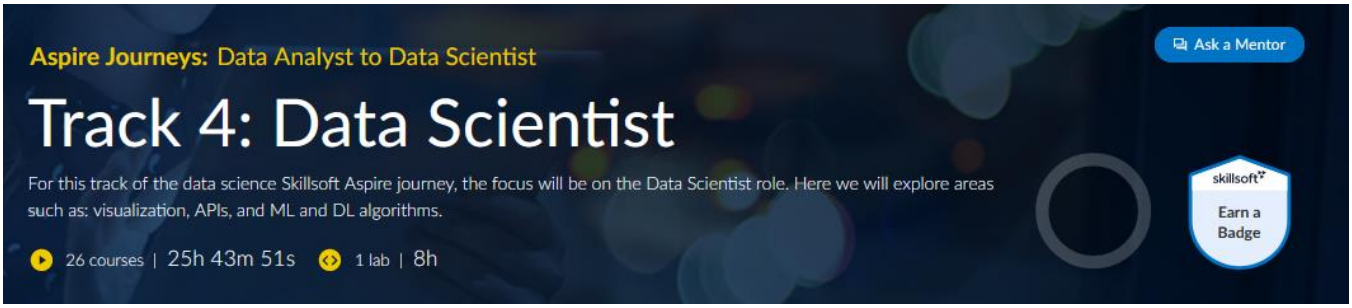
- configure a streaming data source using Netcat and write an application to process the stream
- configure file system object auditing using Group Policy
- connect a web application to AWS IoT using MQTT over WebSockets
- contextual data and collective anomaly detection using scikit-learn
- create an IAM role on AWS that includes the necessary permissions to interact with the Redshift and S3 services
- create charts and dashboards using Qlikview
- create dashboards using ELK
- create tables, load data, and run queries
- demonstrate detecting anomalies using boxplot and scatter plot
- demonstrate how to detect anomalies using R, RCP, and the devtools package
- demonstrate the essential approaches of using IoT Device Simulator
- demonstrate the mathematical approaches of detecting anomalies
- describe different uses for data science visualization tools
- describe how the use of a message transport decouples a streaming application from the sources of streaming data
- describe the cloud architectures of IoT from the perspective of Microsoft Azure, AWS, and GCP
- describe the common compliance standards that a data scientist needs to be familiar with including GDPR, HIPPA, PCI DSS, SOC 2
- describe the different types of data that are used in analysis and types of visualizations that can be created from the data
- describe the various smart data solution implementation frameworks
- describe what DevOps is and some of the common functionalities
- describe why we need data governance
- different uses for data science analytic tools
- discuss the five main requirements for data governance
- enable Microsoft BitLocker to protect data at rest
- generate streams of weather data using the MQTT messaging protocol
- identify how data access can be monitored through SIEM and reports
- identify the approaches and the steps involved in setting up AWS IoT Greengrass
- identify the benefits of rolling out a successful data compliance program
- identify the common compliance standards that a data scientist needs to be familiar with including GDPR, HIPPA, PCI DSS, SOC 3
- identify the essential components that are involved in building a productive dashboard
- identify the role IAM plays in a data governance framework
- identify the steps involved in transforming big data to smart data using k-NN
- identify the types of data that need to be governed
- implement effective security controls to protect data
- implement multi-document transaction management using Replica set in MongoDB
- install the AWS command line interface and use it to create and delete Redshift clusters
- list essential SQL Server change data capture features
- list SQL Server rollback mechanisms
- list the steps in involved in processing streaming data, the transformation of streams, and the materialization of the results of the transformation
- mitigate data breach events by identifying weaknesses
- prominent anomaly detection techniques
- recall methods of encrypting sensitive data
- recognize how to implement clustering on smart data
- recognize how to turn big data to smart data and how to use data volumes
- recognize the critical benefits provided by leaderboards and scorecards
- recognize the differences between batch and streaming data and the types of streaming data sources

- recognize the features of change streams in MongoDB
- recognize the key aspects of working with structured streaming in Spark
- run queries on data in a Redshift cluster and use the query evaluation feature to analyze the query execution metrics
- specify how to design a data governance process
- specify the different types of dashboards and with their associated features and benefits
- understand how data streams are secured
- understand how to deploy a VPN using Azure to secure data in motion
- understand key security concerns related to NoSQL databases
- understand key security risks associated with distributed processing frameworks
- use Microsoft System Center Configuration Manager to view managed device security compliance
- use SQL Server to rollback databases to a specific point in time
- use the AWS console to load datasets to Amazon S3 and then load that data into a table provisioned on a Redshift cluster
- use the QuickSight dashboard to generate a time series plot to visualize sales at a retailer over time
- use the Redshift Query Editor to create tables, load data, and run queries
- work with Spark SQL in order to process streaming data using SQL queries

Implementing Data Ops with Python

Objectives:

- Perform data ops tasks with Python including working with row subsets, creating new columns with Regex, performing joins
  and spreading rows.



The Four Vs of Data

Objectives:

- describe the principle of the four Vs of big data analytics
- specify volume in big data analytics and its role in the principle of the four Vs
- specify variety in big data analytics and its role in the principle of the four Vs
- specify velocity in big data analytics and its role in the principle of the four Vs
- specify veracity in big data analytics and its role in the principle of the four Vs
- discuss the way the four Vs of big data relate to each other
- define variety and data structure and how they relate to the four Vs of big data
- define validity and volatility and how they relate to the four Vs of big data
- discuss how the four Vs should be balanced in order to implement a successful big data strategy
- describe various use cases of big data analytics and the four Vs of big data
- specify how the four Vs can be leveraged to extract value from big data
- describe the four Vs of big data analytics, their differences, and how balance can be achieved

Data Driven Organizations

Objectives:

- describe what it means to be data driven and the importance of it for an organization
- recognize how to enable data-driven decision making
- identify the different levels of analytic maturity
- identify the different types of roles required in data driven organizations
- prioritize resources appropriately
- describe the aspects of data quality
- use PowerBI Desktop to visualize and manipulate a dataset
- describe the importance of dealing with missing data and use Azure Machine Learning Studio to clean it up
- describe the importance of identifying and dealing with duplicates using Azure Data Explorer
- describe what truncated data is and how to remove it using Azure Automation
- describe data provenance
- use Informatica Data Quality

Raw Data to Insights: Data Ingestion & Statistical Analysis

Objectives:

- describe how we can use statistical analysis to add value to data
- recognize the concept of data correction along with the various essential approaches of implementing data correction which includes data detection localization, imputation and correction
- demonstrate how we can facilitate outlier detection using R
- describe the layered architecture of data from the perspective of data ingestion, processing, and visualization
- list and compare the various essential data ingestion tools that we can use to ingest data
- set up Kafka and Apache NiFi to ingest data
- demonstrate the steps involved in ingesting data from databases to Hadoop clusters using Sqoop
- demonstrate how we can ingest data using WaveFront
- detect outliers using R and ingest data using Apache NiFi and WaveFront

Raw Data to Insights: Data Management & Decision Making

Objectives:

- describe the capabilities and advantages provided with the application of data-driven decision making
- load data from databases using R
- demonstrate how to prepare data for analysis
- recall the concept of data correction using the essential approaches of simple transformation rules and deductive correction
- implement data correction using simple transformation rules
- implement data correction using deductive correction
- describe the various essential distributed data management frameworks used to handle big data
- describe the approach of implementing data analytics using Machine Learning
- recognize how to implement exploratory data analysis using R
- recognize how to implement predictive modelling using Machine Learning
- correct data using deductive correction, analyze data in R and facilitate predictive modelling with Machine Learning

Tableau Desktop: Real Time Dashboards

Objectives:

- describe real time dashboards and the differences between real time and streaming data
- identify the different cloud data sources that are available
- build a dashboard
- update your dashboard in real time
- organize your dashboard by adding objects and adjusting the layout
- customize and format different aspects of your dashboard
- add interactivity to a dashboard using actions like filtering
- create a Dashboard Starter to work with cloud data sources
- add extensions to your dashboard such as the Tableau Extensions API
- put dashboards into a story point
- share your dashboard with others
- create Dashboard Starter

Storytelling with Data: Introduction

Objectives:

- identify the process and approaches involved in storytelling with data
- recall the essential approaches of interpreting contexts for storytelling
- recognize the prominent types of analysis that we can use to facilitate data storytelling
- specify the concept of who, what, and how from the context of storytelling
- recognize the relevance of utilizing visualizations in order to facilitate storytelling with data
- list the essential graphical tools that we can use to facilitate data elaboration
- elaborate scenarios to discover who, what, and how
- define the concept of storyboarding along with the prominent storyboarding templates that we can use to implement storyboarding
- recall elements of the storytelling context, specify the analysis types used to facilitate storytelling with data, recognize visualizations used to facilitate storytelling with data, and list graphical tools used for data exploration

Storytelling with Data: Tableau & Power BI

Objectives:

- list the important approaches and criteria involved in selecting effective visuals for data storytelling
- define the concept of slopegraphs and list the essential capabilities and relevance of slopegraphs
- demonstrate how to implement different types of bar charts using PowerBI
- define the concept of clutters and how to identify and eliminate clutters
- describe the Gestalt principles of visual perception
- recall the prominent best practices of story design
- list the prominent tools that we can use to facilitate storytelling with data
- recall the various essential decluttering steps and approaches that we can implement to eliminate clutters
- demonstrate how to craft visual data using Tableau
- recognize the concerns associated with visual designs from the perspective of storytelling
- illustrate the essential building blocks of PowerBI that we can use to facilitate storytelling with data
- create a model visual using Tableau
- load data from a CSV file using PowerBI, create a bar chart using data, and create a pie chart to compare parts of a whole data set

Python for Data Science: Basic Data Visualization Using Seaborn

Objectives:

- describe what Seaborn is and how it relates to other data science libraries in Python
- install Seaborn and load a dataset for analysis
- define and plot the distribution of a single variable using a histogram and kernel density estimate curve
- configure an univariate distribution's appearance, including color, size, and the components of the plot
- analyze the relationship between two variables by plotting a bivariate distribution
- distinguish between scatter plots, hexbin plots, and KDE plots
- use the Seaborn pair plot to generate a grid to plot the relationship between multiple pairs of variables in your dataset
- perform a regression analysis on a pair of variables in your dataset by using the Seaborn lmplot
- describe the basic aesthetic themes and styles available in Seaborn
- recall some of the use cases and features of Seaborn

Python for Data Science: Advanced Data Visualization Using Seaborn

Objectives:

- work with Seaborn to glean patterns in a dataset by visualizing the relationships between several pairs of variables
- define the aesthetic parameters for a plot and make use of Seaborn's built-in templates for creating shareable graphs
- recognize what a normal distribution is and what is defined as an outlier
- use boxplots and violin plots to visualize the distributions of data within specific categories of your dataset
- compare the use cases for swarm plots, bar plots strip plots, and categorical plots
- create a FacetGrid to visualize distributions within a range of categories
- configure a FacetGrid to convey more information and to draw one's focus to specific plots
- describe what a color palette is and select from the built-in color palettes available
- identify the kinds of color palettes to use depending on the type of data it will represent
- recall different ways to visualize data within categories and identify use cases for specific aesthetic parameters

Data Science Statistics: Using Python to Compute & Visualize Statistics

Objectives:

- create and configure simple graphs with lines and markers using the Matplotlib data visualization library
- use the NumPy library to manipulate arrays and the Pandas library to load and analyze a dataset
- generate histograms and pie charts to analyze distributions and create scatter plots to plot the relationship between two variables in a dataset
- apply Python native functions such as max() and sum() to summarize distributions and visualize these values using Matplotlib
- use NumPy to compute statistics such as the mean and median on your data
- calculate statistics such as the mode and standard error of mean using the SciPy library and compute more statistics such as variance and values at various percentiles using NumPy
- use NumPy to compute the correlation and covariance of two distributions and visualize their relationship with scatterplots
- standardize a distribution to express its values as z-scores and use Pandas to generate a correlation and covariance matrix for your dataset
- create and configure a graph using Matplotlib, enumerate the details conveyed in a Boxplot, compute statistical values using the NumPy function, and compute the correlations between all pairs of columns in a Pandas dataframe

Advanced Visualizations & Dashboards: Visualization Using Python

Objectives:

- recognize the importance and relevance of data visualization from the business perspective
- list libraries that can be used in Python to implement data visualization
- set up a data visualization environment using Python tools and libraries
- list the prominent data visualization libraries that we can be used with Matplotlib
- create bar charts using ggplot in Python
- create charts using the bokeh and Pygal libraries in Python
- recognize criteria that should be considered when selecting an appropriate data visualization library
- create interactive graphs and image files
- plot graphs using line and markers
- plot multiple lines in a single graph using different line styles and markers
- create a line chart with Pygal, create an HTML directive to render the line chart, and render the line chart

R for Data Science: Data Visualization

Objectives:

- create a scatter plot
- create a line graph
- create a bar chart
- create a box and whisker plot
- create a histogram
- create a bubble plot
- use an appropriate plot to visualize data

Advanced Visualizations & Dashboards: Visualization Using R

Objectives:

- list the different types of charts that can be implemented and their relevance in data visualization
- demonstrate how to create a stacked bar plot
- create Matplotlib animations
- use NumPy and Plotly to create interactive 3D plots in Jupyter Notebook
- list graphical capabilities of R from the perspective of data visualization
- build heat maps and scatter plots using R
- implement correlogram and build area charts using R
- recognize ggplot2 capabilities from the perspective of data visualization
- build and customize graphs using ggplot2 in R
- create heat maps using R, create scatter plots using R, and create area charts using R

Data Recommendation Engines

Objectives:

- describe what a Recommendation Engine does, how it can be used, and the types and reasons they are used
- compare the different types of Recommendation Engines and how they can be used to solve different recommendation problems
- describe the process of collecting data and why data sets that can be used for learning, training, and evaluating a Recommendation Engine are needed
- use R to import, filter, and massage data into data sets
- describe how Similarity and Neighborhoods can be used to score users and items against another user or a new item
- create an R function that will score a user against another user to compare their similarity
- create an R function that will give a score to an item a user has not seen before based on other users' ratings and similarity scores
- create an R function that finds similar users and finds products they liked which would be good to recommend to the user
- use R to create an Item to Item similarity, or content, score to Recommend similar items
- evaluate a Recommendation Engine by using known data and metrics to calculate the accuracy of recommendations
- validate and score a Recommendation System using R and an evaluation data set
- describe the types and interfaces required to build a Recommendation System

Data Insights, Anomalies, & Verification: Handling Anomalies

Objectives:

- list sources of data anomaly and compare the differences between data verification and validation
- describe approaches of facilitating decomposition and forecasting, and list the steps and formulas used to achieve the desired outcome
- recall data examination approaches, and use randomization tests, null hypothesis, and Monte Carlo
- identify anomaly detection scenarios and categories of anomaly detection techniques
- recognize prominent anomaly detection techniques
- demonstrate how to facilitate contextual data and collective anomaly detection using scikit-learn
- list prominent anomaly detection tools and their key components
- recognize essential rules of anomaly detection
- implement anomaly detection using scikit-learn, R, and boxplot

Data Insights, Anomalies, & Verification: Machine Learning & Visualization Tools

Objectives:

- describe the supervised and unsupervised approaches of anomaly detection
- compare the prominent anomaly detection algorithms
- demonstrate how to detect anomalies using R, RCP, and the devtools package
- identify components of general online anomaly detection systems
- describe the approaches of using time series and windowing to detect anomalies
- recognize the real-world use cases of anomaly detection as well as the steps and approaches adopted to handle the entire process
- demonstrate detecting anomalies using boxplot and scatter plot
- demonstrate the mathematical approaches of detecting anomalies
- implement anomaly detection using a K-means machine learning approach
- implement anomaly detection with visualization, cluster, and mathematical approaches

Data Science Statistics: Applied Inferential Statistics

Objectives:

- test a hypothesis about a sample by comparing it to the general population using the one-sample t-test available in the SciPy library
- compare a sample with another independent sample using the independent t-test and with a related sample using a paired t-test using the SciPy library
- apply independent t-tests on a real dataset to test a hypothesis that managers at a firm have higher salaries than non-managerial employees
- work with Pandas and Matplotlib to analyze the stock price of Volkswagen in 2008, which were affected by some extreme events
- compute the skewness and kurtosis of the returns on Volkswagen stock in 2008 and recognize how it was a few days of extreme behavior which increased those numbers
- perform pre-processing operations on a dataset containing close prices for stocks and indices to analyze it using linear regression
- use the scikit-learn library to fit a linear regression model on the returns on a stock and the returns on the S&P 500 index
- use two explanatory variables - the returns on the S&P 500 index and on an index tracking the strength of the US Dollar - to perform a regression on the returns on individual stocks
- recall different types of T-tests and identify the values they return, calculate percentage returns from time series data using Pandas, and measure the skew and kurtosis values for a series

Data Research Techniques

Objectives:

- recall the fundamental concepts of data research that can be applied on data inference
- identify implementation steps for drawing data hypothesis conclusions
- define the values, variables, and observations that are associated with data from the perspective of quantitative and classification variables
- specify the different scales of standard measurements with a critical comparison between the Generic and JMP model
- identify the key features of non-experimental and experimental research approaches using real-time scenarios
- compare the differences between the descriptive and inferential statistical analysis
- illustrate the prominent usages of the different types of inferential tests
- describe the approaches and the steps involved in the implementation of clinical data research using real-time scenarios
- describe the approaches and the steps involved in the implementation of sales data research using real-time scenarios
- specify the key features of experimental and non-experimental research and recall the differences between descriptive and inferential statistical analysis

Data Research Exploration Techniques

Objectives:

- specify the essential features and benefits provided by implementing exploratory data analysis
- recognize the prominent approaches that can be adopted to implement data exploration
- install and prepare R for data exploration
- demonstrate how to implement data exploration using R
- implement data exploration using plots in R
- specify the essential packages provided by Python that can be used to explore data
- implement data exploration using Python's data exploration packages
- describe the approach of implementing data research using linear algebra
- work with vectors and metrics using Python and R
- explore data using R, explore data with Python packages, and work with vectors using Python

Data Research Statistical Approaches

Objectives:

- describe the features provided by statistical methods and approaches in data research
- identify the relevance of discrete vs continuous distribution in simplifying data research
- recognize the features of PDF and CDF from the perspective of data research
- implement binomial distribution using R
- specify the types of interval estimation that can be used to enhance data research
- implement point and interval estimation using R
- describe the relevance of data visualization techniques in projecting the outcome of data research
- plot visualizations using R to depict the outcome of data research graphically
- recall the data integration techniques that facilitate using statistical methods
- create Histograms, Scatter plots, and Box plots using Python libraries
- implement missing values and outliers using Python
- implement data research using various statistical approaches

Machine & Deep Learning Algorithms: Introduction

Objectives:

- recognize the different kinds of machine learning algorithms such as regression, classification, and clustering, as well as their specific applications
- describe the process involved in learning a relationship between input and output during the training phase of machine learning
- identify the benefits of combining Pandas, scikit-learn, and XGBoost into a single library to ease the task of building and evaluating ML models
- describe what Support Vector Machines are and how they are used to find a hyperplane to divide data points into categories
- recognize the problems associated with a model that is overfitted to training data and how to mitigate the issue
- define what unsupervised learning is, list the features of SVMs, and describe the issues one may run into when using an overfitted model for predictions

Machine & Deep Learning Algorithms: Regression & Clustering

Objectives:

- recognize the application of a confusion matrix and how it can be used to measure the accuracy, precision, and recall of a classification model
- describe how regression works by finding the best fit straight line to model the relationships in your data
- list the characteristics of regression such as simplicity and versatility, which have led to the widespread adoption of this technique in a number of different fields
- distinguish between supervised learning techniques such as regression and classification, and unsupervised learning methods such as clustering
- describe how clustering algorithms are able to find data points containing common attributes and thus create logical groupings of data
- recognize the need to reduce large datasets with many features into a handful of principal components using the PCA technique
- to recall concepts such as precision and recall and the use cases for unsupervised learning

Machine & Deep Learning Algorithms: Data Preparation in Pandas ML

Objectives:

- load data from a CSV file into a Pandas dataframe and prepare the data for training a classification model
- use the scikit-learn library to build and train a LinearSVC classification model and then evaluate its performance using the available model evaluation functions
- install Pandas ML and then define and configure a ModelFrame
- compare training and evaluation in Pandas ML with the equivalent tasks in scikit-learn
- use Pandas for feature extraction and one-hot encoding, load its contents into a ModelFrame, and initialize and train a linear regression model
- evaluate a regression model using metrics such as r-square and mean squared error and visualize its performance using Matplotlib
- work with ModelFrames for feature extraction and label encoding
- configure and build a clustering model using the K-Means algorithm and analyze data clusters to determine characteristics that are unique to them
- distinguish between the use of scikit-learn and Pandas ML when training a model and identify some of the metrics used to evaluate a model

Machine & Deep Learning Algorithms: Imbalanced Datasets Using Pandas ML

Objectives:

- use Pandas ML to explore a dataset where the samples are not evenly distributed across the target classes
- apply the technique of oversampling using the RandomOverSampler class in the imbalanced-learn library, build a classification model with the oversampled data, and evaluate its performance
- create a balanced dataset using the Synthetic Minority Oversampling Technique and build and evaluate a classification model with that data
- perform undersampling operations on a dataset by applying the Near Miss, Cluster Centroids, and Neighborhood Cleaning Rule techniques
- use the EasyEnsembleClassifier and BalancedRandomForestClassifier available in the imbalanced-learn library to build classification models with imbalanced data
- apply a combination of oversampling and undersampling using the SMOTETomek and SMOTEENN techniques
- use Pandas and Seaborn to visualize the correlated fields in a dataset
- train and evaluate a classification model to predict the quality ratings of red wines
- transform a dataset containing multiple features to a handful of principal components and build a classification model using the reduced dimensions of the dataset
- combine the use of oversampling and PCA in building a classification model
- recall the techniques used by algorithms for undersampling and oversampling data and the use of combined samplers

Creating Data APIs Using Node.js

Objectives:

- identify and install the prerequisites to create an API using Node.js
- build a RESTful API using Node.js and Express.js
- build a RESTful API with OAuth in Node.js and describe what OAuth is and why it is required
- create an HTTP server using Hapi.js
- use modules in your API using Node.js
- return data with JSON using Node.js
- use nodemon for Development Workflow with Node.js
- make HTTP requests with Node.js using request library
- use POSTman to test your Node.js API
- deploy your APIs with Node.js
- connect to social media APIs with Node.js to return data
- build a RESTful API for creating tasks in MongoDB that provides notifications on restart

Data Visualization with Python

Objectives:

- Perform data visualization tasks with Python such as creating scatter plots, plotting linear regression, using logistic regression
- and creating decision tree.

Final Exam: Data Scientist

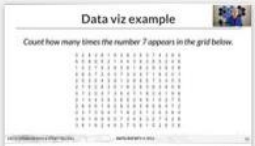
Objectives:

- add extensions to your dashboard such as Tableau Extensions API
- build and customize graphs using ggplot2 in R
- build backup and restore mechanisms in the cloud
- build heat maps and scatter plots using R
- can be leveraged to extract value from big data
- combine the use of oversampling and PCA in building a classification model
- compare the differences between the descriptive and inferential statistical analysis
- compare the different types of Recommendation Engines and how they can be used to solve different recommendation problems
- create an HTTP server using hapi.js
- create an R function that finds similar users and finds products they liked which would be good to recommend to the user
- create Histograms, Scatter plots, and Box plots using Python libraries
- define a port
- define the concept of storyboarding along with the prominent storyboarding templates that we can use to implement storyboarding
- demonstrate how to craft visual data using Tableau
- demonstrate how to create a stacked bar plot
- demonstrate how to implement data exploration using R
- demonstrate how to implement different types of bar charts using PowerBI
- demonstrate how we can ingest data using WaveFront
- demonstrate the steps involved in ingesting data from databases to Hadoop clusters using Sqoop
- describe blockchain
- describe how regression works by finding the best fit straight line to model the relationships in your data
- describe the aspects of data quality
- describe the concept of serverless computing and its benefits
- describe the Gestalt principles of visual perception
- describe the process involved in learning a relationship between input and output during the training phase of machine learning
- describe the various essential distributed data management frameworks used to handle big data
- describe what truncated data is and how to remove it using Azure Automation
- how the four Vs should be balanced in order to implement a successful big data strategy
- identify different cloud data sources available
- identify libraries that can be used in Python to implement data visualization
- identify the process and approaches involved in storytelling with data
- implement correlogram and build area charts using R
- implement Dask arrays in order to manage NumPy APIs
- implement data exploration using plots in R
- implement missing values and outliers using Python
- implement point and interval estimation using R
- implement Python Luigi in order to set up data pipelines
- install and prepare R for data exploration
- integrate Spark and Tableau to manage data pipelines
- Linear regression
- list and compare the various essential data ingestion tools that we can use to ingest data
- list Dask task scheduling and big data collection features
- list libraries that can be used in Python to implement data visualization

- load data from databases using R
- organize your dashboard by adding objects and adjusting the layout
- Pandas ML to explore a dataset where the samples are not evenly distributed across the target classes
- recall cloud migration models from the perspective of architectural preferences
- recall the various essential decluttering steps and approaches that we can implement to eliminate clutters
- recognize how to enable data-driven decision making
- recognize the data pipeline building capabilities provided by Kafka, Spark, and PySpark
- recognize the impact of implementing containerization on cloud hosting environments
- recognize the impact of the implementing Kubernetes and Docker in the cloud
- recognize the problems associated with a model that is overfitted to training data and how to mitigate the issue
- share your dashboard to others
- specify volume in big data analytics and its role in the principle of the four Vs
- use modules in your API using node.js
- use Pandas and Seaborn to visualize the correlated fields in a dataset
- use R to import, filter, and massage data into data sets
- use the scikit-learn library to build and train a LinearSVC classification model and then evaluate its performance using the available model evaluation functions
- work with vectors and metrics using Python and R

Bootcamp Replays

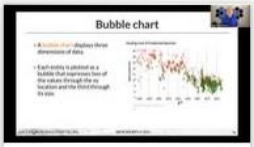
Optional



COURSE

Data Visualization and Storytelling Bootcamp:...

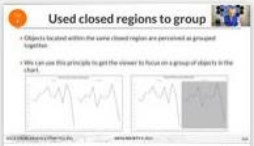
48



COURSE

Data Visualization and Storytelling Bootcamp:...

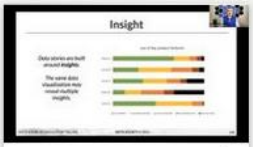
23



COURSE

Data Visualization and Storytelling Bootcamp:...

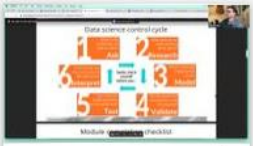
20



COURSE

Data Visualization and Storytelling Bootcamp:...

17



COURSE

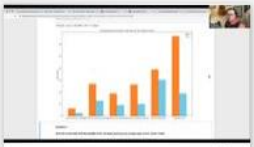
Data Visualization in Python Bootcamp: Session 1 Replay

19

COURSE

Data Visualization in Python Bootcamp: Session 2 Replay

11



COURSE

Data Visualization in Python Bootcamp: Session 3 Replay

9

COURSE

Data Visualization in Python Bootcamp: Session 4 Replay

5

COURSE

Data Wrangling in Python Bootcamp: Session 1 Replay

42

COURSE

Data Wrangling in Python Bootcamp: Session 2 Replay

20

COURSE

Data Wrangling in Python Bootcamp: Session 3 Replay

20

COURSE

Data Wrangling in Python Bootcamp: Session 4 Replay

23

COURSE

Introduction to R and Visualization Bootcamp:...

13

COURSE

Introduction to R and Visualization Bootcamp:...

5

COURSE

Introduction to R and Visualization Bootcamp:...

5

COURSE

Introduction to R and Visualization Bootcamp:...

5

COURSE

Introduction to R and Visualization Bootcamp:...

3

COURSE

Introduction to R and Visualization Bootcamp:...

4

COURSE

Advanced and Interactive Visualization in R Bootcam...

6

COURSE

Advanced and Interactive Visualization in R Bootcam...

6

COURSE

Advanced and Interactive Visualization in R Bootcam...

6

COURSE

Advanced and Interactive Visualization in R Bootcam...

7

COURSE

Advanced and Interactive Visualization in R Bootcam...

3

COURSE

Advanced and Interactive Visualization in R Bootcam...

3

COURSE

Advanced and Interactive Visualization in R Bootcam...

3

COURSE

Advanced and Interactive Visualization in R Bootcam...

3

Business & Leadership for Data Scientists

Optional

COURSE

Developing a Growth Mindset

2402

COURSE

Developing Your Business Acumen

715

COURSE

Using Strategic Thinking to Consider the Big Picture

603

COURSE

Using Active Listening in Workplace Situations

921

COURSE

Choosing the Right Interpersonal...

1267

COURSE

Building a Culture of Design Thinking

454

COURSE

Enabling Business Process Improvement

930

COURSE

The Essential Role of the Agile Product Owner

277

COURSE

Innovating with Lean Product Management

249

COURSE

Six Sigma Measurement System Analysis

191

COURSE

Reaching Sound Conclusions

279

COURSE

Capturing the Attention of Senior Executives

845

Productivity Tools for Data Scientists

Optional

COURSE

Signing in & Navigating within Spaces

41

COURSE

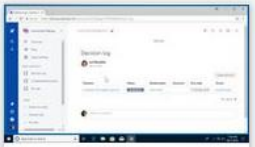
Setting Up & Managing Spaces

34

COURSE

Working with Space

28



COURSE

Working with Team Members

78

COURSE

Configuring Spaces

20

Optional Resources

Optional

LAB

Data Science Sandbox

Data Science 2020

17

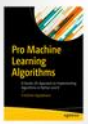
Bookshelf

Optional

BOOK

The Big Book of Dashboards: Visualizing Your Data Using...

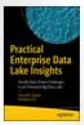
27



BOOK

Pro Machine Learning Algorithms: A Hands-On...

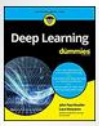
7



BOOK

Practical Enterprise Data Lake Insights: Handle Data-...

6



BOOK

Deep Learning for Dummies

8

BOOK

Python Data Analytics: With Pandas, NumPy, and...

50

BOOK

Data Science Fundamentals for Python and MongoDB

8

BOOK

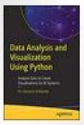
Data Science Using Python and R

27

BOOK

R in Action: Data Analysis and Graphics with R, Secon...

15



BOOK

Data Analysis and Visualization Using Python:...

7

BOOK

Practical Data Science: A Guide to Building the...

21

BOOK

Learn Data Analysis with Python: Lessons in Coding

27

BOOK

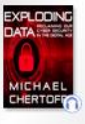
Beginning Apache Spark 2: With Resilient Distributed...

18

BOOK

PySpark SQL Recipes: With HiveQL, Dataframe and...

10



AUDIOBOOK

Exploding Data: Reclaiming Our Cyber Security in the...

14

AUDIOBOOK

Big Data: A Revolution That Will Transform How We Liv...

14

AUDIOBOOK

Predictive Marketing: Easy Ways Every Marketer Can...

7

BOOK

Introduction to MATLAB for Engineers and Scientists:...

3

BOOK

Think Like a Data Scientist: Tackle the Data Science...

3

BOOK

Introducing Data Science: Big Data, Machine Learning, an...

6

BOOK

Learn PySpark: Build Python-based Machine Learning an...

6

BOOK

Applied Reinforcement Learning with Python: With...

3

BOOK

Practical Data Science with Python 3: Synthesizing...

7

AUDIOBOOK

Data Science For Dummies, 2nd Edition

47

BOOK

Data Science for Dummies, 2nd Edition

47

BOOK

Data Science with Python and Dask

7

BOOK

Machine Learning with Spark and Python: Essential...

6

BOOK

Data Lakes

FOLLOW US ON:

www.skilltech.pl

email: biuro@skilltech.pl

tel. +48 22 44 88 827

SkillTech

Technology hired for excellence