Data Scientist Nanodegree Syllabus

Before You Start

Prerequisites: The Data Scientist Nanodegree program is an advanced program designed to prepare you for data scientist jobs. As such, you should have a high comfort level with a variety of topics before starting the program. In order to successfully complete this program, we strongly recommend that the following prerequisites are fulfilled. If you do not have the necessary prerequisites, Udacity has courses and programs that prepare you for this Nanodegree program.

- Programming:
  - Python Programming: Writing functions, logic, control flow, and building basic applications, as well as common data analysis libraries like NumPy and Pandas
  - SQL programming: Querying databases using joins, aggregations, and subqueries
  - Comfortable with using the Terminal, version control in Git, and using Github
- Probability and Statistics
  - Descriptive Statistics: Calculating measures of center and spread, estimation distributions
  - Inferential Statistics: Sampling distributions, hypothesis testing
  - Probability: Probability theory, conditional probability
- Mathematics
  - Calculus: Maximizing and minimizing algebraic equations
  - Linear Algebra: Matrix manipulation and multiplication
- Data wrangling
  - Accessing database, CSV, and JSON data
  - Data cleaning and transformations using Pandas and Sklearn
- Data visualization with matplotlib
  - Exploratory data analysis and visualization
  - Explanatory data visualizations and dashboards

Educational Objectives: The ultimate goal of the Data Scientist Nanodegree program is for you to learn the skills you need to perform well as a data scientist. As a graduate of this program, you will be able to:

- Use Python and SQL to access and analyze data from several different data sources.
- Use principles of statistics and probability to design and execute A/B tests and recommendation engines to assist businesses in making data-automated decisions.
- Build predictive models using a variety of machine learning techniques.
- Perform feature engineering to improve performance of machine learning models.
- Understand how to optimize, tune, and improve algorithms to improve run speed.
- Compare the performances of learned models using suitable metrics.
- Use cloud-based solutions to deploy a data science solution to a basic flask app.
- Manipulate and analyze data at scale using Spark.
- Communicate results effectively to stakeholders.

Length of Program*: Term 1: 3 months (130 hrs), Term 2: 4 months (170 Hours)
Frequency of Classes: The program is self-paced within two terms. Term 1 is 3 months long and Term 2 is 4 months long. All projects must be completed by the end of the term.
Textbooks required: None
Textbooks optional: Elements of Statistical Learning, Machine Learning: A Probabilistic Perspective, Python Machine Learning
Instructional Tools Available: Video lectures, mentors, Slack channel, project reviews

*The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. If you spend about 10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

TERM 1: MACHINE LEARNING FOR DATA SCIENTISTS

Project 1: Find Donors for CharityML with Kaggle

CharityML is a fictitious charity organization that provides financial support for people learning machine learning. In an effort to improve donor outreach effectiveness, you'll build an algorithm that best identifies potential donors. Your goal will be to evaluate and optimize several different supervised learners to determine which algorithm will provide the highest donation yield. You can also submit this project in a Udacity competition on Kaggle to see how you rank vs. your fellow students.

Supporting Lessons: Supervised Learning

Lesson Title	Learning Outcomes
BIRD'S EYE VIEW	➔ Types of machine learning ➔ The history of machine learning ➔ Ethical implications
REGRESSION	➔ Distinguish between Regression and Classification ➔ Learn to predict values with Linear Regression ➔ Learn to predict states using Logistic Regression
PERCEPTRON ALGORITHMS	➔ Learn the definition of a perceptron as a building block for neural networks, and the perceptron algorithm for classification
DECISION TREES	➔ Train Decision Trees to predict states ➔ Use Entropy to build decision trees recursively
NAIVE BAYES	➔ Learn the Bayes rule, and how to apply it to predicting data using the Naive Bayes algorithm ➔ Train models using Bayesian Learning ➔ Use Bayesian Inference to create Bayesian Networks of several variables ➔ Bayes NLP Mini-Project

SUPPORT VECTOR MACHINES	➔ Learn to train a Support Vector Machine to separate data linearly ➔ Use Kernel Methods in order to train SVMs on data that is not linearly separable
ENSEMBLE OF LEARNERS	➔ Enhance traditional algorithms via boosting ➔ Random forests ➔ AdaBoost
EVALUATION METRICS	➔ Learn about metrics such as accuracy, precision, and recall, used to measure the performance of your models
TRAINING AND TUNING	➔ Choose the best model using cross-validation and grid search.

Project 2: Create an Image Classifier

In this project, you will implement an image classification application using a deep learning model on a dataset of images. You will then use the trained model to classify new images. First you will develop your code in a Jupyter notebook, then convert it into a Python application that you will run from the command line of your system.

Supporting Lessons: Introduction to Deep Learning

Supporting Lessons	Learning Outcomes
INTRODUCTION TO NEURAL NETWORKS	➔ Acquire a solid foundation in deep learning and neural networks. Implement gradient descent and backpropagation in Python
IMPLEMENTING GRADIENT DESCENT	➔ Implement gradient descent to train deep learning networks.
TRAINING NEURAL NETWORKS	➔ Learn about techniques for how to improve training of a neural network, such as: early stopping, regularization, and dropout
KERAS	➔ Learn how to use Keras for building deep learning models
DEEP LEARNING WITH PYTORCH	➔ Learn how to use PyTorch for building deep learning models

Project 3: Creating Customer Segments

The data and design for this project was provided by Arvato Financial Services. You will apply unsupervised learning techniques on demographic and spending data for a sample of German households. You will preprocess the data, apply dimensionality reduction techniques, and implement clustering algorithms to segment customers with the goal of optimizing customer outreach for a mail order company.

Supporting Lessons: Unsupervised Learning

Lesson Title	Learning Outcomes
CLUSTERING	➔ Learn the basics of clustering Data ➔ Cluster data with the K-means algorithm
HIERARCHICAL AND DENSITY-BASED CLUSTERING	➔ Cluster data with Single Linkage Clustering ➔ Cluster data with DBSCAN, a clustering method that captures the insight that clusters are a dense group of points
GAUSSIAN MIXTURE MODELS	➔ Cluster data with Gaussian Mixture Models ➔ Optimize Gaussian Mixture Models with Expectation Maximization
PRINCIPAL COMPONENT ANALYSIS	➔ Reduce the dimensionality of the data using Principal Component Analysis through exploring handwritten digits
RANDOM PROJECTIONS & INDEPENDENT COMPONENT ANALYSIS	➔ Explore noise signal data from a cello, a television, and a piano using additional methods of dimensionality reduction

TERM 2: APPLIED DATA SCIENCE

Project 4: Write a Data Science Blog Post

In this project, you will choose a dataset, identify three questions, and analyze the data to find answers to these questions. You will create a GitHub repository with your project, and write a blog post to communicate your findings to the appropriate audience. This project will help you reinforce and extend your knowledge of machine learning, data visualization, and communication.

Supporting Lessons: Solving Problems with Data Science

Supporting Lessons	Learning Outcomes
THE DATA SCIENCE PROCESS	➔ 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
DATA VISUALIZATION WITH PYTHON	➔ Apply data visualization for exploratory and explanatory purposes ➔ Understand how design principles are vital to effective data visualizations ➔ Build univariate, bivariate, and multivariate visualizations in Python

COMMUNICATING WITH STAKEHOLDERS	➔ 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

Project 5: Build Pipelines to Classify Messages with Figure Eight

Figure Eight (formerly Crowdflower) crowdsourced the tagging and translation of messages to apply artificial intelligence to disaster response relief. In this project, you'll build a data pipeline to prepare the message data from major natural disasters around the world. You'll build a machine learning pipeline to categorize emergency text messages based on the need communicated by the sender.

Supporting Lessons: Software Engineering for Data Scientists

Supporting Lessons	Learning Outcomes
SOFTWARE ENGINEERING PRACTICES	➔ 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	➔ 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 ➔ *Portfolio Exercise*: Build your own Python package
WEB DEVELOPMENT	➔ 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

Supporting Lessons: Data Engineering for Data Scientists

Supporting Lessons	Learning Outcomes
ETL PIPELINES	➔ 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
MACHINE LEARNING PIPELINES	→ 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
NATURAL LANGUAGE PROCESSING	→ 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

Project 6: Design a Recommendation Engine with IBM

IBM has an online data science community where members can post tutorials, notebooks, articles, and datasets. In this project, you will build a recommendation engine based on user behavior and social network data, to surface content most likely to be relevant to a user. You'll work with IBM Watson and IBM Cloud to build and deploy the recommendation to a front-end application.

Supporting Lessons: Experiment Design

Supporting Lessons	Learning Outcomes
EXPERIMENT DESIGN	→ Applications of statistics in the real world → Establishing key metrics → Defining control and test conditions → Choosing control and testing groups → SMART experiments: Specific, Measurable, Actionable, Realistic, Timely
A/B TESTING	→ 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 an actual company, analyze the results of an experiment and write up your findings

Supporting Lessons: Recommendations

Supporting Lessons	Learning Outcomes

MATRIX FACTORIZATION FOR RECOMMENDATIONS	➔ Learn about the role of matrix factorization in machine learning ➔ Implement matrix factorization techniques using real data in Python ➔ Interpret the results of matrix factorization to better understand latent features of customer data
RECOMMENDATION ENGINES	➔ Create recommendation engines using common techniques such as collaborative filtering and matrix factorization ➔ Understand common pitfalls of recommendation engines like the cold start problem ➔ Overcome common pitfalls of recommendation engines using industry vetted techniques
DEPLOYING YOUR RECOMMENDATION ENGINE	➔ Integrate your machine learning algorithm into an online application using flask and bootstrap ➔ Put it all together—use machine learning and software engineering to recommend new movies to users

Project 7: Data Science Capstone Project

In this capstone project, you will leverage what you've learned throughout the program to build a data science project of your choosing. You will define the problem you want to solve, investigate and explore the data, identify and explore the data, then perform your analyses and develop a set of conclusions. You will present the analysis and your conclusions in a blog post and GitHub repository. This project will serve as a demonstration of your ability as a data scientist, and will be an important component of your job-ready portfolio.

Supporting Lessons: Data Science Projects

Supporting Lessons	Learning Outcomes
ELECTIVE 1: DOG BREED CLASSIFICATION	➔ 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
ELECTIVE 2: SPARK FOR BIG DATA	➔ Take a Spark course and complete a project using a massive dataset of Spotify data to predict customer churn ➔ Use Spark to wrangle streaming data and make real time predictions
ELECTIVE 3: ARVATO FINANCIAL SERVICES	➔ 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: YOUR CHOICE	➔ Use your skills to tackle any other project of your choice