Course 2: Tools for Data Science

Week 1:

Lesson 1: Languages of Data Science

Course Introduction

- Data science tooling
- Tasks needed to be performed
- Top open source and commercial tools
- How tools overlap, pros and cons
- Data manipulation
- Automation with data science tooling
- Visual programming & modeling
- Cloud computing with Data Science

Languages of Data Science

- Main Ones: Python, R, SQL
- Alternatives: Scala, Java, C++, julia
- Specific use cases: JS, php, go, ruby, visual basic
- Language choice depends on problems needed to be solved
- Roles:
    - Business analyst
    - Database Engineer
    - Data Analyst
    - Data Engineer
    - Data Scientist
    - Research Scientist
    - Software Engineer
    - Statistician
    - Product Manager
    - Project Manager
    - Etc.

Introduction to Python

- Powerhouse language
- Most popular
- 2019: 75% of people use python on regular basis
- Glassdoor: 75% of data science positions require python
- Large orgs that use Python for Data Science: IBM, Wikipedia, Google, Yahoo, CERN, NASA, Facebook, Amazon, Instagram, Spotify, Reddit

- General purpose language
- Large standard library
- **Data Science Uses:**
    - Scientific Computing
        - Pandas
        - NumPy
        - SciPy
        - Matplotlib
    - Artificial Intelligence
        - PyTorch
        - TensorFlow
        - Keras
        - Scikit-learn
    - Natural Language Processing (NLP)
        - Natural Language Toolkit (NLTK)

Introduction to R

- Learning up to 3 languages can increase your salary!
- Python is Open Source (OSI, Open Source Initiative), R is free software (FSF, Free Software Foundation)
- Open Source & Free Software are similar but not the same
- Used by
    - Statisticians
    - Mathematicians
    - Data miners
- Most learn a few year into their career
- 2018: 15000 publicly released packages

Introduction to SQL

- Structured Query Language
- Older than Python and R by 20 years
- Useful in handling structured data
- Why Great?
    - Business and data analyst positions, data engineering
    - Data accessed directly
    - SQL is interpreter between you and database
    - ANSI standard, can apply SQL knowledge to many different databases
- Databases available:
    - MySQL
    - IBM DB2
    - PostgreSQL
    - Apache Open Office Space
    - SQLite
    - Oracle

        - MariaDB
        - Microsoft SQL server

Other Languages

- Scala, Java, C++, julia
- JS, php, go, ruby, visual basic
- Java
    - General purpose object oriented language
    - Designed to be fast and scalable
    - Java applications are compiled to bytecode and run on JVM
    - Data science uses:
        - Weka (data mining)
        - Java-ML (machine learning)
        - Apache MLlib(scaleable ML
        - Deeplearning4j
        - Hadoop: data processing and storage for big data applications running in clustered systems
- Scala
    - General purpose
    - Strong static type system
    - Addresses criticisms of Java
    - Runs on JVM
    - Data Science
        - Apache Spark
            - Shark
            - MLib
            - GraphX
            - Spark Streaming
- C++
    - Improves processing speed
    - System programming
    - TensorFlow made in C++ but runs on Python interface
    - MongoDB (no SQL) made in C++
- JS
    - Tensorflow JS
    - Node.js brings it beyond web programming
- Julia
    - Designed in MIT
    - Speedy development while running fast programs
    - Code is executed directly on processor
    - JuliaDB

Lesson 2: Data Science Tools

Categories of Data Science Tools

- Data Management
    - Retrieving data
- Data Integration and Transformation
    - Extract, transform, load
- Data Visualization
    - Data exploration process
- Model Building
    - Create ML model with appropriate algorithm
- Model Deployment
    - Make model available to 3rd party applications
- Model Monitoring and Assessment
    - Quality checks on model
- Code Asset Management
    - Versioning to facilitate teamwork
- Data Asset Management
    - Supports replication, backup, asset rights
- Development Environments
    - IDEs
    - implement, execute, test, deploy work
- Execution Environments
    - Deployment and model training take place
- Fully Integrated Visual Tooling
    - Covers all other tooling stuff

Open Source Tools

Part 1:

- Data Management
    - Relational Databases
        - MySQL
        - PostgreSQL
    - NoSQL Databases
        - MongoDB
        - CouchDB
        - Cassandra
    - File Based Tools
        - Hadoop (hdfs)
        - Ceph (cloud)
        - Elasticsearch (fast document retrieval)
- Data Integration and Transformation
    - Apache Airflow

        - KubeFlow
        - Kafka
        - Nifi
        - SparkSQL
        - Node-RED
- Data Visualization
    - Hue
    - Kibana
    - Superset
- Model Deployment
    - Apache PredictionIO
    - Seldon
    - mleap
    - TensorFlow Service
    - TensorFlow Lite
    - TensorFlow.js
- Model Monitoring and Assessment
    - ModelDB
    - Prometheus
    - AI Fairness 360 Open Source
    - Adversarial Robustness 360 Toolbox
    - AI Explainability 360
- Code Asset Management
    - Git
    - Github
    - Gitlab
    - bitbucket
- Data Asset Management
    - Apache Atlas
    - ODPI Egeria
    - Kylo

Part 2:

- Development Environments
    - Jupyter
    - Jupyter lab
    - Apache Zeppelin
    - R Studio
    - Spyder
- Execution Environments
    - Apache Spark
    - Apache Flink
    - Riselab Ray
- Fully Integrated Visual Tooling

        - Knime
        - Orange

Commercial Tools for Data Science

- Data Management
    - Oracle
    - Microsoft SQL Server
    - IBM DB2
- Integration and Transformation (ETL tools)
    - Informatica
    - IBM Infosphere DataStage
    - Talend
    - Watson Studio Desktop
- Data Visualization
    - Tableau
    - Microsoft PowerBI
    - IBM Cognos Analytics
- Model Building
    - SPSS
    - SAS
- Model Deployment
    - SPSS
- Model Monitoring and Assessment
    - None available for commercial
- Data Asset Management
    - Informatica
    - IBM Infosphere Information Governance Catalog
- Development Environments
    - Watson Studio Desktop
- Fully Integrated Visual Tools
    - Watson Studio
    - Watson OpenScale
    - H2O.ai

Cloud Based Tools for Data Science

- Fully Integrated Visual Tools and Platforms
    - Watson Studio
    - Watson OpenScale
    - Microsoft Azure ML
    - H2O.ai
- Data Management
    - Amazon DynamoDB (NoSQL database and SaaS(Software as a Service))
    - Cloudant / CouchDB
    - Db2

- Data Integration and Transformation
    - Informatica
    - Data Refinery
- Data Visualization
    - Datameer
    - IBM Cognos Analytics
    - IBM Data Refinery
    - Watson Studio
- Model Building
    - Watson ML
    - AI Platform Training (Google Cloud)
- Model Deployment
    - SPSS
    - Watson Machine Learning (using REST interface)
    - Amazon SageMaker Model
    - Watson Studio

Lesson 3: Packages, APIs, Data Sets and Models

Libraries for Data Science

- Python
  - Scientific computing
    - Pandas (data structures & tools)
    - NumPy (arrays & matrices)
  - Visualization
    - Matplotlib (plots / graphs, most popular)
    - Seaborn (heat maps, time series, violin plots)
  - High-level machine learning
    - Scikit-learn (regression, classification)
    - Keras (deep learning neural networks)
  - Deep learning libraries
    - Keras (deep learning neural networks)
    - TensorFlow (Deep learning: production and deployment)
    - PyTorch (used for testing)
- Apache Spark
  - Process data in clusters simultaneously
  - Python, R, Scala, SQL
- Scala
  - Vegas
  - Deep Learning: Big DL
- R
  - Has built in functionality for ML and data visualization
  - Ggplot2
  - Others which interface with Keras and TensorFlow

Application Programming Interface (API)

- Lets two pieces of software talk to each other
- Your program communicates with other software component using an API (Pandas object for example)
- REST (REpresentational State Transfer) APIs
  - Communication using the internet
  - Advantages:
    - Storage
    - Data access
    - AI algorithms
    - Other resources
  - Your program is called “Client”
  - Interacts with a web service using a REST API

Data Sets: Powering Data Science

- Data set?

        - Collection of data
        - Data structures:
            - Tabular data
            - Hierarchical data, network data
            - Raw files (images or audio)
- Data Ownership
    - Private data
        - Confidential
        - private/personal information
        - Commercially sensitive
    - Open Data
        - Scientific institutions
        - Government
        - Organizations
        - Companies
        - Publicly available
- Find Open Data
    - Datacatalogs.org
    - Data.un.org
    - Kaggle
    - Google
- Community Data License Agreement
    - CDLA - Sharing
        - Permission to use and modify data
        - Publication only under same terms
    - CDLA - Permissive
        - Permission to use and modify data: no obligations

Sharing Enterprise Data

- Data Asset eXchange (DAX)
    - Curated collection of data sets
    - IBM research and 3rd parties
    - Data science friendly licenses
    - Getting started
        - Download data set
        - Explore data set
        - Metadata
        - Example records

Machine Learning Models

- Data contains a wealth of information
- ML models identify patterns in data
- Model must be trained on data before it can be used to make predictions
- Supervised

        - Most commonly used
        - Human provides input data and correct outputs
        - Used to solve regression and classification models
        - Regression
            - Predict real numeric values
            - Home sale prices, stock market
        - Classification
            - Classify things into categories
            - Email spam filters, fraud detection, image classification
- Unsupervised
    - Data is not labeled
    - Model tries to identify patterns without external help
    - Clustering and anomaly detection
- Reinforcement
    - Conceptually similar to human learning processes
    - Robot learning to walk, chess, Go, other games of skill
- Deep Learning
    - Tries to loosely emulate how the human brain works
    - Applications
        - Natural Language Processing
        - Image, Audio, Video analysis
        - Time series forecasting
        - Much more
    - Requires very large data sets of labeled data
    - Computationally intensive
    - Build from scratch or download from public model repositories
    - Built using
        - Tensorflow
        - Pytorch
        - Keras
    - Popular Model Repositories
        - “Model zoo”
        - ONNX model zoo
- Using Models to Solve a Problem
    - What is this?
        - Prepare data
        - Build Model
        - Train Model (iterative process up til this point)
        - Deploy Model
        - Use model
    - This is a teddy bear

Week 2:

Lesson 1: Jupyter Notebook and JupyterLab

Introduction

- Tool for recording data science experiments
- Allows data scientists to combine text and code blocks in a single file
- Generates plots and tables within the file
- Can be exported as pdf and html files
- JupyterLab
    - Extends notebook
    - Interactive environment
    - Real time editing
    - Can be used with a cloud based service
    - “Pip install jupyterlab”

Getting Started

- Can use Skills Network Virtual Environment to run JupyterLab

Jupyter Kernels

- Notebook kernel is a computational engine that executes the code contained in a notebook file
- Kernels for other languages exist

Jupyter Architecture

- Jupyter implements a two-process model: kernel and client
- Client: interface to send code to the kernel
    - It’s the browser when using a Jupyter Notebook
- Kernel: executes code and returns result to client to display

Shortcuts for jupyter:

- Shift+enter -> run
- A -> insert cell above
- B -> insert cell below

Lesson 2: RStudio IDE

Introduction to R and RStudio

- R: statistical programming language
    - Most use by academics, healthcare, and the government
    - Great for visualization
- RStudio: environment to run R code
- Popular R libraries
    - dplyr: Data Manipulation
    - stringr: String Manipulation
    - ggplot: Data Visualization
    - caret: Machine Learning

Plotting with RStudio

- Ggplot
    - Histograms, bar charts, etc.
    - library(ggplot2)
- Plotly
    - Web-based data visualizations
    - Displayed or saved as individual html files
- Lattice
    - Used to implement complex multi-variable data sets
- Leaflet
    - Popular for creating interactive plots
- How to install: install.packages(“package name”)
- Plot function (just plot creates a scatterplot)

Lesson 3: Github

Overview of GIt/Github

- Version control
- Git
    - free/open source
    - Distributed version control system
    - Accessible anywhere in the world
    - Version control images, documents, etc.
- Glossary of Terms
    - SSH protocol
        - Method for secure remote login from one computer to another
    - Repository
        - Folders of your project set up for version control
    - Fork
        - Copy of the repository
    - Pull Request
        - Process used to request that someone reviews and approves your changes before they become final
    - Working Directory
        - Directory on your file system, including its files and subdirectories, that is associated with a git repository
- Basic Git Commands
    - **Init**: starting a new repository / cloning an existing one
    - **Add**: moves changes from working directory to staging area
    - **Status**: state of working directory and state of your changes
    - **Commit**: takes changes and commits them to the project
    - **Reset**: undoes changes made to the working directory
    - **Log**: browse previous changes to a project
    - **Branch**: create isolated environment within repository to make changes
    - **Checkout**: see and change existing branches
    - **Merge**: puts everything back together

Working with Branches

- Branch = snapshot of repository
- Master = main branch
    - Official working version of project
    - Meant to be stable
- Child Branch
    - Changes and experiments are done
    - Build and test, then merge once changes are working and approved

Week 3:

Lesson 1: Watson Studio

What is IBM Watson Studio?

- Simplify data projects
- Collaborative data science and machine learning environment
- Easy to create visualizations
- Works with open source tools

Lesson 2: Other IBM Tools

Other tools for Data Science

- Watson knowledge catalog
- Data refinery
- Modeler Flows in Watson Studio
- IBM SPSS Statistics and IBM SPSS Modeler
- Model deployment strategies
- AutoAI and Watson OpenScale

IBM Watson Knowledge Catalog

- Data Asset Management
- Code Asset Management
- Data Management
- Data Integration and Transformation
- Main Features
    - Find Data
    - Catalog..
    - Govern..
    - Understand..
    - Power..
    - Prepare..
    - Connect..
    - Deploy Anywhere

Data Refinery

- Cleane, Shape, and Prepare data
- Interactive visual interface in order to streamline data preparation
- Kinda works like database operations

SPSS Modeler Flows

- Data Management

- Data Integration and Transformation
- Data Visualization
- Model Building

IBM SPSS Modeler
- Data Management
- Data Integration and Transformation
- Data Visualization
- Model Building
- Model Deployment

Model Deployment with Watson ML
- Model Deployment
- Some Solutions
    - Sage Maker (Amazon)
    - MLFlow (Databricks)
    - Airflow (AirBnB)
    - Kubeflow (Google)
- PMML (Predictive Model Markup Language)
- PFA (Portable Format for Analytics)
- ONNX (Open Neural Network eXchange)

AutoAI in Watson Studio
- Data Preparation
- Model Development
- Feature Engineering
- Hyper-parameter Optimization

IBM Watson OpenScale
- Fairness: monitor bias
- Explainability: explain model decisions
- Model Monitoring: help find causes when there is model drift detected, possibly retrain
- Business Impact: Correlate model metrics and KPIs to measure business impact