Week 2: The Data Science ToolBox

Day 1: Tuesday 10th December 2019

Getting to know your core tools

Lesson 1: Unix (Command Line)

Following week 1 module that introduced you to the profession and its underlying methodology, this week we will dive into a broad collection of tools used in data science.

The most basic tool that allows for quick viewing, exploration and plotting of small data are spreadsheet software such as Microsoft Excel, Google Spreadsheet or Apple Numbers. However, there are technical limitations given that these softwares; they can hardly process data with more than 1m rows/instances and the difficulty to keep a record of what was done.

On the extreme, to overcome the challenges that spreadsheet present Unix provides an alternative way to interact with your computer to execute programs for data processing. Unix is a multitasking, multi-user computer operating system that exists in many variants. It was developed in 1960s in AT&T Bells lab. It was standardised through POSIX (Portable Operating System Interface based on Unix) in 1989. It uses command line tools, stdin and stdout operations to perform basic data manipulation of data files represented as lines of text.

Basic command line tools:

- “man” or “-help” : to access information or manual of a command line. e.g man sort, sort -help
- “cd” : changes working directory. e.g cd path/to/directory
- “mkdir” : make/create directory. e.g mkdir SGA07_DATASCI
- “rmdir” : delete/remove directory. e.g rmdir SGA07_DATASCI
- “cp”, “mv”, “rm” : copy, move/rename, remove files from a directory respectively. e.g cp text.file SGA07_DATASCI
- “ls” : list files in a directory. e.g ls SGA07_DATASCI
- “cat” : reads file to stdin and writes the file to stdout. e.g cat text.file
- “wc” : write word count of file to stdout. Generates 3 values of number of lines (-l), number of words (-w) and number of characters (-c) in text.file. e.g wc text.file

References:

https://www.cl.cam.ac.uk/teaching/1213/UnixTools/materials.html

https://www.datascienceatthecommandline.com/index.html

https://www.howtogeek.com/249966/how-to-install-and-use-the-linux-bash-shell-on-windows-10/

Lesson 2: R and RStudio

While Unix provides a very low level way to interact with your computer during day to day data science tasks, its lacks GUI (Graphic User Interface) that provides some perception to efficiency.

R is an open-source programming language that is focused on delivering a better and user-friendly way to do data analysis, statistics and graphical models. It was developed in 1995 by Ross Ihaka and Robert Gentleman as an implementation of S (statistical programming language) an enterprise solution developed at Bell's Laboratories.

Due to R's adoption for academic and research purposes, it has gathered a huge community that provide online support through CRAN and online forums such as Stack Overflow. CRAN (Comprehensive R Archive Network) is a huge repository of curated R packages - a collection of R functions and data that make it easy to immediately get access to the latest techniques and functionalities without needing to develop everything from scratch yourself.

Some great R packages

- Dplyr for easy data manipulation
- Stringr for easy text processing
- Ggplot for easy data visualisation
- Caret for easy data modelling

R becomes a more functional tool through RStudio - an IDE (Integrated Development Environment) for R programming. It provides a great way to manage your workflow as a data scientist with its intuitive view compartments that includes a console for command line interaction, text editor for direct code execution as well as tools for plotting, history, debugging and workspace management.

References:

https://cran.r-project.org

https://rstudio.com/products/rstudio/

https://www.analyticsvidhya.com/blog/2016/02/complete-tutorial-learn-data-science-scratch/

https://r4ds.had.co.nz/introduction.html

Lesson 3: Python and Anaconda

As was explained in the Lesson 2 of your Day 1 module, a data scientist effectively does more programming than a statistician and more statistics than a programmer. Therefore, it is important to equip yourself with more than R as your go-to programming language.

Python which is also open-source provides a more general-purpose language with readable syntax unlike R that is built by statisticians. While R is effective for statistical analysis, Python is more effective in deployment and large scale implementation of machine learning models.

Python was created by Guido Van Rossem in 1991 with emphasises on productivity and code readability. Similar to CRAN, PyPi is the Python Package index and consists of libraries to which users can contribute. Python has a great community and is growing a rapid dominance within the data science community.

Some great Python packages:

- Numpy for effective manipulation of N-dimension arrays
- Pandas for effective data structure and analysis
- Scikit-learn for effective machine learning
- Seaborn for effective data visualisation

Anaconda is a Python-based data processing and scientific computing platform. It has built in many useful third-party libraries. Installing Anaconda is equivalent to automatically installing Python and some commonly used libraries such as Numpy, Pandas, Scrip, and Matplotlib, so it makes the installation so much easier than regular Python installation. In some way, it serves the same way purpose of RStudio to R and even more.

References:

https://www.python.org

https://www.anaconda.com

https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-learn-data-science-python-scratch-2/

https://tanthiamhuat.files.wordpress.com/2018/04/pythondatasciencehandbook.pdf

TASK: SETUP YOU DATA SCIENCE TOOLBOX

- Make sure you have terminal/shell interface for your computer
  - Windows: https://helpdeskgeek.com/windows-10/how-to-install-use-the-new-windows-10-terminal/
  - Mac: https://www.businessinsider.com/how-to-open-terminal-on-mac?IR=T
  - Linux: https://www.howtogeek.com/140679/beginner-geek-how-to-start-using-the-linux-terminal/
- Create your root directory for Course data products

```
mkdir SGA07_DATASCI
```

- Install R and RStudio
  - R
    - Windows: https://www.youtube.com/watch?v=GAGUDL-4aVw
    - Mac: https://www.youtube.com/watch?v=1PsPfMaLWSk
    - Ubuntu: https://www.youtube.com/watch?v=GsuA5ugYqyw
  - RStudio
    - Windows: https://download1.rstudio.org/desktop/windows/RStudio-1.2.5019.exe
    - Mac: https://download1.rstudio.org/desktop/macos/RStudio-1.2.5019.dmg
    - Ubuntu: https://download1.rstudio.org/desktop/trusty/amd64/rstudio-1.2.5019-amd64.deb
- Install Python and Anaconda
  - Python
    - Windows: https://www.youtube.com/watch?v=dX2-V2BocqQ
    - Mac: https://www.youtube.com/watch?v=TgA4ObrowRg
    - Ubuntu: https://www.youtube.com/watch?v=PSLp7rB6lG4
  - Anaconda
    - Windows: https://repo.anaconda.com/archive/Anaconda3-2019.10-Windows-x86_64.exe
    - Mac: https://repo.anaconda.com/archive/Anaconda3-2019.10-MacOSX-x86_64.pkg
    - Linux: https://repo.anaconda.com/archive/Anaconda3-2019.10-Linux-x86_64.sh
- Create your data profile as either a R or Python script file
  - Create a new Project in RStudio: https://www.youtube.com/watch?v=etkSsF6r2iU
  - Create a new R File in RStudio: https://www.youtube.com/watch?v=rWHV2VlQo2w
  - Refer to Week 1 Day 1 Task and complete with R programming language: https://www.dummies.com/programming/r/how-to-create-a-data-frame-from-scratch-in-r/

Day 2: Thursday 12th December 2019

Getting to know your production tools

Lesson 1: Version Control

For project oriented work, it is important to keep track of all changes made to files. You will soon come to realise that your data science work is rather not literal but follows a more iterative search approach. Therefore, the ability to keep track of changes not only to an individual file but across a host of files will help keep sanity in check. Also, it provides the functionalities to effectively collaborate with other data scientist or developer on a project.

For the purpose of this course, we will be using git for version control. Git is a distributed version control system that allows user to keep entire source code repository and history on their local machine. It was created in 2005 by Linus Torvald to aid the Linux kernel development.

Here are some key concepts of Git as a version control system:

- A repository - This is Git's name for a project. It includes all of the files in the project along with all of the information about how they have changed over time. If you have a full copy of a repository (often referred to as a "repo"), you can view the current state of the project, but you can also view any previous state that the project used to be in.
- A snapshot - This is the way Git keeps track of your code history. It essentially records what all your files look like at a given point in time. However, you decide on when to take a snapshot, of what files and have the ability to go back and visit any snapshot.
- A commit - In Git, history is made up of a series of commits which are stored in the changelog. Every time you make a meaningful set of changes to your project, you should commit them so that you can always get back to the project in that state in the future.
- The Staging Area - This is like a shopping basket for version control. It's where you load up the sets of changes that you'd like to put in your next commit, so if you have edited three files, but want to make one commit with two of them and another commit with the third, you just "stage" the first two using the Git add command, then commit them with an appropriate message and then add and commit the last file separately.

Github (www.github.com) is a web-based git repository hosting service. It allows for code collaboration and storage online as well as adds extra functionality on top of git such as user interface (UI), documentation, bug tracking, feature request, push and pull requests and more!

Reference:

https://www.youtube.com/watch?v=SWYqp7iY_Tc

https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners

https://help.github.com/en/github/using-git/getting-started-with-git-and-github

Lesson 2: Markdown

As outlined in Week 1 Day 2 Lesson 1, one of the important aspects of the data science methodology is reporting and visualisation.

Markdown is a lightweight markup language that you can use to add formatting elements to plaintext text documents. It was developed by John Gruber in 2004 to allows web writers to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML. It can be used to create websites, presentations and technical documentations.

Markdown is portable and platform independent which provides it the flexibility to be used in any development environment (either R or Python). For the most part, as a data scientist you would use it document your methodology, approach and result as well as use it for technical documentation of your model for deployment and production.

Some syntax inc reading a markdown document includes:

- Headings

    This is an H1

    ==========

    This is an H2

    ——————

    # This is an H1

    ## This is an H2

    ###### This is an H6

- Lists

    Unordered

        * Red
        * Green
        * Blue
        - Red
        - Green,.
        - Blue

    Ordered

        1. Bird
        2. McHale
        3. Parish

References:

https://www.markdownguide.org/getting-started/

https://daringfireball.net/projects/markdown/syntax

Lesson 3: Specialised Cloud-based Tools

As you embark on your journey to become a data scientist, it will become important to have knowledge of some specialised tools that provides competitive advantage. These specialised tools provide some greater functionalities to each process of the data science methodology.

Below is a review of 7 most sort after cloud-based technologies:

- Microsoft Power BI is a collection of software services, apps, and connectors that work together to turn your unrelated sources of data into coherent, visually immersive, and interactive insights.
- Tableau Tableau is a powerful and fastest growing data visualisation tool used in the Business Intelligence Industry. It helps in simplifying raw data into the very easily understandable format.
- Google Data Studio is a dashboard and reporting tool that is easy to use, customise, and share. It allows you to transform your data into appealing and informative reports for your audience.
- SAS Business Intelligence is a cloud-based enterprise analysis tool that helps users monitor metrics and manage interactive reports. Designed for large businesses, the platform's features include customisable dashboard, marketing reports, forecasting, data source connectors, ad-hoc analysis and more.
- IBM Watson Studio is an integrated environment designed to make it easy to develop, train, manage models, and deploy AI-powered applications and is a SaaS solution delivered on the IBM Cloud.
- AWS Machine Learning is a robust, cloud-based service that makes it easy for developers of all skill levels to use machine learning technology. Amazon ML provides visualisation tools and wizards that guide you through the process of creating machine learning (ML) models without having to learn complex ML algorithms and technology.

References:

https://powerbi.microsoft.com/en-us/

https://www.tableau.com

https://datastudio.google.com

https://www.sas.com/en_us/solutions/business-intelligence.html

https://www.ibm.com/cloud/watson-studio

https://aws.amazon.com/machine-learning/

TASK: SETUP YOUR PRODUCTION ENVIRONMENT

- Create a GitHub account
- Convert your root directory for Course data products into a version control repository
  - Local Git
  - Push to your Github account
- Create a README.md markdown document with the following sections
  - Title: {Title of Your Final Course Project}
  - Subtitle: Stutern Graduate Accelerator 07 - Data Science Course
  - Author: {Your Name}
  - Heading 1: Motivation
    - Remember the motivation report you write for your pre-course task, you need to revisit and see if that needs an update. If not you can paste the text here
  - Image: Data Profile Visualisation
    - Remember the data science profile visualisation for your week 1 day 1 task, you will need to revisit and export the bar chart (Note: you should have reviewed this task again in Day 1 of this Week)
  - List: References
    - Remember the reference list (1 book, 3 journals and 10 articles) for your week 1 day 2 task, you will need to revisit and create a reference list with what you have so far

    Please refer to the link for template: https://github.com/akinlabiceo/SGA07_DATASCI.git/README.pdf
- Take some time to review the specialised cloud-based tools, create an account as start to get some intuition on how the tools work
  - Note that some of these tools have paid plan, so be mindful of your account creation as this is to your discretion