

Software Engineering for Scientist

Extended curriculum overview (subject to changes)

This document provides an overview on the content and learning objectives from the different streams as well as the capstone projects.

The contents are subject to changes introduced by the steam leads.

As far as possible and for the pilot phase, the program will rely on existing content from renowned platforms (Udacity, Coursera, etc).

0. Onboarding and Introduction			
1 week			
1. Cloud Engineering for Scientists	2. Data Engineering for Scientists	3. ML Engineering for Scientists	4. Data Science for Scientists
Track Lead NN4+2 weeks	Track Lead NN4+2 weeks	Track Lead NN2+1 weeks	Track Lead NN4+2 weeks
Cloud Fundamentals	Data Modeling and Data Base Technologies	Clean Code Principles	Solving Data Science Problems
Cloud Overview	Introduction to Data Modeling	Coding Best Practices	The Data Science Process
Foundational and Compute Services	Relational Data Models	Working with Others Using Version Control	Communicating with Stakeholders
Storage and Content Delivery	NoSQL Data Models	Production Ready Code	Software Engineering for Data Scientists
Security	Key-value stores	Building a Reproducible Model Workflow	Software Engineering Practices
Networking & Elasticity	Document stores	Machine Learning Pipelines	Object Oriented Programming
Messaging & Containers	Graph DBMS	Data Exploration and Preparation	Web Development
AWS Management	Time Series DBMS	Data Validation	Working with transformers (Hugging Face)
Deploy Infrastructure as Code (IAC)	Search engines	Training, Validation and Experiment Tracking	TRANSFORMER MODELS
Getting Started with CloudFormation	RDF stores	Release and Deploy	USING Hugging Face TRANSFORMERS
Infrastructure Diagrams	Wide column stores	Deploying a Scalable ML Pipeline in Production	FINE-TUNING A PRETRAINED MODEL
Networking Infrastructure	Spatial DBMS	Performance Testing / Preparing a Model for Prod	SHARING MODELS AND TOKENIZERS
Servers and Security Groups	Cloud Data Warehouses	Data and Model Versioning	THE DATASETS LIBRARY
Storage and Databases	Introduction to the Data Warehouses	CI/CD	Supervised Learning
Build CI/CD Pipelines, Monitoring & Logging	Introduction to the Cloud with AWS	API Deployment with FastAPI	Regression
Continuous Integration and Continuous Deployment	Implementing Data Warehouses on AWS	Automated model scoring and monitoring	Perceptron Algorithms
Building a Continuous Integration Pipeline	Spark and Data Lakes	Model Training and Deployment	Decision Trees
Enabling Continuous Delivery with Deployment Pipelines	The Power of Spark	Model Scoring and Model Drift	Naive Bayes'
Monitoring Environments	Data Wrangling with Spark	Diagnosing and Fixing Operational Problems	Support Vector Machines
Microservices at Scale Using Kubernetes	Debugging and Optimization	Model Reporting and Monitoring with APIs	Ensemble of Learners
Deploy High-availability Microservice Event-Driven Application	Introduction to Data Lakes		Evaluation Metrics
Use Docker Format	Automate Data Pipelines		Training and Tuning Models
Containerization of Existing App	Data Pipelines		Neural Networks
Operationalize & Orchestrate Kubernetes	Data Quality		Introduction to Neural Networks
Running Containerized Applications	Production Data Pipelines		Implementing Gradient Descent
Deploying Managed Applications			Training Neural Networks
Configuring Networking in Kubernetes			Deep Learning with PyTorch
Customize Deployments for Application Requirements			Unsupervised Learning
Implementing Cloud Deployment Strategies			Clustering
Scaling with Red Hat OpenShift			Hierarchical and Density-Based Clustering
Deploying Applications to Red Hat OpenShift Container Platform			Gaussian Mixture Models
Configuring and Scaling Application Builds in OpenShift			Dimensionality Reduction
5. Closure and placement preparation			
1 week			
6. Placements			

Cloud DevOps	2
Data Engineer	5
Machine Learning DevOps.....	8
Data Scientist	10
Capstone Projects	15

Cloud DevOps

Module	Learning Objectives
Cloud Fundamentals	
Cloud Overview	<ul style="list-style-type: none">• Learn the basics of cloud computing including cloud deployment models, benefits, and popular options• Explore services provided by Amazon Web Services(AWS)
Foundational and Compute Services	<ul style="list-style-type: none">• Learn why we need servers, compute power, and security• Explore AWS compute services like Elastic Cloud Compute (EC2), Virtual Private Cloud (VPC), Lambda for serverless framework, and Elastic Beanstalk in action• Launch a secure EC2 instance, create and execute a Lambda, and deploy an application to Elastic Beanstalk
Storage and Content Delivery	<ul style="list-style-type: none">• Learn why we need storage and content delivery in the cloud• Learn storage services like S3, DynamoDB, Relational Database Service (RDS), and CloudFront• Create a DynamoDB table, launch a MySQL database instance, and create a CloudFront distribution
Security	<ul style="list-style-type: none">• Learn the importance of security in the cloud• See Identity & Access Management (IAM) in action• Secure applications using IAM users, groups, and policies
Networking & Elasticity	<ul style="list-style-type: none">• Learn the basics of networking and elasticity in the cloud• Examine services like Route 53, EC2 Auto Scaling, and Elastic Load Balancing• Add an auto scaling policy to your EC2 instance
Messaging & Containers	<ul style="list-style-type: none">• Learn the basics of messaging and containers in the cloud• Explore services like Simple Notification Service (SNS), Simple Queue Service (SQS), and Elastic Container Service (ECS)• Create cloud notifications using SNS
AWS Management	<ul style="list-style-type: none">• Learn why we need logging, auditing, and resource management in the cloud• Understand services like CloudWatch, CloudTrail, CloudFormation, and the AWS Command Line Interface (CLI)• Explore the CLI
Deploy Infrastructure as Code (IAC)	
Getting Started with CloudFormation	<ul style="list-style-type: none">• Set up the necessary tools to get started with CloudFormation and deploy your first server using CloudFormation
Infrastructure Diagrams	<ul style="list-style-type: none">• Convert business requirements into infrastructure diagrams and understand the principles behind design choices
Networking Infrastructure	<ul style="list-style-type: none">• Implement a virtual private network and subnets and learn how to provide inbound and outbound internet access to your public and private subnets inside your VPC• Use routing table to route the traffic within your virtual private cloud

Servers and Security Groups	<ul style="list-style-type: none"> • Deploy a web server into an autoscaling group • Implement load-balancer to increase capacity of your app • Implement security groups and understand the concept of least-privilege as it applies to network traffic
Storage and Databases	<ul style="list-style-type: none"> • Deploy S3 storage for images, config files, and more. • Deploy relational database and encryption service for your application
Build CI/CD Pipelines, Monitoring & Logging	
Continuous Integration and Continuous Deployment	<ul style="list-style-type: none"> • Understand the fundamentals of CI/CD. • Give examples of business-centered benefits of CI/CD. • Examine the utility of continuous delivery in a dev team. • List best practices. • Differentiate deployment strategies. • Recognize common building blocks of CI/CD pipelines.
Building a Continuous Integration Pipeline	<ul style="list-style-type: none"> • Understand how and why to use configuration management tools. • Utilize a configuration management tool to accomplish deployment to cloud-based servers. • Design a complete CI pipeline.
Enabling Continuous Delivery with Deployment Pipelines	<ul style="list-style-type: none"> • Know what configuration management tools are and how to use them. • Design an Ansible Playbook and control a remote machine. • Build an Ansible Inventory file. • Make various types of CD jobs in a CI/CD pipeline.
Monitoring Environments	<ul style="list-style-type: none"> • Install and configure Prometheus as a monitoring tool. • Get various data sources into Prometheus. • Analyze monitoring data. • Set up alerts.
Microservices at Scale Using Kubernetes	
Deploy High-availability Microservice Event-Driven Application	<ul style="list-style-type: none"> • Understand Serverless (AWS Lambda) concepts • Understand which container abstraction to use: AWS Lambda or Kubernetes • Deploy producer/consumer AWS Lambda applications • Configure CloudWatch events
Use Docker Format	<ul style="list-style-type: none"> • Understand Docker image format Containers • Run and modify Docker containers locally • Deploy customized containers to Amazon ECR
Containerization of Existing App	<ul style="list-style-type: none"> • Use the appropriate Docker base image • Install packages into Docker image • Copy application into Docker image • Configure application setup and start in Docker image

Operationalize & Orchestrate Kubernetes	<ul style="list-style-type: none"> • Understand Kubernetes concepts • Configure monitoring, alerts, and incidence response • Integrate CI/CD Pipeline • Configure Autoscaling
Running Containerized Applications	<ul style="list-style-type: none"> • Contrasting Kubernetes Distributions • Introducing Kubectl • Running and Interacting with Your First Application
Deploying Managed Applications	<ul style="list-style-type: none"> • Managing Containers • Creating a Deployment • Understanding the Schema of a Deployment Resource
Configuring Networking in Kubernetes	<ul style="list-style-type: none"> • Kubernetes Networking • Introducing Kubernetes Services • Discovering Kubernetes Services • Kubernetes Ingress + Kubernetes Ingress Controller • Ingress Resource Configuration + Testing Your Ingress
Customize Deployments for Application Requirements	<ul style="list-style-type: none"> • Defining Resource Requests and Limits for Pods + Viewing Requests, Limits, and Actual Usage • Applying Quotas and Limit Ranges • Proving Liveness, Readiness and Startup • Methods of Checking Application Health + Creating Probes • Externalizing Application Configuration in Kubernetes + Using Secret and Configuration Map Resources • Creating and Managing Secrets and Configuration Maps • Injecting Data from Secrets and Configuration Maps into Applications + Application Configuration Options • Exploring Environment Variables
Implementing Cloud Deployment Strategies:	<ul style="list-style-type: none"> • Deployment Strategies in Kubernetes • Implementing Advanced Deployment Strategies Using the Kubernetes Router • Guided Exercise: Implementing Cloud Deployment Strategies
Scaling with Red Hat OpenShift	
Deploying Applications to Red Hat OpenShift Container Platform	<ul style="list-style-type: none"> • Introducing OpenShift Container Platform • Introducing the Developer Web Console for OpenShift
Configuring and Scaling Application Builds in OpenShift	<ul style="list-style-type: none"> • Updating an Application • Configuring Application Secrets • Connecting an Application to a Database • Scaling Applications in OpenShift

Data Engineer

Module	Learning Objectives
Data Modeling and Data Base Technologies	
Introduction to Data Modeling	<ul style="list-style-type: none">• Understand the purpose of data modeling• Identify the strengths and weaknesses of different types of databases and data storage techniques• Create a table in Postgres and Apache Cassandra
Relational Data Models	<ul style="list-style-type: none">• Understand when to use a relational database• Understand the difference between OLAP and OLTP databases• Create normalized data tables• Implement denormalized schemas (e.g. STAR, Snowflake)
NoSQL Data Models	<ul style="list-style-type: none">• Understand when to use NoSQL databases and how they differ from relational databases• Select the appropriate primary key and clustering columns for a given use case• Create a NoSQL database in Apache Cassandra
Key-value stores	<ul style="list-style-type: none">• Understand when to use a key value store• Overview of the main key-value store technologies• Create a key-value store database in Redis
Document stores	<ul style="list-style-type: none">• Understand when to use a document store• Overview of the main document store technologies• Create a document store database in MongoDB
Graph DBMS	<ul style="list-style-type: none">• Understand when to use a graph database• Overview of the main graph database technologies• Create a graph database in Neo4j (or ArangoDB)
Time Series DBMS	<ul style="list-style-type: none">• Understand when to use a time series database• Overview of the main time series database technologies• Create a time series database in InfluxDB (or TimescaleDB)
Search engines	<ul style="list-style-type: none">• Understand when to use a search engine• Overview of the main search engine technologies• Create a search engine using Elasticsearch
RDF stores	<ul style="list-style-type: none">• Understand when to use a RDF store• Overview of the main RDF store technologies• Create a RDF store using Virtuoso (or MarkLogic)
Wide column stores	<ul style="list-style-type: none">• Understand when to use a wide column database• Overview of the main wide column database technologies• Create a wide column database using Cassandra (or HBase)

Spatial DBMS	<ul style="list-style-type: none"> • Understand when to use a Spatial DBMS • Overview of the main Spatial DBMS technologies • Create a Spatial DBMS using PostGIS
Cloud Data Warehouses	
Introduction to the Data Warehouses	<ul style="list-style-type: none"> • Understand Data Warehousing architecture • Run an ETL process to denormalize a database (3NF to Star) • Create an OLAP cube from facts and dimensions • Compare columnar vs. row oriented approaches
Introduction to the Cloud with AWS (refresher)	<ul style="list-style-type: none"> • Understand cloud computing • Create an AWS account and understand their services • Set up Amazon S3, IAM, VPC, EC2, RDS PostgreSQL
Implementing Data Warehouses on AWS	<ul style="list-style-type: none"> • Identify components of the Redshift architecture • Run ETL process to extract data from S3 into Redshift • Set up AWS infrastructure using Infrastructure as Code (IaC) • Design an optimized table by selecting the appropriate distribution style and sorting key
Spark and Data Lakes	
The Power of Spark	<ul style="list-style-type: none"> • Understand the big data ecosystem • Understand when to use Spark and when not to use it
Data Wrangling with Spark	<ul style="list-style-type: none"> • Manipulate data with SparkSQL and Spark Dataframes • Use Spark for ETL purposes
Debugging and Optimization	<ul style="list-style-type: none"> • Troubleshoot common errors and optimize their code using the Spark WebUI
Introduction to Data Lakes	<ul style="list-style-type: none"> • Understand the purpose and evolution of data lakes • Implement data lakes on Amazon S3, EMR, Athena, and Amazon Glue • Use Spark to run ELT processes and analytics on data of diverse sources, structures, and vintages • Understand the components and issues of data lakes
Automate Data Pipelines	
Data Pipelines	<ul style="list-style-type: none"> • Create data pipelines with Apache Airflow • Set up task dependencies • Create data connections using hooks
Data Quality	<ul style="list-style-type: none"> • Track data lineage • Set up data pipeline schedules • Partition data to optimize pipelines • Write tests to ensure data quality • Backfill data
Production Data Pipelines	<ul style="list-style-type: none"> • Build reusable and maintainable pipelines • Build your own Apache Airflow plugins • Implement subDAGs • Set up task boundaries • Monitor data pipelines

Machine Learning DevOps

Module	Learning Objectives
Clean Code Principles	
Coding Best Practices	<ul style="list-style-type: none">• Write clean, modular and well-documented code• Refactor code for efficiency• Follow PEP8 Standards• Automate use of PEP8 standards using PyLint and AutoPEP8
Working with Others Using Version Control	<ul style="list-style-type: none">• Work independently using git and Github• Work with teams using git and Github• Create branches for isolating changes in git and Github• Open pull requests for making changes to production code• Conduct and receive code reviews using best practices
Production Ready Code	<ul style="list-style-type: none">• Correctly use try-except blocks to identify errors• Create unit tests to test programs• Track actions and results of processes with logging• Identify model drift and when automated or non-automated retraining should be used to make model updates
Building a Reproducible Model Workflow	
Machine Learning Pipelines	<ul style="list-style-type: none">• MLOps fundamentals• Version data and artifacts• Write a ML pipeline component• Link together ML components
Data Exploration and Preparation	<ul style="list-style-type: none">• Execute and track the Exploratory Data Analysis(EDA)• Clean and pre-process the data• Segregate(split)datasets
Data Validation	<ul style="list-style-type: none">• Use pytest with parameters for reproducible and automatic data tests• Perform deterministic and non-deterministic data tests
Training, Validation and Experiment Tracking	<ul style="list-style-type: none">• Tame the chaos with experiment, code and data tracking• Track experiments with W&B• Validate and choose best-performing model• Export model as an inference artifact• Test final inference artifact
Release and Deploy	<ul style="list-style-type: none">• Release pipeline code• Options for deployment and how to deploy a model
Deploying a Scalable ML Pipeline in Production	

Performance Testing and Preparing a Model for Production	<ul style="list-style-type: none"> • Analyze slices of data when training and testing models • Probe a model for bias using common frameworks such as Aequitas • Write model cards that explain the purpose, provenance and pitfalls of a model
Data and Model Versioning	<ul style="list-style-type: none"> • Version control data/models/etc locally using DVC • Setup remote storage for use with DVC • Create pipelines and track experiments with DVC
CI/CD	<ul style="list-style-type: none"> • Follow software engineering principles by automating, testing and versioning code • Setup Continuous Integration using GitHub Actions • Setup Continuous Deployment using Heroku
API Deployment with FastAPI	<ul style="list-style-type: none"> • Write an API for machine learning inference using FastAPI • Deploy a machine learning inference API to Heroku • Write unit tests for APIs using the requests module
Automated model scoring and monitoring	
Model Training and Deployment	<ul style="list-style-type: none"> • Ingest data • Automatically train models • Deploy models to production • Keep records about processes • Automate processes using cronjobs
Model Scoring and Model Drift	<ul style="list-style-type: none"> • Automatically score ML models • Keep records of model scores • Check for model drift using several different model drift tests • Determine whether models need to be retrained and re-deployed
Diagnosing and Fixing Operational Problems	<ul style="list-style-type: none"> • Check data integrity and stability • Check for dependency issues • Check for timing issues • Resolve operational issues
Model Reporting and Monitoring with APIs	<ul style="list-style-type: none"> • Create API endpoints that enable users to access model results, metrics and diagnostics • Set up APIs with multiple, complex endpoints • Call APIs and work with their results

Data Scientist

Module	Learning Objectives
Solving Data Science Problems	
The Data Science Process	<ul style="list-style-type: none">• Apply the CRISP-DM process to business applications• Wrangle, explore, and analyze a dataset• Apply machine learning for prediction• Apply statistics for descriptive and inferential understanding• Draw conclusions that motivate others to act on your results
Communicating with Stakeholders	<ul style="list-style-type: none">• Implement best practices in sharing your code and written summaries• Learn what makes a great data science blog• Learn how to create your ideas with the data science community
Software Engineering for Data Scientists	
Software Engineering Practices	<ul style="list-style-type: none">• Write clean, modular, and well-documented code• Refactor code for efficiency• Create unit tests to test programs• Write useful programs in multiple scripts• Track actions and results of processes with logging• Conduct and receive code reviews
Object Oriented Programming	<ul style="list-style-type: none">• Understand when to use object oriented programming• Build and use classes• Understand magic methods• Write programs that include multiple classes, and follow good code structure• Learn how large, modular Python packages, such as pandas and scikit-learn, use object oriented programming <ul style="list-style-type: none">• Portfolio Exercise: Build your own Python package
Web Development	<ul style="list-style-type: none">• Learn about the components of a web app• Build a web application that uses Flask, Plotly, and the Bootstrap framework• Portfolio Exercise: Build a data dashboard using a dataset of your choice and deploy it to a web application
Data Engineering for Data Scientists	

ETL Pipelines	<ul style="list-style-type: none"> • Understand what ETL pipelines are • Access and combine data from CSV, JSON, logs, APIs, and databases • Standardize encodings and columns • Normalize data and create dummy variables • Handle outliers, missing values, and duplicated data • Engineer new features by running calculations • Build a SQLite database to store cleaned data
Natural Language Processing	<ul style="list-style-type: none"> • Prepare text data for analysis with tokenization, lemmatization, and removing stop words • Use scikit-learn to transform and vectorize text data • Build features with bag of words and tf-idf • Extract features with tools such as named entity recognition and part of speech tagging • Build an NLP model to perform sentiment analysis
Machine Learning Pipelines	<ul style="list-style-type: none"> • Understand the advantages of using machine learning pipelines to streamline the data preparation and modeling process • Chain data transformations and an estimator with scikit-learn's Pipeline • Use feature unions to perform steps in parallel and create more complex workflows • Grid search over pipeline to optimize parameters for entire workflow • Complete a case study to build a full machine learning pipeline that prepares data and creates a model for a dataset
Experiment Design and Recommendations	
Experiment Design	<ul style="list-style-type: none"> • Understand how to set up an experiment, and the ideas associated with experiments vs. observational studies • Defining control and test conditions • Choosing control and testing groups
Statistical Concerns of Experimentation	<ul style="list-style-type: none"> • Applications of statistics in the real world • Establishing key metrics • SMART experiments: Specific, Measurable, Actionable, Realistic, Timely
A/B Testing	<ul style="list-style-type: none"> • How it works and its limitations • Sources of Bias: Novelty and Recency Effects • Multiple Comparison Techniques (FDR, Bonferroni, Tukey) • Portfolio Exercise: Using a technical screener from Starbucks to analyze the results of an experiment and write up your findings
Introduction to Recommendation Engines	<ul style="list-style-type: none"> • Distinguish between common techniques for creating recommendation engines including knowledge based, content based, and collaborative filtering based methods. • Implement each of these techniques in python. • List business goals associated with recommendation engines, and be able to recognize which of these goals are most easily met with existing recommendation techniques.

Matrix Factorization for Recommendations	<ul style="list-style-type: none"> • Understand the pitfalls of traditional methods and pitfalls of measuring the influence of recommendation engines under traditional regression and classification techniques. • Create recommendation engines using matrix factorization and FunkSVD • Interpret the results of matrix factorization to better understand latent features of customer data • Determine common pitfalls of recommendation engines like the cold start problem and difficulties associated with usual tactics for assessing the effectiveness of recommendation engines using usual techniques, and potential solutions.
Data Science Projects	
Elective 1: Dog Breed Classification	<ul style="list-style-type: none"> • Use convolutional neural networks to classify different dogs according to their breeds • Deploy your model to allow others to upload images of their dogs and send them back the corresponding breeds. • Complete one of the most popular projects in Udacity history, and show the world how you can use your deep learning skills to entertain an audience!
Elective 2: Starbucks	<ul style="list-style-type: none"> • Use purchasing habits to arrive at discount measures to obtain and retain customers • Identify groups of individuals that are most likely to be responsive to rebates.
Elective 3: Arvato Financial Services	<ul style="list-style-type: none"> • Work through a real-world dataset and challenge provided by Arvato Financial Services, a Bertelsmann company • Top performers have a chance at an interview with Arvato or another Bertelsmann company!
Elective 4: Spark for Big Data	<ul style="list-style-type: none"> • Take a course on Apache Spark and complete a project using a massive, distributed dataset to predict customer churn • Learn to deploy your Spark cluster on either AWS or IBM Cloud
Elective 5: Your Choice	• Use your skills to tackle any other project of your choice
Data Science Projects	
Elective 1: Dog Breed Classification	<ul style="list-style-type: none"> • Use convolutional neural networks to classify different dogs according to their breeds • Deploy your model to allow others to upload images of their dogs and send them back the corresponding breeds. • Complete one of the most popular projects in Udacity history, and show the world how you can use your deep learning skills to entertain an audience!
Elective 2: Starbucks	<ul style="list-style-type: none"> • Use purchasing habits to arrive at discount measures to obtain and retain customers • Identify groups of individuals that are most likely to be responsive to rebates.
Elective 3: Arvato Financial Services	<ul style="list-style-type: none"> • Work through a real-world dataset and challenge provided by Arvato Financial Services, a Bertelsmann company • Top performers have a chance at an interview with Arvato or another Bertelsmann company!
Elective 4: Spark for Big Data	<ul style="list-style-type: none"> • Take a course on Apache Spark and complete a project using a massive, distributed dataset to predict customer churn • Learn to deploy your Spark cluster on either AWS or IBM Cloud
Elective 5: Your Choice	• Use your skills to tackle any other project of your choice

Working with transformers	
TRANSFORMER MODELS	<ul style="list-style-type: none"> • Introduction • Natural Language Processing • Transformers, what can they do? • How do Transformers work? • Encoder models • Decoder models • Sequence-to-sequence models • Bias and limitations
USING Hugging Face TRANSFORMERS	<ul style="list-style-type: none"> • Introduction • Behind the pipeline • Models • Tokenizers • Handling multiple sequences
FINE-TUNING A PRETRAINED MODEL	<ul style="list-style-type: none"> • Introduction • Processing the data • Fine-tuning a model with the Trainer API or Keras • A full training
SHARING MODELS AND TOKENIZERS	<ul style="list-style-type: none"> • The Hugging Face Hub • Using pretrained models • Sharing pretrained models • Building a model card
THE DATASETS LIBRARY	<ul style="list-style-type: none"> • Introduction • What if my dataset isn't on the Hub? • Time to slice and dice • Big data? • Creating your own dataset • Semantic search with FAISS
Supervised Learning	
Regression	<ul style="list-style-type: none"> • Learn the difference between Regression and Classification • Train a Linear Regression model to predict values • Learn to predict states using Logistic Regression
Perceptron Algorithms	<ul style="list-style-type: none"> • Learn the definition of a perceptron as a building block for neural networks, and the perceptron algorithm for classification.
Decision Trees	<ul style="list-style-type: none"> • Train Decision Trees to predict states • Use Entropy to build decision trees, recursively
Naive Bayes'	<ul style="list-style-type: none"> • Learn Bayes' rule, and apply it to predict cases of spam messages using the Naive Bayes algorithm. • Train models using Bayesian Learning • Complete an exercise that uses Bayesian Learning for natural language processing
Support Vector Machines	<ul style="list-style-type: none"> • Learn to train a Support Vector Machines to separate data, linearly • Use Kernel Methods in order to train SVMs on data that is not linearly separable

Ensemble of Learners	<ul style="list-style-type: none"> • Build data visualizations for quantitative and categorical data. • Create pie, bar, line, scatter, histogram, and boxplot charts. • Build professional presentations.
Evaluation Metrics	<ul style="list-style-type: none"> • Learn about different metrics to measure model success. • Calculate accuracy, precision, and recall to measure the performance of your models.
Training and Tuning Models	<ul style="list-style-type: none"> • Train and test models with Scikit-learn. • Choose the best model using evaluation techniques like cross-validation and grid search.
Neural Networks	
Introduction to Neural Networks	<ul style="list-style-type: none"> • Learn the foundations of deep learning and neural networks. • Implement gradient descent and backpropagation in Python
Implementing Gradient Descent	<ul style="list-style-type: none"> • Implement gradient descent using NumPy matrix multiplication.
Training Neural Networks	<ul style="list-style-type: none"> • Learn several techniques to effectively train a neural network. • Prevent overfitting of training data and learn best practices for minimizing the error of a network.
Deep Learning with PyTorch	<ul style="list-style-type: none"> • Learn how to use PyTorch for building deep learning models.
Unsupervised Learning	
Clustering	<ul style="list-style-type: none"> • Learn the basics of clustering data • Cluster data with the K-means algorithm
Hierarchical and Density-Based Clustering	<ul style="list-style-type: none"> • Cluster data with Single Linkage Clustering. • Cluster data with DBSCAN, a clustering method that captures the insight that clusters are dense group of points
Gaussian Mixture Models	<ul style="list-style-type: none"> • Cluster data with Gaussian Mixture Models • Optimize Gaussian Mixture Models with and Expectation Maximization
Dimensionality Reduction	<ul style="list-style-type: none"> • Reduce the dimensionality of the data using Principal • Component Analysis and Independent Component Analysis

Capstone Projects

The participants will complete a capstone project featuring a science challenge for each stream (usually 1 or 2 weeks)

The capstone projects will be presented by the end of the stream completion period to the peers and to selected members of the pillars.

The potential capstone projects will be advertise during the stream. Participants are also encourage to propose their own ones.